Boosting

Statistical Learning
Master in Big Data. University of Santiago de Compostela
Manuel Mucientes

Introduction

- "Boosting is one of the most powerful learning ideas introduced in the last twenty years" (Hastie et al., 2009)
- Idea: combine the outputs of many "weak" classifiers to produce a powerful "committee"
- Weak classifier: its error rate is only slightly better than random guessing

Introduction (ii)

- Boosting is a way of fitting an additive expansion in a set of elementary basis functions:

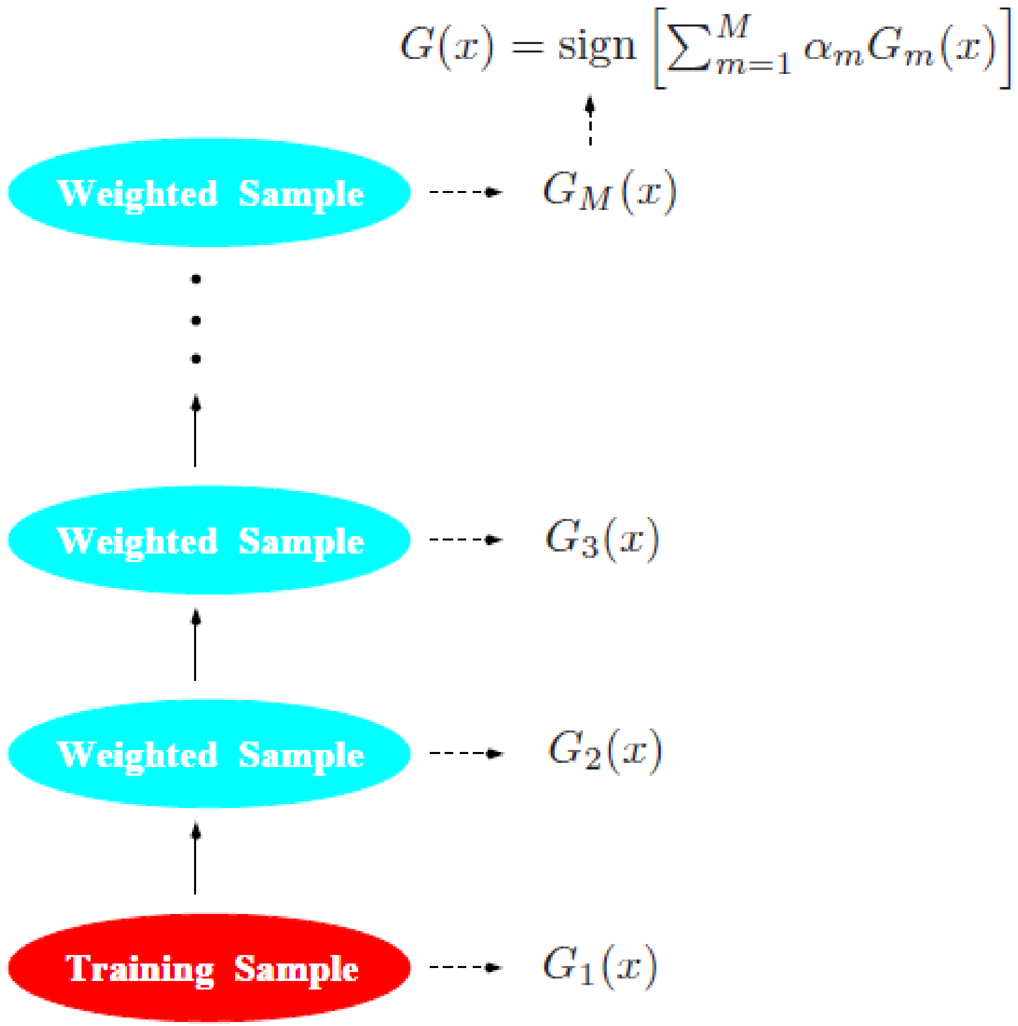
$$f(x) = \sum_{m=1}^{M} \beta_m b(x; \gamma_m)$$

  - Basis functions: weak classifiers
  - β_m (expansion coefficients), γ_m (parameters of the functions)
- Loss function:

$$\min_{\{\beta_m, \gamma_m\}_1^M} \sum_{i=1}^{N} L\left(y_i, \sum_{m=1}^{M} \beta_m b(x_i; \gamma_m)\right)$$

AdaBoost

- Most popular boosting algorithm: AdaBoost.M1 (Freund and Schapire, 1997)
- Two-class problem: output variable in {-1, 1}
- Boosting: sequentially apply the weak classification algorithm to repeatedly modified versions of the data
- Final prediction: weighted majority vote
  - Give a higher influence to the more accurate classifiers



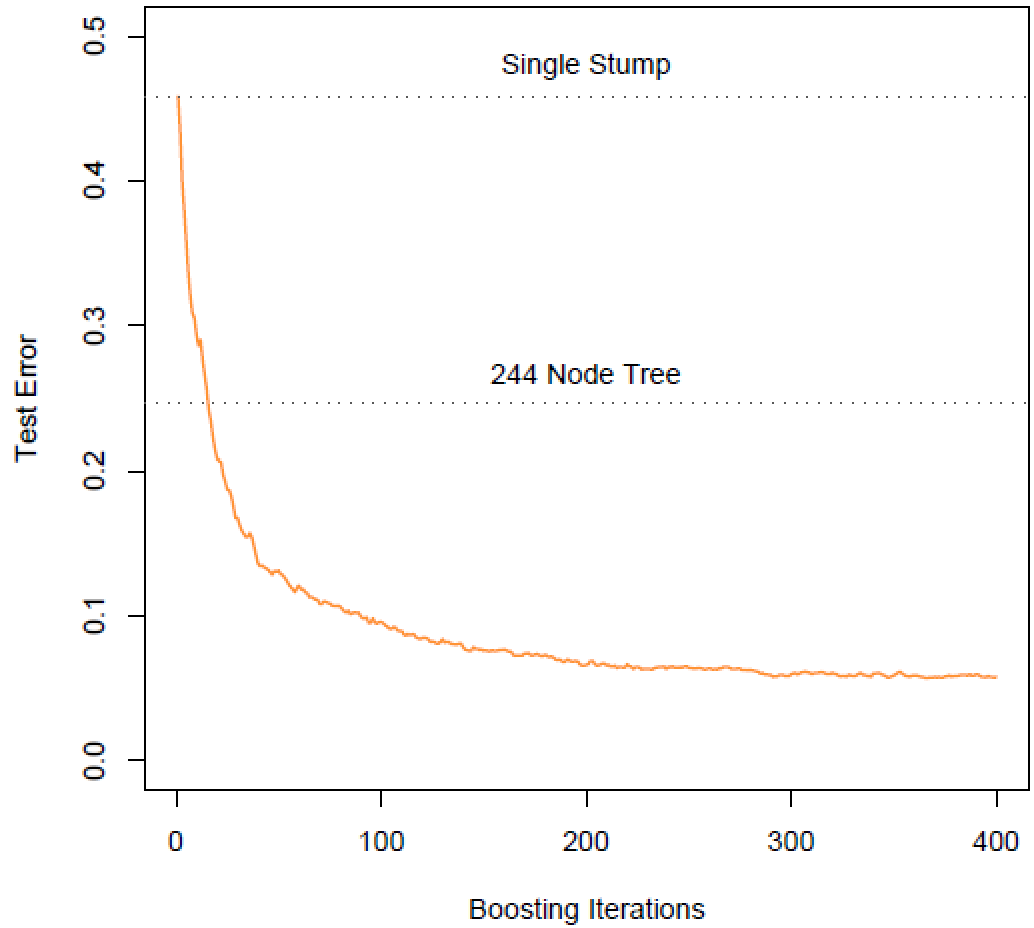
AdaBoost (ii)

Algorithm 10.1 *AdaBoost.M1.*

1. Initialize the observation weights $w_i = 1/N$, $i = 1, 2, \ldots, N$.
2. For $m = 1$ to M:
   (a) Fit a classifier $G_m(x)$ to the training data using weights w_i.
   (b) Compute
   $$\text{err}_m = \frac{\sum_{i=1}^{N} w_i I(y_i \neq G_m(x_i))}{\sum_{i=1}^{N} w_i}.$$
   (c) Compute $\alpha_m = \log((1 - \text{err}_m)/\text{err}_m)$.
   (d) Set $w_i \leftarrow w_i \cdot \exp[\alpha_m \cdot I(y_i \neq G_m(x_i))]$, $i = 1, 2, \ldots, N$.
3. Output $G(x) = \text{sign}\left[\sum_{m=1}^{M} \alpha_m G_m(x)\right]$.

AdaBoost (iii)

- Example:
  - Target: $Y = \begin{cases} 1 & \text{if } \sum_{j=1}^{10} X_j^2 > \chi_{10}^2(0.5), \\ -1 & \text{otherwise.} \end{cases}$
  - Ten independent Gaussian features
  - Training: 2,000 cases
  - Test: 10,000 cases
  - Weak classifier: stump (two-terminal node tree)
    - Single stump: 45.8% test error
- Adaboost with trees: "best off-the-shelf classifier in the world" (Breiman, 1998)



"Off-the-Shelf" Procedures for Data Mining

Characteristic	Neural Nets	SVM	Trees	MARS	k-NN, Kernels
Natural handling of data of "mixed" type	▼	▼	▲	▲	▼
Handling of missing values	▼	▼	▲	▲	▲
Robustness to outliers in input space	▼	▼	▲	▼	▲
Insensitive to monotone transformations of inputs	▼	▼	▲	▼	▼
Computational scalability (large N)	▼	▼	▲	▲	▼
Ability to deal with irrelevant inputs	▼	▼	▲	▲	▼
Ability to extract linear combinations of features	▲	▲	▼	▼	◆
Interpretability	▼	▼	◆	▲	▼
Predictive power	▲	▲	▼	◆	▲

- MARS (Multivariate Adaptive Regression Splines)

- Boosting trees improves their accuracy, maintaining most of their desirable properties

Gradient Boosting

- Originally called MART (Multiple Additive Regression Trees)
  - Also known as Gradient Tree Boosting
- Idea:
  - At each step the solution tree is the one that maximally reduces:

$$\hat{\Theta}_m = \arg\min_{\Theta_m} \sum_{i=1}^{N} L\left(y_i, f_{m-1}(x_i) + T(x_i; \Theta_m)\right)$$

  - Fit the tree to the components of the negative gradient:

$$r_{im} = -\left[\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)}\right]_{f=f_{m-1}}$$

    - This components are referred to as generalized or pseudo residuals

Gradient Boosting (ii)

Algorithm 10.3 *Gradient Tree Boosting Algorithm.*

1. Initialize $f_0(x) = \arg\min_\gamma \sum_{i=1}^{N} L(y_i, \gamma)$.
2. For $m = 1$ to M:
   (a) For $i = 1, 2, \ldots, N$ compute
   $$r_{im} = -\left[\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)}\right]_{f=f_{m-1}}.$$
   (b) Fit a regression tree to the targets r_{im} giving terminal regions $R_{jm},\ j = 1, 2, \ldots, J_m$.
   (c) For $j = 1, 2, \ldots, J_m$ compute
   $$\gamma_{jm} = \arg\min_{\gamma} \sum_{x_i \in R_{jm}} L\left(y_i, f_{m-1}(x_i) + \gamma\right).$$
   (d) Update $f_m(x) = f_{m-1}(x) + \sum_{j=1}^{J_m} \gamma_{jm} I(x \in R_{jm})$.
3. Output $\hat{f}(x) = f_M(x)$.

Gradient Boosting (iii)

- Tree size: restrict all trees to be the same size
  - Cross-validation to select J seldom improves over using J=6 (Hastie et al., 2009, p. 363)
  - In real problems larger J might be necessary

- Optimal number of trees (M): validation sample

- Shrinkage: another way to regularize
  - Scale the contribution of each tree: line 2(d) of gradient boosting

$$f_m(x) = f_{m-1}(x) + \nu \cdot \sum_{j=1}^{J} \gamma_{jm} I(x \in R_{jm})$$

Gradient Boosting (iv)

- Subsampling: Stochastic gradient boosting (Friedman, 1999)
  - At each iteration sample a fraction (η) of the training set without replacement
- Four hyper-parameters: *J*, *M*, ν, η
  - Determine suitable values for *J*, $\nu\ (< 0.1)$, η (0.5)
  - Pick *M* through validation

Variable importance of additive trees

- Contribution of each input variable in predicting the response

- For a single tree (Breiman et al., 1984):

$$\mathcal{I}_\ell^2(T) = \sum_{t=1}^{J-1} \hat{\imath}_t^2 \, I(v(t) = \ell)$$

  - J-1: number of internal nodes
  - $\hat{\imath}_t^2$: improvement of RSS (regression), Gini index or cross-entropy (classification)

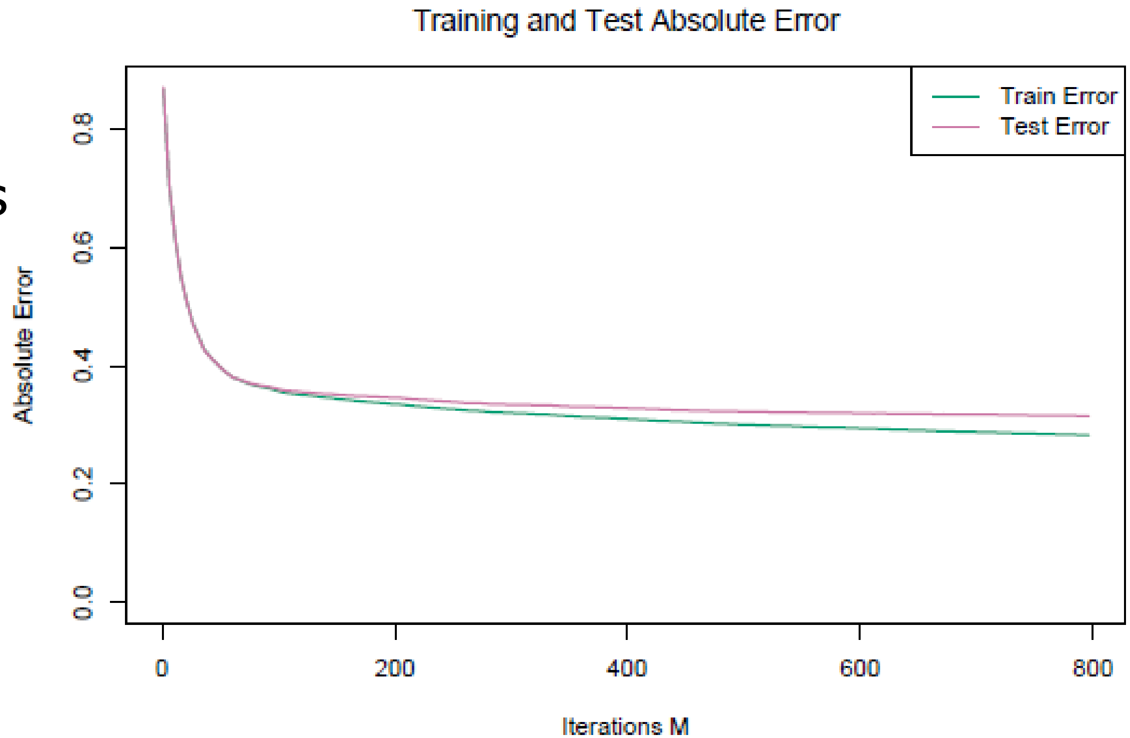
- For additive trees: $$\mathcal{I}_\ell^2 = \frac{1}{M} \sum_{m=1}^{M} \mathcal{I}_\ell^2(T_m)$$

  - More reliable than for a single tree

Example: California Housing

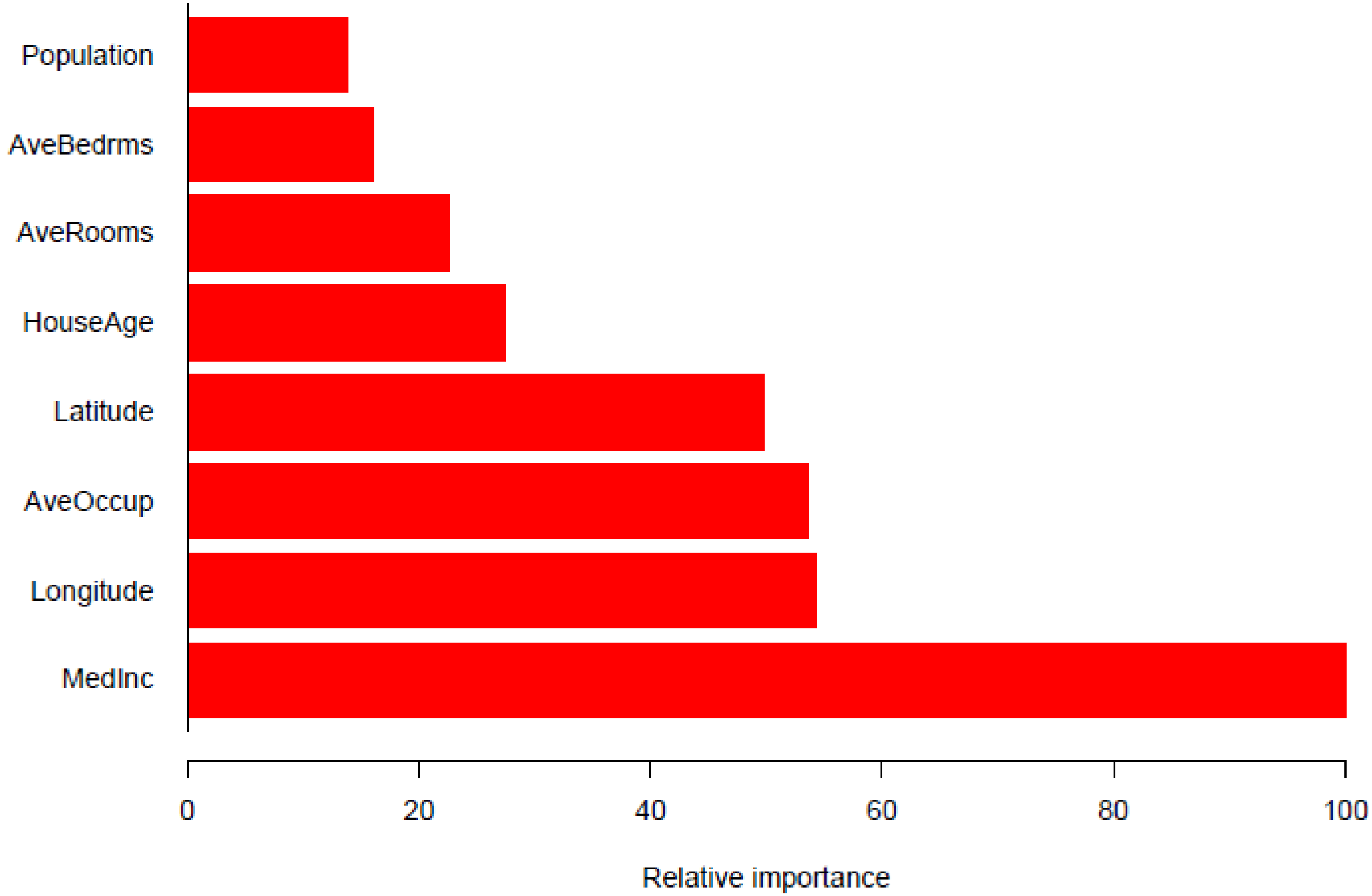
- Pace and Barry, 1997. StatLib repository
- 20,460 neighborhoods in California: 80% training, 20% test
- Response variable: median house value in each neighborhood in units of $100,000
- Eight numerical predictors: median income (MedInc), housing density (House), etc.
- Gradient boosting with

  J=6, ν=0.1, Huber loss



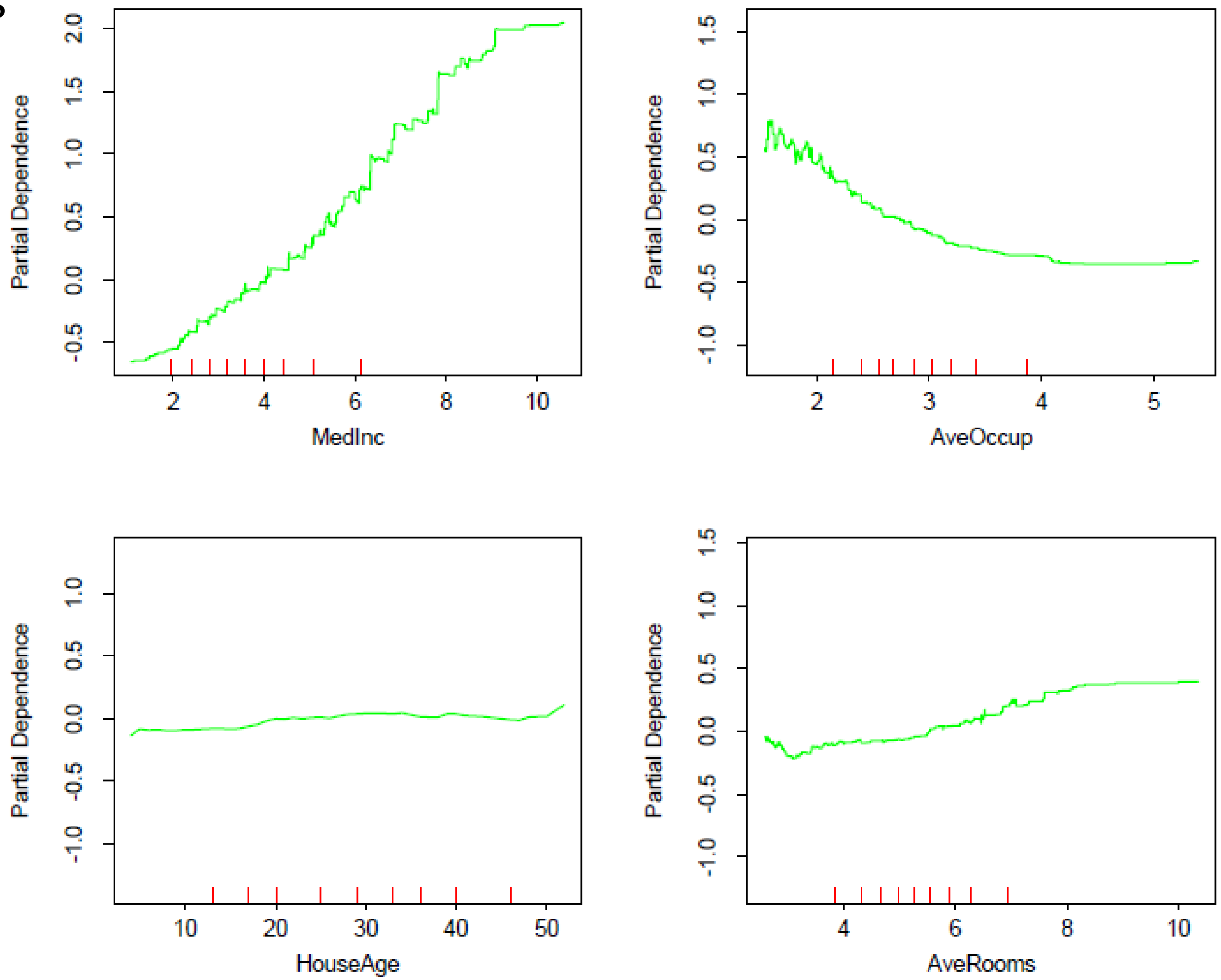
Example: California Housing (ii)

- Variable importance



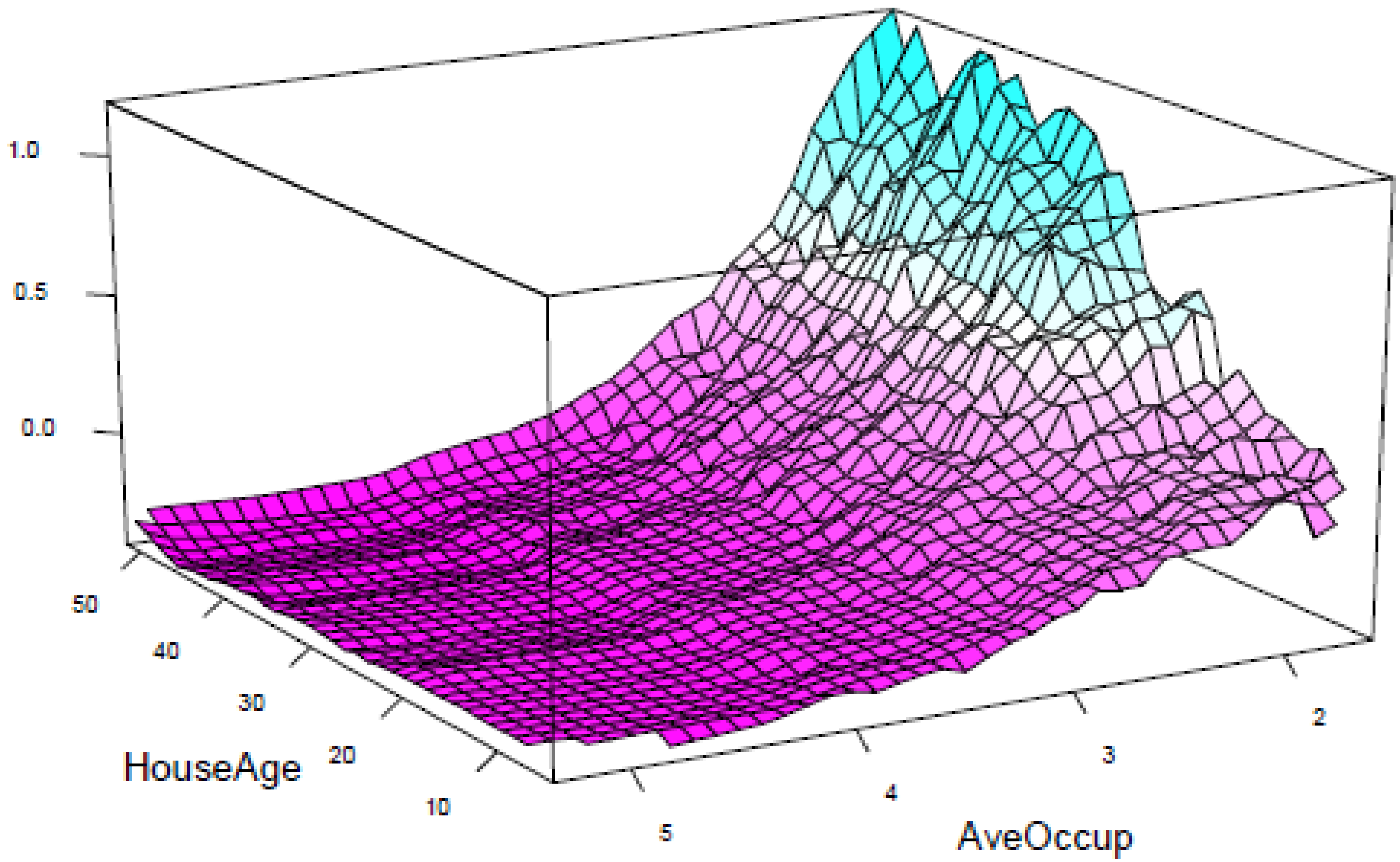
Example: California Housing (iii)

- Partial dependence plots (one variable)
  - Effect of a variable taking into account the (average) effects of the other variables



Example: California Housing (iv)

- Partial dependence plot (two variables)



Bibliography

- T. Hastie, R. Tibshirani, y J. Friedman, The elements of statistical learning. Springer, 2009.
  - Chapter 10
- G. James, D. Witten, T. Hastie, y R. Tibshirani, An Introduction to Statistical Learning with Applications in R. Springer, 2013.
  - Chapter 8, Sec. 8.2.3