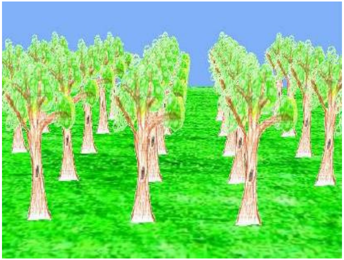
Random forests

Gérard Biau

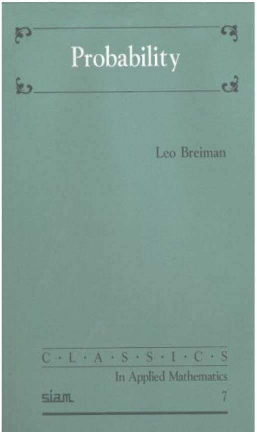


Hervelee, September 2012

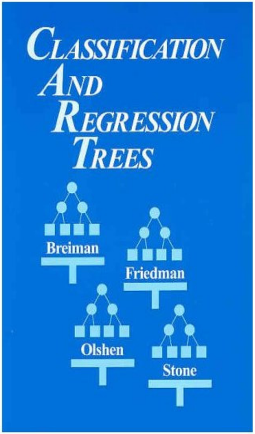
Outline

Outline

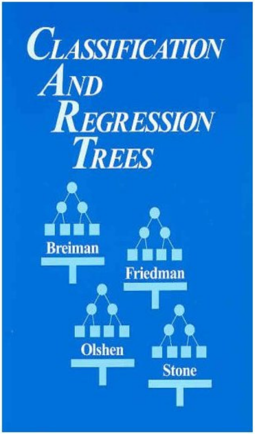
Leo Breiman (1928-2005)

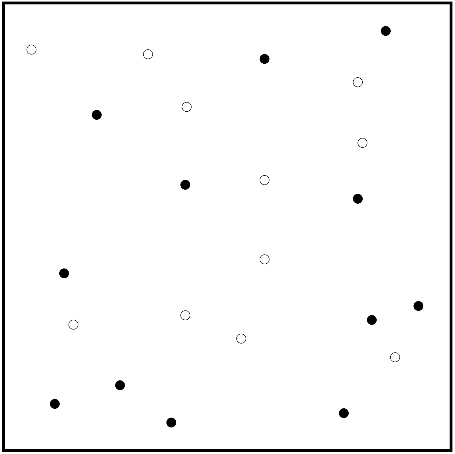


Leo Breiman (1928-2005)

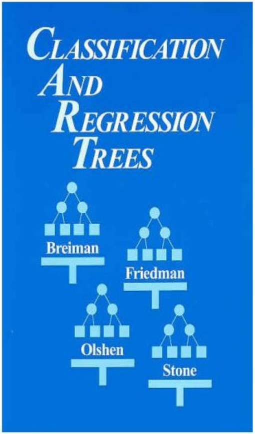


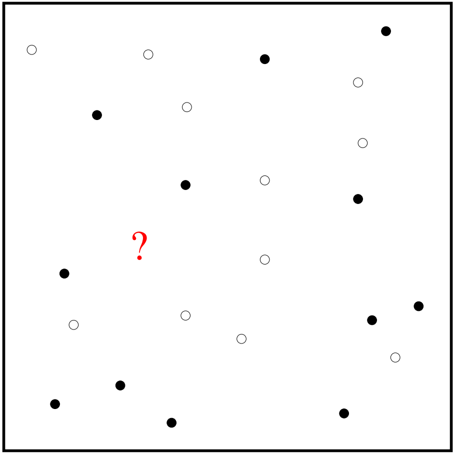
Classification

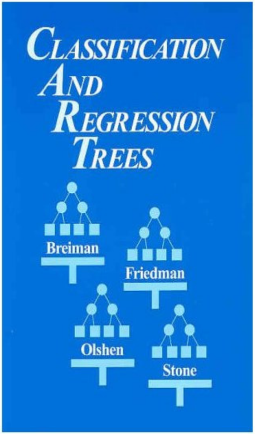


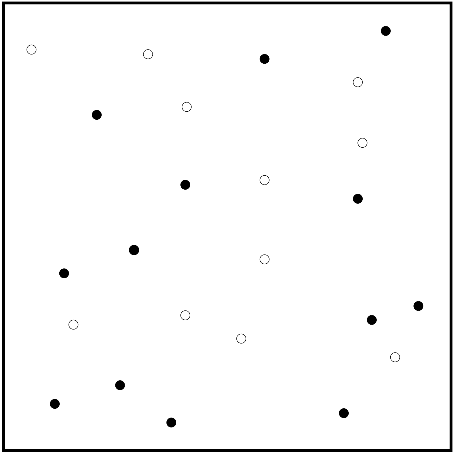


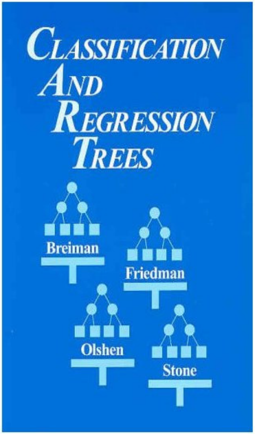
Classification

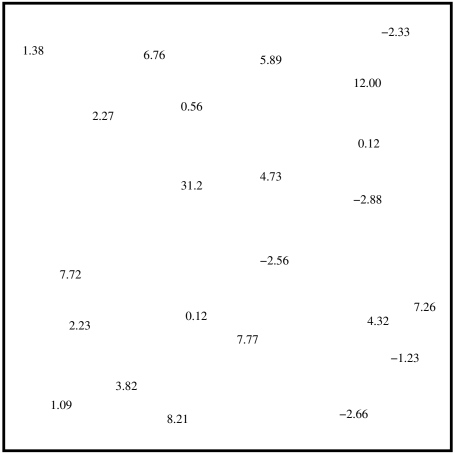




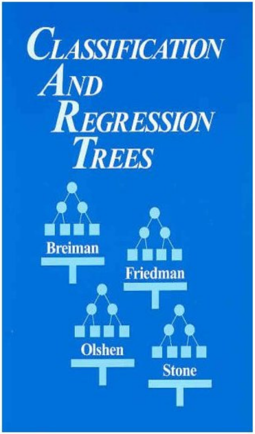

CLASSIFICATION
AND
REGRESSION
TREES
Breiman
Friedman
Olshen
Stone

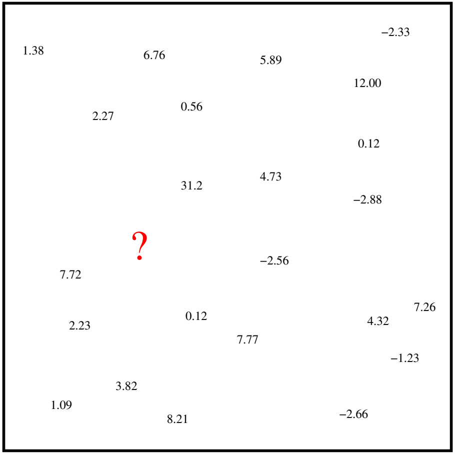



CLASSIFICATION
AND
REGRESSION
TREES
Breiman
Friedman
Olshen
Stone

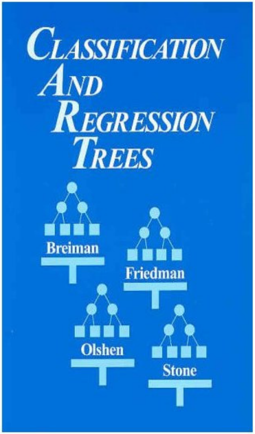

−2.33
1.38
6.76
5.89
12.00
2.27
0.56
0.12
31.2
4.73
−2.88
−2.56
7.72
7.26
2.23
0.12
4.32
7.77
−1.23
3.82
1.09
8.21
−2.66

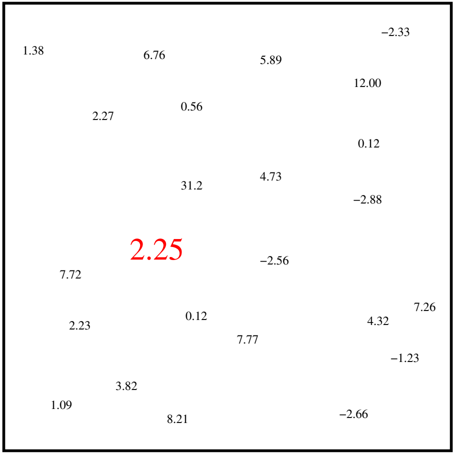
Regression


CLASSIFICATION
AND
REGRESSION
TREES
Breiman
Friedman
Olshen
Stone

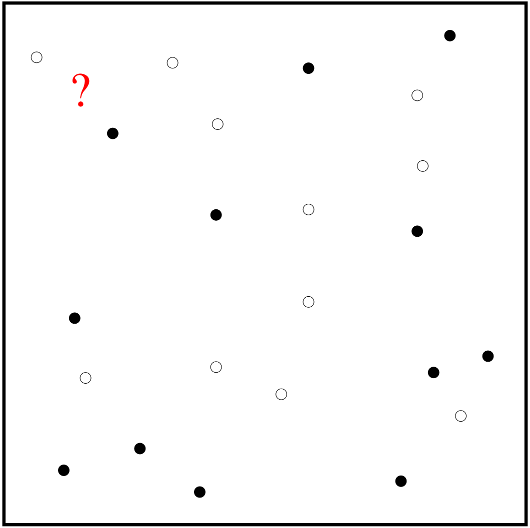

−2.33
1.38
6.76
5.89
12.00
2.27
0.56
0.12
4.73
31.2
−2.88
?
−2.56
7.72
7.26
0.12
4.32
2.23
7.77
−1.23
3.82
1.09
8.21
−2.66

Regression

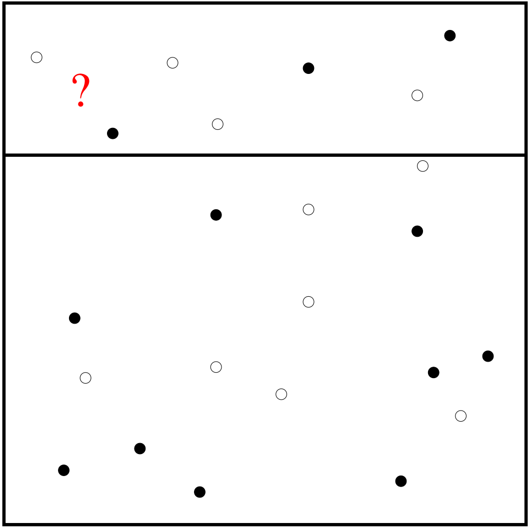

CLASSIFICATION
AND
REGRESSION
TREES
Breiman
Friedman
Olshen
Stone


−2.33
1.38
6.76
5.89
12.00
2.27
0.56
0.12
31.2
4.73
−2.88
2.25
−2.56
7.72
7.26
2.23
0.12
4.32
7.77
−1.23
3.82
1.09
8.21
−2.66

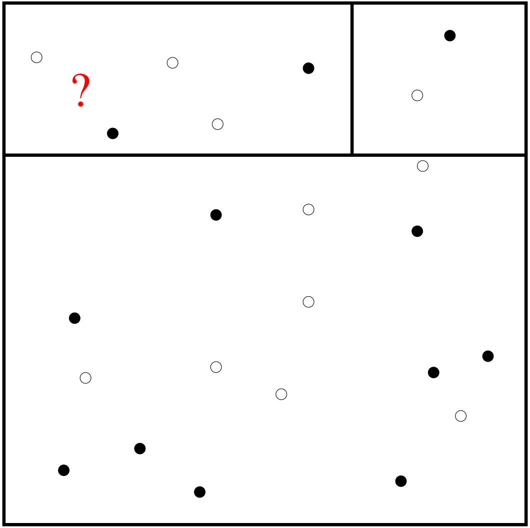
Trees



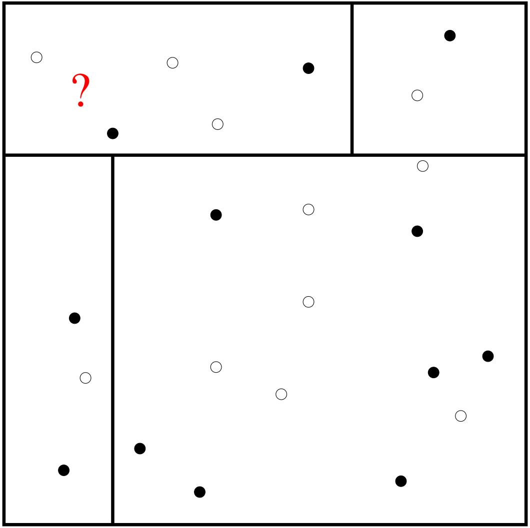
Trees



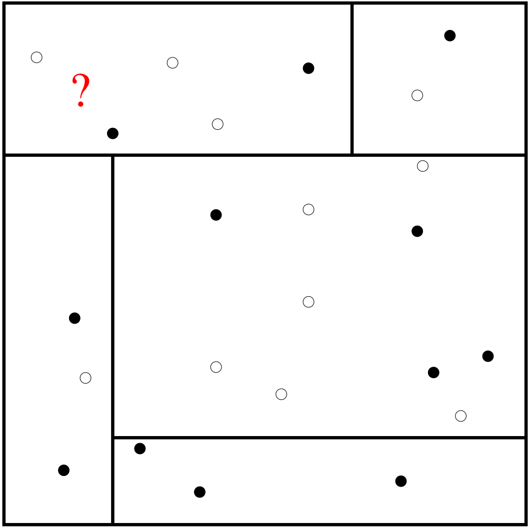
Trees



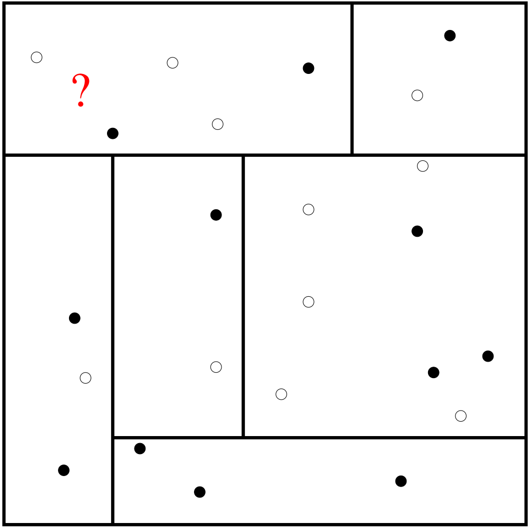
Trees



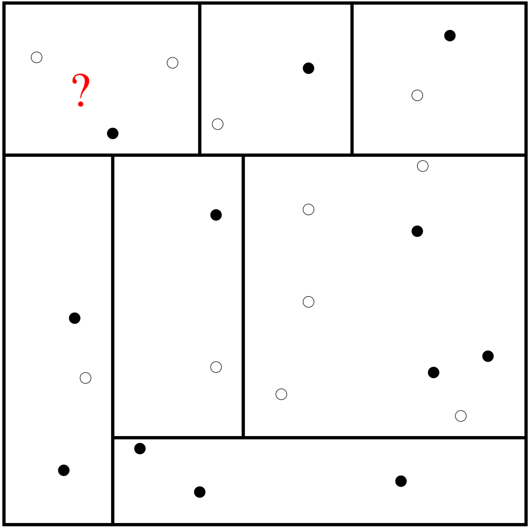
Trees



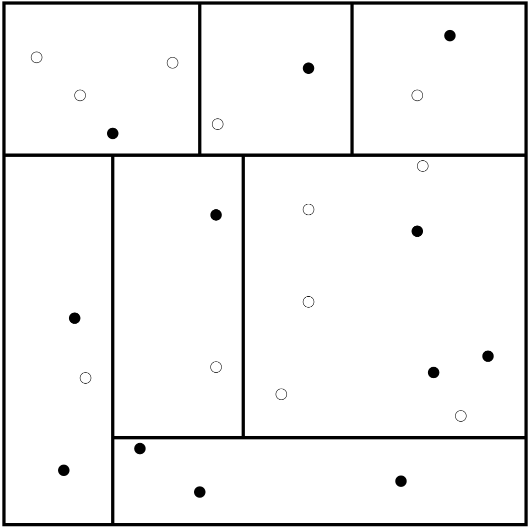
Trees



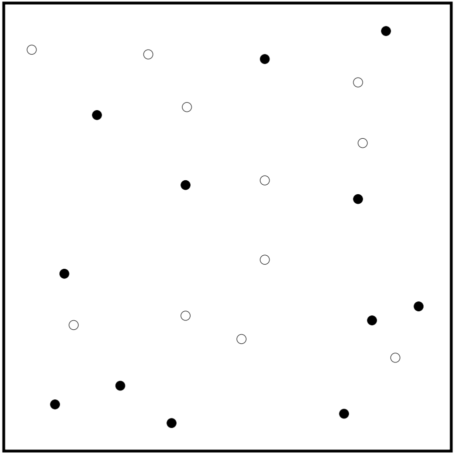
Trees



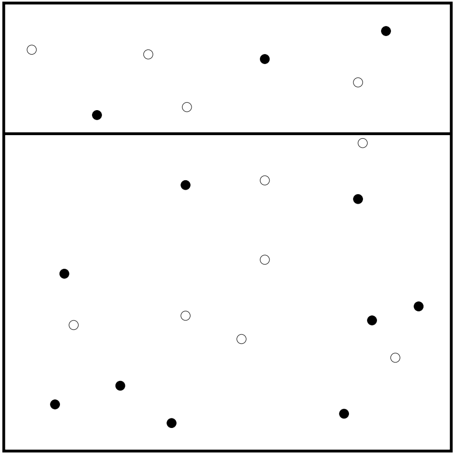
Trees

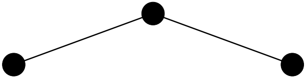


Trees

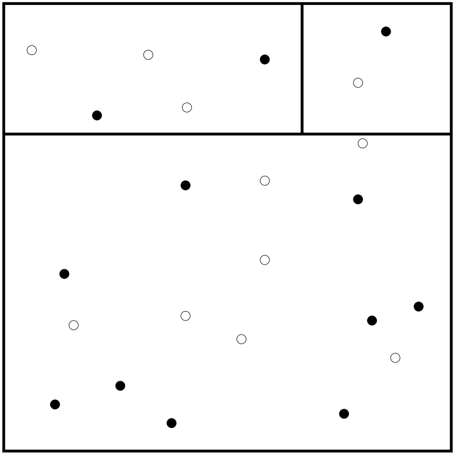


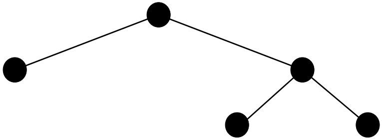
Trees



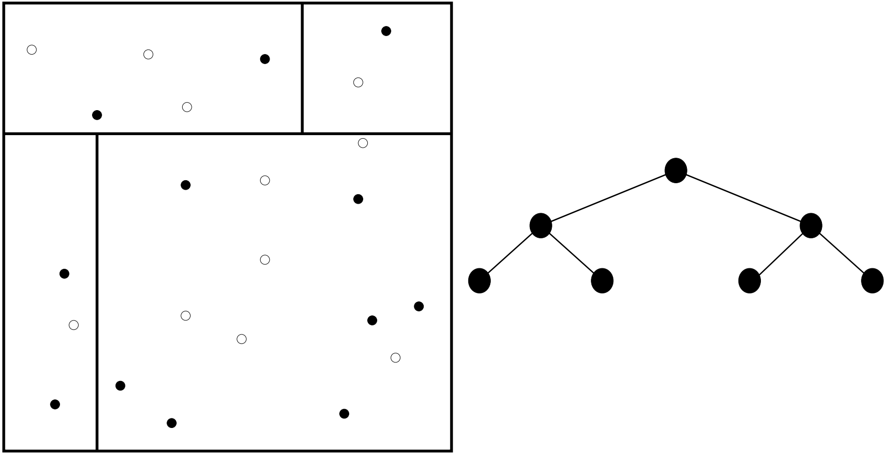


Trees

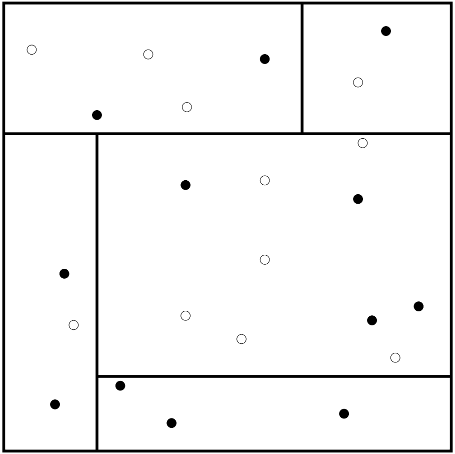


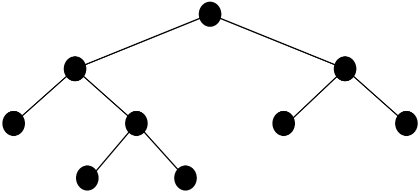


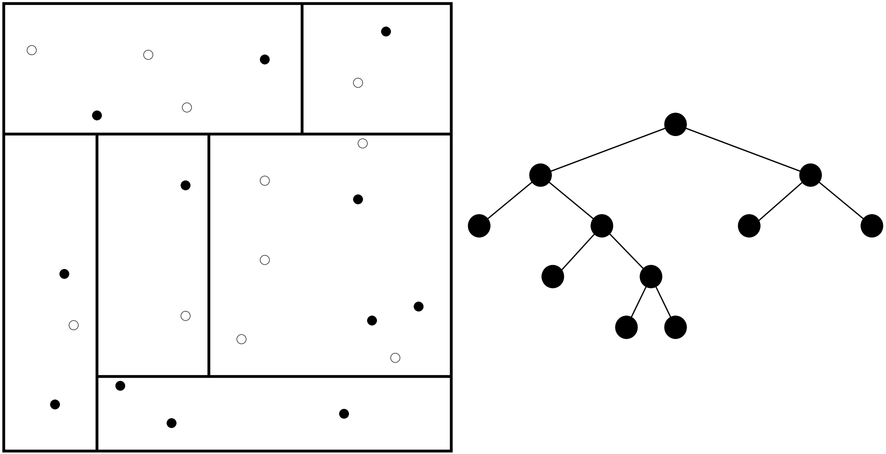
Trees



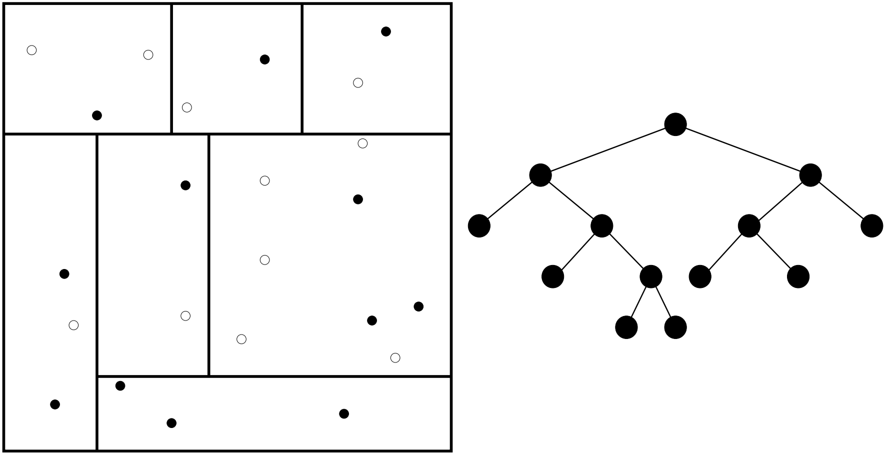
Trees



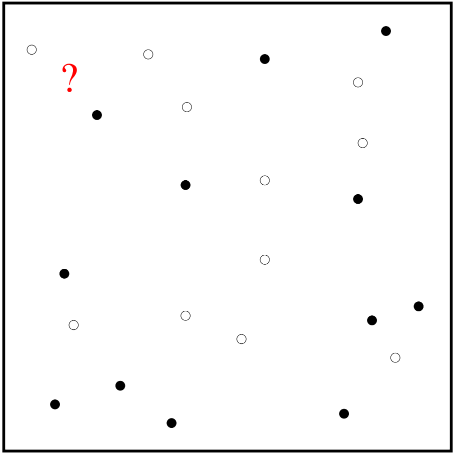




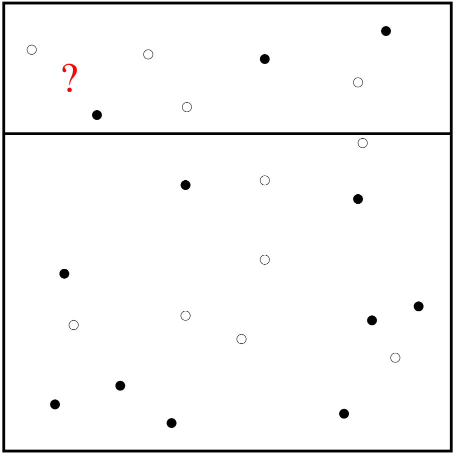
Trees

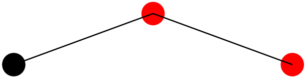


Trees

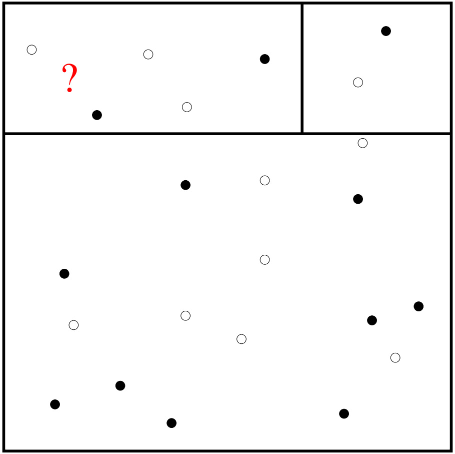


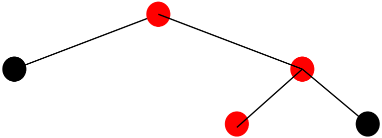
Trees

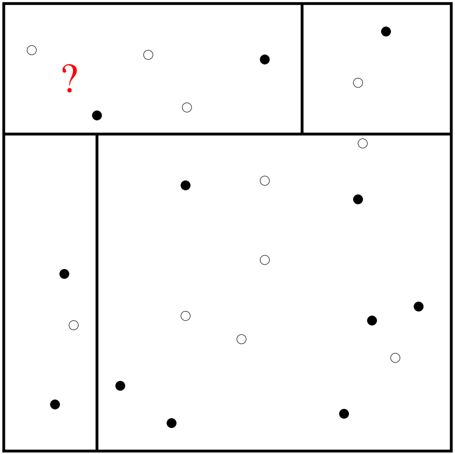


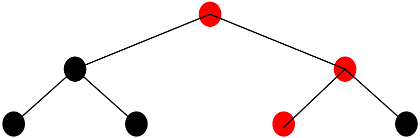


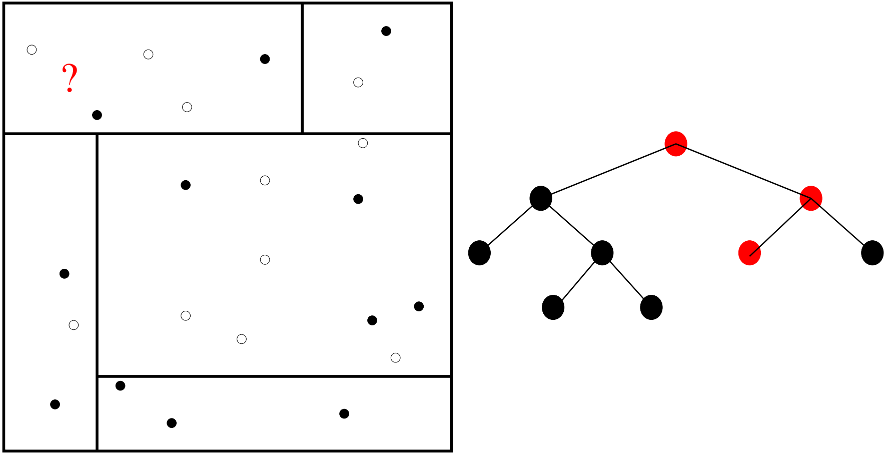
Trees



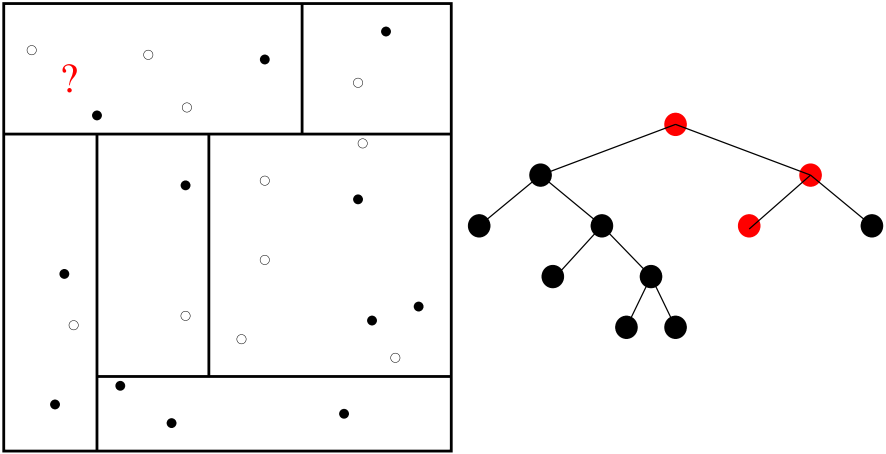




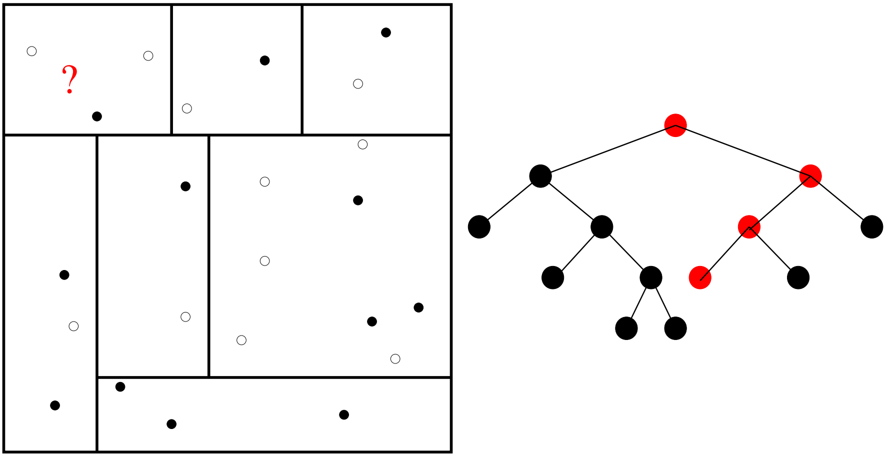




Trees



Trees



- Leo Breiman promoted random forests.
- Idea: Using tree averaging as a means of obtaining good rules.
- The base trees are simple and randomized.

Breiman's ideas were decisively influenced by

- **Amit and Geman** (1997, geometric feature selection).
- **Ho** (1998, random subspace method).
- **Dietterich** (2000, random split selection approach).

`stat.berkeley.edu/users/breiman/RandomForests`

From trees to forests

- Leo Breiman promoted random forests.
- Idea: Using tree averaging as a means of obtaining good rules.
- The base trees are simple and randomized.

Breiman's ideas were decisively influenced by

- **Amit and Geman** (1997, geometric feature selection).
- **Ho** (1998, random subspace method).
- **Dietterich** (2000, random split selection approach).

`stat.berkeley.edu/users/breiman/RandomForests`

From trees to forests

- Leo Breiman promoted random forests.
- Idea: Using tree averaging as a means of obtaining good rules.
- The base trees are simple and randomized.

Breiman's ideas were decisively influenced by

- **Amit and Geman** (1997, geometric feature selection).
- **Ho** (1998, random subspace method).
- **Dietterich** (2000, random split selection approach).

stat.berkeley.edu/users/breiman/RandomForests

From trees to forests

- Leo Breiman promoted random forests.
- Idea: Using tree averaging as a means of obtaining good rules.
- The base trees are simple and randomized.

Breiman's ideas were decisively influenced by

- **Amit and Geman** (1997, geometric feature selection).
- **Ho** (1998, random subspace method).
- **Dietterich** (2000, random split selection approach).

stat.berkeley.edu/users/breiman/RandomForests

- Leo Breiman promoted random forests.
- Idea: Using tree averaging as a means of obtaining good rules.
- The base trees are simple and randomized.

Breiman's ideas were decisively influenced by

- **Amit and Geman** (1997, geometric feature selection).
- **Ho** (1998, random subspace method).
- **Dietterich** (2000, random split selection approach).

`stat.berkeley.edu/users/breiman/RandomForests`

Random forests

- They have emerged as serious competitors to state of the art methods.
- They are fast and easy to implement, produce highly accurate predictions and can handle a very large number of input variables without overfitting.
- In fact, forests are among the most accurate general-purpose learners available.

- The algorithm is difficult to analyze and its mathematical properties remain to date largely unknown.
- Most theoretical studies have concentrated on isolated parts or stylized versions of the procedure.

Random forests

- They have emerged as serious competitors to state of the art methods.
- They are fast and easy to implement, produce highly accurate predictions and can handle a very large number of input variables without overfitting.
- In fact, forests are among the most accurate general-purpose learners available.

- The algorithm is difficult to analyze and its mathematical properties remain to date largely unknown.
- Most theoretical studies have concentrated on isolated parts or stylized versions of the procedure.

Random forests

- They have emerged as serious competitors to state of the art methods.
- They are fast and easy to implement, produce highly accurate predictions and can handle a very large number of input variables without overfitting.
- In fact, forests are among the most accurate general-purpose learners available.

- The algorithm is difficult to analyze and its mathematical properties remain to date largely unknown.
- Most theoretical studies have concentrated on isolated parts or stylized versions of the procedure.

Random forests

- They have emerged as serious competitors to state of the art methods.
- They are fast and easy to implement, produce highly accurate predictions and can handle a very large number of input variables without overfitting.
- In fact, forests are among the most accurate general-purpose learners available.

- The algorithm is difficult to analyze and its mathematical properties remain to date largely unknown.
- Most theoretical studies have concentrated on isolated parts or stylized versions of the procedure.

Random forests

- They have emerged as serious competitors to state of the art methods.
- They are fast and easy to implement, produce highly accurate predictions and can handle a very large number of input variables without overfitting.
- In fact, forests are among the most accurate general-purpose learners available.

- The algorithm is difficult to analyze and its mathematical properties remain to date largely unknown.
- Most theoretical studies have concentrated on isolated parts or stylized versions of the procedure.

Key-references

- **Breiman** (2000, 2001, 2004).
  - ▷ Definition, experiments and intuitions.
- **Lin and Jeon** (2006).
  - ▷ Link with layered nearest neighbors.
- **Biau, Devroye and Lugosi** (2008).
  - ▷ Consistency results for stylized versions.
- **Biau** (2012).
  - ▷ Sparsity and random forests.

Key-references

- **Breiman** (2000, 2001, 2004).
  - ▷ Definition, experiments and intuitions.
- **Lin and Jeon** (2006).
  - ▷ Link with layered nearest neighbors.
- **Biau, Devroye and Lugosi** (2008).
  - ▷ Consistency results for stylized versions.
- **Biau** (2012).
  - ▷ Sparsity and random forests.

Key-references

- **Breiman** (2000, 2001, 2004).
  - ▷ Definition, experiments and intuitions.
- **Lin and Jeon** (2006).
  - ▷ Link with layered nearest neighbors.
- **Biau, Devroye and Lugosi** (2008).
  - ▷ Consistency results for stylized versions.
- **Biau** (2012).
  - ▷ Sparsity and random forests.

Key-references

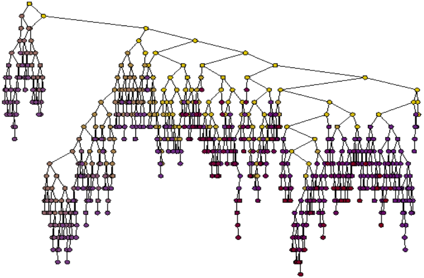
- **Breiman** (2000, 2001, 2004).
  - ▷ Definition, experiments and intuitions.
- **Lin and Jeon** (2006).
  - ▷ Link with layered nearest neighbors.
- **Biau, Devroye and Lugosi** (2008).
  - ▷ Consistency results for stylized versions.
- **Biau** (2012).
  - ▷ Sparsity and random forests.

Outline

Three basic ingredients

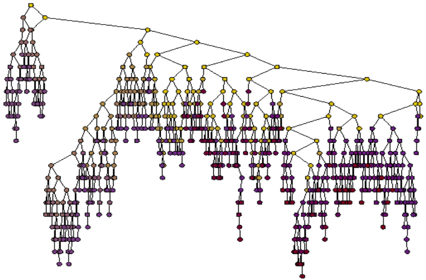
1-Randomization and no-pruning

- For each tree, select at random, at each node, a small group of input coordinates to split.
- Calculate the best split based on these features and cut.
- The tree is grown to maximum size, without pruning.



1-Randomization and no-pruning

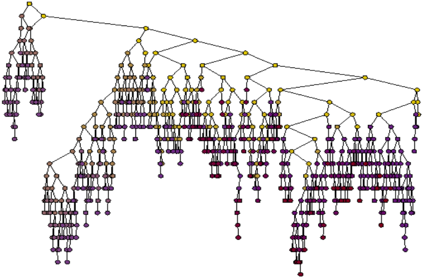
- For each tree, select at random, at each node, a small group of input coordinates to split.
- Calculate the best split based on these features and cut.
- The tree is grown to maximum size, without pruning.



Three basic ingredients

1-Randomization and no-pruning

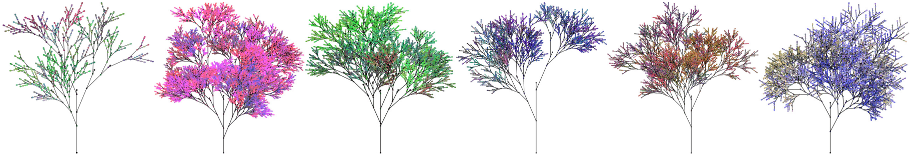
- ▷ For each tree, select at random, at each node, a small group of input coordinates to split.
- ▷ Calculate the best split based on these features and cut.
- ▷ The tree is grown to maximum size, without pruning.



Three basic ingredients

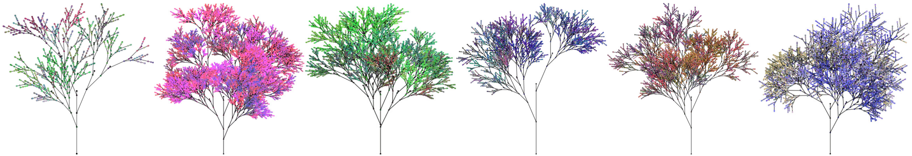
2-Aggregation

- Final predictions are obtained by aggregating over the ensemble.
- It is fast and easily parallelizable.



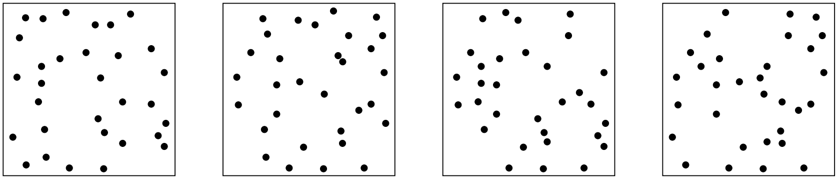
2-Aggregation

- Final predictions are obtained by aggregating over the ensemble.
- It is fast and easily parallelizable.



3-Bagging

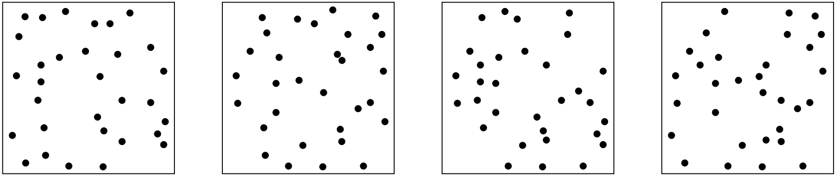
- ▷ The subspace randomization scheme is blended with bagging.
- ▷ Breiman (1996).
- ▷ Bühlmann and Yu (2002).
- ▷ Biau, Cérou and Guyader (2010).



Three basic ingredients

3-Bagging

- ▷ The subspace randomization scheme is blended with bagging.
- ▷ **Breiman** (1996).
- ▷ **Bühlmann and Yu** (2002).
- ▷ **Biau, Cérou and Guyader** (2010).



Mathematical framework

- A training sample: $\mathcal{D}_n = \{(\mathbf{X}_1, Y_1), \ldots, (\mathbf{X}_n, Y_n)\}$ i.i.d. $[0,1]^d \times \mathbb{R}$-valued random variables.
- A generic pair: $(\mathbf{X}, Y)$ satisfying $\mathbb{E}Y^2 < \infty$.
- Our mission: For fixed $\mathbf{x} \in [0,1]^d$, estimate the regression function $r(\mathbf{x}) = \mathbb{E}[Y|\mathbf{X} = \mathbf{x}]$ using the data.
- Quality criterion: $\mathbb{E}[r_n(\mathbf{X}) - r(\mathbf{X})]^2$.

Mathematical framework

- A training sample: $\mathcal{D}_n = \{(\mathbf{X}_1, Y_1), \ldots, (\mathbf{X}_n, Y_n)\}$ i.i.d. $[0,1]^d \times \mathbb{R}$-valued random variables.
- A generic pair: $(\mathbf{X}, Y)$ satisfying $\mathbb{E}Y^2 < \infty$.
- Our mission: For fixed $\mathbf{x} \in [0,1]^d$, estimate the regression function $r(\mathbf{x}) = \mathbb{E}[Y|\mathbf{X} = \mathbf{x}]$ using the data.
- Quality criterion: $\mathbb{E}[r_n(\mathbf{X}) - r(\mathbf{X})]^2$.

Mathematical framework

- A training sample: $\mathcal{D}_n = \{(\mathbf{X}_1, Y_1), \ldots, (\mathbf{X}_n, Y_n)\}$ i.i.d. $[0,1]^d \times \mathbb{R}$-valued random variables.
- A generic pair: $(\mathbf{X}, Y)$ satisfying $\mathbb{E}Y^2 < \infty$.
- Our mission: For fixed $\mathbf{x} \in [0,1]^d$, estimate the regression function $r(\mathbf{x}) = \mathbb{E}[Y|\mathbf{X} = \mathbf{x}]$ using the data.
- Quality criterion: $\mathbb{E}[r_n(\mathbf{X}) - r(\mathbf{X})]^2$.

Mathematical framework

- A training sample: $\mathcal{D}_n = \{(\mathbf{X}_1, Y_1), \ldots, (\mathbf{X}_n, Y_n)\}$ i.i.d. $[0,1]^d \times \mathbb{R}$-valued random variables.
- A generic pair: $(\mathbf{X}, Y)$ satisfying $\mathbb{E}Y^2 < \infty$.
- Our mission: For fixed $\mathbf{x} \in [0,1]^d$, estimate the regression function $r(\mathbf{x}) = \mathbb{E}[Y|\mathbf{X} = \mathbf{x}]$ using the data.
- Quality criterion: $\mathbb{E}[r_n(\mathbf{X}) - r(\mathbf{X})]^2$.

The model

- A random forest is a collection of randomized base regression trees $\{r_n(\mathbf{x}, \Theta_m, \mathcal{D}_n), m \geq 1\}$.
- These random trees are combined to form the aggregated regression estimate

$$\bar{r}_n(\mathbf{X}, \mathcal{D}_n) = \mathbb{E}_\Theta \left[r_n(\mathbf{X}, \Theta, \mathcal{D}_n) \right].$$

- Θ is assumed to be independent of $\mathbf{X}$ and the training sample $\mathcal{D}_n$.
- However, we allow Θ to be based on a test sample, independent of, but distributed as, $\mathcal{D}_n$.

The model

- A random forest is a collection of randomized base regression trees $\{r_n(\mathbf{x}, \Theta_m, \mathcal{D}_n), m \geq 1\}$.
- These random trees are combined to form the aggregated regression estimate

$$\bar{r}_n(\mathbf{X}, \mathcal{D}_n) = \mathbb{E}_{\Theta}\left[r_n(\mathbf{X}, \Theta, \mathcal{D}_n)\right].$$

- Θ is assumed to be independent of $\mathbf{X}$ and the training sample $\mathcal{D}_n$.
- However, we allow Θ to be based on a test sample, independent of, but distributed as, $\mathcal{D}_n$.

The model

- A random forest is a collection of randomized base regression trees $\{r_n(\mathbf{x}, \Theta_m, \mathcal{D}_n), m \geq 1\}$.
- These random trees are combined to form the aggregated regression estimate

$$\bar{r}_n(\mathbf{X}, \mathcal{D}_n) = \mathbb{E}_\Theta \left[r_n(\mathbf{X}, \Theta, \mathcal{D}_n) \right].$$

- Θ is assumed to be independent of $\mathbf{X}$ and the training sample $\mathcal{D}_n$.
- However, we allow Θ to be based on a test sample, independent of, but distributed as, $\mathcal{D}_n$.

The model

- A random forest is a collection of randomized base regression trees $\{r_n(\mathbf{x}, \Theta_m, \mathcal{D}_n), m \geq 1\}$.
- These random trees are combined to form the aggregated regression estimate

$$\bar{r}_n(\mathbf{X}, \mathcal{D}_n) = \mathbb{E}_{\Theta}\left[r_n(\mathbf{X}, \Theta, \mathcal{D}_n)\right].$$

- Θ is assumed to be independent of $\mathbf{X}$ and the training sample $\mathcal{D}_n$.
- However, we allow Θ to be based on a test sample, independent of, but distributed as, $\mathcal{D}_n$.

The procedure

- Fix $k_n \geq 2$ and repeat the following procedure $\lceil \log_2 k_n \rceil$ times:

1. At each node, a coordinate of $\mathbf{X} = (X^{(1)}, \ldots, X^{(d)})$ is selected, with the j-th feature having a probability $p_{nj} \in (0, 1)$ of being selected.
2. At each node, once the coordinate is selected, the split is at the midpoint of the chosen side.

- Thus

$$\bar{r}_n(\mathbf{X}) = \mathbb{E}_\Theta \left[\frac{\sum_{i=1}^n Y_i \mathbf{1}_{[\mathbf{X}_i \in A_n(\mathbf{X},\Theta)]}}{\sum_{i=1}^n \mathbf{1}_{[\mathbf{X}_i \in A_n(\mathbf{X},\Theta)]}} \mathbf{1}_{\mathcal{E}_n(\mathbf{X},\Theta)} \right],$$

where

$$\mathcal{E}_n(\mathbf{X}, \Theta) = \left[\sum_{i=1}^n \mathbf{1}_{[\mathbf{X}_i \in A_n(\mathbf{X},\Theta)]} \neq 0 \right].$$

The procedure

▷ Fix $k_n \geq 2$ and repeat the following procedure $\lceil \log_2 k_n \rceil$ times:

1. At each node, a coordinate of $\mathbf{X} = (X^{(1)}, \ldots, X^{(d)})$ is selected, with the j-th feature having a probability $p_{nj} \in (0, 1)$ of being selected.

2. At each node, once the coordinate is selected, the split is at the midpoint of the chosen side.

▷ Thus

$$\bar{r}_n(\mathbf{X}) = \mathbb{E}_\Theta \left[\frac{\sum_{i=1}^n Y_i \mathbf{1}_{[\mathbf{X}_i \in A_n(\mathbf{X},\Theta)]}}{\sum_{i=1}^n \mathbf{1}_{[\mathbf{X}_i \in A_n(\mathbf{X},\Theta)]}} \mathbf{1}_{\mathcal{E}_n(\mathbf{X},\Theta)} \right],$$

where

$$\mathcal{E}_n(\mathbf{X}, \Theta) = \left[\sum_{i=1}^n \mathbf{1}_{[\mathbf{X}_i \in A_n(\mathbf{X},\Theta)]} \neq 0 \right].$$

The procedure

- Fix $k_n \geq 2$ and repeat the following procedure $\lceil \log_2 k_n \rceil$ times:

1. At each node, a coordinate of $\mathbf{X} = (X^{(1)}, \ldots, X^{(d)})$ is selected, with the j-th feature having a probability $p_{nj} \in (0,1)$ of being selected.
2. At each node, once the coordinate is selected, the split is at the midpoint of the chosen side.

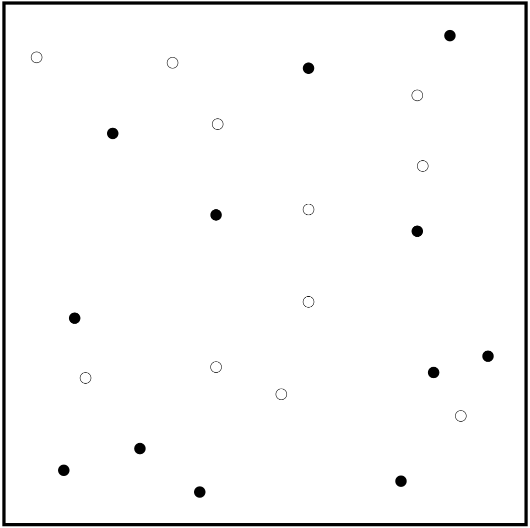
- Thus

$$\bar{r}_n(\mathbf{X}) = \mathbb{E}_\Theta \left[\frac{\sum_{i=1}^n Y_i \mathbf{1}_{[\mathbf{X}_i \in A_n(\mathbf{X},\Theta)]}}{\sum_{i=1}^n \mathbf{1}_{[\mathbf{X}_i \in A_n(\mathbf{X},\Theta)]}} \mathbf{1}_{\mathcal{E}_n(\mathbf{X},\Theta)} \right],$$

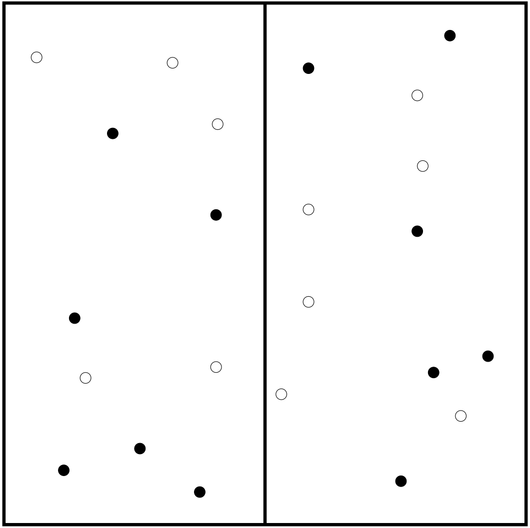
where

$$\mathcal{E}_n(\mathbf{X},\Theta) = \left[\sum_{i=1}^n \mathbf{1}_{[\mathbf{X}_i \in A_n(\mathbf{X},\Theta)]} \neq 0 \right].$$

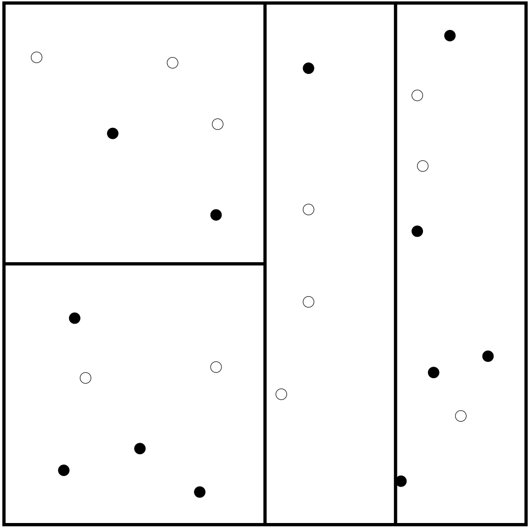
Binary trees



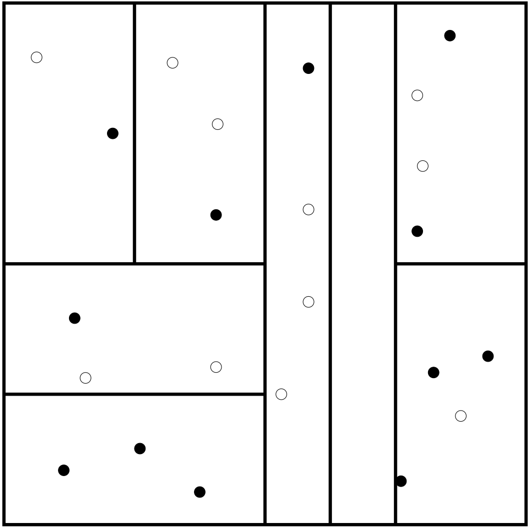
Binary trees



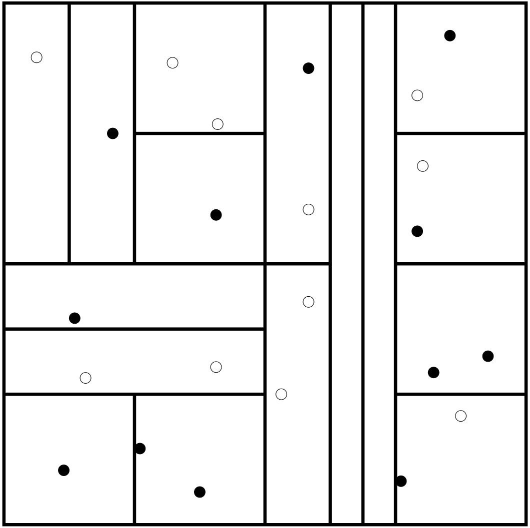
Binary trees



Binary trees



Binary trees



General comments

- Each individual tree has exactly $2^{\lceil \log_2 k_n \rceil}$ ($\approx k_n$) terminal nodes, and each leaf has Lebesgue measure $2^{-\lceil \log_2 k_n \rceil}$ ($\approx 1/k_n$).
- If $\mathbf{X}$ has uniform distribution on $[0, 1]^d$, there will be on average about n/k_n observations per terminal node.
- The choice $k_n = n$ induces a very small number of cases in the final leaves.

This scheme is close to what the original random forests algorithm does.

General comments

- Each individual tree has exactly $2^{\lceil \log_2 k_n \rceil}$ ($\approx k_n$) terminal nodes, and each leaf has Lebesgue measure $2^{-\lceil \log_2 k_n \rceil}$ ($\approx 1/k_n$).
- If $\mathbf{X}$ has uniform distribution on $[0,1]^d$, there will be on average about n/k_n observations per terminal node.
- The choice $k_n = n$ induces a very small number of cases in the final leaves.

This scheme is close to what the original random forests algorithm does.

General comments

- Each individual tree has exactly $2^{\lceil \log_2 k_n \rceil}$ ($\approx k_n$) terminal nodes, and each leaf has Lebesgue measure $2^{-\lceil \log_2 k_n \rceil}$ ($\approx 1/k_n$).
- If $\mathbf{X}$ has uniform distribution on $[0,1]^d$, there will be on average about n/k_n observations per terminal node.
- The choice $k_n = n$ induces a very small number of cases in the final leaves.

This scheme is close to what the original random forests algorithm does.

General comments

- Each individual tree has exactly $2^{\lceil \log_2 k_n \rceil}$ ($\approx k_n$) terminal nodes, and each leaf has Lebesgue measure $2^{-\lceil \log_2 k_n \rceil}$ ($\approx 1/k_n$).
- If $\mathbf{X}$ has uniform distribution on $[0, 1]^d$, there will be on average about n/k_n observations per terminal node.
- The choice $k_n = n$ induces a very small number of cases in the final leaves.

This scheme is close to what the original random forests algorithm does.

Consistency

Theorem

Assume that the distribution of $\mathbf{X}$ has support on $[0,1]^d$. Then the random forests estimate $\bar{r}_n$ is consistent whenever $p_{nj} \log k_n \to \infty$ for all $j = 1, \ldots, d$ and $k_n/n \to 0$ as $n \to \infty$.

- In the purely random model, $p_{nj} = 1/d$, independently of n and j, and consistency is ensured as long as $k_n \to \infty$ and $k_n/n \to 0$.
- This is however a radically simplified version of the random forests used in practice.
- A more in-depth analysis is needed.

Consistency

Theorem

Assume that the distribution of $\mathbf{X}$ has support on $[0,1]^d$. Then the random forests estimate $\bar{r}_n$ is consistent whenever $p_{nj} \log k_n \to \infty$ for all $j = 1, \ldots, d$ and $k_n/n \to 0$ as $n \to \infty$.

- In the purely random model, $p_{nj} = 1/d$, independently of n and j, and consistency is ensured as long as $k_n \to \infty$ and $k_n/n \to 0$.
- This is however a radically simplified version of the random forests used in practice.
- A more in-depth analysis is needed.

Consistency

Theorem

Assume that the distribution of $\mathbf{X}$ has support on $[0,1]^d$. Then the random forests estimate $\bar{r}_n$ is consistent whenever $p_{nj} \log k_n \to \infty$ for all $j = 1, \ldots, d$ and $k_n/n \to 0$ as $n \to \infty$.

- In the purely random model, $p_{nj} = 1/d$, independently of n and j, and consistency is ensured as long as $k_n \to \infty$ and $k_n/n \to 0$.
- This is however a radically simplified version of the random forests used in practice.
- A more in-depth analysis is needed.

Consistency

Theorem

Assume that the distribution of $\mathbf{X}$ has support on $[0,1]^d$. Then the random forests estimate $\bar{r}_n$ is consistent whenever $p_{nj} \log k_n \to \infty$ for all $j = 1, \ldots, d$ and $k_n/n \to 0$ as $n \to \infty$.

- In the purely random model, $p_{nj} = 1/d$, independently of n and j, and consistency is ensured as long as $k_n \to \infty$ and $k_n/n \to 0$.
- This is however a radically simplified version of the random forests used in practice.
- A more in-depth analysis is needed.

Sparsity

- There is empirical evidence that many signals in high-dimensional spaces admit a sparse representation.
  - Images wavelet coefficients.
  - High-throughput technologies.
- Sparse estimation is playing an increasingly important role in the statistics and machine learning communities.
- Several methods have recently been developed in both fields, which rely upon the notion of sparsity.

Sparsity

- There is empirical evidence that many signals in high-dimensional spaces admit a sparse representation.
  - ▷ Images wavelet coefficients.
  - ▷ High-throughput technologies.
- Sparse estimation is playing an increasingly important role in the statistics and machine learning communities.
- Several methods have recently been developed in both fields, which rely upon the notion of sparsity.

Sparsity

- There is empirical evidence that many signals in high-dimensional spaces admit a sparse representation.
  - ▷ Images wavelet coefficients.
  - ▷ High-throughput technologies.
- Sparse estimation is playing an increasingly important role in the statistics and machine learning communities.
- Several methods have recently been developed in both fields, which rely upon the notion of sparsity.

Sparsity

- There is empirical evidence that many signals in high-dimensional spaces admit a sparse representation.
  - Images wavelet coefficients.
  - High-throughput technologies.
- Sparse estimation is playing an increasingly important role in the statistics and machine learning communities.
- Several methods have recently been developed in both fields, which rely upon the notion of sparsity.

Sparsity

- There is empirical evidence that many signals in high-dimensional spaces admit a sparse representation.
  - ▷ Images wavelet coefficients.
  - ▷ High-throughput technologies.
- Sparse estimation is playing an increasingly important role in the statistics and machine learning communities.
- Several methods have recently been developed in both fields, which rely upon the notion of sparsity.

Our vision

- The regression function $r(\mathbf{X}) = \mathbb{E}[Y|\mathbf{X}]$ depends in fact only on a nonempty subset $\mathcal{S}$ (for $\mathcal{S}$trong) of the d features.
- In other words, letting $\mathbf{X}_{\mathcal{S}} = (X_j : j \in \mathcal{S})$ and $S = \text{Card}\,\mathcal{S}$, we have
$$r(\mathbf{X}) = \mathbb{E}[Y|\mathbf{X}_{\mathcal{S}}].$$
- In the dimension reduction scenario we have in mind, the ambient dimension d can be very large, much larger than n.
- As such, the value S characterizes the sparsity of the model: The smaller S, the sparser r.

Our vision

- The regression function $r(\mathbf{X}) = \mathbb{E}[Y|\mathbf{X}]$ depends in fact only on a nonempty subset $\mathcal{S}$ (for $\mathcal{S}$trong) of the d features.
- In other words, letting $\mathbf{X}_{\mathcal{S}} = (X_j : j \in \mathcal{S})$ and $S = \text{Card}\, \mathcal{S}$, we have

$$r(\mathbf{X}) = \mathbb{E}[Y|\mathbf{X}_{\mathcal{S}}].$$

- In the dimension reduction scenario we have in mind, the ambient dimension d can be very large, much larger than n.
- As such, the value S characterizes the sparsity of the model: The smaller S, the sparser r.

Our vision

- The regression function $r(\mathbf{X}) = \mathbb{E}[Y|\mathbf{X}]$ depends in fact only on a nonempty subset $\mathcal{S}$ (for $\mathcal{S}$trong) of the d features.
- In other words, letting $\mathbf{X}_{\mathcal{S}} = (X_j : j \in \mathcal{S})$ and $S = \text{Card}\,\mathcal{S}$, we have

$$r(\mathbf{X}) = \mathbb{E}[Y|\mathbf{X}_{\mathcal{S}}].$$

- In the dimension reduction scenario we have in mind, the ambient dimension d can be very large, much larger than n.
- As such, the value S characterizes the sparsity of the model: The smaller S, the sparser r.

Our vision

- The regression function $r(\mathbf{X}) = \mathbb{E}[Y|\mathbf{X}]$ depends in fact only on a nonempty subset $\mathcal{S}$ (for $\mathcal{S}$trong) of the d features.
- In other words, letting $\mathbf{X}_{\mathcal{S}} = (X_j : j \in \mathcal{S})$ and $S = \text{Card}\,\mathcal{S}$, we have

$$r(\mathbf{X}) = \mathbb{E}[Y|\mathbf{X}_{\mathcal{S}}].$$

- In the dimension reduction scenario we have in mind, the ambient dimension d can be very large, much larger than n.
- As such, the value S characterizes the sparsity of the model: The smaller S, the sparser r.

- Ideally, $p_{nj} = 1/S$ for $j \in \mathcal{S}$.
- To stick to reality, we will rather require that $p_{nj} = (1/S)(1 + \xi_{nj})$.
- Such a randomization mechanism may be designed on the basis of a test sample.

Action plan

$$\mathbb{E}\left[\bar{r}_n(\mathbf{X}) - r(\mathbf{X})\right]^2 = \mathbb{E}\left[\bar{r}_n(\mathbf{X}) - \tilde{r}_n(\mathbf{X})\right]^2 + \mathbb{E}\left[\tilde{r}_n(\mathbf{X}) - r(\mathbf{X})\right]^2,$$

where

$$\tilde{r}_n(\mathbf{X}) = \sum_{i=1}^{n} \mathbb{E}_\Theta\left[W_{ni}(\mathbf{X}, \Theta)\right] r(\mathbf{X}_i).$$

Sparsity and random forests

- Ideally, $p_{nj} = 1/S$ for $j \in \mathcal{S}$.
- To stick to reality, we will rather require that $p_{nj} = (1/S)(1 + \xi_{nj})$.
- Such a randomization mechanism may be designed on the basis of a test sample.

Action plan

$$\mathbb{E}\left[\bar{r}_n(\mathbf{X}) - r(\mathbf{X})\right]^2 = \mathbb{E}\left[\bar{r}_n(\mathbf{X}) - \tilde{r}_n(\mathbf{X})\right]^2 + \mathbb{E}\left[\tilde{r}_n(\mathbf{X}) - r(\mathbf{X})\right]^2,$$

where

$$\tilde{r}_n(\mathbf{X}) = \sum_{i=1}^{n} \mathbb{E}_\Theta\left[W_{ni}(\mathbf{X}, \Theta)\right] r(\mathbf{X}_i).$$

Sparsity and random forests

- Ideally, $p_{nj} = 1/S$ for $j \in \mathcal{S}$.
- To stick to reality, we will rather require that $p_{nj} = (1/S)(1 + \xi_{nj})$.
- Such a randomization mechanism may be designed on the basis of a test sample.

Action plan

$$\mathbb{E}\left[\bar{r}_n(\mathbf{X}) - r(\mathbf{X})\right]^2 = \mathbb{E}\left[\bar{r}_n(\mathbf{X}) - \tilde{r}_n(\mathbf{X})\right]^2 + \mathbb{E}\left[\tilde{r}_n(\mathbf{X}) - r(\mathbf{X})\right]^2,$$

where

$$\tilde{r}_n(\mathbf{X}) = \sum_{i=1}^{n} \mathbb{E}_\Theta\left[W_{ni}(\mathbf{X}, \Theta)\right] r(\mathbf{X}_i).$$

- Ideally, $p_{nj} = 1/S$ for $j \in \mathcal{S}$.
- To stick to reality, we will rather require that $p_{nj} = (1/S)(1+\xi_{nj})$.
- Such a randomization mechanism may be designed on the basis of a test sample.

Action plan

$$\mathbb{E}\left[\bar{r}_n(\mathbf{X}) - r(\mathbf{X})\right]^2 = \mathbb{E}\left[\bar{r}_n(\mathbf{X}) - \tilde{r}_n(\mathbf{X})\right]^2 + \mathbb{E}\left[\tilde{r}_n(\mathbf{X}) - r(\mathbf{X})\right]^2,$$

where

$$\tilde{r}_n(\mathbf{X}) = \sum_{i=1}^{n} \mathbb{E}_\Theta\left[W_{ni}(\mathbf{X}, \Theta)\right] r(\mathbf{X}_i).$$

Proposition

Assume that $\mathbf{X}$ is uniformly distributed on $[0,1]^d$ and, for all $\mathbf{x} \in \mathbb{R}^d$,

$$\sigma^2(\mathbf{x}) = \mathbb{V}[Y \,|\, \mathbf{X} = \mathbf{x}] \leq \sigma^2$$

for some positive constant σ^2. Then, if $p_{nj} = (1/S)(1 + \xi_{nj})$ for $j \in \mathcal{S}$,

$$\mathbb{E}\left[\bar{r}_n(\mathbf{X}) - \tilde{r}_n(\mathbf{X})\right]^2 \leq C\sigma^2 \left(\frac{S^2}{S-1}\right)^{S/2d} (1 + \xi_n) \frac{k_n}{n(\log k_n)^{S/2d}},$$

where

$$C = \frac{288}{\pi}\left(\frac{\pi \log 2}{16}\right)^{S/2d}.$$

Bias

Proposition

Assume that $\mathbf{X}$ is uniformly distributed on $[0,1]^d$ and r is L-Lipschitz. Then, if $p_{nj} = (1/S)(1+\xi_{nj})$ for $j \in \mathcal{S}$,

$$\mathbb{E}\left[\tilde{r}_n(\mathbf{X}) - r(\mathbf{X})\right]^2 \leq \frac{2SL^2}{k_n^{\frac{0.75}{S\log 2}(1+\gamma_n)}} + \left[\sup_{\mathbf{x}\in[0,1]^d} r^2(\mathbf{x})\right] e^{-n/2k_n}.$$

- The rate at which the bias decreases to 0 depends on the number of strong variables, not on d.
- $k_n^{-(0.75/(S\log 2))(1+\gamma_n)} = \mathrm{o}(k_n^{-2/d})$ as soon as $S \leq \lfloor 0.54d \rfloor$.
- The term $e^{-n/2k_n}$ prevents the extreme choice $k_n = n$.

Bias

Proposition

Assume that $\mathbf{X}$ is uniformly distributed on $[0,1]^d$ and r is L-Lipschitz. Then, if $p_{nj} = (1/S)(1+\xi_{nj})$ for $j \in \mathcal{S}$,

$$\mathbb{E}\left[\tilde{r}_n(\mathbf{X}) - r(\mathbf{X})\right]^2 \le \frac{2SL^2}{k_n^{\frac{0.75}{S\log 2}(1+\gamma_n)}} + \left[\sup_{\mathbf{x}\in[0,1]^d} r^2(\mathbf{x})\right] e^{-n/2k_n}.$$

- The rate at which the bias decreases to 0 depends on the number of strong variables, not on d.
- $k_n^{-(0.75/(S\log 2))(1+\gamma_n)} = \mathrm{o}(k_n^{-2/d})$ as soon as $S \le \lfloor 0.54d \rfloor$.
- The term $e^{-n/2k_n}$ prevents the extreme choice $k_n = n$.

Bias

Proposition

Assume that $\mathbf{X}$ is uniformly distributed on $[0,1]^d$ and r is L-Lipschitz. Then, if $p_{nj} = (1/S)(1+\xi_{nj})$ for $j \in \mathcal{S}$,

$$\mathbb{E}\left[\tilde{r}_n(\mathbf{X}) - r(\mathbf{X})\right]^2 \leq \frac{2SL^2}{k_n^{\frac{0.75}{S\log 2}(1+\gamma_n)}} + \left[\sup_{\mathbf{x}\in[0,1]^d} r^2(\mathbf{x})\right] e^{-n/2k_n}.$$

- The rate at which the bias decreases to 0 depends on the number of strong variables, not on d.
- $k_n^{-(0.75/(S\log 2))(1+\gamma_n)} = \mathrm{o}(k_n^{-2/d})$ as soon as $S \leq \lfloor 0.54d \rfloor$.
- The term $e^{-n/2k_n}$ prevents the extreme choice $k_n = n$.

Bias

Proposition

Assume that $\mathbf{X}$ is uniformly distributed on $[0,1]^d$ and r is L-Lipschitz. Then, if $p_{nj} = (1/S)(1+\xi_{nj})$ for $j \in \mathcal{S}$,

$$\mathbb{E}\left[\tilde{r}_n(\mathbf{X}) - r(\mathbf{X})\right]^2 \leq \frac{2SL^2}{k_n^{\frac{0.75}{S\log 2}(1+\gamma_n)}} + \left[\sup_{\mathbf{x}\in[0,1]^d} r^2(\mathbf{x})\right] e^{-n/2k_n}.$$

- The rate at which the bias decreases to 0 depends on the number of strong variables, not on d.
- ${k_n}^{-(0.75/(S\log 2))(1+\gamma_n)} = \mathrm{o}({k_n}^{-2/d})$ as soon as $S \leq \lfloor 0.54d \rfloor$.
- The term $e^{-n/2k_n}$ prevents the extreme choice $k_n = n$.

Theorem

If $p_{nj} = (1/S)(1 + \xi_{nj})$ for $j \in \mathcal{S}$, with $\xi_{nj} \log n \to 0$ as $n \to \infty$, then for the choice

$$k_n \propto \left(\frac{L^2}{\Xi}\right)^{1/(1+\frac{0.75}{S \log 2})} n^{1/(1+\frac{0.75}{S \log 2})},$$

we have

$$\limsup_{n\to\infty} \sup_{(\mathbf{X},Y)\in\mathcal{F}_S} \frac{\mathbb{E}\left[\bar{r}_n(\mathbf{X}) - r(\mathbf{X})\right]^2}{\left(\Xi L^{\frac{2S\log 2}{0.75}}\right)^{\frac{0.75}{S\log 2+0.75}} n^{\frac{-0.75}{S\log 2+0.75}}} \leq \Lambda.$$

Take-home message

The rate $n^{\frac{-0.75}{S\log 2+0.75}}$ is strictly faster than the usual minimax rate $n^{-2/(d+2)}$ as soon as $S \leq \lfloor 0.54d \rfloor$.

Main result

Theorem

If $p_{nj} = (1/S)(1 + \xi_{nj})$ for $j \in \mathcal{S}$, with $\xi_{nj} \log n \to 0$ as $n \to \infty$, then for the choice

$$k_n \propto \left(\frac{L^2}{\Xi}\right)^{1/(1+\frac{0.75}{S \log 2})} n^{1/(1+\frac{0.75}{S \log 2})},$$

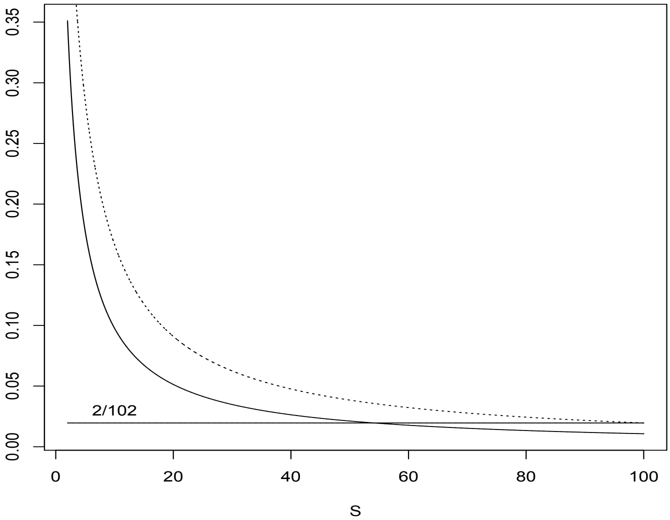
we have

$$\limsup_{n\to\infty} \sup_{(\mathbf{X},Y)\in\mathcal{F}_S} \frac{\mathbb{E}\left[\bar{r}_n(\mathbf{X}) - r(\mathbf{X})\right]^2}{\left(\Xi L^{\frac{2S\log 2}{0.75}}\right)^{\frac{0.75}{S\log 2+0.75}} n^{\frac{-0.75}{S\log 2+0.75}}} \leq \Lambda.$$

Take-home message

The rate $n^{\frac{-0.75}{S\log 2+0.75}}$ is strictly faster than the usual minimax rate $n^{-2/(d+2)}$ as soon as $S \leq \lfloor 0.54d \rfloor$.

Dimension reduction



Discussion — Bagging

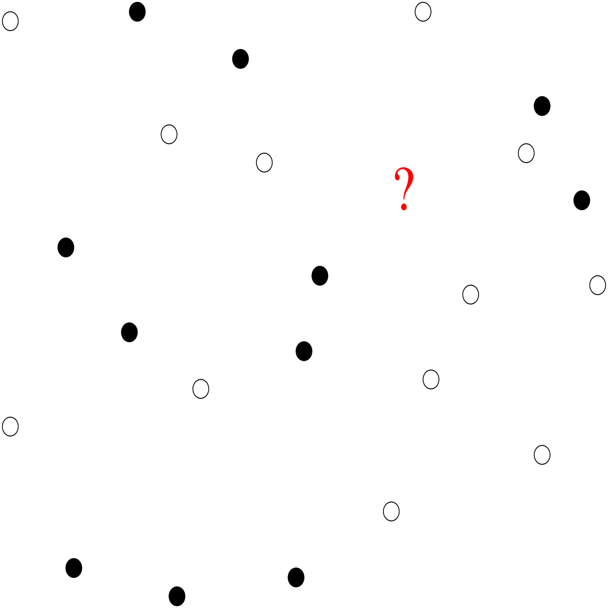
- The optimal parameter k_n depends on the unknown distribution of $(\mathbf{X}, Y)$.
- To correct this situation, adaptive choices of k_n should preserve the rate of convergence of the estimate.
- Another route we may follow is to analyze the effect of bagging.
- Biau, Cérou and Guyader (2010).

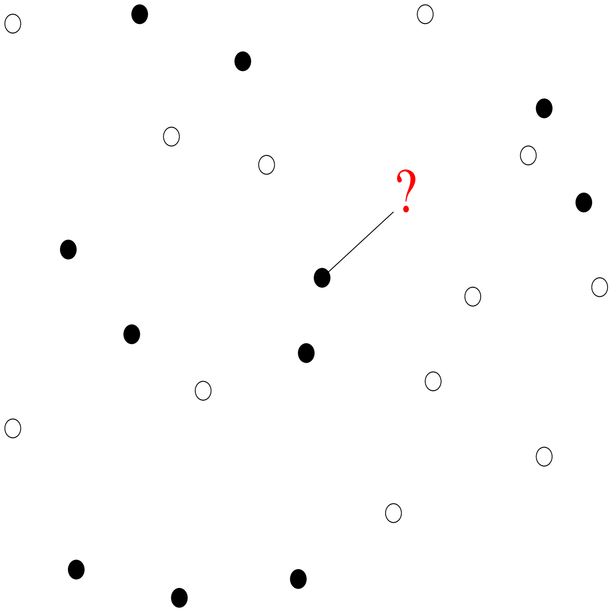
- The optimal parameter k_n depends on the unknown distribution of $(\mathbf{X}, Y)$.
- To correct this situation, adaptive choices of k_n should preserve the rate of convergence of the estimate.
- Another route we may follow is to analyze the effect of bagging.
- Biau, Cérou and Guyader (2010).

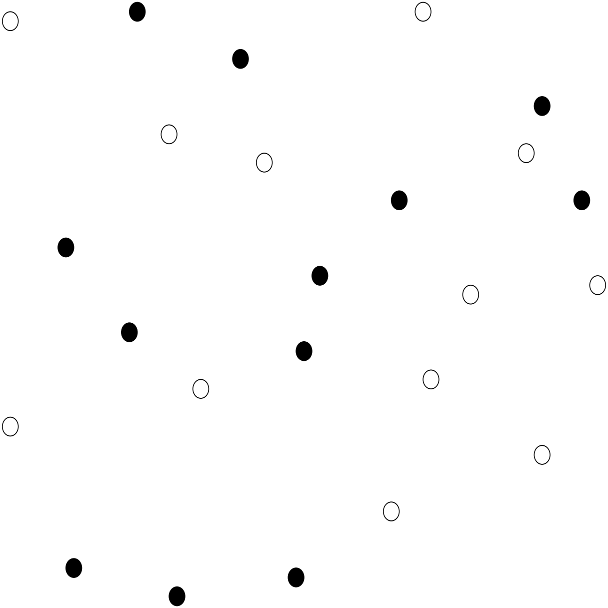
Discussion — Bagging

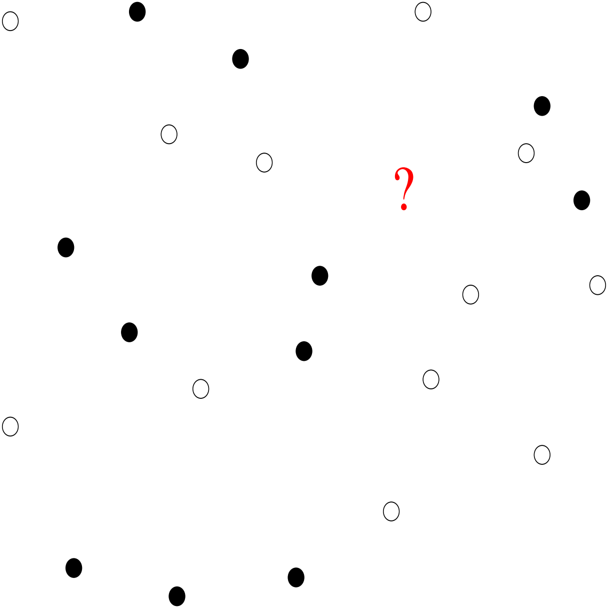
- The optimal parameter k_n depends on the unknown distribution of $(\mathbf{X}, Y)$.
- To correct this situation, adaptive choices of k_n should preserve the rate of convergence of the estimate.
- Another route we may follow is to analyze the effect of bagging.
- Biau, Cérou and Guyader (2010).

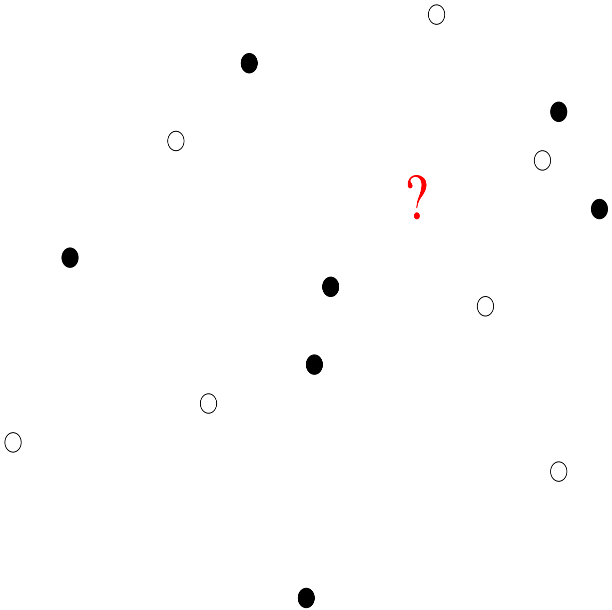
- The optimal parameter k_n depends on the unknown distribution of $(\mathbf{X}, Y)$.
- To correct this situation, adaptive choices of k_n should preserve the rate of convergence of the estimate.
- Another route we may follow is to analyze the effect of bagging.
- **Biau, Cérou and Guyader** (2010).

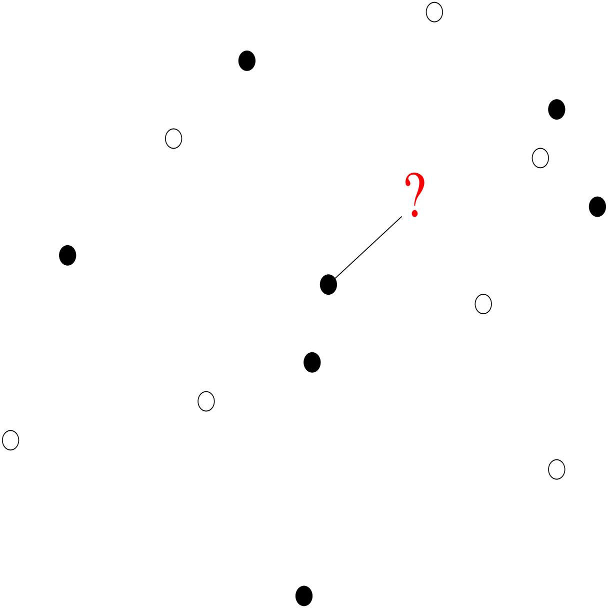


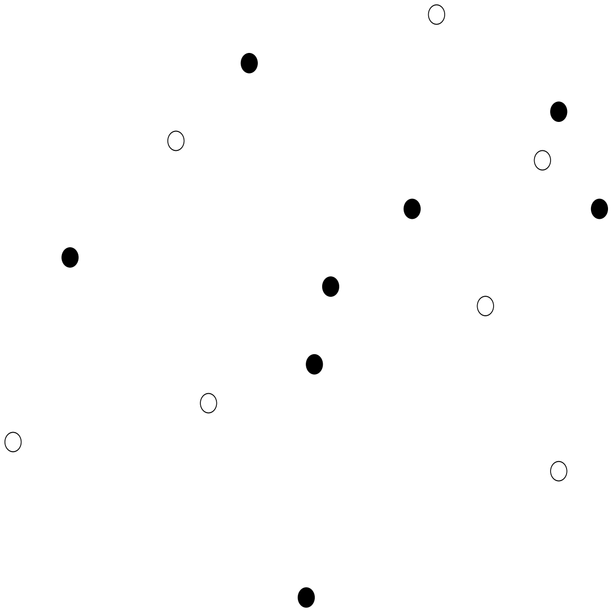


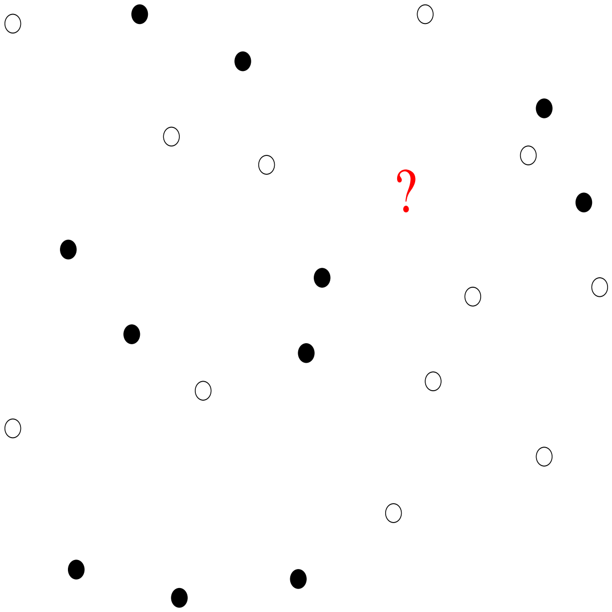


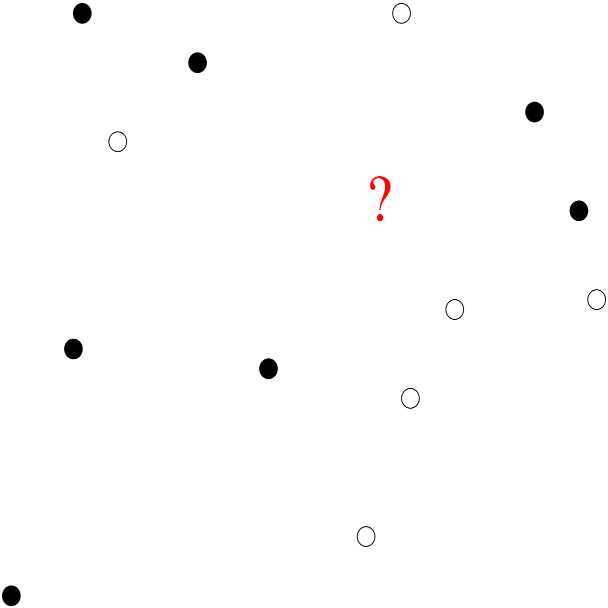


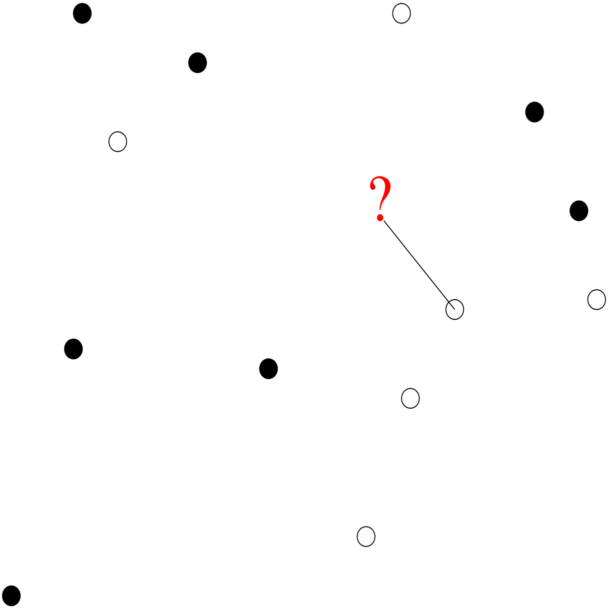


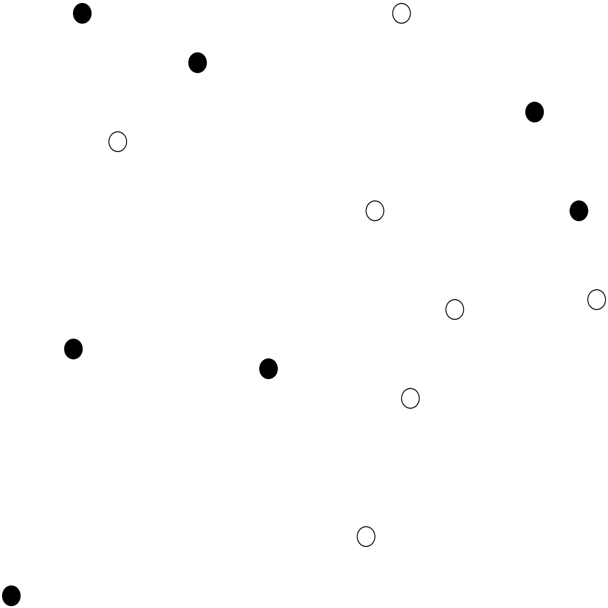












Imaginary scenario

The following splitting scheme is iteratively repeated at each node:

1. Select at random M_n candidate coordinates to split on.
2. If the selection is all weak, then choose one at random to split on.
3. If there is more than one strong variable elected, choose one at random and cut.

Discussion — Choosing the p_{nj}'s

Imaginary scenario

The following splitting scheme is iteratively repeated at each node:

1. Select at random M_n candidate coordinates to split on.
2. If the selection is all weak, then choose one at random to split on.
3. If there is more than one strong variable elected, choose one at random and cut.

Imaginary scenario

The following splitting scheme is iteratively repeated at each node:

1. Select at random M_n candidate coordinates to split on.
2. If the selection is all weak, then choose one at random to split on.
3. If there is more than one strong variable elected, choose one at random and cut.

Discussion — Choosing the p_{nj}'s

Imaginary scenario

The following splitting scheme is iteratively repeated at each node:

1. Select at random M_n candidate coordinates to split on.
2. If the selection is all weak, then choose one at random to split on.
3. If there is more than one strong variable elected, choose one at random and cut.

Discussion — Choosing the p_{nj}'s

- Each coordinate in $\mathcal{S}$ will be cut with the "ideal" probability

$$p_n^\star = \frac{1}{S}\left[1 - \left(1 - \frac{S}{d}\right)^{M_n}\right].$$

- The parameter M_n should satisfy

$$\left(1 - \frac{S}{d}\right)^{M_n} \log n \to 0 \quad \text{as } n \to \infty.$$

- This is true as soon as

$$M_n \to \infty \quad \text{and} \quad \frac{M_n}{\log n} \to \infty \quad \text{as } n \to \infty.$$

Discussion — Choosing the p_{nj}'s

- Each coordinate in $\mathcal{S}$ will be cut with the "ideal" probability

$$p_n^\star = \frac{1}{S}\left[1 - \left(1 - \frac{S}{d}\right)^{M_n}\right].$$

- The parameter M_n should satisfy

$$\left(1 - \frac{S}{d}\right)^{M_n} \log n \to 0 \quad \text{as } n \to \infty.$$

- This is true as soon as

$$M_n \to \infty \quad \text{and} \quad \frac{M_n}{\log n} \to \infty \quad \text{as } n \to \infty.$$

Discussion — Choosing the p_{nj}'s

- Each coordinate in $\mathcal{S}$ will be cut with the "ideal" probability

$$p_n^\star = \frac{1}{S}\left[1 - \left(1 - \frac{S}{d}\right)^{M_n}\right].$$

- The parameter M_n should satisfy

$$\left(1 - \frac{S}{d}\right)^{M_n} \log n \to 0 \quad \text{as } n \to \infty.$$

- This is true as soon as

$$M_n \to \infty \quad \text{and} \quad \frac{M_n}{\log n} \to \infty \quad \text{as } n \to \infty.$$

Assumptions

- We have at hand an independent test set $\mathcal{D}'_n$.
- The model is linear:

$$Y = \sum_{j \in \mathcal{S}} a_j X^{(j)} + \varepsilon.$$

- For a fixed node $A = \prod_{j=1}^{d} A_j$, fix a coordinate j and look at the weighted conditional variance $\mathbb{V}[Y | X^{(j)} \in A_j] \, \mathbb{P}(X^{(j)} \in A_j)$.
- If $j \in \mathcal{S}$, then the best split is at the midpoint of the node, with a variance decrease equal to $a_j^2/16 > 0$.
- If $j \in \mathcal{W}$, the decrease of the variance is always 0, whatever the location of the split.

Assumptions

- We have at hand an independent test set $\mathcal{D}'_n$.
- The model is linear:

$$Y = \sum_{j \in \mathcal{S}} a_j X^{(j)} + \varepsilon.$$

- For a fixed node $A = \prod_{j=1}^{d} A_j$, fix a coordinate j and look at the weighted conditional variance $\mathbb{V}[Y | X^{(j)} \in A_j] \, \mathbb{P}(X^{(j)} \in A_j)$.
- If $j \in \mathcal{S}$, then the best split is at the midpoint of the node, with a variance decrease equal to $a_j^2/16 > 0$.
- If $j \in \mathcal{W}$, the decrease of the variance is always 0, whatever the location of the split.

Assumptions

- We have at hand an independent test set $\mathcal{D}'_n$.
- The model is linear:

$$Y = \sum_{j \in \mathcal{S}} a_j X^{(j)} + \varepsilon.$$

- For a fixed node $A = \prod_{j=1}^{d} A_j$, fix a coordinate j and look at the weighted conditional variance $\mathbb{V}[Y|X^{(j)} \in A_j]\,\mathbb{P}(X^{(j)} \in A_j)$.
- If $j \in \mathcal{S}$, then the best split is at the midpoint of the node, with a variance decrease equal to $a_j^2/16 > 0$.
- If $j \in \mathcal{W}$, the decrease of the variance is always 0, whatever the location of the split.

Assumptions

- We have at hand an independent test set $\mathcal{D}'_n$.
- The model is linear:

$$Y = \sum_{j \in \mathcal{S}} a_j X^{(j)} + \varepsilon.$$

- For a fixed node $A = \prod_{j=1}^{d} A_j$, fix a coordinate j and look at the weighted conditional variance $\mathbb{V}[Y | X^{(j)} \in A_j] \, \mathbb{P}(X^{(j)} \in A_j)$.
- If $j \in \mathcal{S}$, then the best split is at the midpoint of the node, with a variance decrease equal to $a_j^2/16 > 0$.
- If $j \in \mathcal{W}$, the decrease of the variance is always 0, whatever the location of the split.

Assumptions

- We have at hand an independent test set $\mathcal{D}'_n$.
- The model is linear:

$$Y = \sum_{j \in \mathcal{S}} a_j X^{(j)} + \varepsilon.$$

- For a fixed node $A = \prod_{j=1}^{d} A_j$, fix a coordinate j and look at the weighted conditional variance $\mathbb{V}[Y | X^{(j)} \in A_j] \, \mathbb{P}(X^{(j)} \in A_j)$.
- If $j \in \mathcal{S}$, then the best split is at the midpoint of the node, with a variance decrease equal to $a_j^2/16 > 0$.
- If $j \in \mathcal{W}$, the decrease of the variance is always 0, whatever the location of the split.

Assumptions

- We have at hand an independent test set $\mathcal{D}'_n$.
- The model is linear:

$$Y = \sum_{j \in \mathcal{S}} a_j X^{(j)} + \varepsilon.$$

- For a fixed node $A = \prod_{j=1}^{d} A_j$, fix a coordinate j and look at the weighted conditional variance $\mathbb{V}[Y | X^{(j)} \in A_j] \, \mathbb{P}(X^{(j)} \in A_j)$.
- If $j \in \mathcal{S}$, then the best split is at the midpoint of the node, with a variance decrease equal to $a_j^2/16 > 0$.
- If $j \in \mathcal{W}$, the decrease of the variance is always 0, whatever the location of the split.

Near-reality scenario

The following splitting scheme is iteratively repeated at each node:

1. Select at random M_n candidate coordinates to split on.
2. For each of the M_n elected coordinates, calculate the best split.
3. Select the coordinate which outputs the best within-node sum of squares decrease, and cut.

Conclusion

For $j \in \mathcal{S}$,

$$p_{nj} \approx \frac{1}{S}\left(1 + \xi_{nj}\right),$$

where $\xi_{nj} \to 0$ and satisfies the constraint $\xi_{nj} \log n \to 0$ as n tends to infinity, provided $k_n \log n/n \to 0$, $M_n \to \infty$ and $M_n/\log n \to \infty$.

Discussion — Choosing the p_{nj}'s

Near-reality scenario

The following splitting scheme is iteratively repeated at each node:

1. Select at random M_n candidate coordinates to split on.
2. For each of the M_n elected coordinates, calculate the best split.
3. Select the coordinate which outputs the best within-node sum of squares decrease, and cut.

Conclusion

For $j \in \mathcal{S}$,

$$p_{nj} \approx \frac{1}{S}\left(1 + \xi_{nj}\right),$$

where $\xi_{nj} \to 0$ and satisfies the constraint $\xi_{nj} \log n \to 0$ as n tends to infinity, provided $k_n \log n/n \to 0$, $M_n \to \infty$ and $M_n/\log n \to \infty$.

Near-reality scenario

The following splitting scheme is iteratively repeated at each node:

1. Select at random M_n candidate coordinates to split on.
2. For each of the M_n elected coordinates, calculate the best split.
3. Select the coordinate which outputs the best within-node sum of squares decrease, and cut.

Conclusion

For $j \in \mathcal{S}$,

$$p_{nj} \approx \frac{1}{S}\left(1 + \xi_{nj}\right),$$

where $\xi_{nj} \to 0$ and satisfies the constraint $\xi_{nj} \log n \to 0$ as n tends to infinity, provided $k_n \log n/n \to 0$, $M_n \to \infty$ and $M_n/\log n \to \infty$.

Discussion — Choosing the p_{nj}'s

Near-reality scenario

The following splitting scheme is iteratively repeated at each node:

1. Select at random M_n candidate coordinates to split on.
2. For each of the M_n elected coordinates, calculate the best split.
3. Select the coordinate which outputs the best within-node sum of squares decrease, and cut.

Conclusion

For $j \in \mathcal{S}$,

$$p_{nj} \approx \frac{1}{S}\left(1 + \xi_{nj}\right),$$

where $\xi_{nj} \to 0$ and satisfies the constraint $\xi_{nj} \log n \to 0$ as n tends to infinity, provided $k_n \log n/n \to 0$, $M_n \to \infty$ and $M_n/\log n \to \infty$.

Discussion — Choosing the p_{nj}'s

Near-reality scenario

The following splitting scheme is iteratively repeated at each node:

1. Select at random M_n candidate coordinates to split on.
2. For each of the M_n elected coordinates, calculate the best split.
3. Select the coordinate which outputs the best within-node sum of squares decrease, and cut.

Conclusion

For $j \in \mathcal{S}$,

$$p_{nj} \approx \frac{1}{S}\left(1 + \xi_{nj}\right),$$

where $\xi_{nj} \to 0$ and satisfies the constraint $\xi_{nj} \log n \to 0$ as n tends to infinity, provided $k_n \log n/n \to 0$, $M_n \to \infty$ and $M_n/\log n \to \infty$.

Outline

Tested models

$$Y = r(\mathbf{X}) + \varepsilon, \quad \text{with } \mathbf{X} \sim \mathcal{U}([0,1]^d) \text{ and } \varepsilon \sim \mathcal{N}(0,1).$$

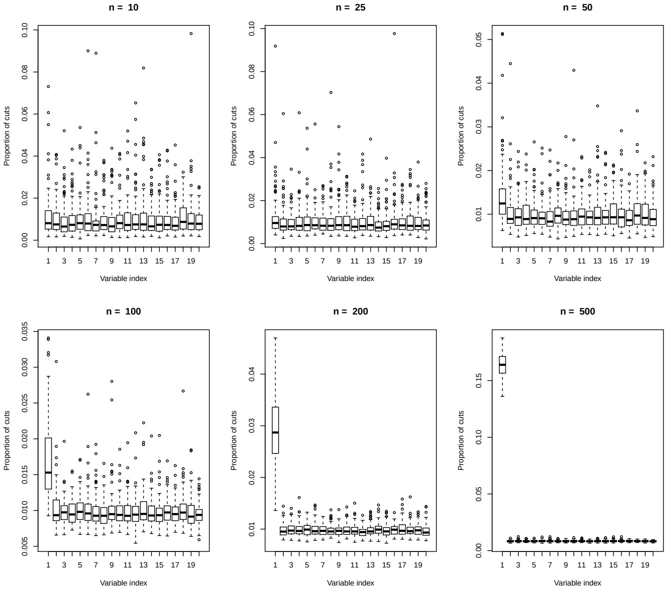
1. [Sinus] For $\mathbf{x} \in [0,1]^d$,

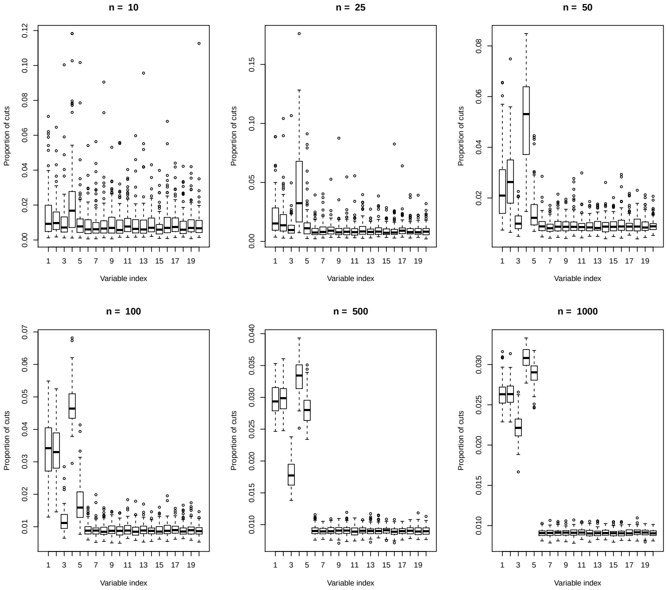
$$r(\mathbf{x}) = 10 \sin(10\pi x^{(1)}).$$

2. [Friedman #1] Here,

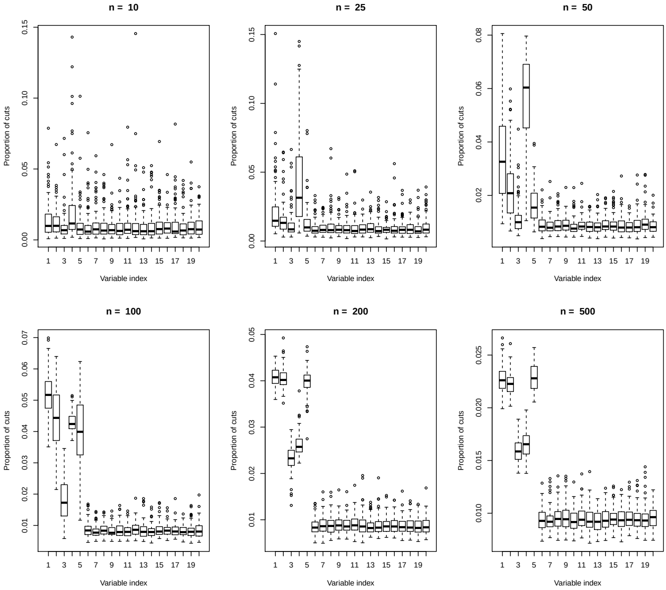
$$r(\mathbf{x}) = 10 \sin(\pi x^{(1)} x^{(2)}) + 20(x^{(3)} - .05)^2 + 10x^{(4)} + 5x^{(5)}.$$

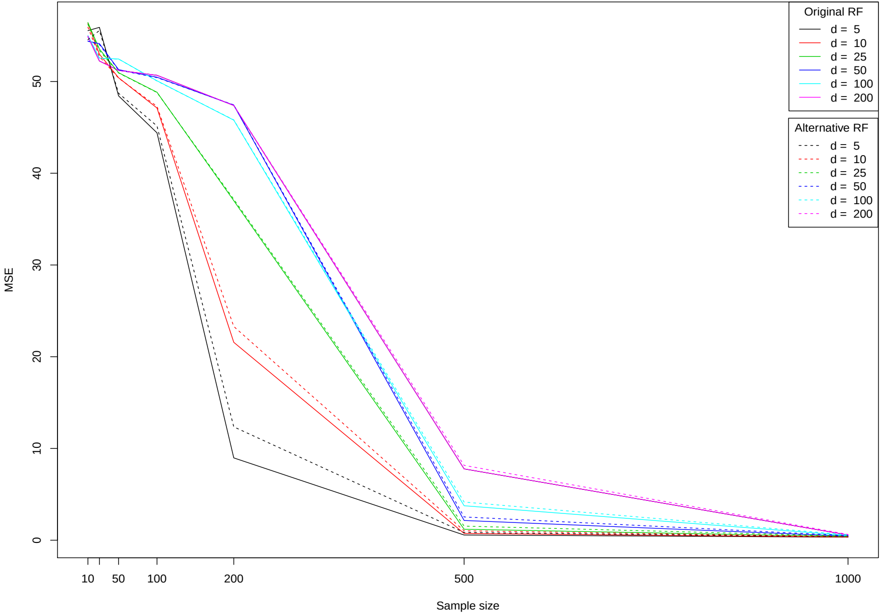
3. [Tree] In this example, the function r has itself a tree structure.

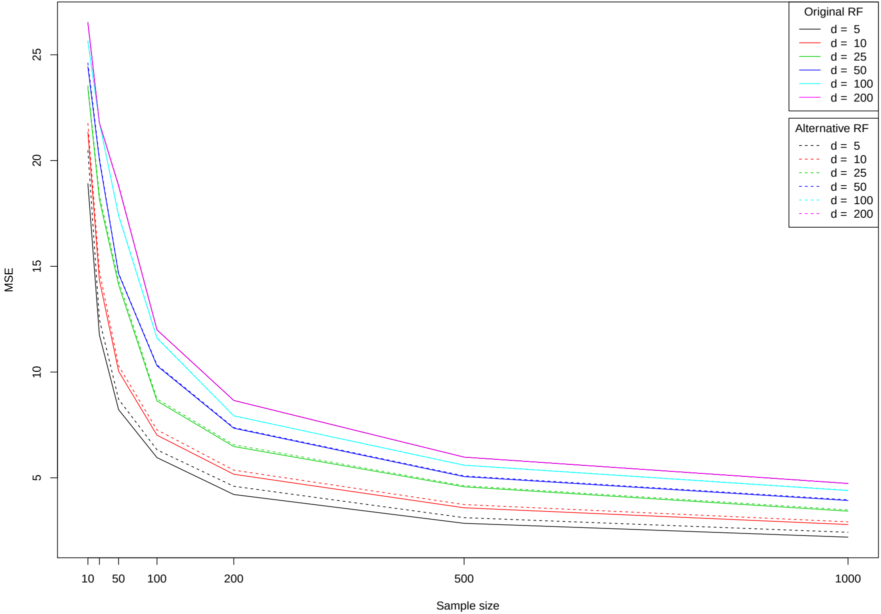

n = 10
n = 25
n = 50
n = 100
n = 200
n = 500
Proportion of cuts
Variable index

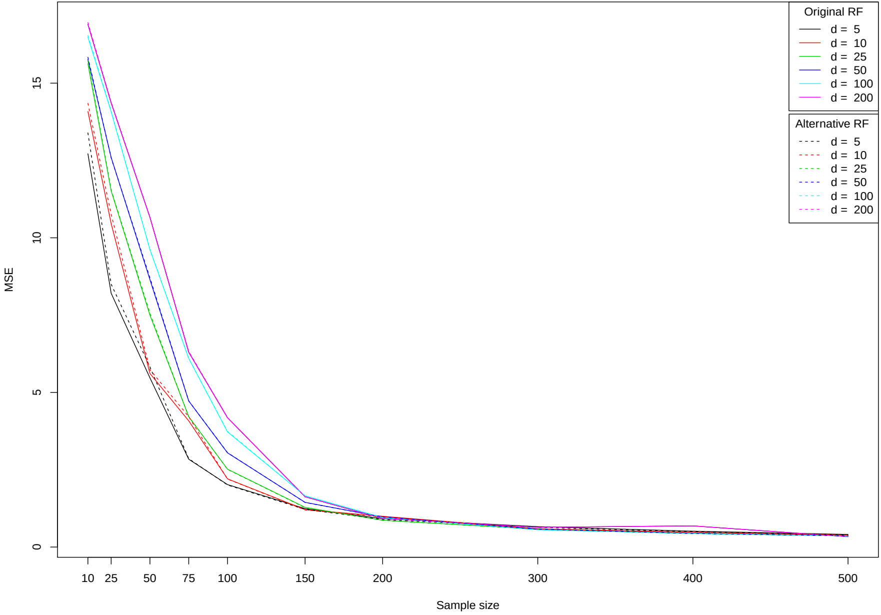


n = 10
n = 25
n = 50
n = 100
n = 500
n = 1000
Proportion of cuts
Variable index
1 3 5 7 9 11 13 15 17 19


n = 10
n = 25
n = 50
n = 100
n = 200
n = 500
Proportion of cuts
Variable index

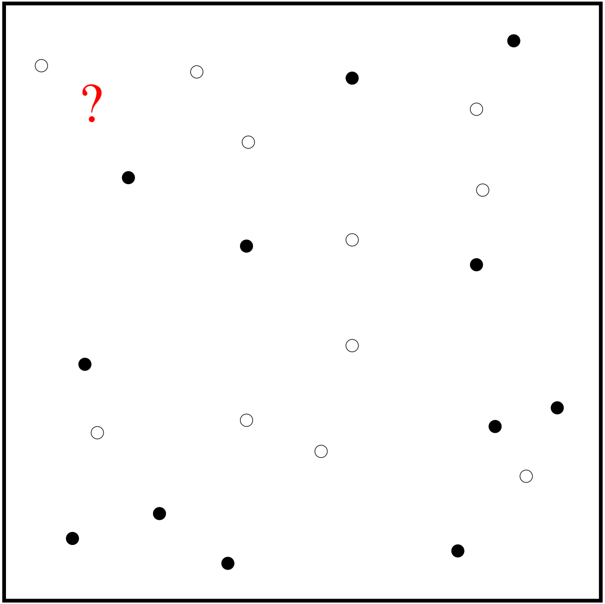

Original RF
d = 5
d = 10
d = 25
d = 50
d = 100
d = 200
Alternative RF
d = 5
d = 10
d = 25
d = 50
d = 100
d = 200
MSE
0
10
20
30
40
50
10
50
100
200
500
1000
Sample size

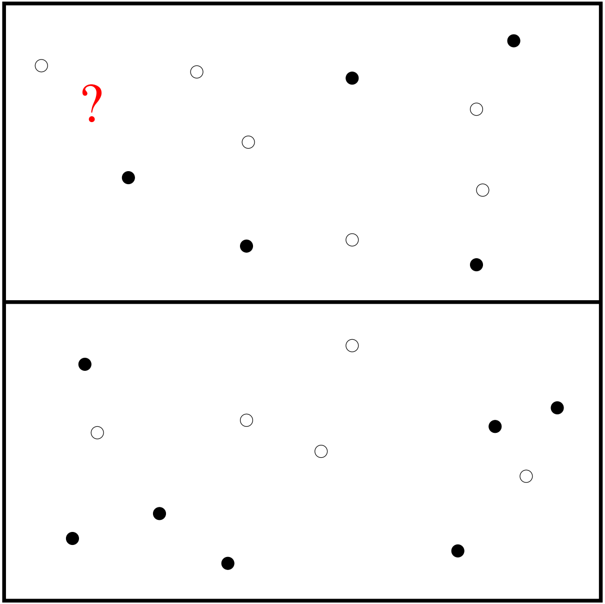

Original RF
d = 5
d = 10
d = 25
d = 50
d = 100
d = 200
Alternative RF
d = 5
d = 10
d = 25
d = 50
d = 100
d = 200
MSE
5
10
15
20
25
10
50
100
200
500
1000
Sample size

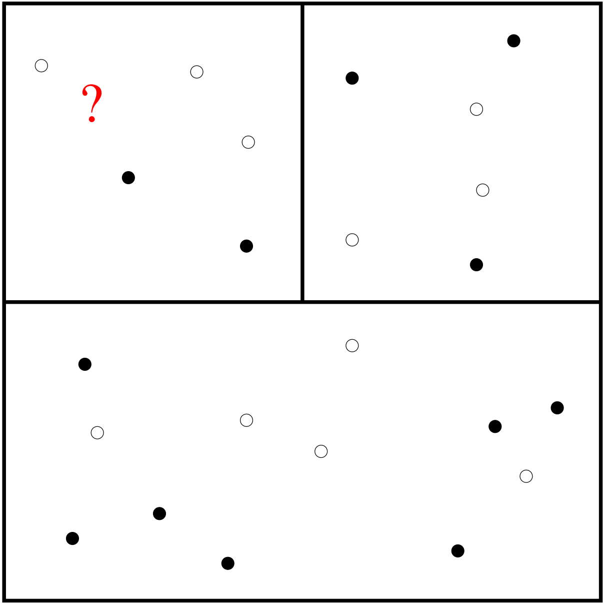


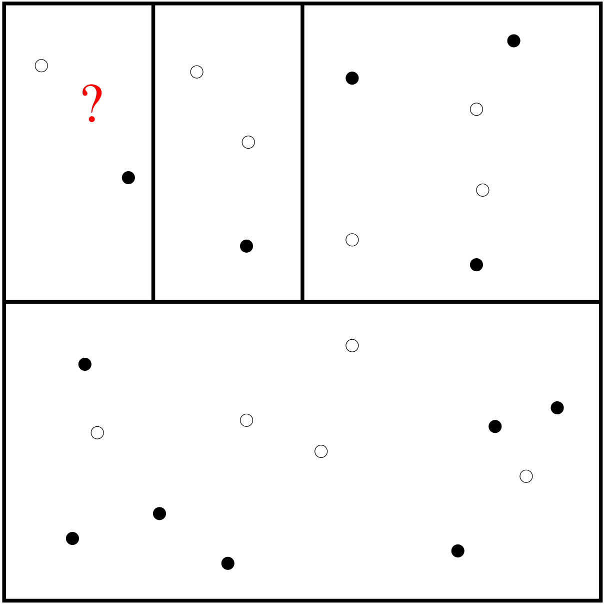
Original RF
d = 5
d = 10
d = 25
d = 50
d = 100
d = 200
Alternative RF
d = 5
d = 10
d = 25
d = 50
d = 100
d = 200
MSE
0
5
10
15
10
25
50
75
100
150
200
300
400
500
Sample size

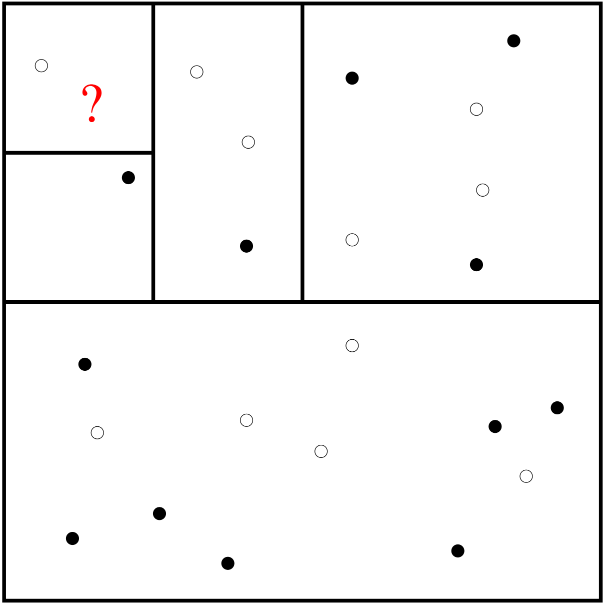
Outline

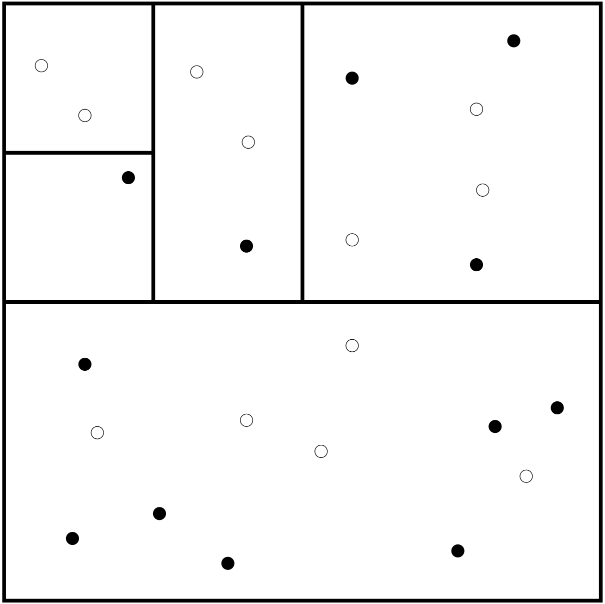


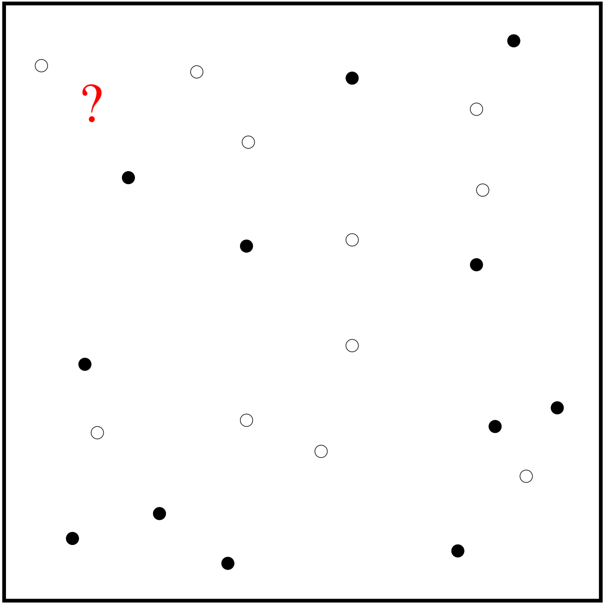


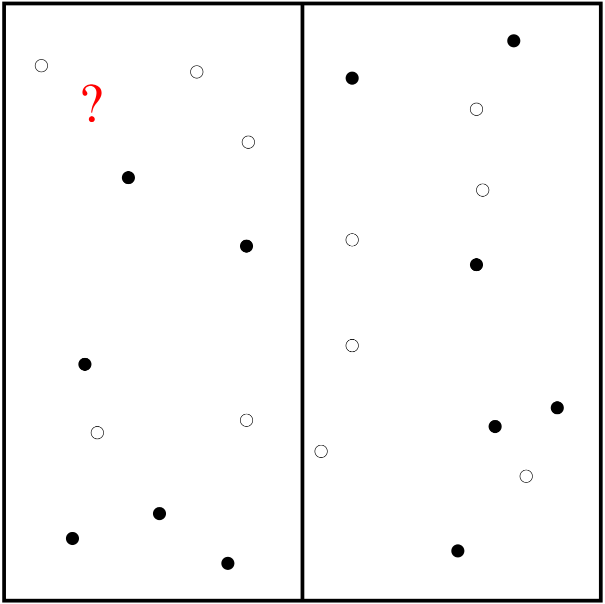


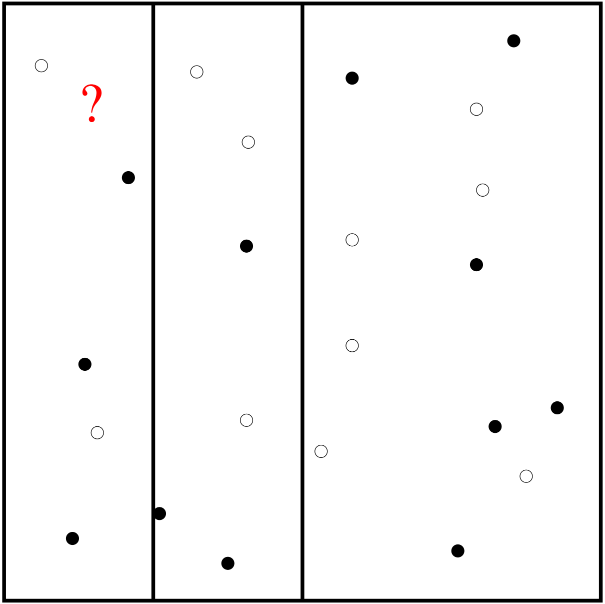


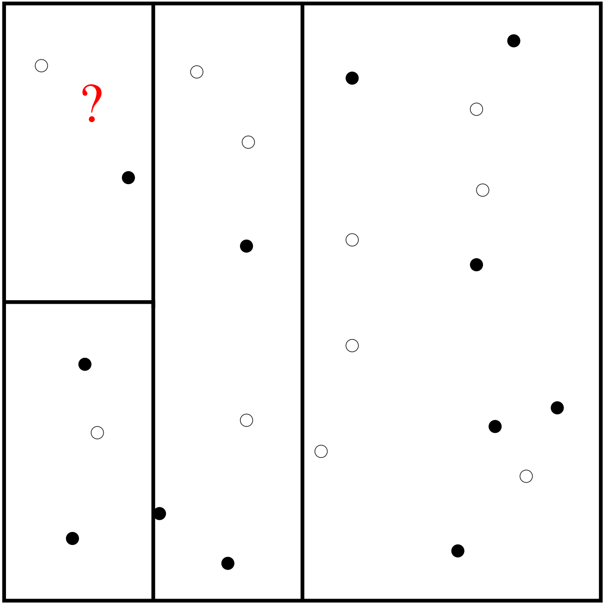


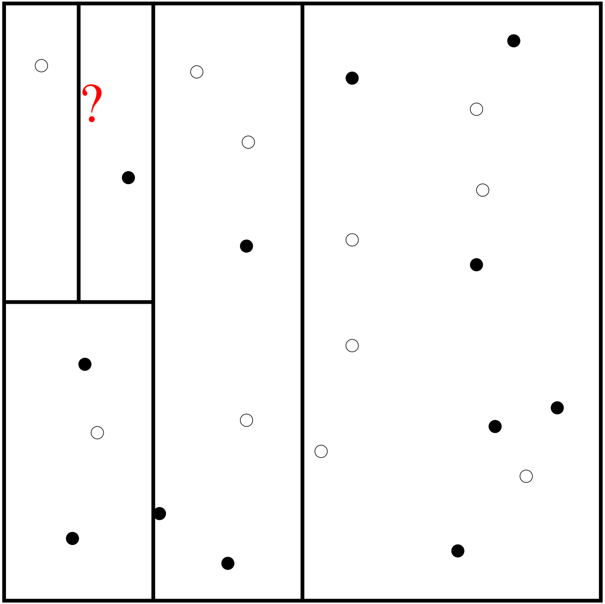


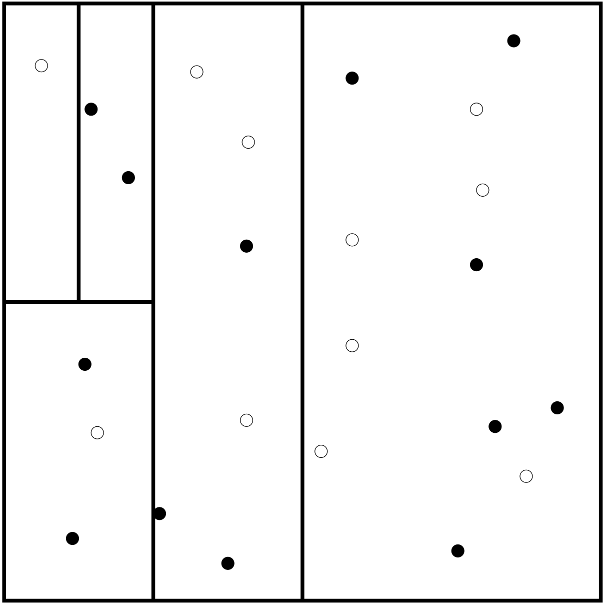


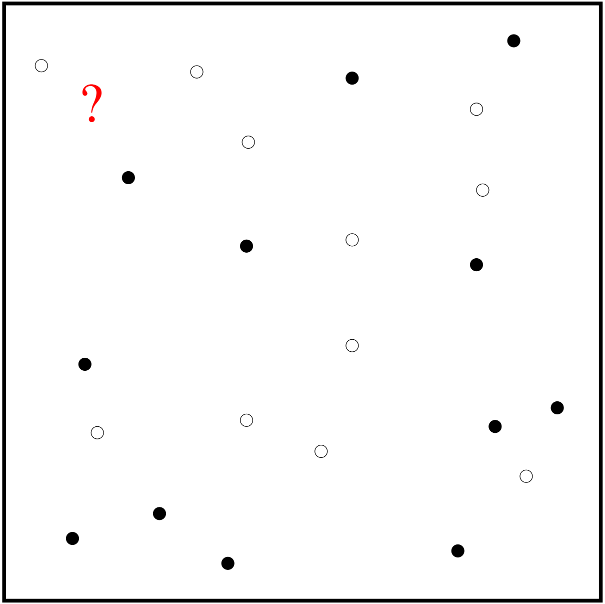


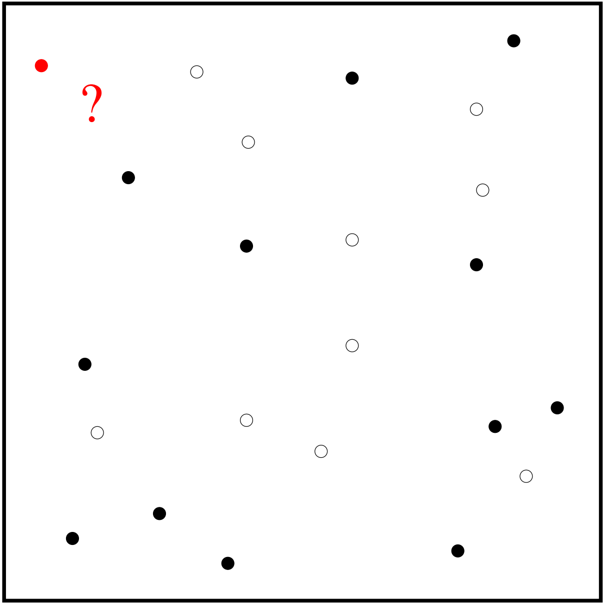


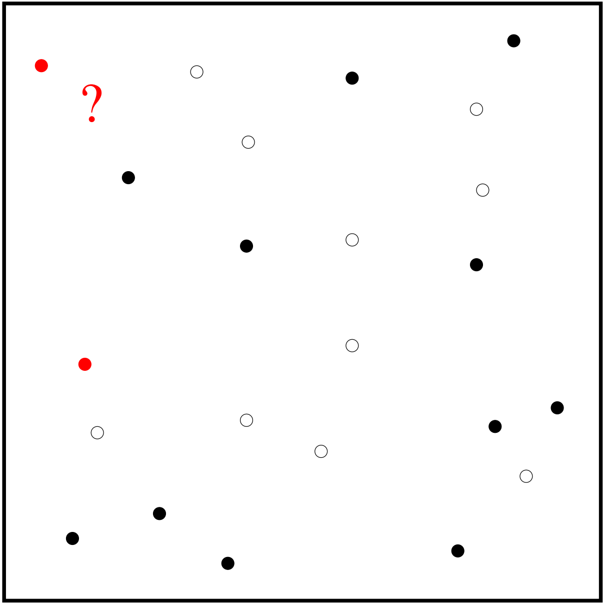


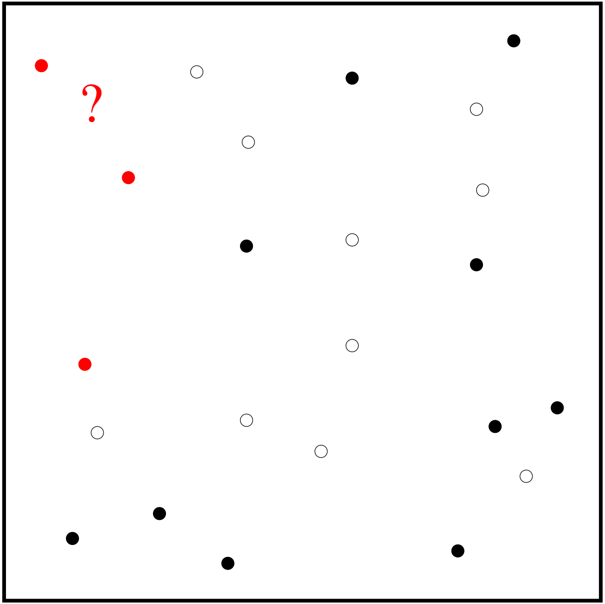


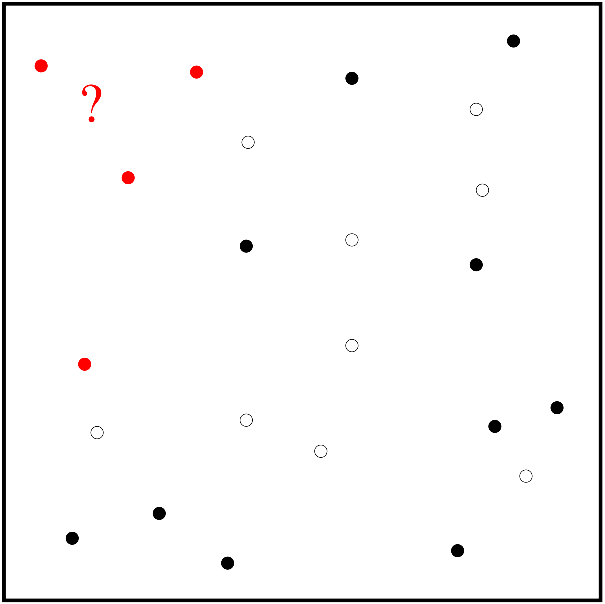


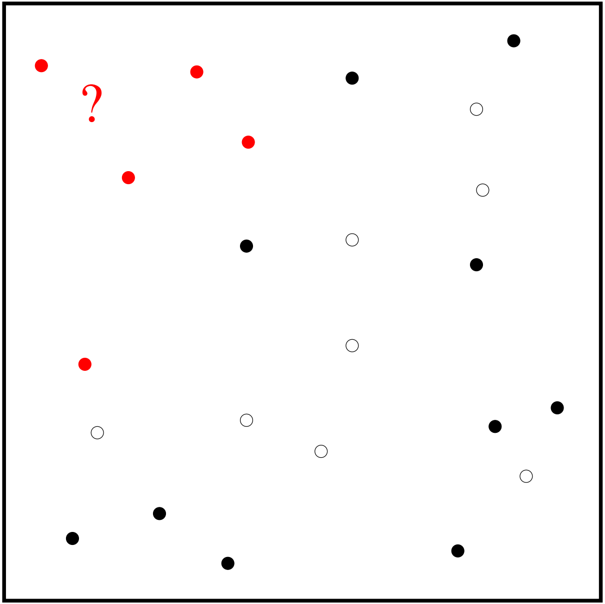


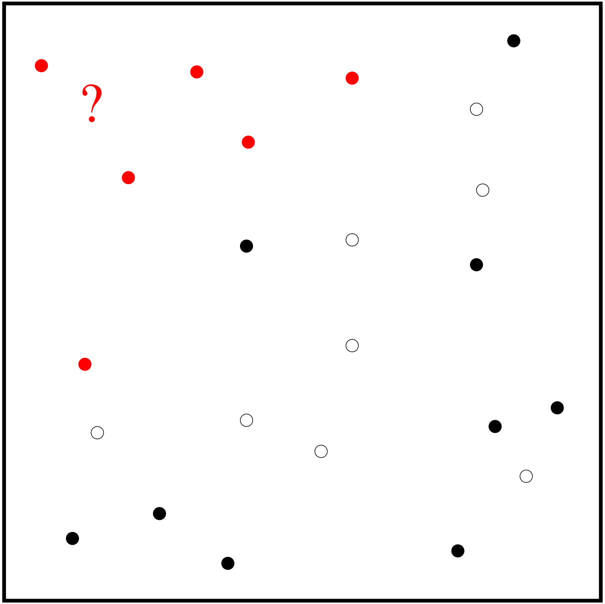


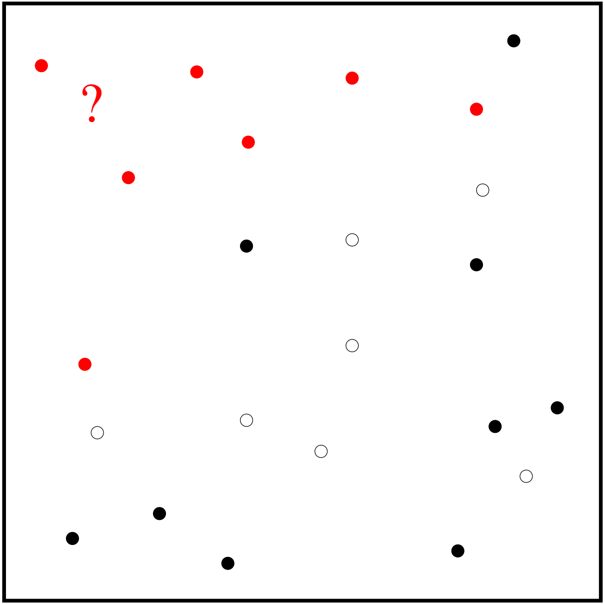


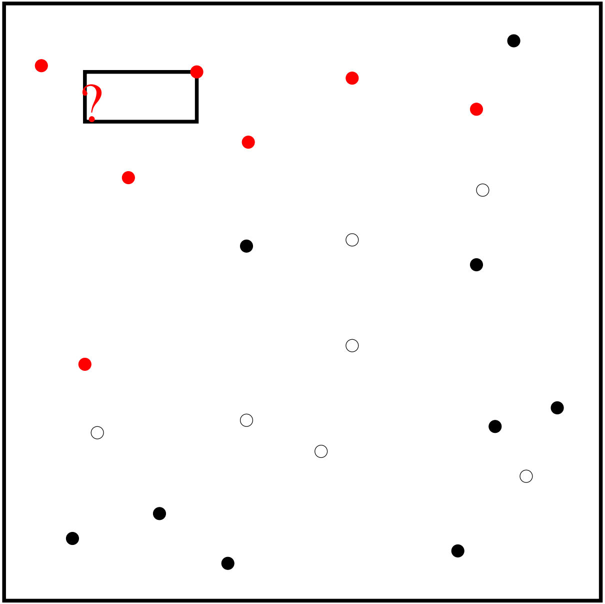


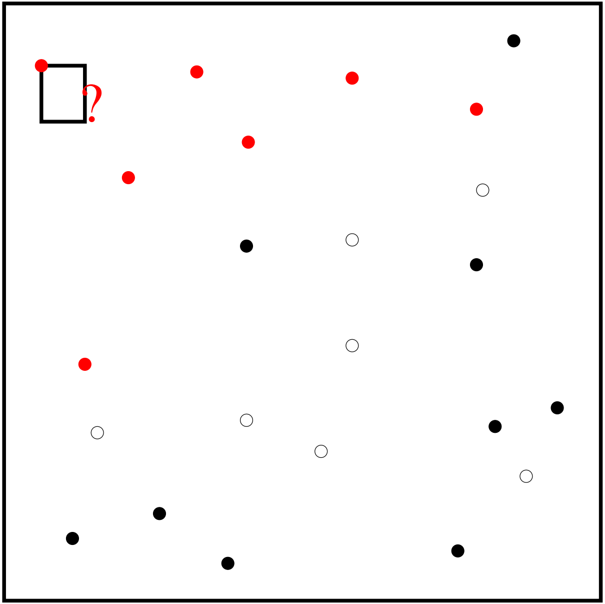


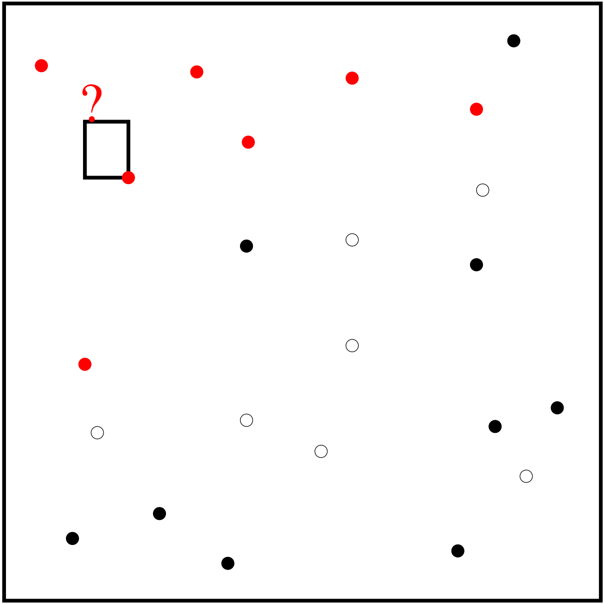


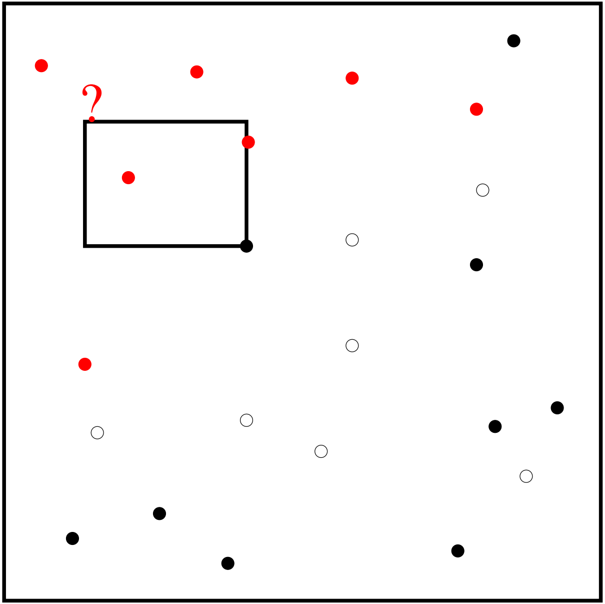








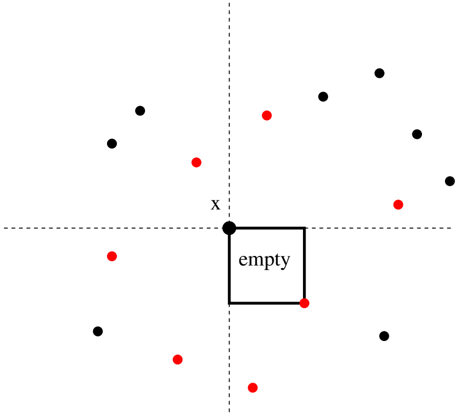




Layered Nearest Neighbors

Definition

Let $\mathbf{X}_1, \ldots, \mathbf{X}_n$ be a sample of i.i.d. random vectors in $\mathbb{R}^d$, $d \geq 2$. An observation $\mathbf{X}_i$ is said to be a LNN of a point $\mathbf{x}$ if the hyperrectangle defined by $\mathbf{x}$ and $\mathbf{X}_i$ contains no other data points.



What is known about $L_n(\mathbf{x})$?

- ... a lot when $\mathbf{X}_1, \ldots, \mathbf{X}_n$ are uniformly distributed over $[0, 1]^d$.
- For example,

$$\mathbb{E}L_n(\mathbf{x}) = \frac{2^d(\log n)^{d-1}}{(d-1)!} + \mathcal{O}\left((\log n)^{d-2}\right)$$

and

$$\frac{(d-1)!\, L_n(\mathbf{x})}{2^d(\log n)^{d-1}} \to 1 \quad \text{in probability as } n \to \infty.$$

- This is the problem of maxima in random vectors (**Barndorff-Nielsen and Sobel**, 1966).

What is known about $L_n(\mathbf{x})$?

- ... a lot when $\mathbf{X}_1, \ldots, \mathbf{X}_n$ are uniformly distributed over $[0, 1]^d$.
- For example,

$$\mathbb{E} L_n(\mathbf{x}) = \frac{2^d (\log n)^{d-1}}{(d-1)!} + \mathcal{O}\left((\log n)^{d-2}\right)$$

and

$$\frac{(d-1)!\, L_n(\mathbf{x})}{2^d (\log n)^{d-1}} \to 1 \quad \text{in probability as } n \to \infty.$$

- This is the problem of maxima in random vectors (**Barndorff-Nielsen and Sobel**, 1966).

What is known about $L_n(\mathbf{x})$?

- ... a lot when $\mathbf{X}_1, \ldots, \mathbf{X}_n$ are uniformly distributed over $[0,1]^d$.
- For example,

$$\mathbb{E}L_n(\mathbf{x}) = \frac{2^d (\log n)^{d-1}}{(d-1)!} + \mathcal{O}\left((\log n)^{d-2}\right)$$

and

$$\frac{(d-1)!\, L_n(\mathbf{x})}{2^d (\log n)^{d-1}} \to 1 \quad \text{in probability as } n \to \infty.$$

- This is the problem of maxima in random vectors (**Barndorff-Nielsen and Sobel**, 1966).

Two results (Biau and Devroye, 2010)

Model

$\mathbf{X}_1, \ldots, \mathbf{X}_n$ are independently distributed according to some probability density f (with probability measure μ).

Theorem

For μ-almost all $\mathbf{x} \in \mathbb{R}^d$, one has

$$L_n(\mathbf{x}) \to \infty \quad \text{in probability as } n \to \infty.$$

Theorem

Suppose that f is λ-almost everywhere continuous. Then

$$\frac{(d-1)!\, \mathbb{E} L_n(\mathbf{x})}{2^d (\log n)^{d-1}} \to 1 \quad \text{as } n \to \infty,$$

at μ-almost all $\mathbf{x} \in \mathbb{R}^d$.

Two results (Biau and Devroye, 2010)

Model

$\mathbf{X}_1, \ldots, \mathbf{X}_n$ are independently distributed according to some probability density f (with probability measure μ).

Theorem

For μ-almost all $\mathbf{x} \in \mathbb{R}^d$, one has

$$L_n(\mathbf{x}) \to \infty \quad \text{in probability as } n \to \infty.$$

Theorem

Suppose that f is λ-almost everywhere continuous. Then

$$\frac{(d-1)!\,\mathbb{E}L_n(\mathbf{x})}{2^d(\log n)^{d-1}} \to 1 \quad \text{as } n \to \infty,$$

at μ-almost all $\mathbf{x} \in \mathbb{R}^d$.

Two results (Biau and Devroye, 2010)

Model

$\mathbf{X}_1, \dots, \mathbf{X}_n$ are independently distributed according to some probability density f (with probability measure μ).

Theorem

For μ-almost all $\mathbf{x} \in \mathbb{R}^d$, one has

$$L_n(\mathbf{x}) \to \infty \quad \text{in probability as } n \to \infty.$$

Theorem

Suppose that f is λ-almost everywhere continuous. Then

$$\frac{(d-1)!\,\mathbb{E}L_n(\mathbf{x})}{2^d(\log n)^{d-1}} \to 1 \quad \text{as } n \to \infty,$$

at μ-almost all $\mathbf{x} \in \mathbb{R}^d$.

LNN regression estimation

Model

$(\mathbf{X}, Y), (\mathbf{X}_1, Y_1), \ldots, (\mathbf{X}_n, Y_n)$ are i.i.d. random vectors of $\mathbb{R}^d \times \mathbb{R}$. Moreover, $|Y|$ is bounded and $\mathbf{X}$ has a density.

The regression function $r(\mathbf{x}) = \mathbb{E}[Y|\mathbf{X} = \mathbf{x}]$ may be estimated by

$$r_n(\mathbf{x}) = \frac{1}{L_n(\mathbf{x})} \sum_{i=1}^{n} Y_i \mathbf{1}_{[\mathbf{X}_i \in \mathcal{L}_n(\mathbf{x})]}.$$

1. No smoothing parameter.
2. A scale-invariant estimate.
3. Intimately connected to Breiman's random forests.

Model

$(\mathbf{X}, Y), (\mathbf{X}_1, Y_1), \ldots, (\mathbf{X}_n, Y_n)$ are i.i.d. random vectors of $\mathbb{R}^d \times \mathbb{R}$.
Moreover, $|Y|$ is bounded and $\mathbf{X}$ has a density.

The regression function $r(\mathbf{x}) = \mathbb{E}[Y|\mathbf{X} = \mathbf{x}]$ may be estimated by

$$r_n(\mathbf{x}) = \frac{1}{L_n(\mathbf{x})} \sum_{i=1}^{n} Y_i \mathbf{1}_{[\mathbf{X}_i \in \mathcal{L}_n(\mathbf{x})]}.$$

1. No smoothing parameter.
2. A scale-invariant estimate.
3. Intimately connected to Breiman's random forests.

LNN regression estimation

Model

$(\mathbf{X}, Y), (\mathbf{X}_1, Y_1), \ldots, (\mathbf{X}_n, Y_n)$ are i.i.d. random vectors of $\mathbb{R}^d \times \mathbb{R}$. Moreover, $|Y|$ is bounded and $\mathbf{X}$ has a density.

The regression function $r(\mathbf{x}) = \mathbb{E}[Y|\mathbf{X} = \mathbf{x}]$ may be estimated by

$$r_n(\mathbf{x}) = \frac{1}{L_n(\mathbf{x})} \sum_{i=1}^{n} Y_i \mathbf{1}_{[\mathbf{X}_i \in \mathcal{L}_n(\mathbf{x})]}.$$

1. No smoothing parameter.
2. A scale-invariant estimate.
3. Intimately connected to Breiman's random forests.

LNN regression estimation

Model

$(\mathbf{X}, Y), (\mathbf{X}_1, Y_1), \ldots, (\mathbf{X}_n, Y_n)$ are i.i.d. random vectors of $\mathbb{R}^d \times \mathbb{R}$. Moreover, $|Y|$ is bounded and $\mathbf{X}$ has a density.

The regression function $r(\mathbf{x}) = \mathbb{E}[Y|\mathbf{X} = \mathbf{x}]$ may be estimated by

$$r_n(\mathbf{x}) = \frac{1}{L_n(\mathbf{x})} \sum_{i=1}^{n} Y_i \mathbf{1}_{[\mathbf{X}_i \in \mathcal{L}_n(\mathbf{x})]}.$$

1. No smoothing parameter.
2. A scale-invariant estimate.
3. Intimately connected to Breiman's random forests.

LNN regression estimation

Model

$(\mathbf{X}, Y), (\mathbf{X}_1, Y_1), \ldots, (\mathbf{X}_n, Y_n)$ are i.i.d. random vectors of $\mathbb{R}^d \times \mathbb{R}$. Moreover, $|Y|$ is bounded and $\mathbf{X}$ has a density.

The regression function $r(\mathbf{x}) = \mathbb{E}[Y|\mathbf{X} = \mathbf{x}]$ may be estimated by

$$r_n(\mathbf{x}) = \frac{1}{L_n(\mathbf{x})} \sum_{i=1}^{n} Y_i \mathbf{1}_{[\mathbf{X}_i \in \mathcal{L}_n(\mathbf{x})]}.$$

1. No smoothing parameter.
2. A scale-invariant estimate.
3. Intimately connected to Breiman's random forests.

Consistency

Theorem (Pointwise L_p-consistency)

Assume that the regression function r is λ-almost everywhere continuous and that Y is bounded. Then, for μ-almost all $\mathbf{x} \in \mathbb{R}^d$ and all $p \geq 1$,

$$\mathbb{E}\,|r_n(\mathbf{x}) - r(\mathbf{x})|^p \to 0 \quad \text{as } n \to \infty.$$

Theorem (Gobal L_p-consistency)

Under the same conditions, for all $p \geq 1$,

$$\mathbb{E}\,|r_n(\mathbf{X}) - r(\mathbf{X})|^p \to 0 \quad \text{as } n \to \infty.$$

1. No universal consistency result with respect to r is possible.
2. The results do not impose any condition on the density.
3. They are also scale-free.

Consistency

Theorem (Pointwise L_p-consistency)

Assume that the regression function r is λ-almost everywhere continuous and that Y is bounded. Then, for μ-almost all $\mathbf{x} \in \mathbb{R}^d$ and all $p \geq 1$,

$$\mathbb{E}\,|r_n(\mathbf{x}) - r(\mathbf{x})|^p \to 0 \quad \text{as } n \to \infty.$$

Theorem (Gobal L_p-consistency)

Under the same conditions, for all $p \geq 1$,

$$\mathbb{E}\,|r_n(\mathbf{X}) - r(\mathbf{X})|^p \to 0 \quad \text{as } n \to \infty.$$

1. No universal consistency result with respect to r is possible.
2. The results do not impose any condition on the density.
3. They are also scale-free.

Consistency

Theorem (Pointwise L_p-consistency)

Assume that the regression function r is λ-almost everywhere continuous and that Y is bounded. Then, for μ-almost all $\mathbf{x} \in \mathbb{R}^d$ and all $p \geq 1$,

$$\mathbb{E}\,|r_n(\mathbf{x}) - r(\mathbf{x})|^p \to 0 \quad \textit{as } n \to \infty.$$

Theorem (Gobal L_p-consistency)

Under the same conditions, for all $p \geq 1$,

$$\mathbb{E}\,|r_n(\mathbf{X}) - r(\mathbf{X})|^p \to 0 \quad \textit{as } n \to \infty.$$

1. No universal consistency result with respect to r is possible.
2. The results do not impose any condition on the density.
3. They are also scale-free.

Consistency

Theorem (Pointwise L_p-consistency)

Assume that the regression function r is λ-almost everywhere continuous and that Y is bounded. Then, for μ-almost all $\mathbf{x} \in \mathbb{R}^d$ and all $p \geq 1$,

$$\mathbb{E}\,|r_n(\mathbf{x}) - r(\mathbf{x})|^p \to 0 \quad \text{as } n \to \infty.$$

Theorem (Gobal L_p-consistency)

Under the same conditions, for all $p \geq 1$,

$$\mathbb{E}\,|r_n(\mathbf{X}) - r(\mathbf{X})|^p \to 0 \quad \text{as } n \to \infty.$$

1. No universal consistency result with respect to r is possible.
2. The results do not impose any condition on the density.
3. They are also scale-free.

Consistency

Theorem (Pointwise L_p-consistency)

Assume that the regression function r is λ-almost everywhere continuous and that Y is bounded. Then, for μ-almost all $\mathbf{x} \in \mathbb{R}^d$ and all $p \geq 1$,

$$\mathbb{E}\,|r_n(\mathbf{x}) - r(\mathbf{x})|^p \to 0 \quad \text{as } n \to \infty.$$

Theorem (Gobal L_p-consistency)

Under the same conditions, for all $p \geq 1$,

$$\mathbb{E}\,|r_n(\mathbf{X}) - r(\mathbf{X})|^p \to 0 \quad \text{as } n \to \infty.$$

1. No universal consistency result with respect to r is possible.
2. The results do not impose any condition on the density.
3. They are also scale-free.

Back to random forests

A random forest can be viewed as a weighted LNN regression estimate

$$\bar{r}_n(\mathbf{x}) = \sum_{i=1}^{n} Y_i W_{ni}(\mathbf{x}),$$

where the weights concentrate on the LNN and satisfy

$$\sum_{i=1}^{n} W_{ni}(\mathbf{x}) = 1.$$

Consider the non-adaptive random forests estimate

$$\bar{r}_n(\mathbf{x}) = \sum_{i=1}^{n} Y_i W_{ni}(\mathbf{x}).$$

Proposition

For any $\mathbf{x} \in \mathbb{R}^d$, assume that $\sigma^2 = \mathbb{V}[Y|\mathbf{X} = \mathbf{x}]$ is independent of $\mathbf{x}$. Then

$$\mathbb{E}\left[\bar{r}_n(\mathbf{x}) - r(\mathbf{x})\right]^2 \geq \frac{\sigma^2}{\mathbb{E}L_n(\mathbf{x})}.$$

Non-adaptive strategies

Consider the non-adaptive random forests estimate

$$\bar{r}_n(\mathbf{x}) = \sum_{i=1}^{n} Y_i W_{ni}(\mathbf{x}).$$

Proposition

For any $\mathbf{x} \in \mathbb{R}^d$, assume that $\sigma^2 = \mathbb{V}[Y|\mathbf{X} = \mathbf{x}]$ is independent of $\mathbf{x}$. Then

$$\mathbb{E}\left[\bar{r}_n(\mathbf{x}) - r(\mathbf{x})\right]^2 \geq \frac{\sigma^2}{\mathbb{E}L_n(\mathbf{x})}.$$

Rate of convergence

At μ-almost all **x**, when f is λ-almost everywhere continuous,

$$\mathbb{E}\left[\bar{r}_n(\mathbf{x}) - r(\mathbf{x})\right]^2 \gtrsim \frac{\sigma^2(d-1)!}{2^d(\log n)^{d-1}}.$$

Improving the rate of convergence

1. Stop as soon as a future rectangle split would cause a sub-rectangle to have fewer than k_n points.
2. Resort to bagging and randomize using random subsamples.

Rate of convergence

At μ-almost all $\mathbf{x}$, when f is λ-almost everywhere continuous,

$$\mathbb{E}\left[\bar{r}_n(\mathbf{x}) - r(\mathbf{x})\right]^2 \gtrsim \frac{\sigma^2(d-1)!}{2^d(\log n)^{d-1}}.$$

Improving the rate of convergence

1. Stop as soon as a future rectangle split would cause a sub-rectangle to have fewer than k_n points.
2. Resort to bagging and randomize using random subsamples.

At μ-almost all $\mathbf{x}$, when f is λ-almost everywhere continuous,

$$\mathbb{E}\left[\bar{r}_n(\mathbf{x}) - r(\mathbf{x})\right]^2 \gtrsim \frac{\sigma^2(d-1)!}{2^d(\log n)^{d-1}}.$$

Improving the rate of convergence

1. Stop as soon as a future rectangle split would cause a sub-rectangle to have fewer than k_n points.
2. Resort to bagging and randomize using random subsamples.