Escaping the curse of dimensionality with a tree-based regressor

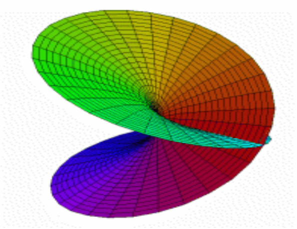
Samory Kpotufe

UCSD CSE

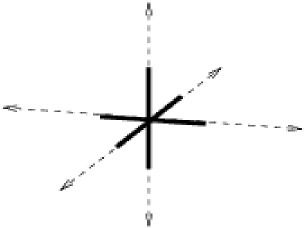
Curse of dimensionality

- **In general:** Computational and/or prediction performance deteriorate as the dimension D increases.
- **For nonparametric regression:** Worst case bounds on excess risk $\|f_n - f\|^2$ are of the form $n^{-2/(2+D)}$.
  Here $\|f_n - f\|^2 = \mathbb{E}_X \|f_n(X) - f(X)\|^2$.

Reasons for hope: data often has low intrinsic complexity



d-dimensional manifold

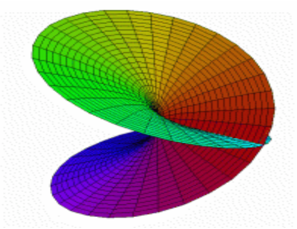


d-sparse data

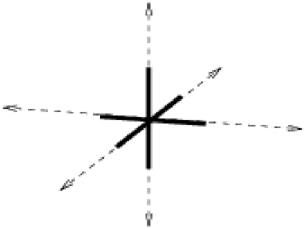
Curse of dimensionality

- **In general:** Computational and/or prediction performance deteriorate as the dimension D increases.
- **For nonparametric regression:** Worst case bounds on excess risk $\|f_n - f\|^2$ are of the form $n^{-2/(2+D)}$.
  Here $\|f_n - f\|^2 = \mathbb{E}_X \|f_n(X) - f(X)\|^2$.

Reasons for hope: data often has low intrinsic complexity



d-dimensional manifold

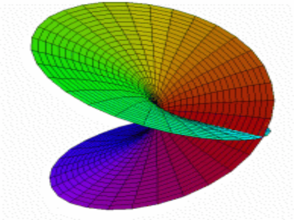


d-sparse data

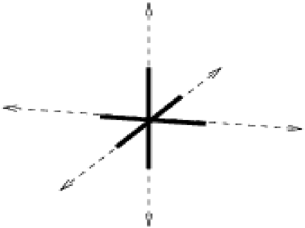
Curse of dimensionality

- **In general:** Computational and/or prediction performance deteriorate as the dimension D increases.
- **For nonparametric regression:** Worst case bounds on excess risk $\|f_n - f\|^2$ are of the form $n^{-2/(2+D)}$.
  Here $\|f_n - f\|^2 = \mathbb{E}_X \|f_n(X) - f(X)\|^2$.

Reasons for hope: data often has low intrinsic complexity



d-dimensional manifold



d-sparse data

How do we take advantage of such situations?

- **Manifold learning (e.g. LLE, Isomap)**: embed the data in a lower dimensional space where traditional learners might perform well.
- **Adaptivity**: can we design learners which run in $\mathbb{R}^D$ but whose performance depend just on the intrinsic complexity of the data?

How do we take advantage of such situations?

- **Manifold learning (e.g. LLE, Isomap)**: embed the data in a lower dimensional space where traditional learners might perform well.
- **Adaptivity**: can we design learners which run in $\mathbb{R}^D$ but whose performance depend just on the intrinsic complexity of the data?

Recent adaptivity results

- **Classification with dyadic trees** (Scott and Nowak, 2006).
  Manifold data.
- **Kernel regression** (Bickel and Li, 2006).
  Manifold data.
- **Vector Quantization with RPtree partitioning** (Dasgupta and Freund, 2008).
  Data with low Assouad dimension.

Tree-based regressors are computationally inexpensive relative to kernel regressors. Is there an adaptive tree-based regressor?

Recent adaptivity results

- **Classification with dyadic trees** (Scott and Nowak, 2006).
  Manifold data.
- **Kernel regression** (Bickel and Li, 2006).
  Manifold data.
- **Vector Quantization with RPtree partitioning** (Dasgupta and Freund, 2008).
  Data with low Assouad dimension.

Tree-based regressors are computationally inexpensive relative to kernel regressors. Is there an adaptive tree-based regressor?

Recent adaptivity results

- **Classification with dyadic trees** (Scott and Nowak, 2006).
  Manifold data.
- **Kernel regression** (Bickel and Li, 2006).
  Manifold data.
- **Vector Quantization with RPtree partitioning** (Dasgupta and Freund, 2008).
  Data with low Assouad dimension.

Tree-based regressors are computationally inexpensive relative to kernel regressors. Is there an adaptive tree-based regressor?

Recent adaptivity results

- **Classification with dyadic trees** (Scott and Nowak, 2006).
  Manifold data.
- **Kernel regression** (Bickel and Li, 2006).
  Manifold data.
- **Vector Quantization with RPtree partitioning** (Dasgupta and Freund, 2008).
  Data with low Assouad dimension.

Tree-based regressors are computationally inexpensive relative to kernel regressors. Is there an adaptive tree-based regressor?

Recent adaptivity results

- **Classification with dyadic trees** (Scott and Nowak, 2006).
  Manifold data.
- **Kernel regression** (Bickel and Li, 2006).
  Manifold data.
- **Vector Quantization with RPtree partitioning** (Dasgupta and Freund, 2008).
  Data with low Assouad dimension.

Tree-based regressors are computationally inexpensive relative to kernel regressors. Is there an adaptive tree-based regressor?

Recent adaptivity results

- **Classification with dyadic trees** (Scott and Nowak, 2006).
  Manifold data.
- **Kernel regression** (Bickel and Li, 2006).
  Manifold data.
- **Vector Quantization with RPtree partitioning** (Dasgupta and Freund, 2008).
  Data with low Assouad dimension.

Tree-based regressors are computationally inexpensive relative to kernel regressors. Is there an adaptive tree-based regressor?

Recent adaptivity results

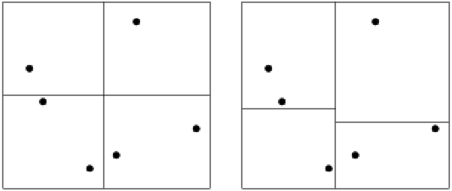
- **Classification with dyadic trees** (Scott and Nowak, 2006).
  Manifold data.
- **Kernel regression** (Bickel and Li, 2006).
  Manifold data.
- **Vector Quantization with RPtree partitioning** (Dasgupta and Freund, 2008).
  Data with low Assouad dimension.

Tree-based regressors are computationally inexpensive relative to kernel regressors. Is there an adaptive tree-based regressor?

Tree-based regression

Build a hierarchy of nested partitions of $\mathcal{X}$, somehow pick a partition $\mathbf{A}$:

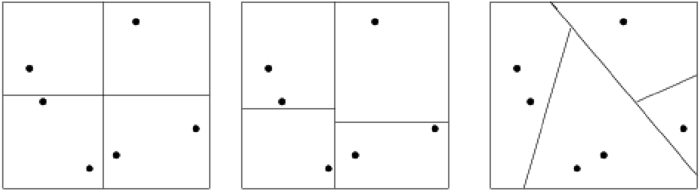
$f_{n,\mathbf{A}}(x) \doteq$ average Y value in $\mathbf{A}(x)$, the cell of $\mathbf{A}$ in which x falls.



Dyadic tree $\qquad$ k-d tree

Tree-based regression

Build a hierarchy of nested partitions of $\mathcal{X}$, somehow pick a partition $\mathbf{A}$:
$f_{n,\mathbf{A}}(x) \doteq$ average Y value in $\mathbf{A}(x)$, the cell of $\mathbf{A}$ in which x falls.



Dyadic tree $\quad$ k-d tree $\quad$ RPtree

Random Partition tree (RPtree)

Recursively bisect the data near the median along a random direction.

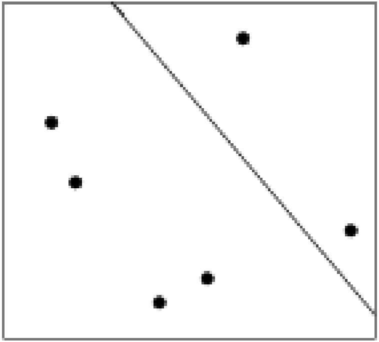


Figure: First level.

Random Partition tree (RPtree)

Recursively bisect the data near the median along a random direction.

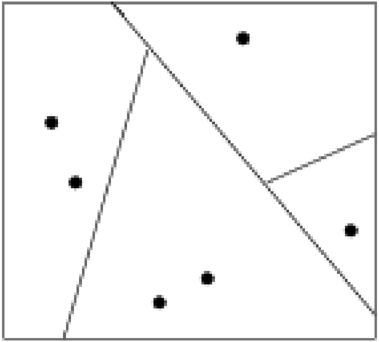


Figure: Second level.

Our results:

We show how to use the RPtree for regression and obtain rates that depend just on the intrinsic complexity of the data, namely its Assouad dimension.

This is the first such adaptivity result for tree-based regression.

Our results:

We show how to use the RPtree for regression and obtain rates that depend just on the intrinsic complexity of the data, namely its Assouad dimension.

This is the first such adaptivity result for tree-based regression.

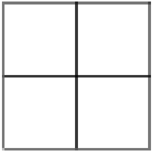
What we'll see next:

- **Preliminaries: (1) Assouad dimension. (2) Algorithmic issues regarding the choice of a partition.**
- How we choose an RPtree partition for regression.
- Analysis overview.

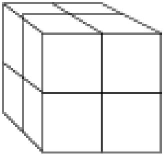
Assouad dimension.

Definition

The Assouad dimension (or doubling dimension) of $\mathcal{X}$ is the smallest d such that any ball $B \subset \mathcal{X}$ can be covered by 2^d balls of half its radius.

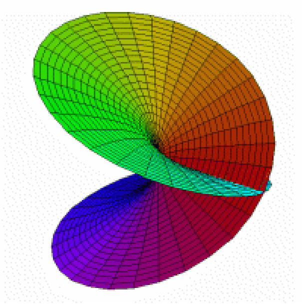


2^2 balls

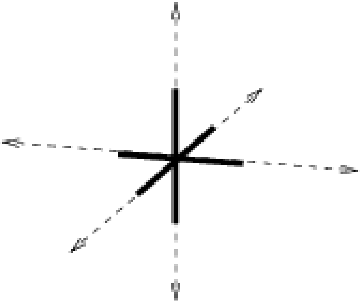


2^3 balls

Examples of data with low Assouad dimension



d-dimensional manifold



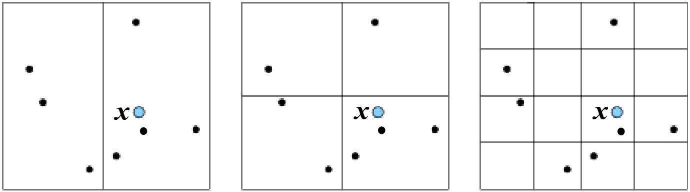
d-sparse data

Choosing a good partition: bias-variance tradeoff

Partition $\mathbf{A}$ *with small cell diameters?*

Low bias: x is near all the data in $\mathbf{A}(x) \implies$ similar Y values.
High variance: fewer data in $\mathbf{A}(x) \implies$ unstable estimates.

So choose partition with mid-range cell diameters.



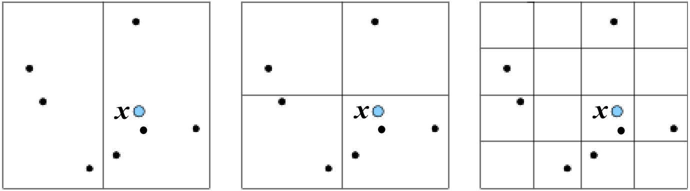
Choosing a good partition: bias-variance tradeoff

Partition $\mathbf{A}$ *with small cell diameters?*

Low bias: x is near all the data in $\mathbf{A}(x) \implies$ similar Y values.

High variance: fewer data in $\mathbf{A}(x) \implies$ unstable estimates.

So choose partition with mid-range cell diameters.

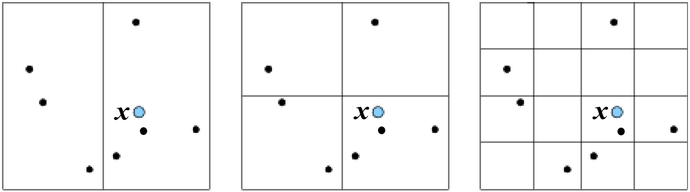


Choosing a good partition: bias-variance tradeoff

Partition $\mathbf{A}$ with small cell diameters?

Low bias: x is near all the data in $\mathbf{A}(x) \implies$ similar Y values.
High variance: fewer data in $\mathbf{A}(x) \implies$ unstable estimates.

So choose partition with mid-range cell diameters.



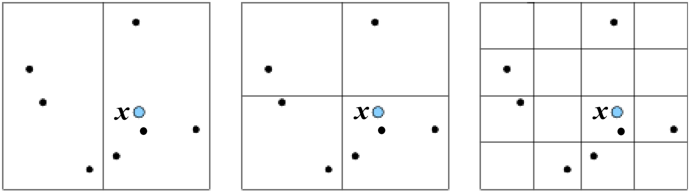
Choosing a good partition: bias-variance tradeoff

Partition $\mathbf{A}$ *with small cell diameters?*

Low bias: x is near all the data in $\mathbf{A}(x) \implies$ similar Y values.

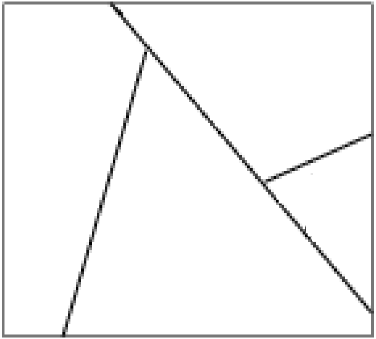
High variance: fewer data in $\mathbf{A}(x) \implies$ unstable estimates.

So choose partition with mid-range cell diameters.



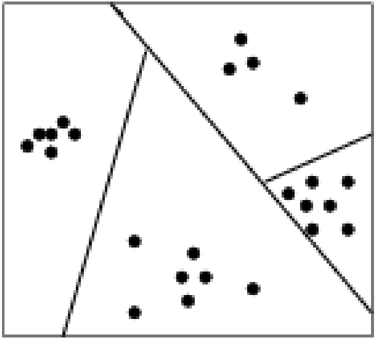
Problem with RPtree cells:

- Cell diameters are hard to assess.
- Cell diameters may not decrease at all.



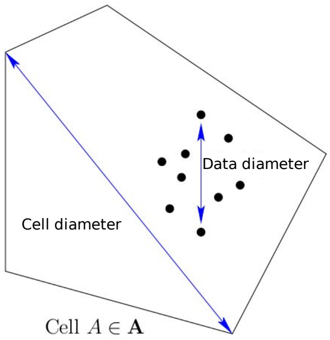
Problem with RPtree cells:

- Cell diameters are hard to assess.
- Cell diameters may not decrease at all.



Problem with RPtree cells:

- Cell diameters are hard to assess.
- Cell diameters may not decrease at all.



Choosing a partition based on data diameter:

This is of general interest for algorithm design and risk analysis: many trees have misbehaved cell diameters like RPtree does.

Data diameters aren't stable $\implies$ hard to generalize from.

RPtree quickly decreases data diameters from the root down: data diameters are halved every $d \log d$ levels [DF08].

Fast data diameter decrease rate implies: We reach cells with small data diameters without decreasing the number of points per cell too much.

Choosing a partition based on data diameter:

This is of general interest for algorithm design and risk analysis: many trees have misbehaved cell diameters like RPtree does.

Data diameters aren't stable $\implies$ hard to generalize from.

RPtree quickly decreases data diameters from the root down: data diameters are halved every $d \log d$ levels [DF08].

Fast data diameter decrease rate implies: We reach cells with small data diameters without decreasing the number of points per cell too much.

Choosing a partition based on data diameter:

This is of general interest for algorithm design and risk analysis: many trees have misbehaved cell diameters like RPtree does.

Data diameters aren't stable $\Longrightarrow$ hard to generalize from.

RPtree quickly decreases data diameters from the root down: data diameters are halved every $d \log d$ levels [DF08].

Fast data diameter decrease rate implies: We reach cells with small data diameters without decreasing the number of points per cell too much.

Choosing a partition based on data diameter:

This is of general interest for algorithm design and risk analysis: many trees have misbehaved cell diameters like RPtree does.

Data diameters aren't stable $\implies$ hard to generalize from.

RPtree quickly decreases data diameters from the root down: data diameters are halved every $d \log d$ levels [DF08].

Fast data diameter decrease rate implies: We reach cells with small data diameters without decreasing the number of points per cell too much.

Choosing a partition based on data diameter:

This is of general interest for algorithm design and risk analysis: many trees have misbehaved cell diameters like RPtree does.

Data diameters aren't stable $\Longrightarrow$ hard to generalize from.

RPtree quickly decreases data diameters from the root down: data diameters are halved every $d \log d$ levels [DF08].

Fast data diameter decrease rate implies: We reach cells with small data diameters without decreasing the number of points per cell too much.

What we'll see next:

- Preliminaries: (1) Assouad dimension. (2) Algorithmic issues regarding the choice of a partition.
- **How we choose an RPtree partition for regression.**
- Analysis overview.

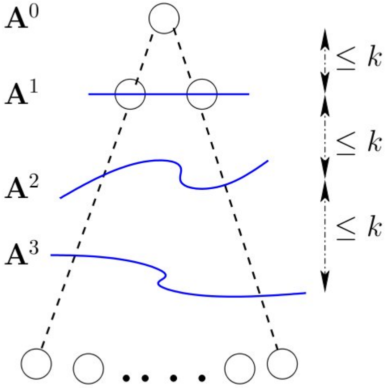
How we choose an RPtree partition for regression

Identify candidate partitions $\mathbf{A}^i$ such that data diameters are halved from $\mathbf{A}^i$ to $\mathbf{A}^{i+1}$.

Diameter decrease rate k:
max levels between $\mathbf{A}^i$ and $\mathbf{A}^{i+1}$.

Two stopping options:
I: Stop after $O(\log n)$ levels, and test all $f_{n,\mathbf{A}^i}$ on new sample.

II: Stop when data diameters are small enough. **Requires no testing**.



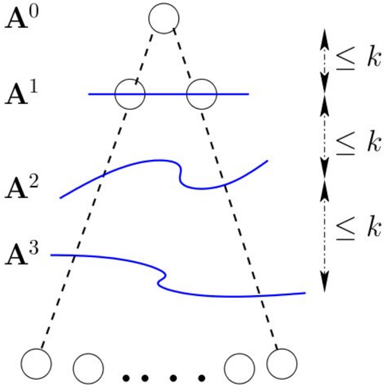
How we choose an RPtree partition for regression

Identify candidate partitions $\mathbf{A}^i$ such that data diameters are halved from $\mathbf{A}^i$ to $\mathbf{A}^{i+1}$.

Diameter decrease rate k:
max levels between $\mathbf{A}^i$ and $\mathbf{A}^{i+1}$.

Two stopping options:
I: Stop after $O(\log n)$ levels, and test all $f_{n,\mathbf{A}^i}$ on new sample.

II: Stop when data diameters are small enough. **Requires no testing**.



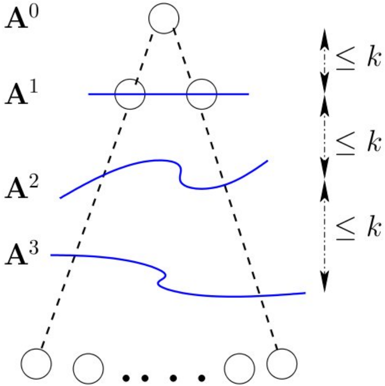
How we choose an RPtree partition for regression

Identify candidate partitions $\mathbf{A}^i$ such that data diameters are halved from $\mathbf{A}^i$ to $\mathbf{A}^{i+1}$.

Diameter decrease rate k:
max levels between $\mathbf{A}^i$ and $\mathbf{A}^{i+1}$.

Two stopping options:
I: Stop after $O(\log n)$ levels, and test all $f_{n,\mathbf{A}^i}$ on new sample.

II: Stop when data diameters are small enough. **Requires no testing**.



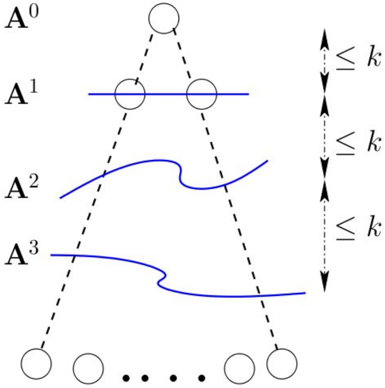
How we choose an RPtree partition for regression

Identify candidate partitions $\mathbf{A}^i$ such that data diameters are halved from $\mathbf{A}^i$ to $\mathbf{A}^{i+1}$.

Diameter decrease rate k:
max levels between $\mathbf{A}^i$ and $\mathbf{A}^{i+1}$.

Two stopping options:
I: Stop after $O(\log n)$ levels, and test all $f_{n,\mathbf{A}^i}$ on new sample.

II: Stop when data diameters are small enough. **Requires no testing**.



How we choose an RPtree partition for regression

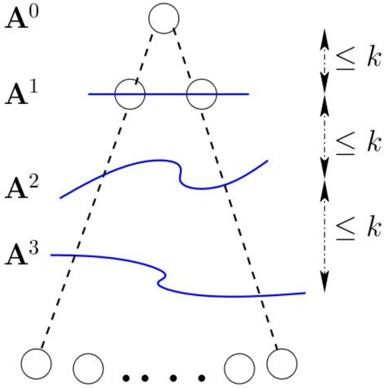
Identify candidate partitions $\mathbf{A}^i$ such that data diameters are halved from $\mathbf{A}^i$ to $\mathbf{A}^{i+1}$.

Diameter decrease rate k:
max levels between $\mathbf{A}^i$ and $\mathbf{A}^{i+1}$.

Two stopping options:

I: Stop after $O(\log n)$ levels, and test all $f_{n,\mathbf{A}^i}$ on new sample.

II: Stop when data diameters are small enough. **Requires no testing**.



Theorem

With probability at least $1 - \delta$, *under either stopping option,*

$$\|f_n - f\|^2 \leq C \left(\frac{\log^2 n + \log 1/\delta}{n} \right)^{2/(2+k)},$$

where $k \leq C' d \log d$ *is the observed diameter decrease rate.*

Assumptions:

Regression function $f(x) = \mathbb{E}[Y|X = x]$ is Lipschitz, X and Y are bounded. No distributional assumption.

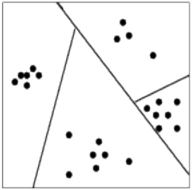
What we'll see next:

- Preliminaries: (1) Assouad dimension. (2) Algorithmic issues regarding the choice of a partition.
- How we choose an RPtree partition for regression.
- **Analysis overview.**

Risk analysis: Handling data diameters

Remember: Data diameters don't generalize well to distribution.

Solution: $\forall \mathbf{A} \in \{\mathbf{A}^i\}$, replace $\mathbf{A}$ with alternate partition $\mathbf{A}'$.

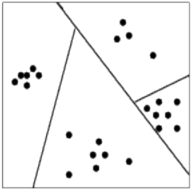


Partition $\mathbf{A}$

- Dense cells of $\mathbf{A}'$ have manageable diameters: cell diameters approximate data diameters.
- $\mathrm{Risk}(f_{n,\mathbf{A}'}) \approx \mathrm{Risk}(f_{n,\mathbf{A}})$.

Risk analysis: Handling data diameters

Remember: Data diameters don't generalize well to distribution.
Solution: $\forall \mathbf{A} \in \{\mathbf{A}^i\}$, replace $\mathbf{A}$ with alternate partition $\mathbf{A}'$.

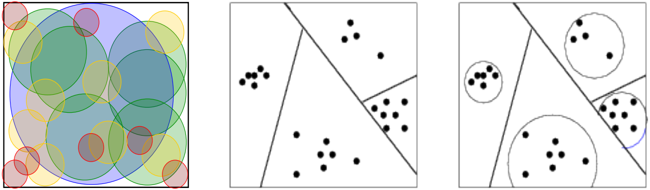


Partition $\mathbf{A}$

- Dense cells of $\mathbf{A}'$ have manageable diameters: cell diameters approximate data diameters.
- $\mathrm{Risk}(f_{n,\mathbf{A}'}) \approx \mathrm{Risk}(f_{n,\mathbf{A}})$.

Risk analysis: Handling data diameters

Remember: Data diameters don't generalize well to distribution.
Solution: $\forall \mathbf{A} \in \{\mathbf{A}^i\}$, replace $\mathbf{A}$ with alternate partition $\mathbf{A}'$.

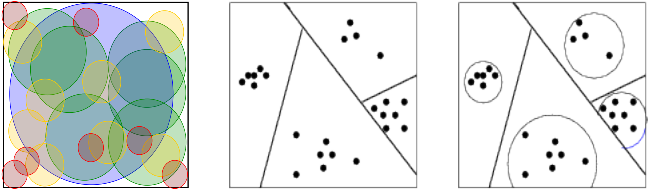


Cover $\mathcal{B}$ Partition $\mathbf{A}$ Partition $\mathbf{A}'$

- Dense cells of $\mathbf{A}'$ have manageable diameters: cell diameters approximate data diameters.
- $\text{Risk}(f_{n,\mathbf{A}'}) \approx \text{Risk}(f_{n,\mathbf{A}})$.

Risk analysis: Handling data diameters

Remember: Data diameters don't generalize well to distribution.
Solution: $\forall \mathbf{A} \in \{\mathbf{A}^i\}$, replace $\mathbf{A}$ with alternate partition $\mathbf{A}'$.



Cover $\mathcal{B}$ Partition $\mathbf{A}$ Partition $\mathbf{A}'$

- Dense cells of $\mathbf{A}'$ have manageable diameters: cell diameters approximate data diameters.
- $\mathrm{Risk}(f_{n,\mathbf{A}'}) \approx \mathrm{Risk}(f_{n,\mathbf{A}})$.

So we can just analyze the risk of $f_{n,\mathbf{A}'}$, *for every* $\mathbf{A} \in \{\mathbf{A}^i\}$

Handle the empty cells and the dense cells separately.

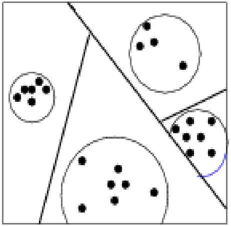


Figure: Partition $\mathbf{A}'$.

So we can just analyze the risk of $f_{n,\mathbf{A}'}$, *for every* $\mathbf{A} \in \{\mathbf{A}^i\}$

Handle the empty cells and the dense cells separately.

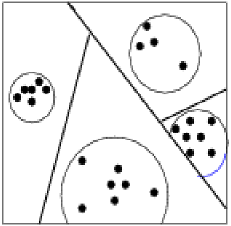


Figure: Partition $\mathbf{A}'$.

Relative VC bounds for $A' \in \mathbf{A}'$

- $\mu_n(A') \approx 0 \implies \mu(A') \lesssim \frac{1}{n}$: empty cells don't affect risk.
- $\mu_n(A') \gg 0 \implies \mu_n(A') \approx \mu(A')$: dense cells will be stable.

Need to exhibit VC class containing all $A' \in \mathbf{A}'$

- Cells $\mathbf{A}'$ have random shapes.
- Define a VC class $\mathcal{C} \supset \mathbf{A}'$ by conditioning on the randomness in the algorithm and n.

Relative VC bounds for $A' \in \mathbf{A}'$

- $\mu_n(A') \approx 0 \implies \mu(A') \lesssim \frac{1}{n}$: empty cells don't affect risk.
- $\mu_n(A') \gg 0 \implies \mu_n(A') \approx \mu(A')$: dense cells will be stable.

Need to exhibit VC class containing all $A' \in \mathbf{A}'$

- Cells $\mathbf{A}'$ have random shapes.
- Define a VC class $\mathcal{C} \supset \mathbf{A}'$ by conditioning on the randomness in the algorithm and n.

Relative VC bounds for $A' \in \mathbf{A}'$

- $\mu_n(A') \approx 0 \implies \mu(A') \lesssim \frac{1}{n}$: empty cells don't affect risk.
- $\mu_n(A') \gg 0 \implies \mu_n(A') \approx \mu(A')$: dense cells will be stable.

Need to exhibit VC class containing all $A' \in \mathbf{A}'$

- Cells $\mathbf{A}'$ have random shapes.
- Define a VC class $\mathcal{C} \supset \mathbf{A}'$ by conditioning on the randomness in the algorithm and n.

Relative VC bounds for $A' \in \mathbf{A}'$

- $\mu_n(A') \approx 0 \implies \mu(A') \lesssim \frac{1}{n}$: empty cells don't affect risk.
- $\mu_n(A') \gg 0 \implies \mu_n(A') \approx \mu(A')$: dense cells will be stable.

Need to exhibit VC class containing all $A' \in \mathbf{A}'$

- Cells $\mathbf{A}'$ have random shapes.
- Define a VC class $\mathcal{C} \supset \mathbf{A}'$ by conditioning on the randomness in the algorithm and n.

Risk bound for $f_{n,\mathbf{A}'}$, $\forall \mathbf{A} \in \{\mathbf{A}^i\}$:

$$\left\| f_{n,\mathbf{A}'} - f \right\|^2 \lesssim \frac{|\mathbf{A}'|}{n} + \mathrm{diam}_n^2(\mathbf{A}').$$

Bound is minimized when:

$\mathrm{diam}_n^2(\mathbf{A}') \approx \frac{|\mathbf{A}'|}{n}$, in which case $\left\| f_{n,\mathbf{A}'} - f \right\|^2 \lesssim n^{-2/(2+k)}$.

Show that we can find $\mathbf{A} \in \{\mathbf{A}^i\}$ s.t. $\mathbf{A}'$ minimizes the bound, and conclude. □

Risk bound for $f_{n,\mathbf{A}'}$, $\forall \mathbf{A} \in \{\mathbf{A}^i\}$:

$$\left\| f_{n,\mathbf{A}'} - f \right\|^2 \lesssim \frac{|\mathbf{A}'|}{n} + \text{diam}_n^2(\mathbf{A}').$$

Bound is minimized when:

$\text{diam}_n^2(\mathbf{A}') \approx \frac{|\mathbf{A}'|}{n}$, in which case $\left\| f_{n,\mathbf{A}'} - f \right\|^2 \lesssim n^{-2/(2+k)}$.

Show that we can find $\mathbf{A} \in \{\mathbf{A}^i\}$ s.t. $\mathbf{A}'$ minimizes the bound, and conclude. □

Risk bound for $f_{n,\mathbf{A}'}$, $\forall \mathbf{A} \in \{\mathbf{A}^i\}$:

$$\left\|f_{n,\mathbf{A}'} - f\right\|^2 \lesssim \frac{|\mathbf{A}'|}{n} + \mathrm{diam}_n^2\left(\mathbf{A}'\right).$$

Bound is minimized when:

$\mathrm{diam}_n^2\left(\mathbf{A}'\right) \approx \frac{|\mathbf{A}'|}{n}$, in which case $\left\|f_{n,\mathbf{A}'} - f\right\|^2 \lesssim n^{-2/(2+k)}$.

Show that we can find $\mathbf{A} \in \{\mathbf{A}^i\}$ s.t. $\mathbf{A}'$ minimizes the bound, and conclude. □

Recap:

If the data space has low Assouad dimension d, the excess risk of an RPtree regressor depends just on d.

Extending to a more general setting

What if the data has high Assouad dimension overall, but the intrinsic dimension is low in smaller regions?