JST CREST 学習/数理モデルに基づく時空間展開型アーキテクチャの創出と応用 (本村CREST)
機械学習班 フォレストワークショップ 話題提供

決定森回帰の信頼区間推定, Benign Overfitting, 多変量木とReLUネットの入力空間分割

瀧川 一学

ichigaku.takigawa@riken.jp

理化学研究所 革新知能統合研究センター@京阪奈ATR (iPS細胞連携班)
北海道大学 化学反応創成研究拠点 (ICReDD)

2022年2月24日

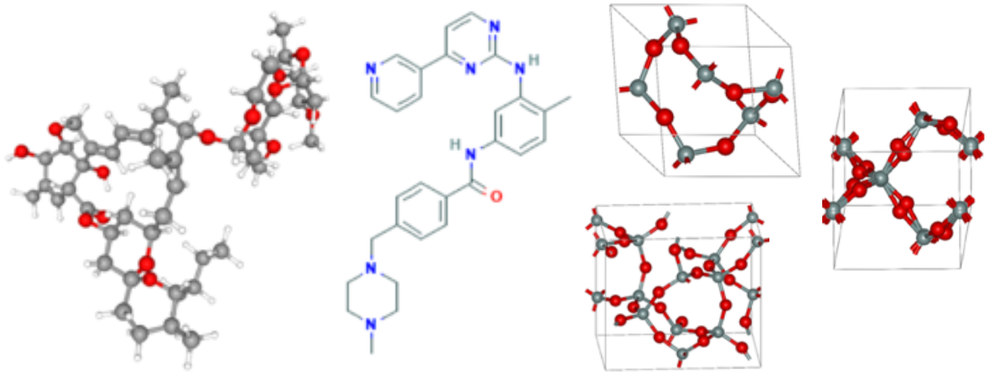
私の関心：離散構造を伴う機械学習＋データ中心型自然科学

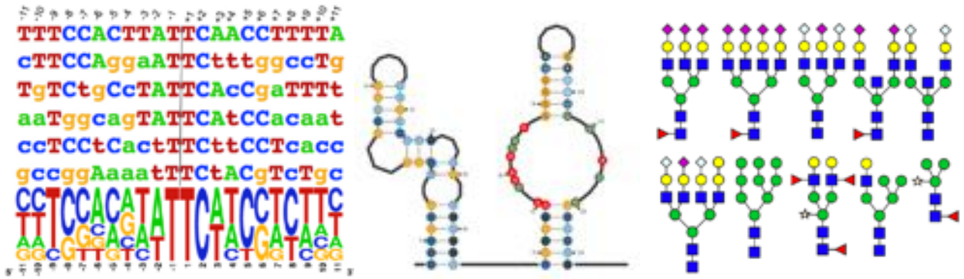
実務は幹細胞画像の深層学習(理研)と化学(北大)

離散構造 (組合せ的構造や代数的構造)

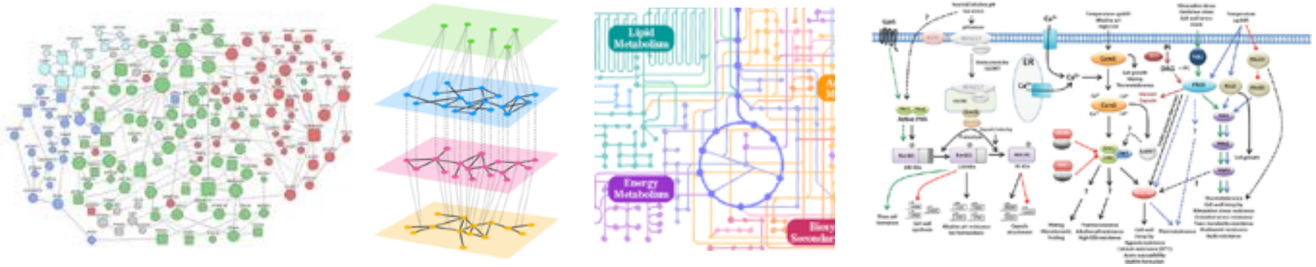
= 集合、論理、群、順列・組合せ、系列/文字列、ツリー、グラフ、組合せ幾何、…

対象に離散構造



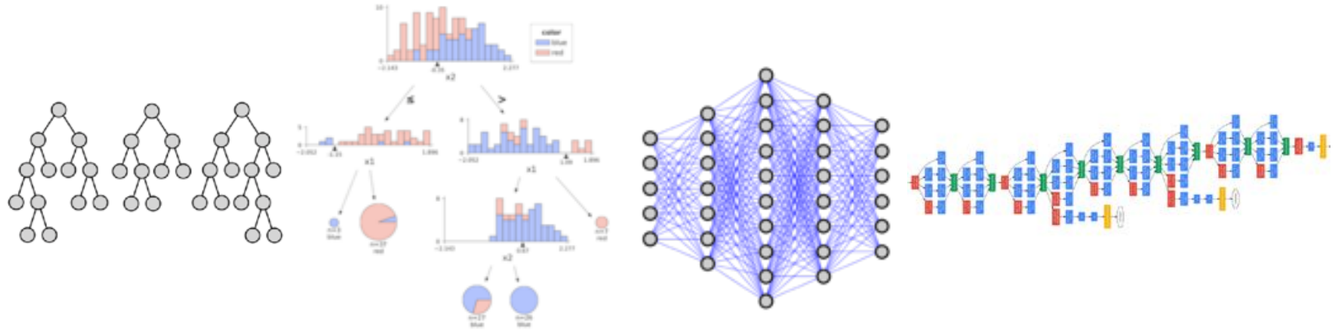


対象間に離散構造



機械学習モデルに離散構造

本日の話題



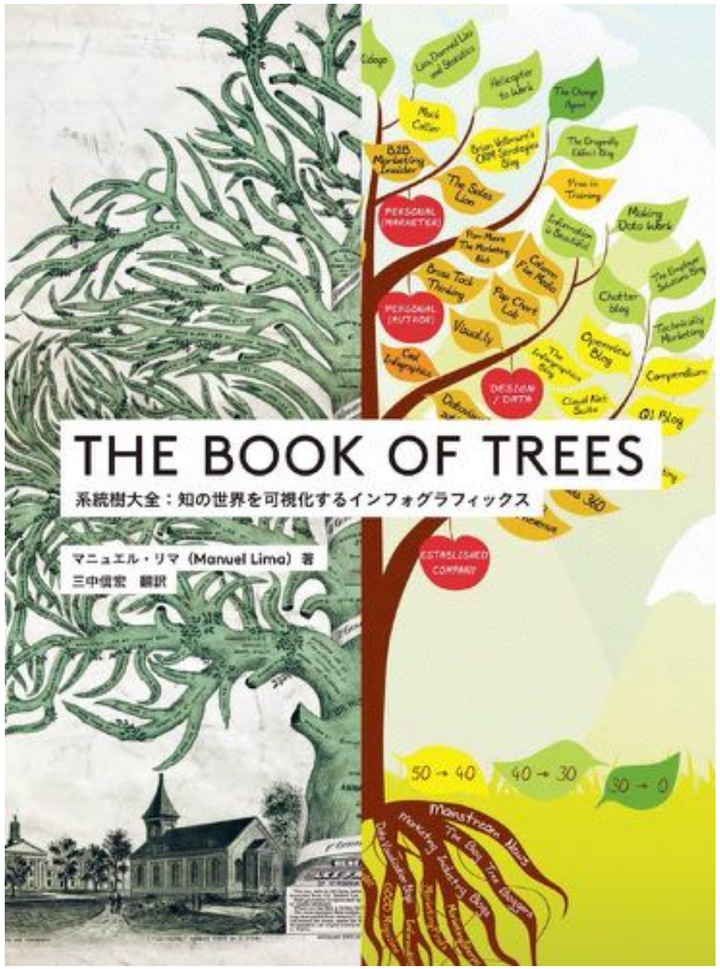
今日の話題提供

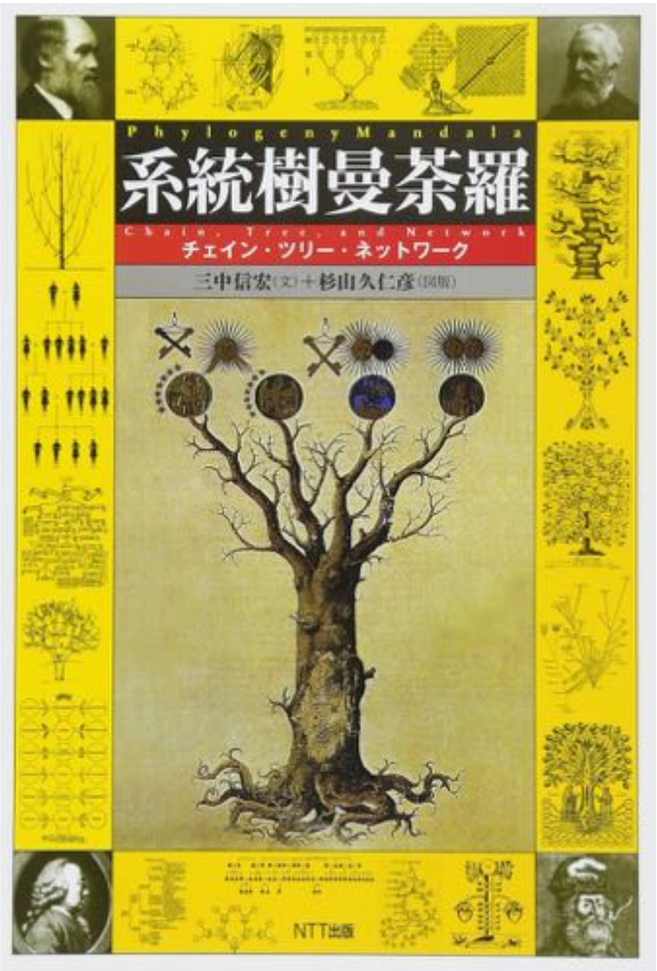
業務(自然科学での機械学習利活用)でユーザとして**決定木アンサンブル(とニューラルネット)**を使っていて出会った現象と問題の紹介

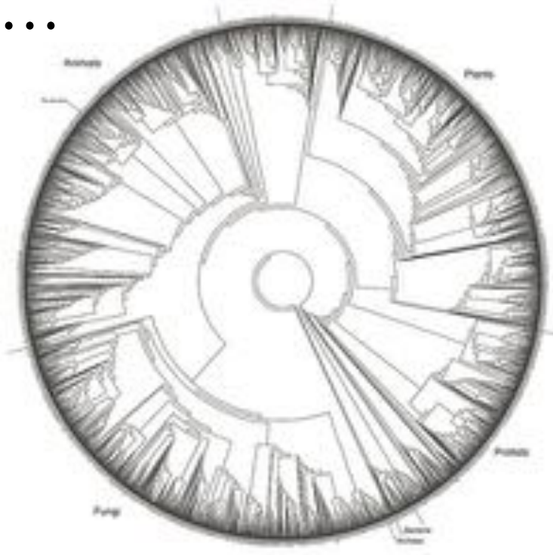
- 決定森回帰の**信頼区間**推定・Benign Overfitting
- 多変量木とReLUネットの**入力空間分割**

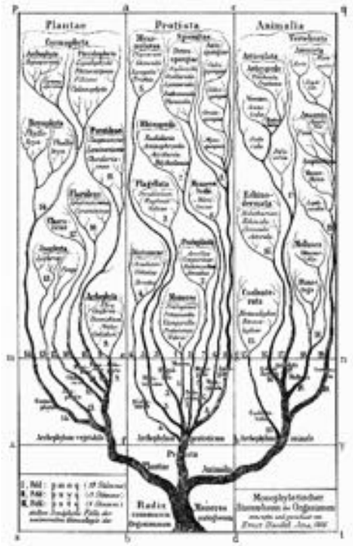
ツリーって良いよね…

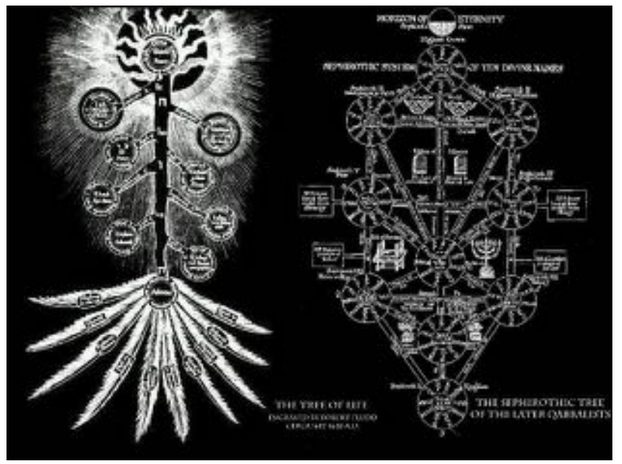
決定木、系統樹、データ構造、ファイルシステム、XML、…





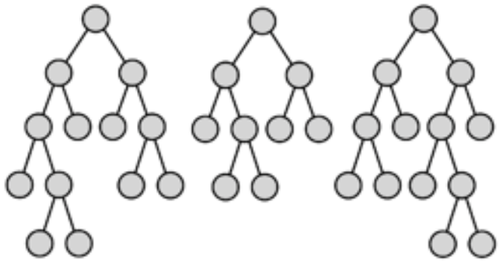


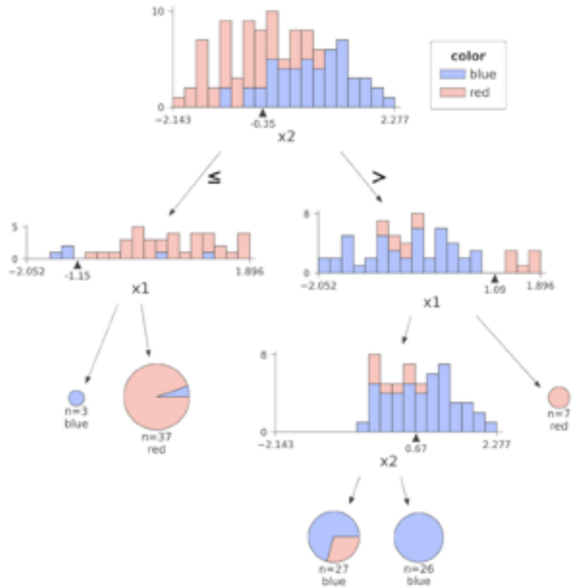




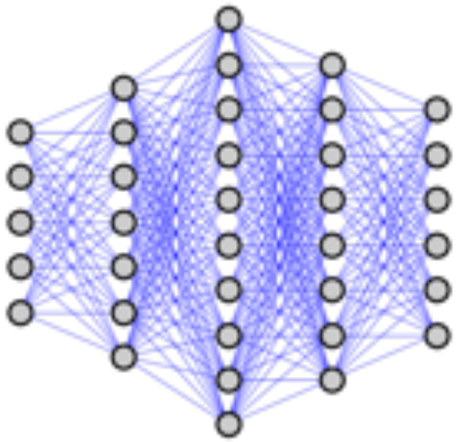
離散構造で定義される機械学習モデル

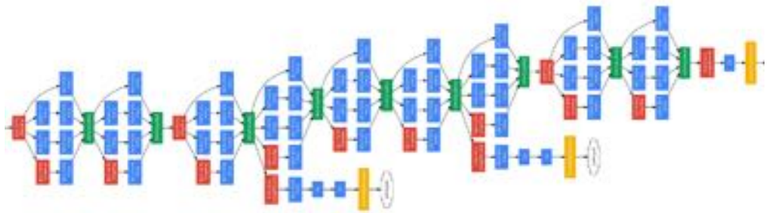
決定森
決定木アンサンブル





ニューラルネットワーク





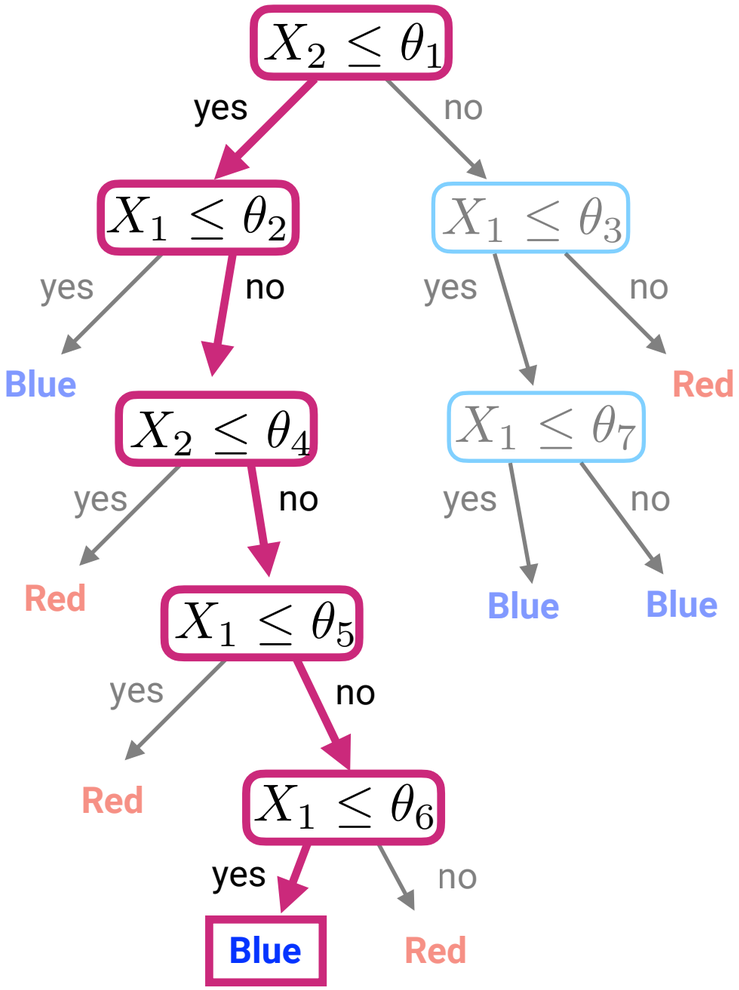
歴史的に密接な関係

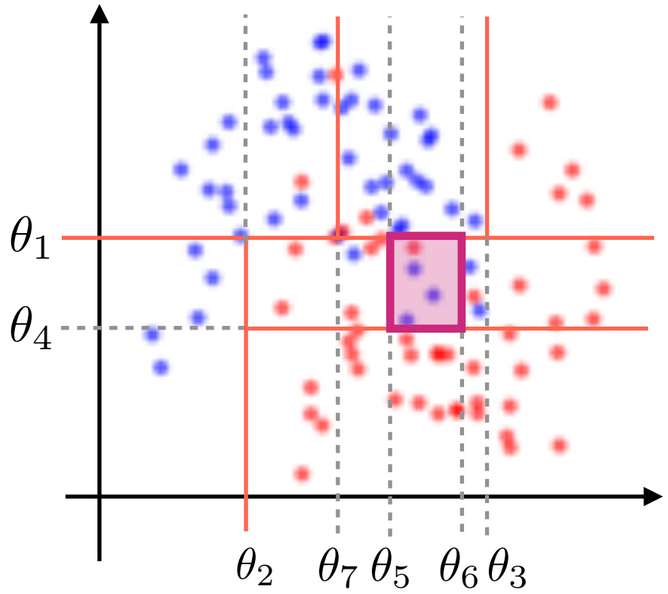
- 融合・発展が人工知能分野の最大の関心事
- オートマトン・論理回路と歴史的には同じ出発点

＋ **論理推論・ルールベース・手続き型などの(旧世代型の?)機械学習モデル**

決定木と論理関数

```
if x2 ≤ θ1 then
  if x1 ≤ θ2 then
    return Blue
  else
    if x2 ≤ θ4 then
      return Red
    else
      if x1 ≤ θ5 then
        return Red
      else
        if x1 ≤ θ6 then
          return Blue
        else
          return Red
else
  if x1 ≤ θ3 then
    if x2 ≤ θ7 then
      return Blue
    else
      return Blue
  else
    return Red
```





決定木は論理式として**選言標準形(積和形)**

$$\begin{cases} T = 1 & \textbf{Blue} \\ T = 0 & \textbf{Not Blue (= Red)} \end{cases}$$

$$T := P_1 \vee P_2 \vee \dots$$

Path

$$P_i := Q_1 \wedge Q_2 \wedge Q_3 \wedge \dots$$

Query

$$Q_j := \begin{cases} X_k \leq \theta_l \\ X_k > \theta_l \end{cases}$$

蛇足：論理とコンピュータと計算機科学と神経回路と人工知能

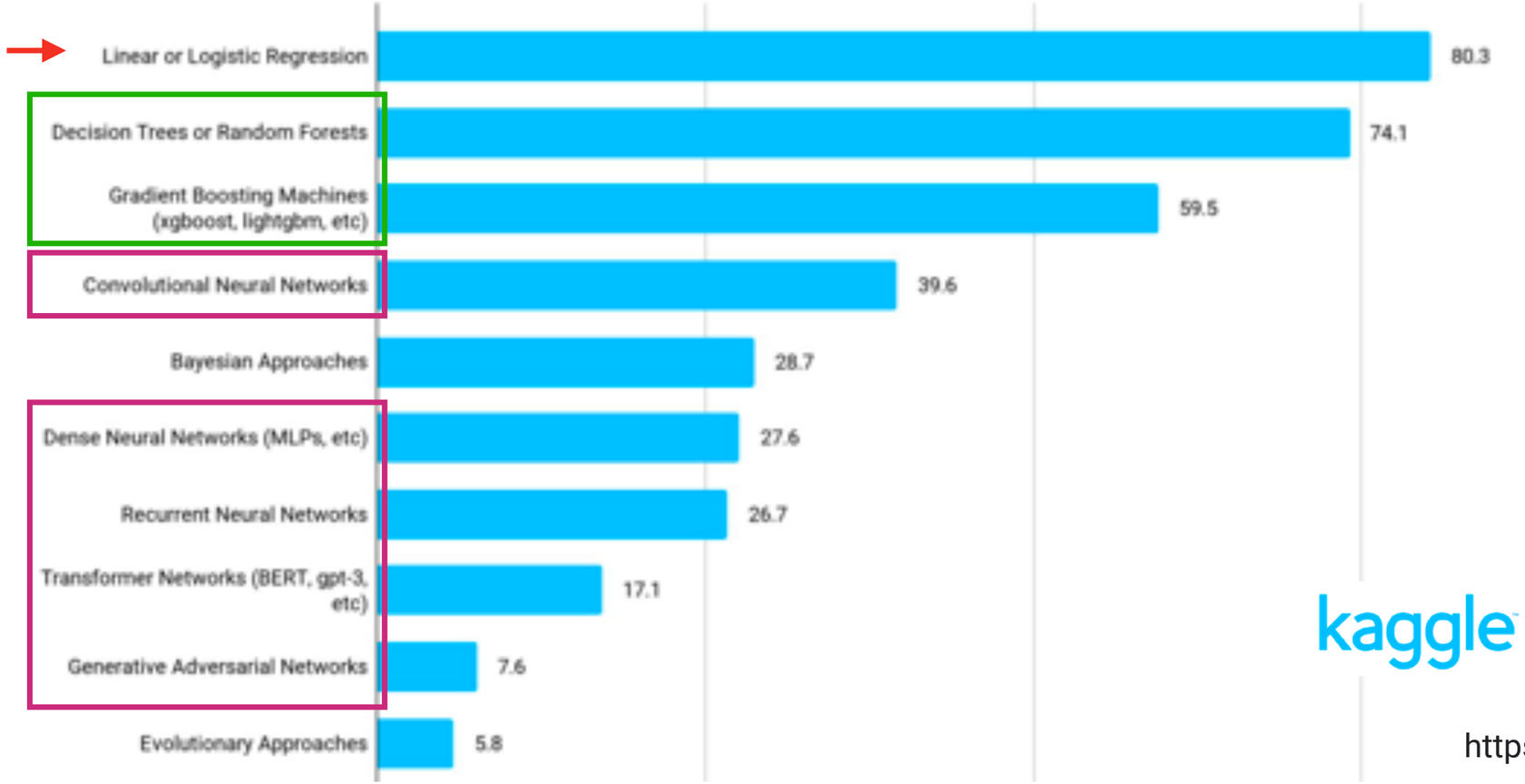
AFSA京都会議で山本章博先生たちに教えてもらった

はじめて人工知能ということばが定義されたことになっている「ダートマス会議」までの計算機概念の歴史

- 「人間のように思考できる機械」を目指して現在の「計算機」概念に至るまで
- **ニューラルネットワーク、オートマトン、形式言語、計算理論、論理回路、などはすべての同じ出発点を持つ**
- バークリー、ウィーナー、フォン・ノイマン、チューリング、クリーネ、シャノン、マカロック・ピッツ、たちの歴史

この２つは線形モデルに次ぐ現代の応用データ科学の主道具

Q17. Which of the following ML algorithms do you use on a regular basis? (Select all that apply)



- ✓ 高精度
- ✓ 高速
- ✓ 非線形
- ✓ カテゴリカル変数の扱い
- ✓ シンプル・解釈しやすい
- ✓ 並列化しやすい

kaggle **State of Data Science and Machine Learning 2021**

https://www.kaggle.com/kaggle-survey-2021

The 2021 Kaggle DS & ML Survey received **25,973 usable responses** from participants in 171 different countries and territories.

今日の話題提供

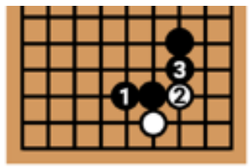
業務(自然科学での機械学習利活用)でユーザとして**決定木アンサンブル(とニューラルネット)**を使っていて出会った現象と問題の紹介

- 決定森回帰の**信頼区間**推定・Benign Overfitting
- 多変量木とReLUネットの**入力空間分割**

Recap 1) 機械学習は新しい(雑な)プログラミング

プログラマがハードコードするのではなく、入出力見本例をたくさん見せて、その入出力を再現することで「プログラム」を生成。Software 2.0、微分可能プログラミング、etc

J’aime la musique



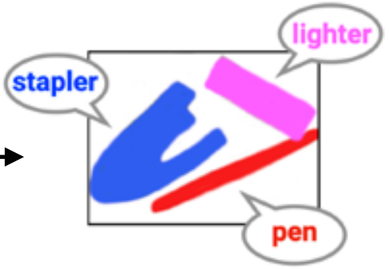
Object recognition

Speech recognition

Machine translation

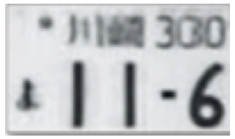
Super resolution

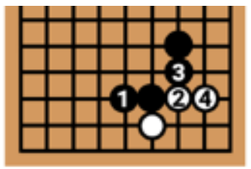
Game play



“ありがとう”

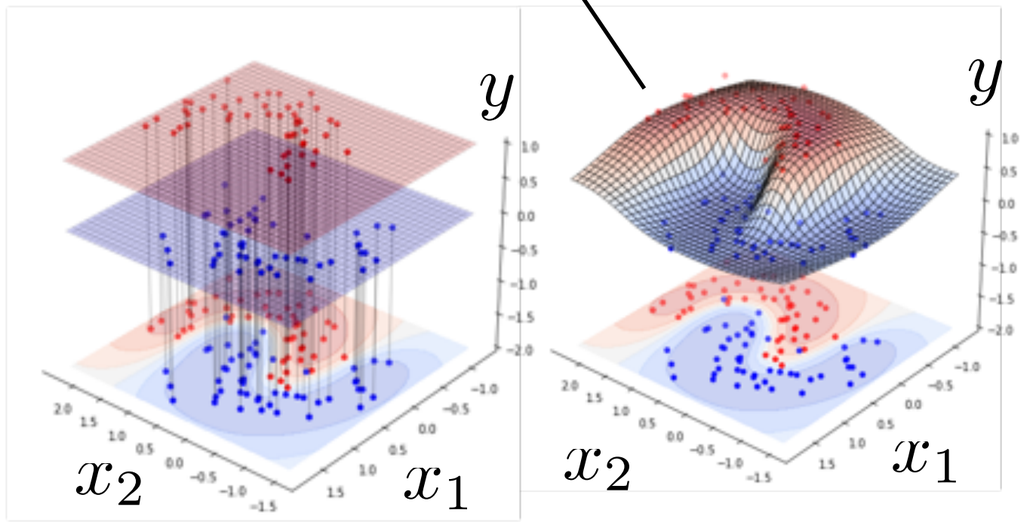
I love music

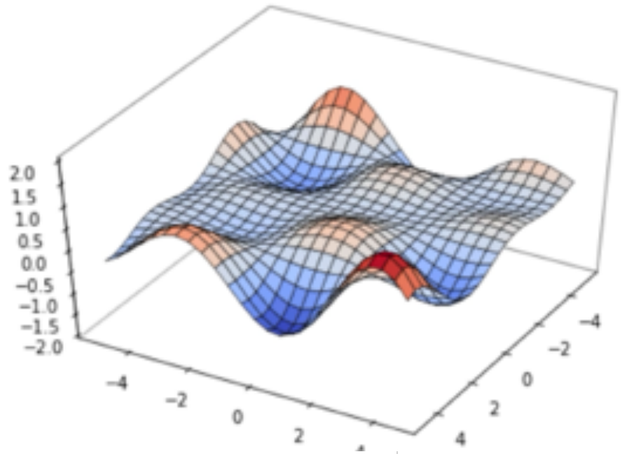


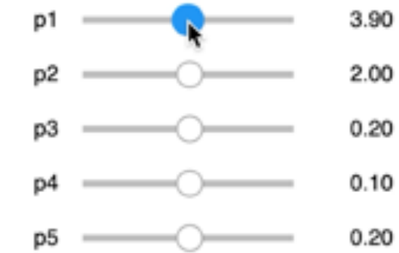


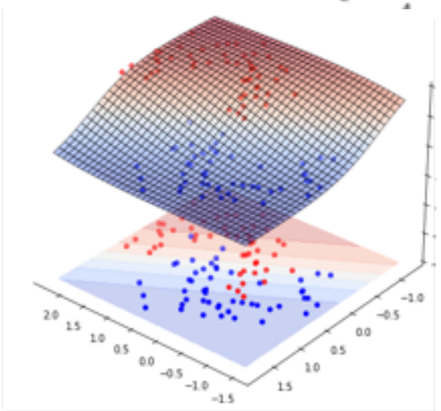
Recap 2) 機械学習は関数モデルによるデータ内挿での予測

関数モデル = プログラム (定義済みの基本演算の合成で作れる入力→出力のマッピング)

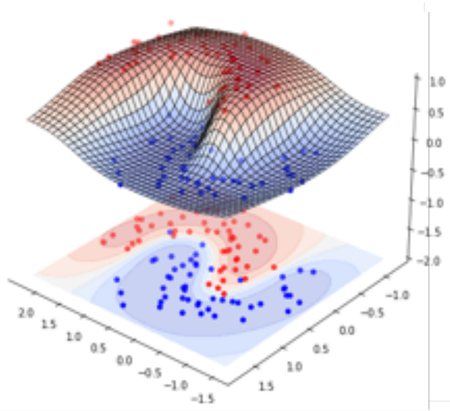




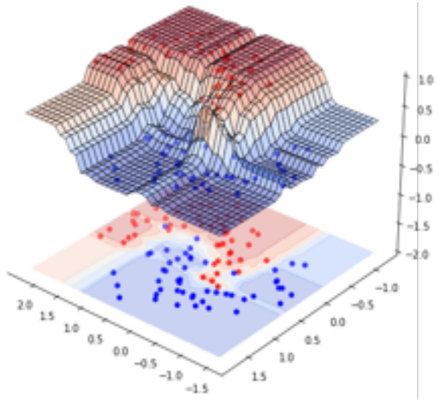




Logistic Regression



Gaussian Process



Random Forest

Recap 3) Breimanの3つの教訓

7.3 Recent Lessons

The advances in methodology and increases in predictive accuracy since the mid-1980s that have occurred in the research of machine learning has been phenomenal. There have been particularly exciting developments in the last five years. What has been learned? The three lessons that seem most important to one:

Rashomon: the multiplicity of good models;
Occam: the conflict between simplicity and accuracy;
Bellman: dimensionality—curse or blessing.

Rashomon

良い機械学習モデルの多重性(非一意性)

Occam

予測精度とシンプルさ(解釈性)のコンフリクト

Bellman

高次元性は呪いか？祝福か？

Breiman L, Statistical Modeling: The Two Cultures. *Statist. Sci.* 16(3): 199-231, 2001.
https://doi.org/10.1214/ss/1009213726

Recap 3) Breimanの3つの教訓

7.3 Recent Lessons

The advances in methodology and increases in predictive accuracy since the mid-1980s that have occurred in the research of machine learning has been phenomenal. There have been particularly exciting developments in the last five years. What has been learned? The three lessons that seem most important to one:

Rashomon: the multiplicity of good models;
Occam: the conflict between simplicity and accuracy;
Bellman: dimensionality—curse or blessing.

Rashomon

良い機械学習モデルの多重性(非一意性)

Occam

予測精度とシンプルさ(解釈性)のコンフリクト

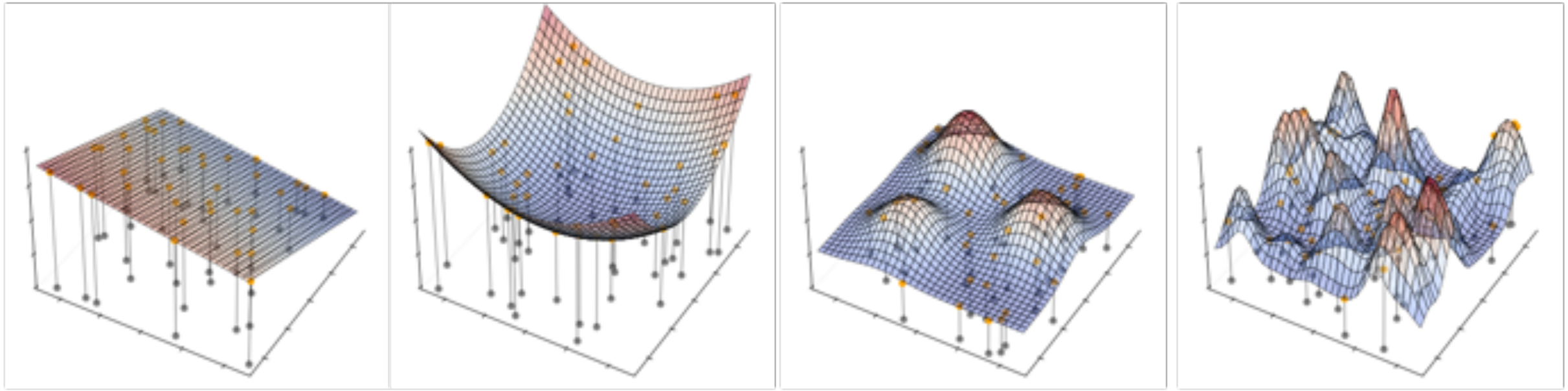
Bellman

高次元性は呪いか？祝福か？

Breiman L, Statistical Modeling: The Two Cultures. *Statist. Sci.* 16(3): 199-231, 2001.
https://doi.org/10.1214/ss/1009213726

Recap 4) 次元の呪い

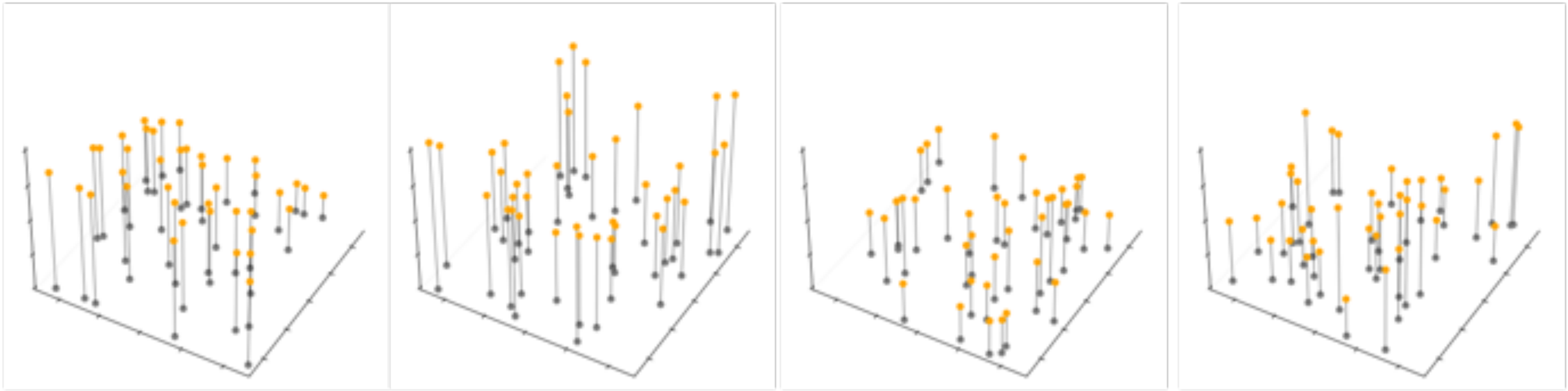
関数面を推定したり予測が合ってるかの検証をしたりするのに見本例は何点くらい必要？



注意：Trainingに必要なのは勿論、ValidationやTestにも必要 (精度推定も統計的推定なので)

Recap 4) 次元の呪い

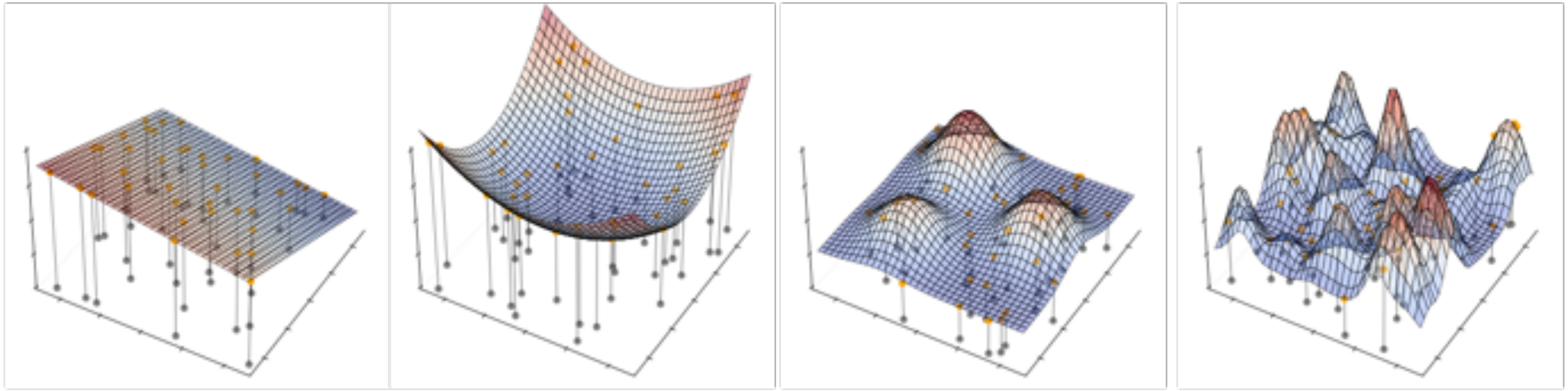
関数面を推定したり予測が合ってるかの検証をしたりするのに見本例は何点くらい必要？



注意：Trainingに必要なのは勿論、ValidationやTestにも必要 (精度推定も統計的推定なので)

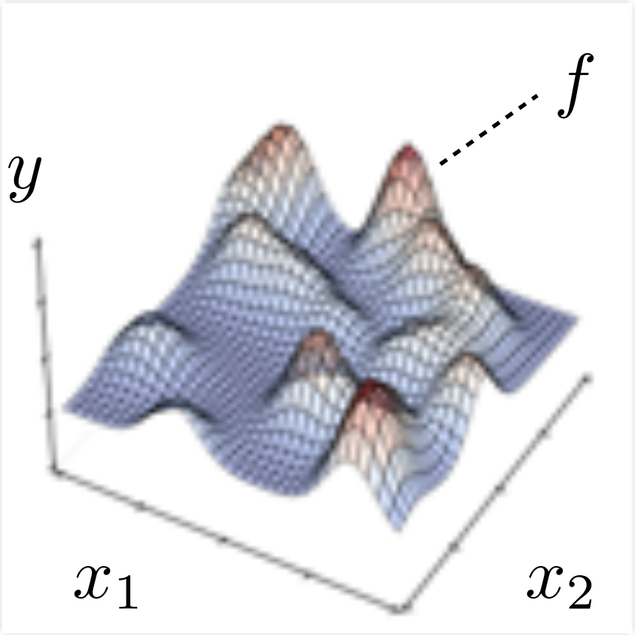
Recap 4) 次元の呪い

関数面を推定したり予測が合ってるかの検証をしたりするのに見本例は何点くらい必要？



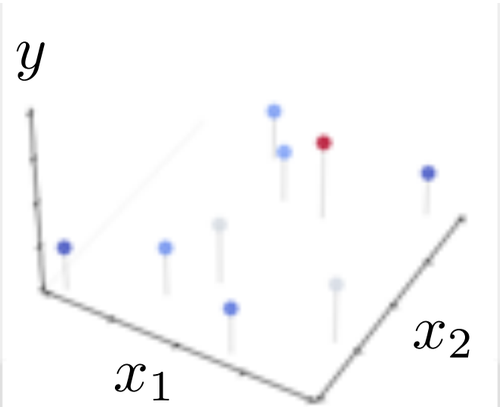
注意：Trainingに必要なのは勿論、ValidationやTestにも必要 (精度推定も統計的推定なので)

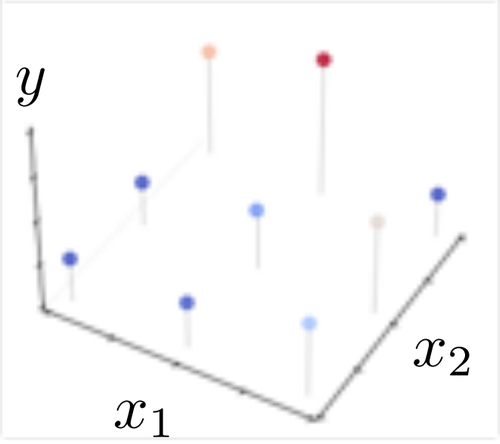
Recap 4) 次元の呪い



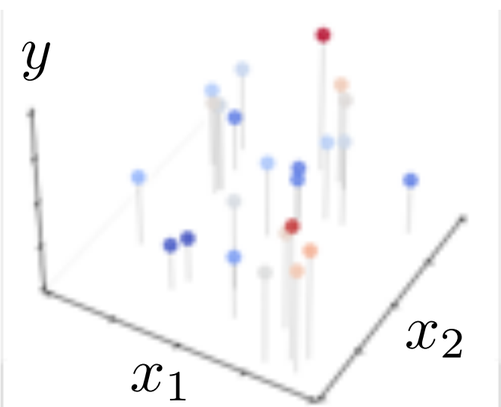
$$x_1, x_2 \longrightarrow f \longrightarrow y$$

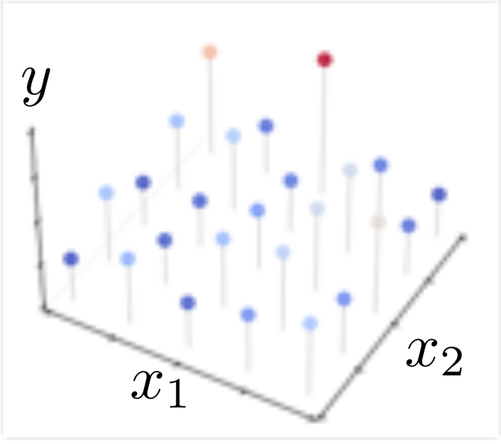
N = 3^2 = 9



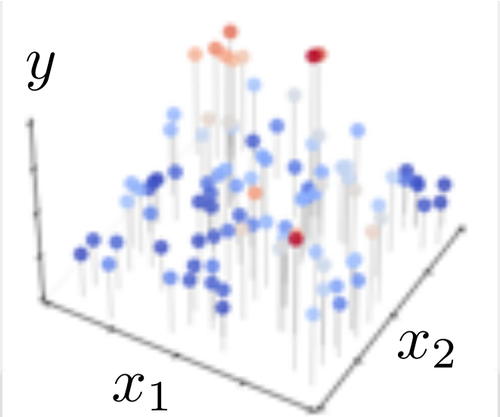


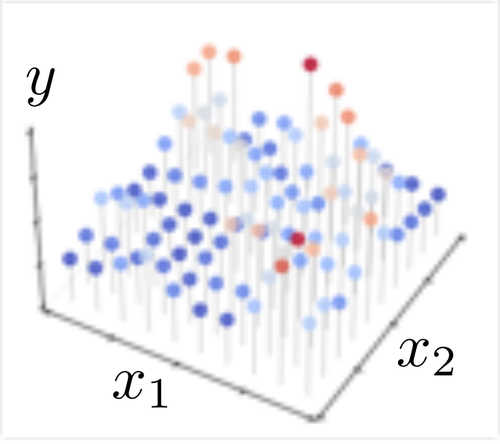
N = 5^2 = 25





N = 10^2 = 100



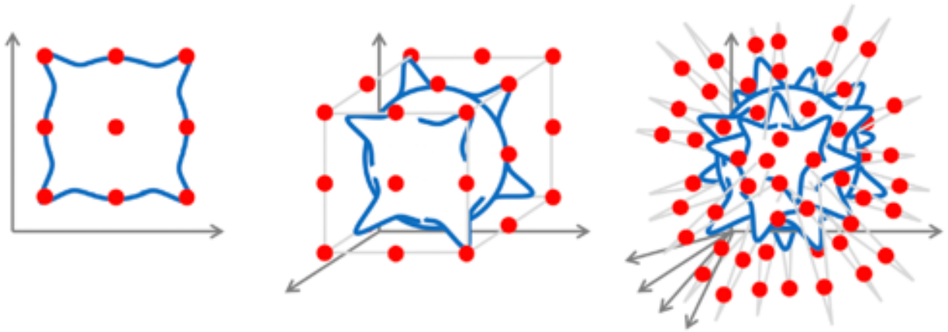


この例で関数概形知るには「100点くらい」は要るかな…？

Recap 4) 次元の呪い: Approximation Theoryの古典的知見

もし**似た入力には似た出力**をという緩い連続性(Lipschitz連続性)しか課さない場合には一様なε近似に$(1/\varepsilon)^d$オーダのサンプル数が必要になる。Sobolev classとかにしても良くならない。

例：d=2で「100点くらい」の精度を求めるなら**d=10では10^{10}=100億点必要**で非現実的…



L-Lipschitz $f : \mathbb{R}^d \to \mathbb{R}$

$$|f(x) - f(x')| \leqslant L\|x - x'\| \quad \text{for all } x, x' \in \mathbb{R}^d$$

Donoho DL,
High-dimensional data analysis: The curses and blessings of dimensionality.
Plenary Lecture, AMS National Meeting on Mathematical Challenges of the 21st Century. 2000.

Bronstein MM, Bruna J, Cohen T, Veličković P.
Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges.
arXiv [cs.LG]. 2021. http://arxiv.org/abs/2104.13478

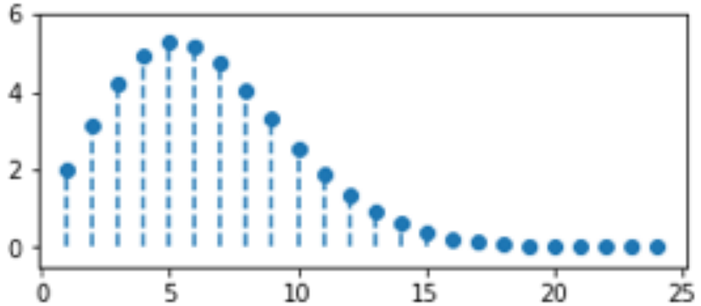
Recap 4) 次元の呪い：高次元空間の非直感性

機械学習やパターン認識の教科書によく出てくる例.
(私は学生時代, C.Bishop "Neural Networks for Pattern Recognition" (1996)で最初に見た)

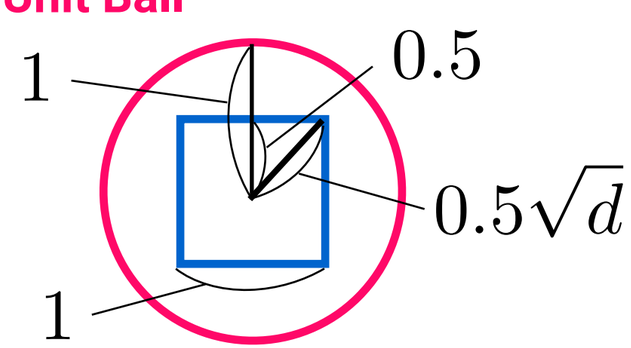
- d次元空間における単位球(r=1の超球)の体積

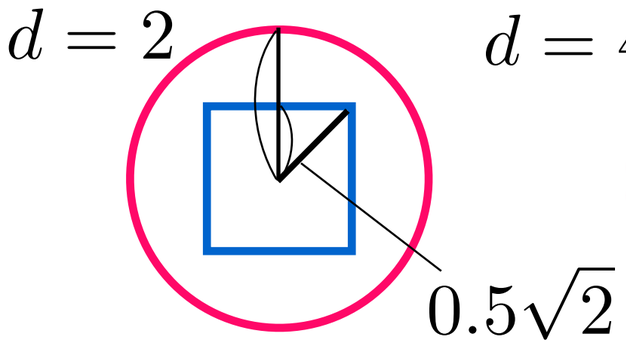
$$V(d) = \frac{\pi^{\frac{d}{2}}}{\frac{d}{2} \cdot \Gamma\left(\frac{d}{2}\right)} \to 0 \quad (d \to \infty)$$

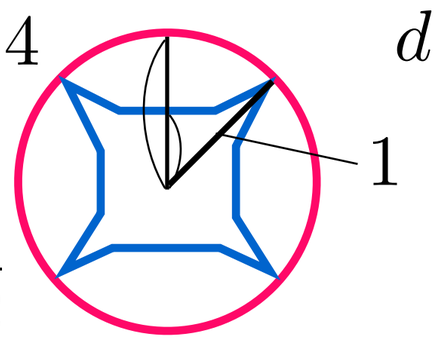
!?

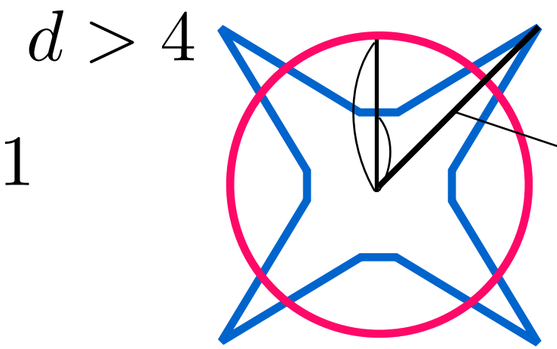


Unit Cube in Unit Ball









$0.5\sqrt{d} > 1$

!?

https://www.math.ucdavis.edu/~strohmer/courses/180BigData/180lecture1.pdf

Recap 4) 次元の呪い：理論上はうまくいくわけなさそう

機械学習の最大の関心：なぜこの無理ゲー設定にも関わらず、わりとうまくいっちゃうの！？

現在の機械学習は**「数百万次元で数千万パラメタの関数フィッティングをしている」**状況で
人間にとっての「ビッグ」データすら全く足りないはずで、**理論上はうまくいくはずがない**

① 高次元性：入力変数が多すぎ！

✓ 機械学習は入力されてない情報を全く考慮してくれない… (擬似相関リスク)

✓ とりあえず色々な変数を入れがち

画像そのままを入力する場合
20×20 ピクセルのカラー画像 → 1200変数
1000×1000 ピクセルのカラー画像 → 300万変数

② 過剰パラメタ化：パラメタ数が多すぎ！

画像
ResNet50: 2600万パラメタ
ResNet101: 4500万パラメタ
EfficientNet-B7: 6600万パラメタ
VGG19: 1億4400万パラメタ

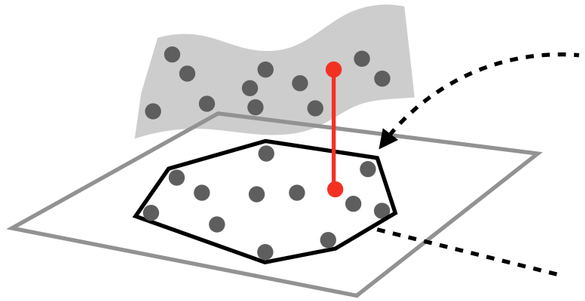
言語
12-layer, 12-heads BERT: 1億1000万パラメタ
24-layer, 16-heads BERT: 3億3600万パラメタ
GPT-2 XL: 15億5800万パラメタ
GPT-3: 1750億パラメタ
Gopher: 2800億パラメタ

Recap 4) 次元の呪い：高次元で内挿なんて起こるわけない？

高次元では内挿なんて起こる確率はゼロ！というネタ.
高次元(d>100)では「内挿とは何か」「内挿か外挿か」
という基本的議論すら**実は極めて非自明**

- *"on any high-dimensional (>100) dataset, interpolation almost surely never happens."*
- *"Those results challenge the validity of our current interpolation/extrapolation definition as an indicator of generalization performances. "*

内挿 = 検査点が**訓練データ点の凸包に落ちた**ときに
周りの点の値からそのy値を決めること



3時間以上に渡る著者対談@MLStreetTalk
https://youtu.be/86ib0sfdFtw

Balestriero R, Pesenti J, LeCun Y.
Learning in High Dimension Always Amounts to Extrapolation.
arXiv [cs.LG]. 2021. http://arxiv.org/abs/2110.09485

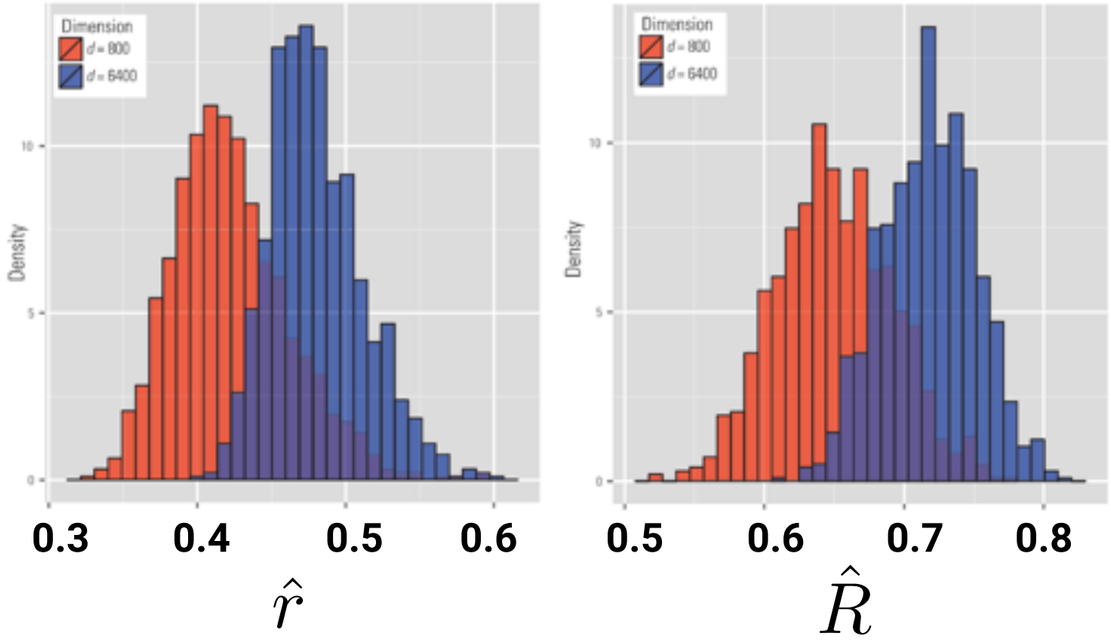
Recap 4) 次元の呪い：高い擬似相関リスク

- **擬似相関のリスク**：大きな変数プール(n変数)からBest Subset回帰(m変数)を探すと
  「本当は全く相関がないにも関わらず」ほぼ常に良い回帰モデルが見つかってしまう！😫
- ケモインフォ界隈では非常に古くから知られているアーチファクト (Topliss 1972, 1979)

$$(X_1, \ldots, X_d) \sim N_d(0, I)$$

$$\hat{r} = \max_{j \geqslant 2} |\mathrm{corr}(X_1, X_j)|$$

$$\hat{R} = \max_{|S|=4} \max_{\beta_j} \left| \mathrm{corr} \left(X_1, \sum_{j \in S} \beta_j X_j \right) \right|$$



n=60を1000回でシミュレート

Fan J, Han F, Liu H.
Challenges of Big Data Analysis. *Natl Sci Rev*. 2014;1: 293−314.

Recap 4) 次元の呪い：測度の集中現象

- **測度の集中現象**：高次元空間ではサンプル点間の距離がすべてほとんど同じになってしまう
- 距離尺度で情報フィルタリングをする場合、**高次元になるとほぼ全検索に近くなる**ことがデータベースや情報検索業界で指摘されてきた。

  Beyer+ 1999の例:

  n+1個のd次元点　$Q, Z_1, \ldots, Z_n \overset{\text{iid}}{\sim} \prod_{i=1}^{d} P$

$$\lim_{d\to\infty} P\left[\frac{\max_i d(Q,\{Z_i\}) - \min_i d(Q,\{Z_i\})}{\min_i d(Q,\{Z_i\})} \leqslant \varepsilon\right] = 1 \text{ for every } \varepsilon > 0$$

K. Beyer+, When Is “Nearest Neighbor” Meaningful? ICDT’99
V. Pestov, On the geometry of similarity search: dimensionality curse and concentration of measure, *Information Processing Letters*, 1999.

現在の機械学習の主関心：どうやって高次元性を手懐ける？

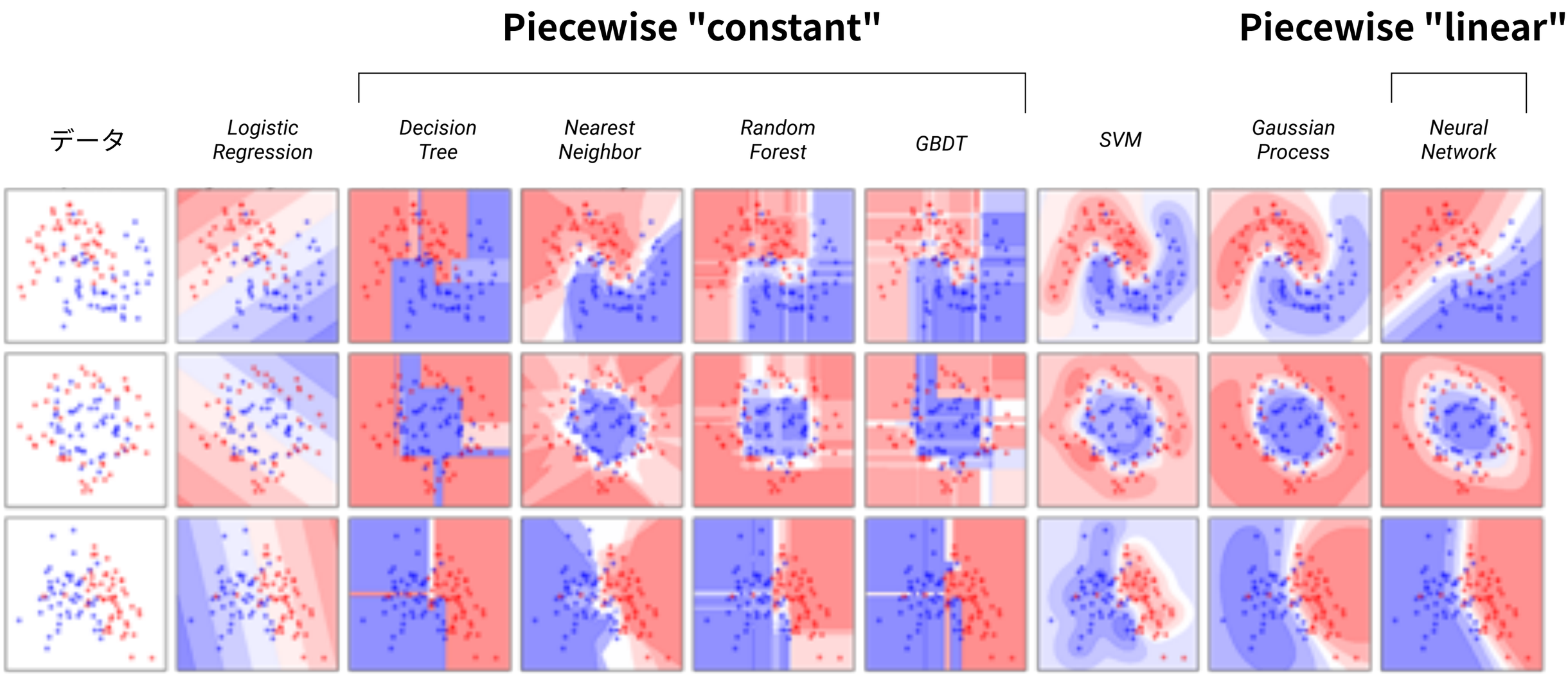
いろいろな研究がある…
が、基本的には目前のタスクによくマッチする「良い帰納バイアス」のデザインの問題？

Inductive Bias

高い自由度を持つ機械学習モデルが不都合・不適当な関数を意図せず表現してしまわないようモデル空間や学習方式やモデル構造を**制限・制約・制御する(意図的バイアスを課す)**

1. **正則化のデザイン (局所的安定性・スパース性・連続性・ロバスト性・etc)**
   Explicitな正則化だけではなく確率的摂動(SGD等)によるImplicit regularizationも
2. **良い感じの初期値のデザイン**
   大規模事前学習の転移などによる良い「Warm Start」の設計
3. **モデル構造や入力変数・入力表現やタスク構造のデザイン**
   深層学習の構造デザインパターン、特徴量エンジニアリング、幾何的深層学習、多モーダル

Recap: 決定森と最近隣法 = Piecewise constant predictors



Recap: 決定森と最近隣法 = Piecewise constant predictors

"We introduce a concept of potential nearest neighbors (*k*-PNNs) and show that random forests can be viewed as adaptively weighted *k*-PNN methods. "

Random Forests and Adaptive Nearest Neighbors

Yi LIN and Yongho JEON

In this article we study random forests through their connection with a new framework of adaptive nearest-neighbor methods. We introduce a concept of potential nearest neighbors (*k*-PNNs) and show that random forests can be viewed as adaptively weighted *k*-PNN methods. Various aspects of random forests can be studied from this perspective. We study the effect of terminal node sizes on the prediction accuracy of random forests. We further show that random forests with adaptive splitting schemes assign weights to *k*-PNNs in a desirable way: for the estimation at a given target point, these random forests assign voting weights to the *k*-PNNs of the target point according to the local importance of different input variables. We propose a new simple splitting scheme that achieves desirable adaptivity in a straightforward fashion. This simple scheme can be combined with existing algorithms. The resulting algorithm is computationally faster and gives comparable results. Other possible aspects of random forests, such as using linear combinations in splitting, are also discussed. Simulations and real datasets are used to illustrate the results.

KEY WORDS: Adaptive estimation; Boosting; Classification trees; Randomized trees; Regression trees.

Journal of the American Statistical Association , Jun., 2006, Vol. 101, No. 474 (Jun., 2006), pp. 578-590

https://www.jstor.org/stable/27590719

Recap: 決定森と最近隣法 = Piecewise constant predictors

Journal of Multivariate Analysis 101 (2010) 2499–2518

Contents lists available at ScienceDirect

Journal of Multivariate Analysis

journal homepage: www.elsevier.com/locate/jmva

On the layered nearest neighbour estimate, the bagged nearest neighbour estimate and the random forest method in regression and classification

Gérard Biau [a,b,*], Luc Devroye [c]

[a] LSTA & LPMA, Université Pierre et Marie Curie—Paris VI, Boîte 158, Tour 15-25, 2ème étage, 4 place Jussieu, 75252 Paris Cedex 05, France
[b] DMA, Ecole Normale Supérieure, 45 rue d'Ulm, 75230 Paris Cedex 05, France
[c] School of Computer Science, McGill University, Montreal, Canada H3A 2K6

ARTICLE INFO

Article history:

ABSTRACT

Let $\mathbf{X}_1, \ldots, \mathbf{X}_n$ be identically distributed random vectors in $\mathbb{R}^d$, independently drawn

複雑な関数に柔軟にフィットするためには加えて

- **ランダムネス**
- **アンサンブル**

が鍵になっている?

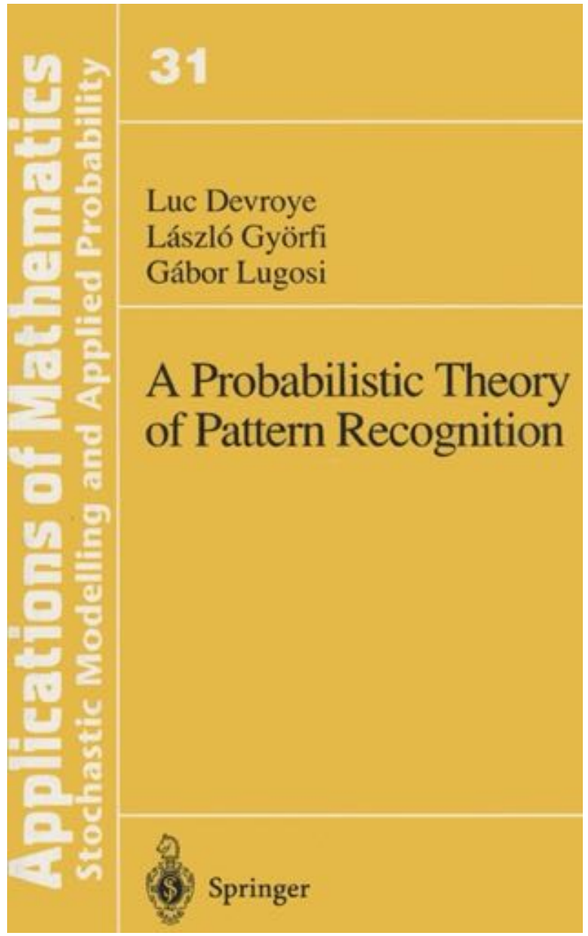
Recap: 決定森と最近隣法

最近隣法なんて…と思うと思いますが、アホでも分かる簡素さと
実用性＋非常に優れた理論的性質を兼ね備えている…

Stoneのユニバーサル一致性定理(1977)の影響で最近隣法は80年代の
ノンパラメトリック統計の理論コミュニティの一大関心事

Interestingly, until 1977, it was not known if a universally consistent rule existed. All pre-1977 consistency results came with restrictions on (X, Y). In 1977, Stone showed that one could just take any k-nearest neighbor rule with $k = k(n) \to \infty$ and $k/n \to 0$. The k-nearest neighbor classifier $g_n(x)$ takes a majority vote over the Y_i's in the subset of k pairs (X_i, Y_i) from $(X_1, Y_1), \ldots, (X_n, Y_n)$ that have the smallest values for $\|X_i - x\|$ (i.e., for which X_i is closest to x). Since Stone's proof of the universal consistency of the k-nearest neighbor rule, several other rules have been shown to be universally consistent as well. This book stresses universality and hopefully gives a reasonable account of the developments in this direction.

The Bible **or "The Yellow Terror"**
by Luc, Laci, Gábor



Recap: 決定森と最近隣法

最近隣法なんて…と思うと思いますが、アホでも分かる簡素さと
実用性＋非常に優れた理論的性質を兼ね備えている…

Stoneのユニバーサル一致性定理(1977)の影響で最近隣法は80年代の
ノンパラメトリック統計の理論コミュニティの一大関心事

↕

└→ **生成器の「型」(正規分布とか)を仮定しない**

古典的な推測統計学 (パラメトリック統計)

ゲーム設定＝観測例の値だけから生成器のパラメタ値(μとσ)を当てられる？

$N(\mu = \square, \sigma = \square)$

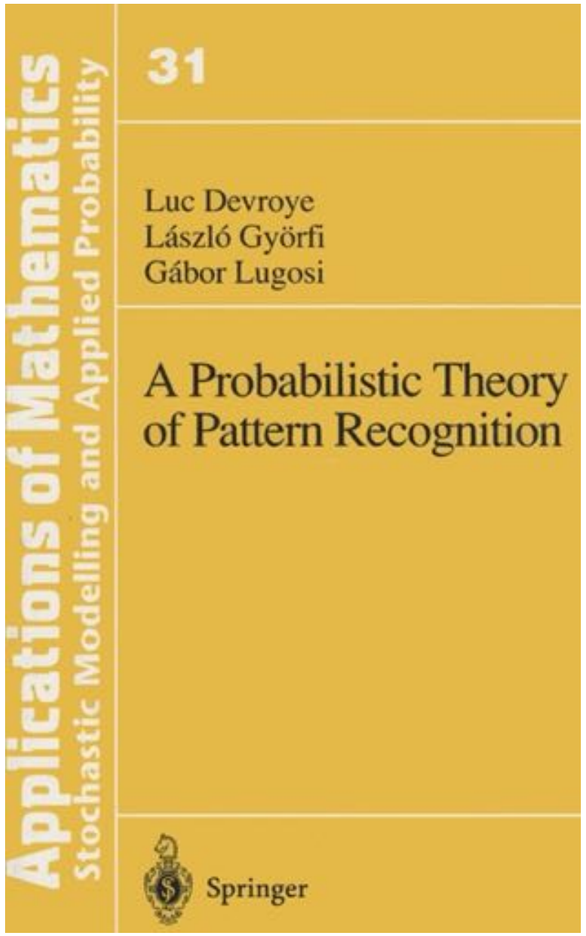
```
np.random.normal(loc=[redacted],
                 scale=[redacted]
                 size=50)
```

？

乱数生成

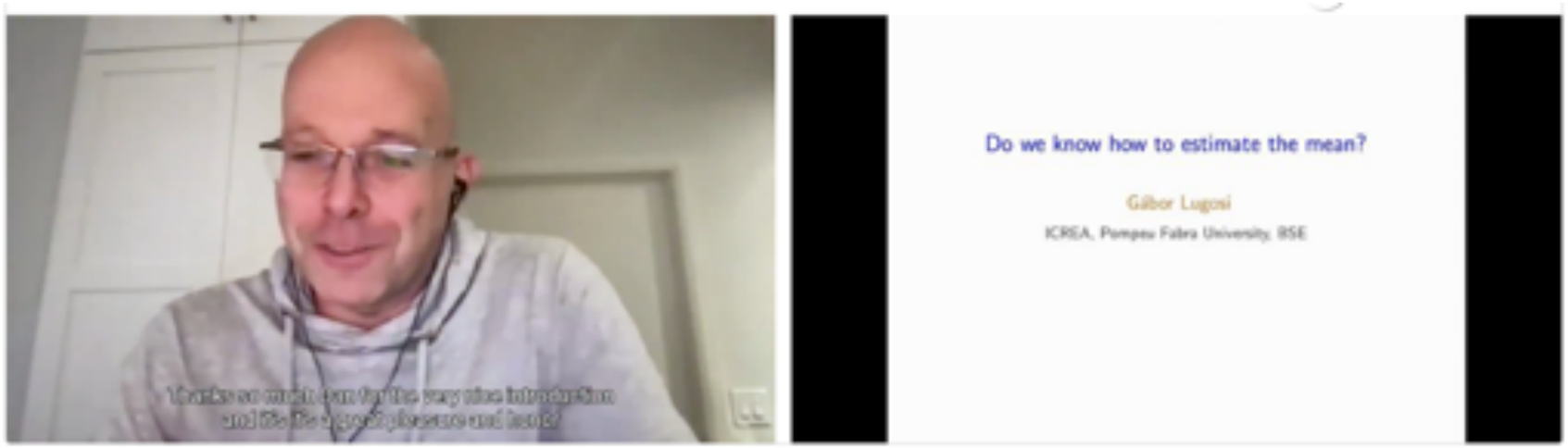
```
[-5.91856619, -4.67299272, -2.98381106, -3.98874017, -7.07799836,
 -0.92568268, -1.74559329, -3.18887676, -1.10733331, -1.15926761,
  2.30246681, -0.82358339, -1.3801766 , -0.54597068, -0.89262521,
 -0.27022552, -1.44010658,  0.17471008, -0.351082  , -2.41667565,
 -1.47623191, -4.38890872, -1.21671228,  1.33280656,  1.69434986,
  0.19073689,  0.94328555, -3.38047511, -0.99864808, -2.69899999,
 -0.71580737,  2.89350466, -0.41668164, -2.47111573, -1.30012492,
 -0.33235115, -2.28290987, -2.69598786, -3.92504614, -2.9469401 ,
 -3.12949327,  2.39505039,  3.11259008, -5.86736858, -3.09089171,
  2.36510819, -2.27716495, -1.47832675, -2.94300436,  0.37291792]
```

The Bible **or "The Yellow Terror"**
by Luc, Laci, Gábor



蛇足: Gábor Lugosi

NeurIPS 2021 Invited Talk (Breiman Lecture)
Do we know how to estimate the mean?



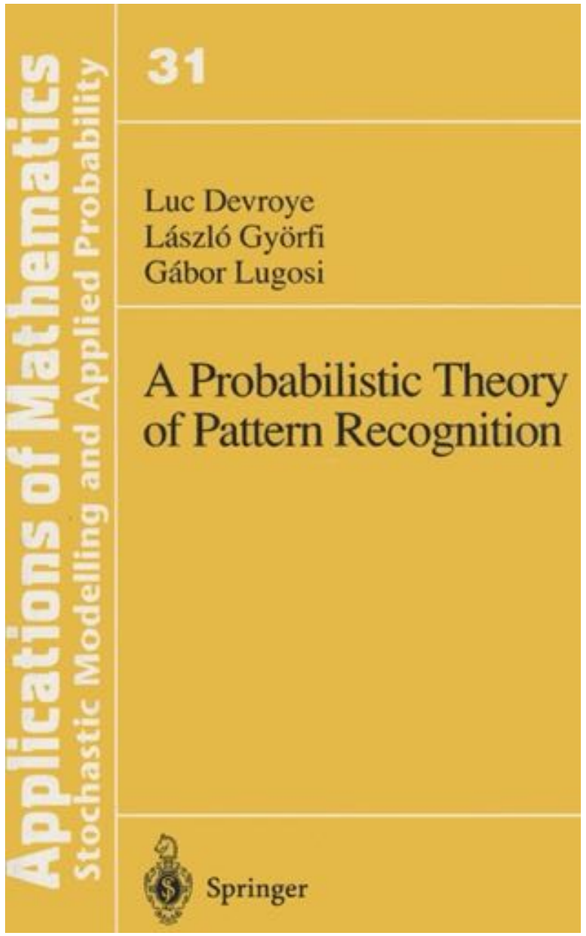
Given $X_1, \ldots, X_n$, a real i.i.d. sequence, estimate $\mu = \mathbb{E}X_1$

↑ ただiid確率変数の**平均値を推定**する話 (!)

$(\sum_{i=1}^{n} X_i)/n$ より良いのあるの？

アンサンブルと
密な関係！

The Bible **or "The Yellow Terror"**
by Luc, Laci, Gábor



Recap: 決定森と最近隣法

Journal of Machine Learning Research, 9, 2015-2033, 2008

Consistency of Random Forests and Other Averaging Classifiers

Gérard Biau GERARD.BIAU@UPMC.FR
LSTA & LPMA
Université Pierre et Marie Curie – Paris VI
Boîte 158, 175 rue du Chevaleret
75013 Paris, France

Luc Devroye LUC@CS.MCGILL.CA
School of Computer Science
McGill University
Montreal, Canada H3A 2K6

Gábor Lugosi LUGOSI@UPF.ES
ICREA and Department of Economics
Pompeu Fabra University
Ramon Trias Fargas 25-27
08005 Barcelona, Spain

Editor: Peter Bartlett

Abstract

In the last years of his life, Leo Breiman promoted random forests for use in classification. He suggested using averaging as a means of obtaining good discrimination rules. The base classifiers used for averaging are simple and randomized, often based on random samples from the data. He left a few questions unanswered regarding the consistency of such rules. In this paper, we give a number of theorems that establish the universal consistency of averaging rules. We also show that some popular classifiers, including one suggested by Breiman, are not universally consistent.

Keywords: random forests, classification trees, consistency, bagging

This paper is dedicated to the memory of Leo Breiman.

The Annals of Statistics, 43(4), 1716-1741, 2015.

The Annals of Statistics
2015, Vol. 43, No. 4, 1716–1741
DOI: 10.1214/15-AOS1321

CONSISTENCY OF RANDOM FORESTS[1]

BY ERWAN SCORNET*, GÉRARD BIAU* AND JEAN-PHILIPPE VERT†

Sorbonne Universités and MINES ParisTech, PSL-Research University†*

Random forests are a learning algorithm proposed by Breiman [*Mach. Learn.* **45** (2001) 5–32] that combines several randomized decision trees and aggregates their predictions by averaging. Despite its wide usage and outstanding practical performance, little is known about the mathematical properties of the procedure. This disparity between theory and practice originates in the difficulty to simultaneously analyze both the randomization process and the highly data-dependent tree structure. In the present paper, we take a step forward in forest exploration by proving a consistency result for Breiman's [*Mach. Learn.* **45** (2001) 5–32] original algorithm in the context of additive regression models. Our analysis also sheds an interesting light on how random forests can nicely adapt to sparsity.

今日の話題提供

業務(自然科学での機械学習利活用)でユーザとして**決定木アンサンブル(とニューラルネット)**を使っていて出会った現象と問題の紹介

- 決定森回帰の**信頼区間**推定・Benign Overfitting
- 多変量木とReLUネットの**入力空間分割**

Recap 3) Breimanの3つの教訓

7.3 Recent Lessons

The advances in methodology and increases in predictive accuracy since the mid-1980s that have occurred in the research of machine learning has been phenomenal. There have been particularly exciting developments in the last five years. What has been learned? The three lessons that seem most important to one:

Rashomon: the multiplicity of good models;
Occam: the conflict between simplicity and accuracy;
Bellman: dimensionality—curse or blessing.

Rashomon

良い機械学習モデルの多重性(非一意性)

Occam

予測精度とシンプルさ(解釈性)のコンフリクト

Bellman

高次元性は呪いか？祝福か？

Breiman L, **Statistical Modeling: The Two Cultures.** *Statist. Sci.* 16(3): 199-231, 2001.
https://doi.org/10.1214/ss/1009213726

Rashomon Effect

Rashomon Effect: 良い機械学習モデルの多重性 (非一意性)

例　機械学習コンペでは一つのデータセットに対して**多様な**モデルが提出される。
　　予測精度は上位はだんご状態になり**実用上はほぼ同等に良い**モデルとみなせる。

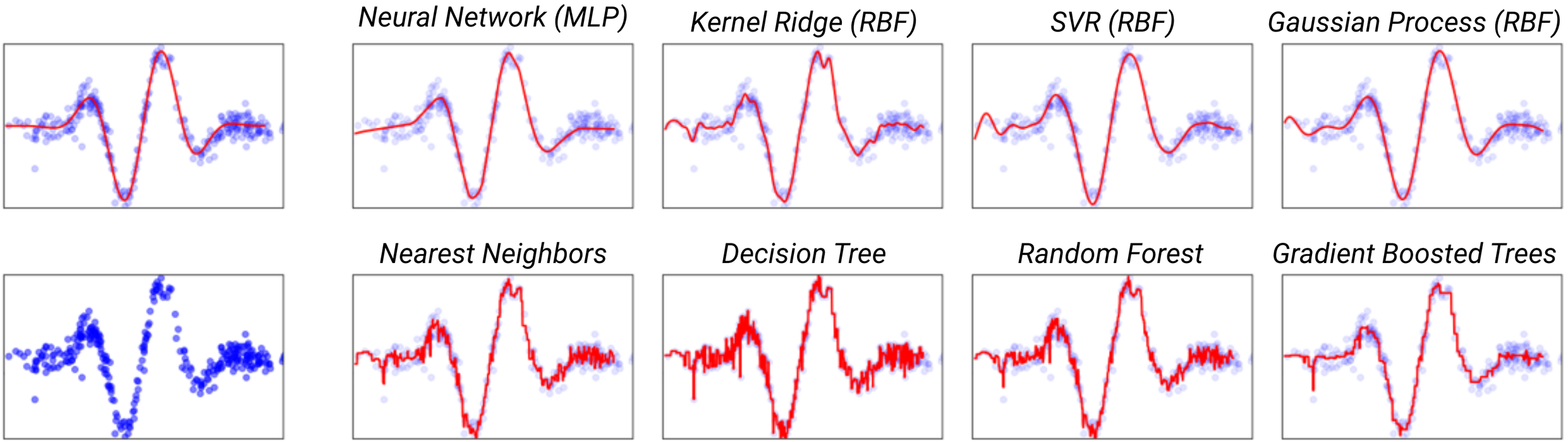
- *"What I call the **Rashomon Effect** is that there is often a multitude of different descriptions [equations f(x)] in a class of functions giving about the same minimum error rate. The most easily understood example is subset selection in linear regression."*
- *"The Rashomon Effect also occurs with decision trees and neural nets."*
- *"This effect is closely connected to what I call **instability** (Breiman,1996a) that occurs when there are many different models crowded together that have about the same training or test set error."*

Breiman L, **Statistical Modeling: The Two Cultures.** *Statist. Sci.* 16(3): 199-231, 2001.
https://doi.org/10.1214/ss/1009213726

Rashomon EffectとUnderspecification

結局、訓練データも(検証データも)テストデータも有限だけど「高次元空間」では
真の関数があるとしても見本数不足でそれを特定(specify)しきれないから？

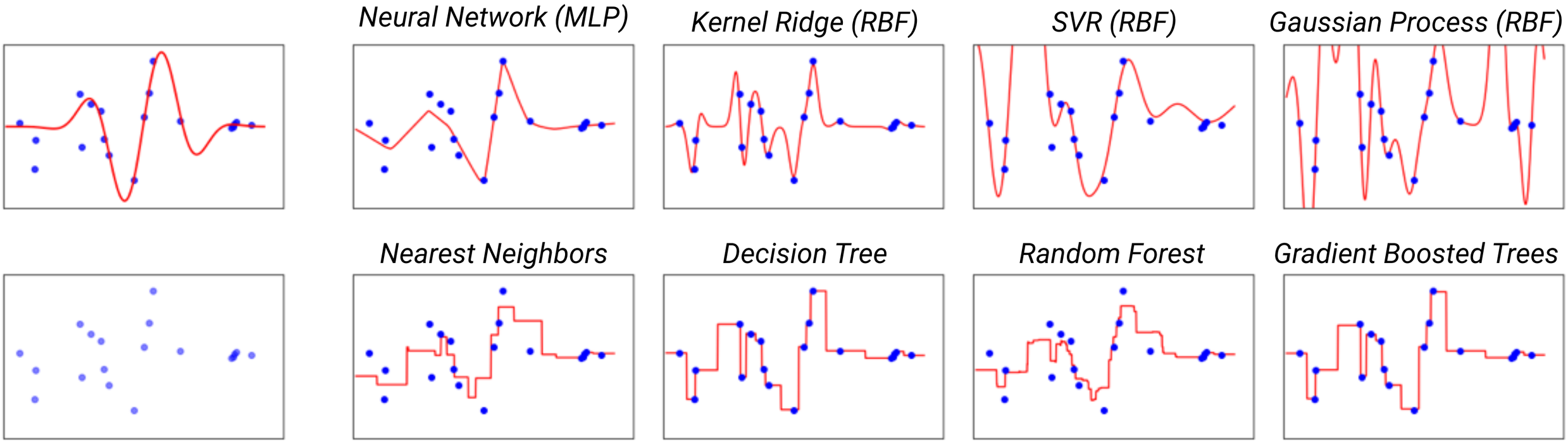
もしサンプル数が十分なら現代的な非線形手法ならだいたいどれを選んでもOKなはず



Rashomon EffectとUnderspecification

結局、訓練データも(検証データも)テストデータも有限だけど「高次元空間」では
真の関数があるとしても見本数不足でそれを特定(specify)しきれないから？

Underspecifiedな状況の場合、用いる手法やハイパーパラメタによってだいぶ異なる結果に



Underspecificationはビッグデータ事例でも起こっている？

「良い教師ラベルがつけられた大規模データの実例でも」起こっているのかも？

"While ML models are validated on **held-out data**, this validation is often insufficient to guarantee that the models will have well-defined behavior when they are used **in a new setting.** "

https://ai.googleblog.com/2021/10/how-underspecification-presents.html

Google AI Blog

The latest from Google Research

How Underspecification Presents Challenges for Machine Learning

Monday, October 18, 2021

Posted by Alex D'Amour and Katherine Heller, Research Scientists, Google Research

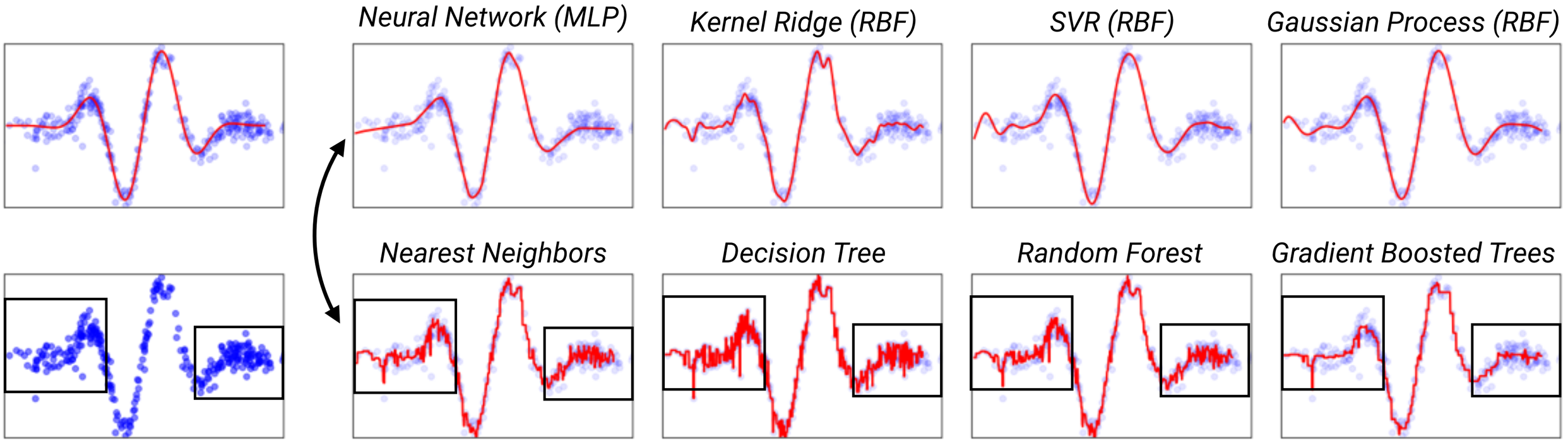
Machine learning (ML) models are being used more widely today than ever before and are becoming increasingly impactful. However, they often exhibit unexpected behavior when they are used in real-world domains. For example, computer vision models can exhibit surprising sensitivity to irrelevant features, while natural language processing models can depend unpredictably on demographic correlations not directly indicated by the text. Some reasons for these failures are well-known: for example, training ML models on poorly curated data, or training models to solve

https://arxiv.org/abs/2011.03395

考えたい点①

「決定木アンサンブルでは**回帰曲線まわりで回帰値がぶれやすい**」はマズい特性か？

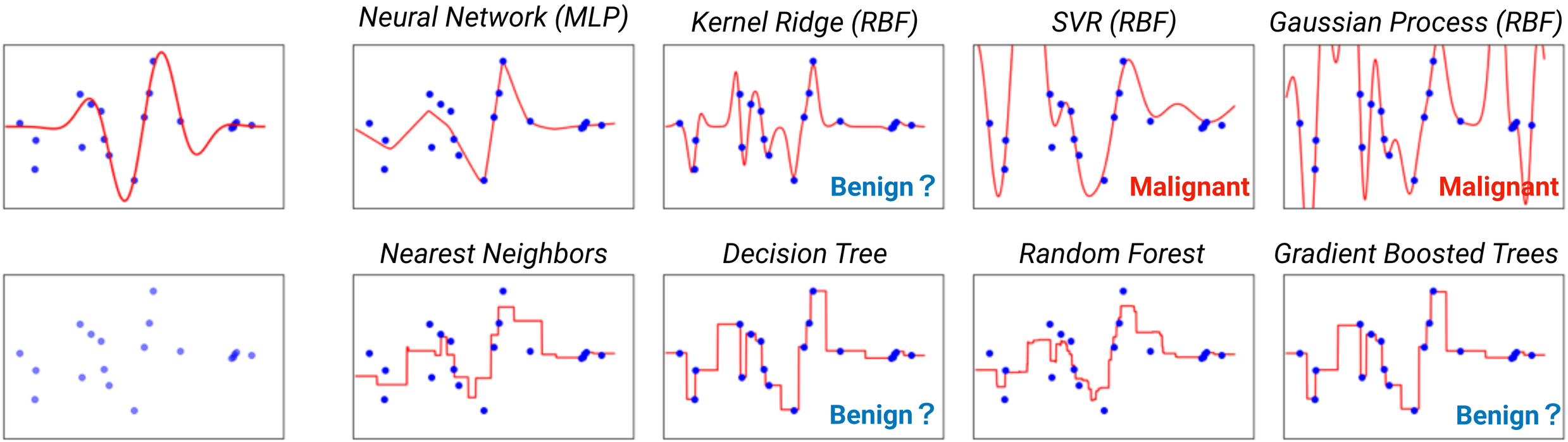
→ もともとのサンプル値が真の傾向からランダム変動しているのだからこの程度予測値が振れるのは逆に健全では？ **「予測分散」を計算して**付与すればむしろ実用用途でもGoodなはず



考えたい点②

「決定木は**Overfitting**しやすい」はマズい特性か？

→ アンサンブル学習への主動機になっている。ただし、この「Overfitting」は必ずしも有害とは言えない (Benign overfitting)。**ノイズあり事例で訓練誤差0**でもそれなりに役に立つ！

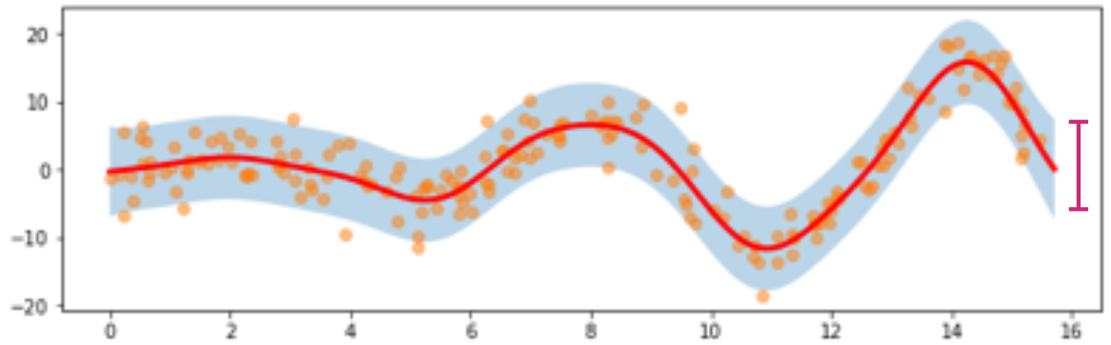


考えたい点① 決定森回帰の予測分散(信頼区間)の推定

回帰を実際に意思決定に活用する際には、予測値だけでなくその**信頼度(不確実性)の情報**がとても重要

A. この商品の売り上げ予測値は21.3 (±0.5)です！
B. この商品の売り上げ予測値は21.3 (±10.9)です！

✓ **予測値の分散**さえ算出できれば(正規近似で)**信頼区間**や逐次実験計画に活用する**期待値改善量(Expected Improvement)**などの量も計算できる！



95%信頼区間

$$\rightarrow \quad \bar{y}(x) \pm 1.96\,\sigma(x)$$

考えたい点① 決定森回帰の予測分散 (Random Forest型)

- ✓ 回帰木単体で自然な予測分散を持つ
  ➡ 予測点が落ちた領域にある訓練サンプルのyの「分散」
- ✓ Random Forest型の場合は決定木が基本的に独立なので、各々の回帰木の予測分散を統合してアンサンブルの予測分散を算出

$$
\begin{aligned}
\mu = \mathbb{E}[L] &= \frac{1}{B}\sum_{b=1}^{B}\mu_b; \\
\sigma^2 = \mathrm{Var}(L) &= \mathbb{E}[\mathrm{Var}(L|Y)] + \mathrm{Var}(\mathbb{E}[L|Y]) \\
&= \left(\frac{1}{B}\sum_{b=1}^{B}\sigma_b^2\right) + \left(\mathbb{E}[\mathbb{E}(L|Y)^2] - \mathbb{E}[\mathbb{E}(L|Y)]^2\right) \\
&= \left(\frac{1}{B}\sum_{b=1}^{B}\sigma_b^2\right) + \left(\frac{1}{B}\sum_{b=1}^{B}\mu_b^2\right) - \mathbb{E}[L]^2 \\
&= \left(\frac{1}{B}\sum_{b=1}^{B}\sigma_b^2 + \mu_b^2\right) - \mu^2.
\end{aligned}
$$

全分散の法則 (Law of total variance)
全分散 = グループ内分散 + グループ外分散

Hutter et al, Algorithm Runtime Prediction: Methods & Evaluation (2014) https://arxiv.org/abs/1211.0906

考えたい点① 決定森回帰の予測分散 (Random Forest型)

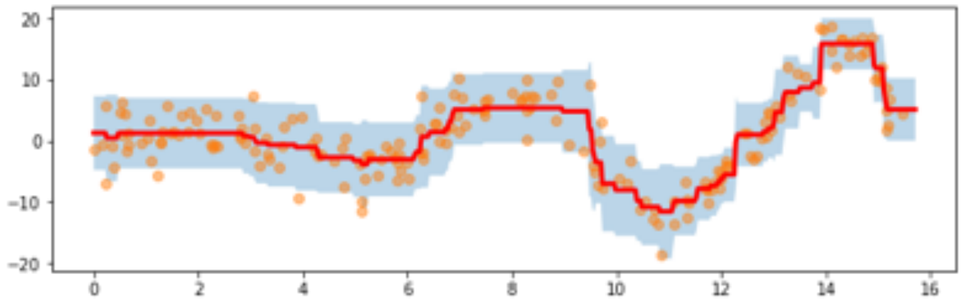
https://scikit-optimize.github.io/stable/_modules/skopt/learning/forest.html

```
def _return_std(X, trees, predictions, min_variance):
    # This derives std(y | x) as described in 4.3.2 of arXiv:1211.0906
    std = np.zeros(len(X))

    for tree in trees:
        var_tree = tree.tree_.impurity[tree.apply(X)]

        # This rounding off is done in accordance with the
        # adjustment done in section 4.3.3
        # of http://arxiv.org/pdf/1211.0906v2.pdf to account
        # for cases such as leaves with 1 sample in which there
        # is zero variance.
        var_tree[var_tree < min_variance] = min_variance
        mean_tree = tree.predict(X)
        std += var_tree + mean_tree ** 2

    std /= len(trees)
    std -= predictions ** 2.0
    std[std < 0.0] = 0.0
    std = std ** 0.5
    return std
```

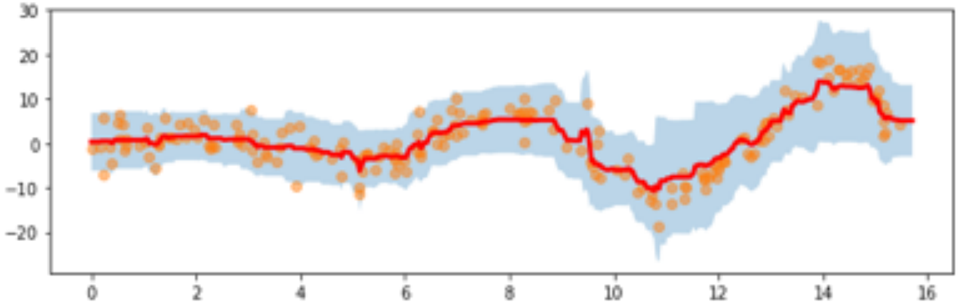
Hutter et al, Algorithm Runtime Prediction: Methods & Evaluation (2014) https://arxiv.org/abs/1211.0906

考えたい点① 決定森回帰の予測分散(信頼区間)の推定

95%信頼区間 $\bar{y}(x) \pm 1.96\,\sigma(x)$

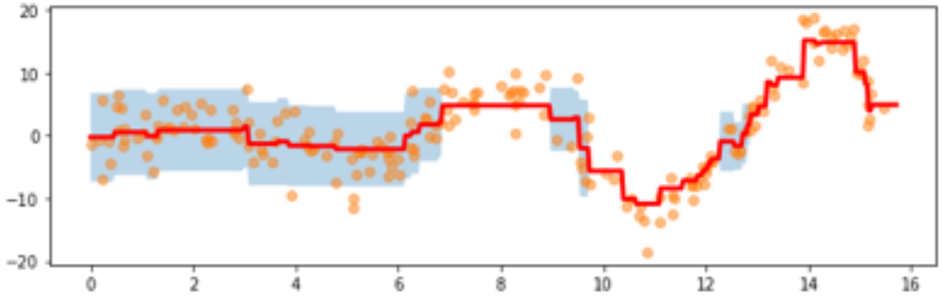


RandomForestRegressor
(n_estimators=10, max_leaf_nodes=12)



ExtraTreesRegressor
(n_estimators=10, max_leaf_nodes=32)

これと同じ方法をそのままGradient Boostingの決定木集合に適用すると…



GradientBoostingRegressor
(n_estimators=30, learning_rate=0.1)

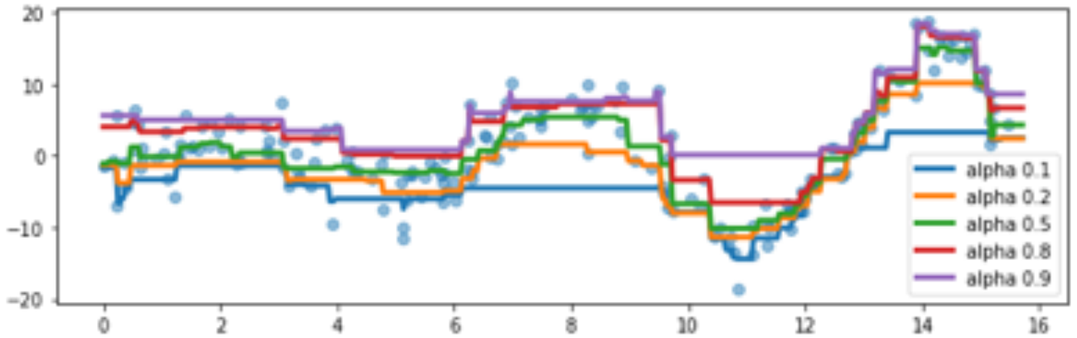
なんかヘン!?

またGradient Boostingは学習率による重み付き平均なので重みの考慮も必要!?

考えたい点① 決定森回帰の予測分散 (Gradient Boosting型)

- ✓ Gradient Boostingでは損失関数を自由に変えられる点に着目して通常は**分位点回帰(Quantile Regression)**で予測分散相当量を算出する。
- ✓ 正規分布するデータの場合、標準偏差に相当する分位点は0.16 (下側)と0.84 (上側)になるので、この分位点に対して回帰すれば標準偏差の上側と下側の値が得られる
- ✓ **分位点回帰 (Quantile Regression)**

GradientBoostingRegressor(loss='quantile', alpha=α)



データの真ん中(平均値)で
はなく特定の裾(q-分位点)
を直接狙った回帰

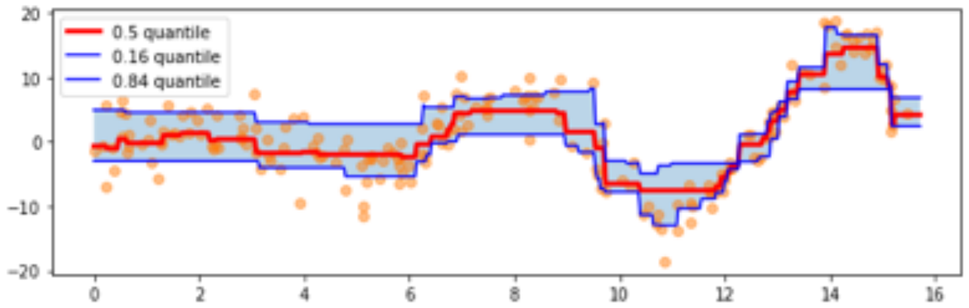
考えたい点① 決定森回帰の予測分散 (Gradient Boosting型)

✓ 損失関数を分位点ロス (Quantile loss)に変更し τ=0.16, 0.5, 0.84の分位点qを予測する回帰をそれぞれ行う。

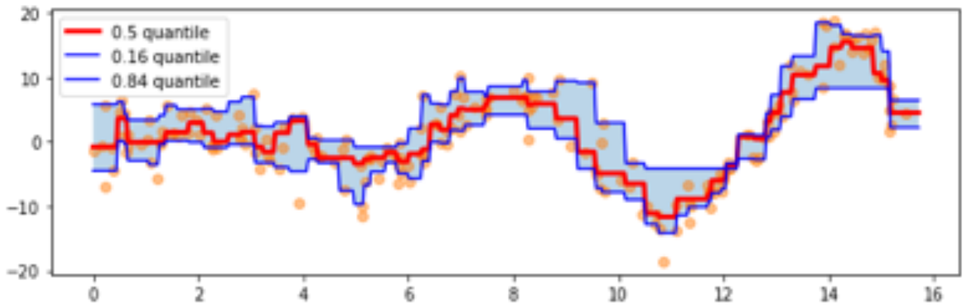
$$L_\tau(y_i, q) = \left[(\tau - 1) \sum_{y_j < q} (y_i - q) + \tau \sum_{y_j \geq q_i} (y_i - q) \right]$$

aka "Pinball loss"

✓ 0.5分位点が平均値、0.16〜0.84の間が標準偏差と近似できるので、信頼区間や期待値改善量なども算出できる

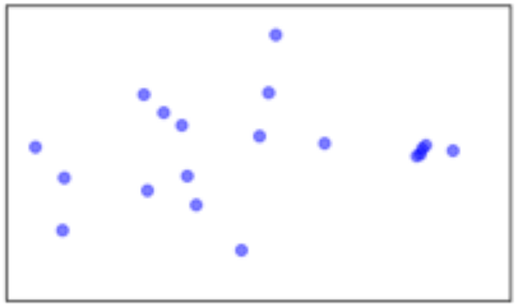


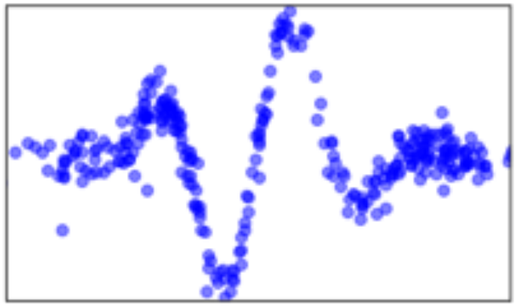
GradientBoostingRegressor(
(n_estimators=30, learning_rate=0.1)



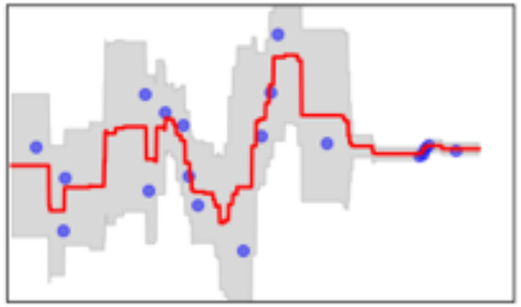
LGBMRegressor (n_estimators=30, learning_rate=0.1,
max_leaf_nodes=8, min_child_samples=5)

最初の例

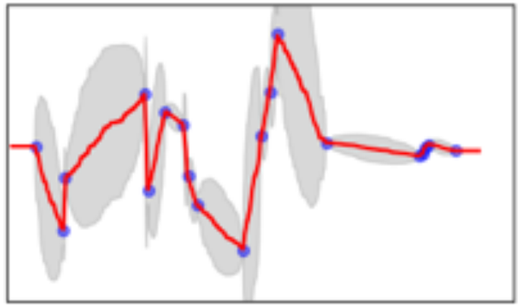




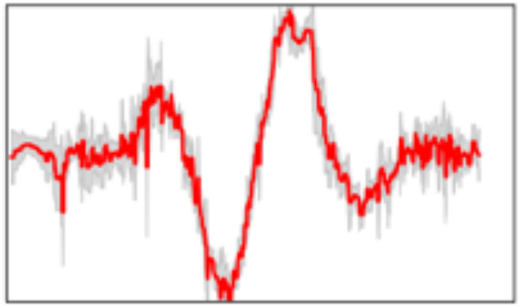
Rando Forest



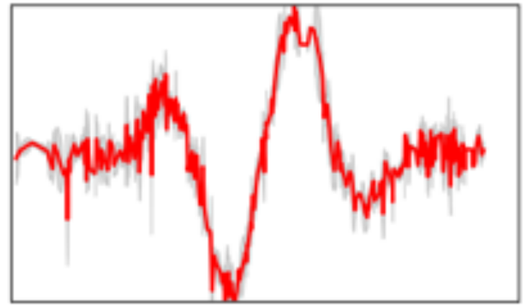
ExtraTrees (w/o bootstrap)



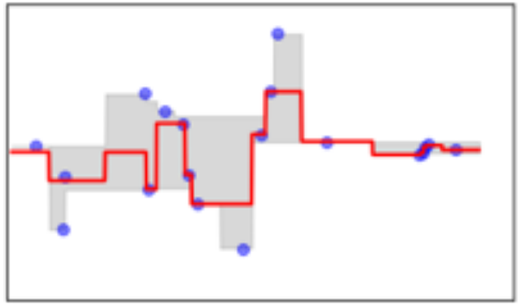
Rando Forest



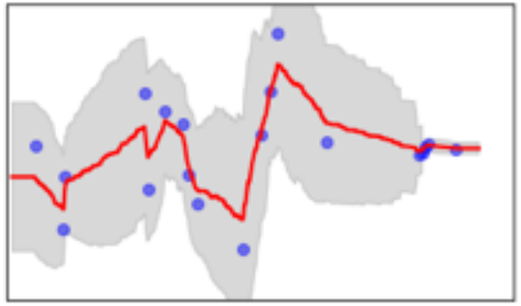
ExtraTrees (w/o bootstrap)



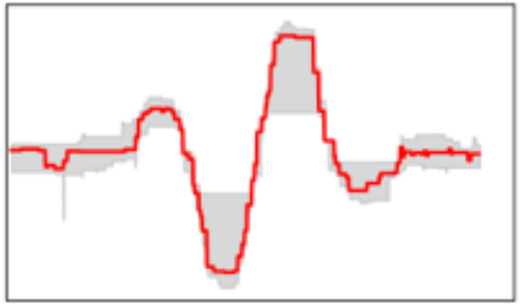
Gradient Boosted Trees



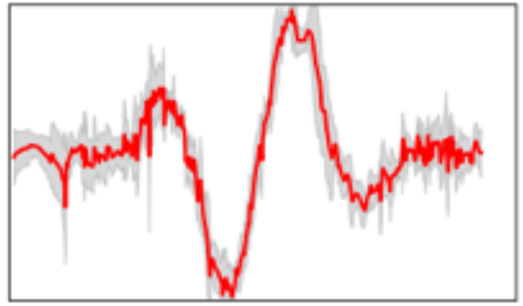
ExtraTrees (w/ bootstrap)



Gradient Boosted Trees

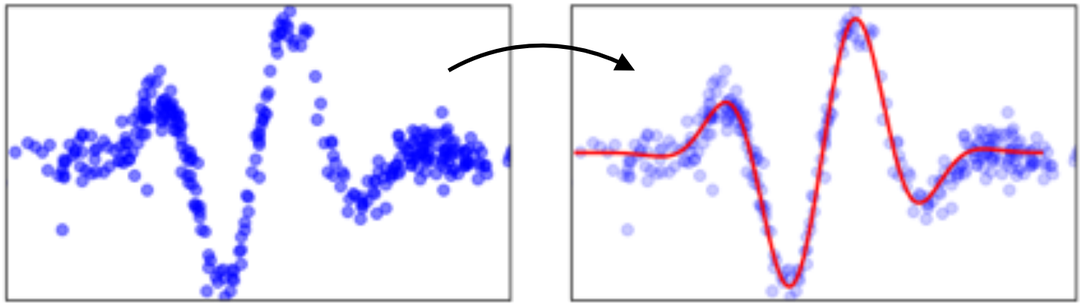


ExtraTrees (w/ bootstrap)

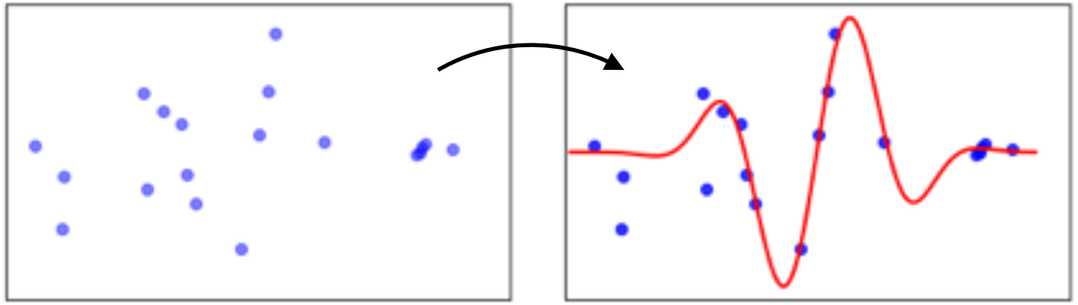


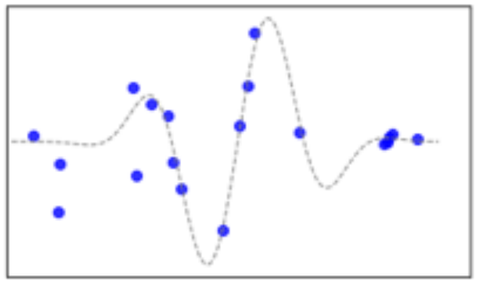
ちなみに…

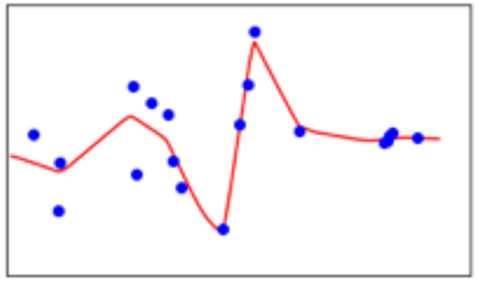
これならともかく…

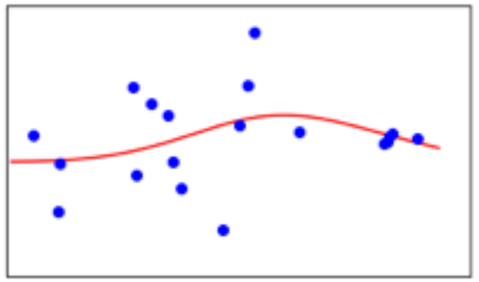


そもそも これは出来て良いのか？









これは人工データで正解があるデータなので

その意味ではこれのほうが良いとは言え

```
MLPRegressor(hidden_layer_sizes=(100,50,30,30,30),
             activation='relu',
             learning_rate_init=1e-3,
             alpha=0.0)
```

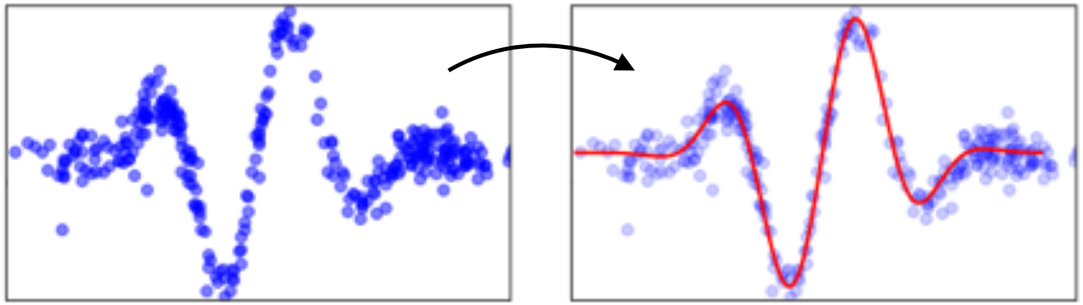
活性化関数だけ
ReLU→Tanhに変更

これでも自然な気もする…

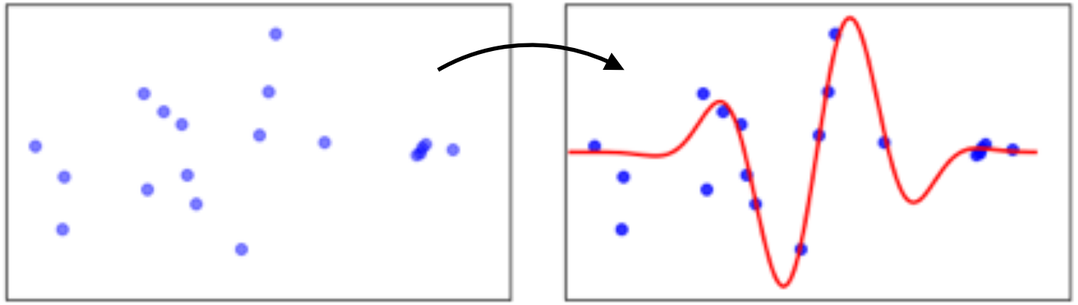
```
MLPRegressor(hidden_layer_sizes=(100,50,30,30,30),
             activation='tanh',
             learning_rate_init=1e-3,
             alpha=0.0)
```

ちなみに↓のようなそもそも論も大いにある…

これならともかく…



そもそも これは出来て良いのか？



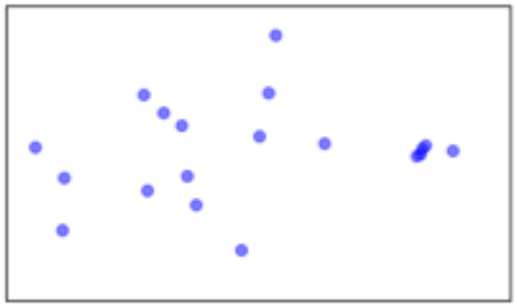
そもそも論として、統計学的にはサンプル不足であるUnderspecified な状況で何か建設的な議論は可能なのか？？

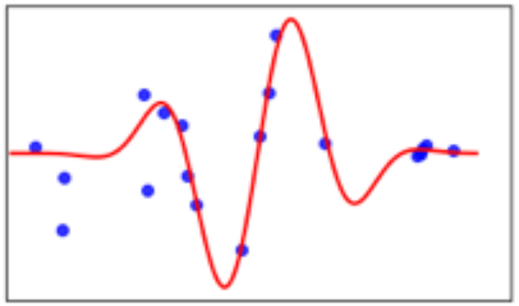
機械学習モデルのdeploy後に出会うデータ(テストデータ)は**無限**、訓練データ・検証データは**有限**、という無理設定では結局**「モデルの帰納バイアス(inductive bias)」が目前の問題にマッチするか**だけが重要？

⟶ この意味でもモデルの挙動の深い理解とそれを実用ツールと実問題に還元する実践のiterationは大事！

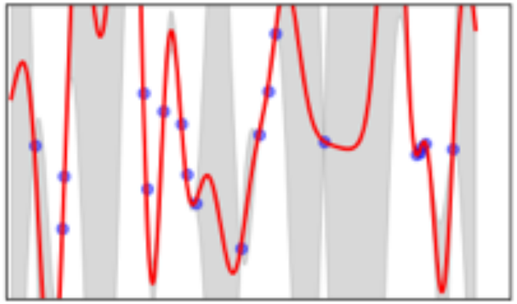
ベイズでええやん…という意見はあると思うけど…

いずれにせよ**何らかの非線形モデリングは必要**。例えば、カーネル法でそれを行うガウス過程回帰で良い感じにするのはかなり職人的なカーネルスキルが要る予感…

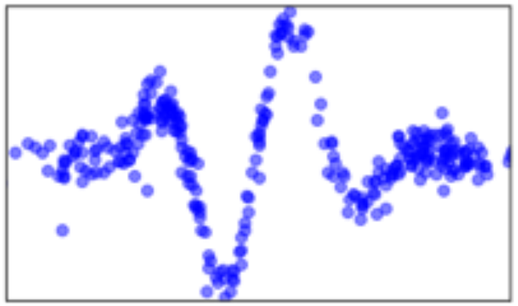


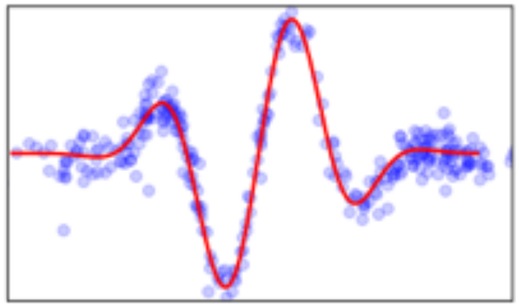


Gaussian Process

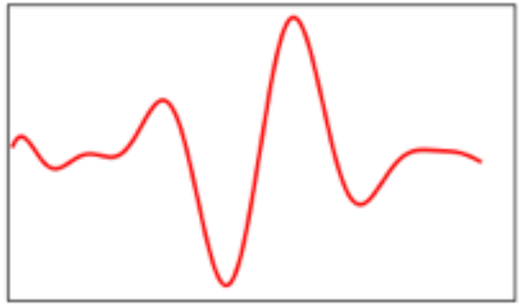


単なるRBFやMaternでは無理?





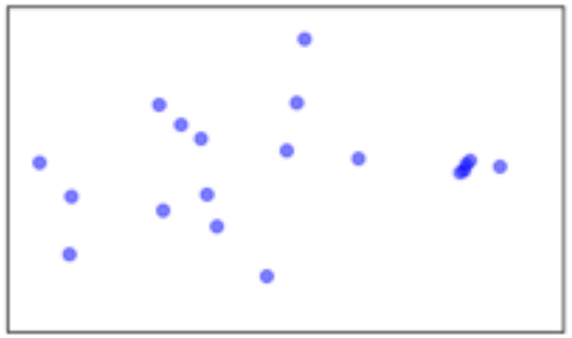
Gaussian Process



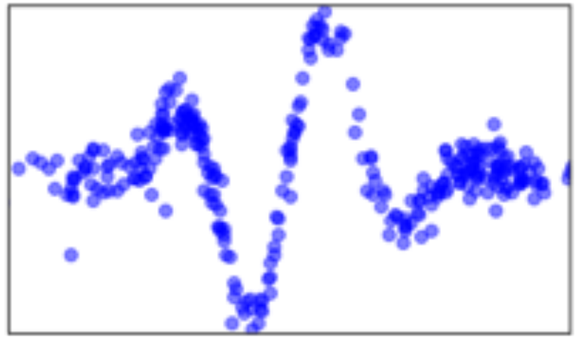
サンプルがたくさんあると予測分散**ほぼ0になりがち**

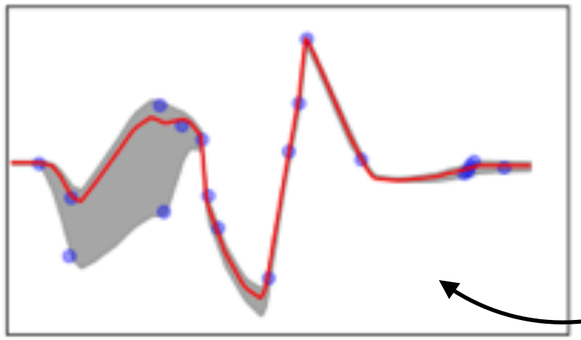
私感：もしカーネル法でもうまく出来るならこれだけ現場で決定木アンサンブルが重宝されることもないのではという気もする。現代的データをカーネル法でうまく扱うには職人技が必要？

Neural Networks (MLP) + 信頼区間推定



arch:
(Linear(1, 100), ReLU,
Dropout(p),
Linear(100, 100), ReLU,
Dropout(p),
Linear(100, 1))

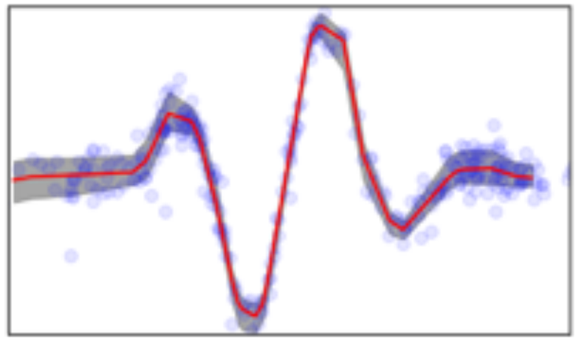




MLP
(Quantile Regression)

p=0.0

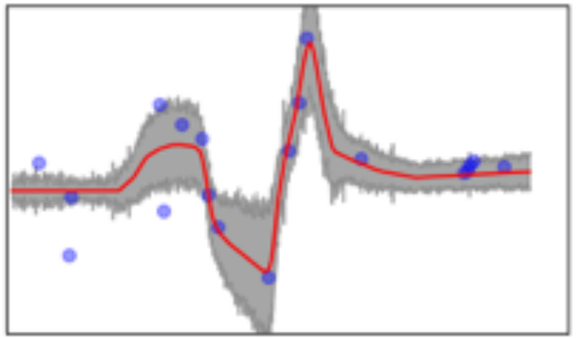
ちょっと変な感じ?



MLP
(Quantile Regression)

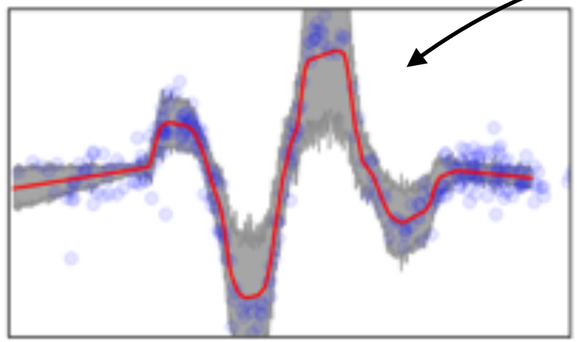
p=0.0

Dropoutのせいで
端っこの値はどう
しても分散大きめに?



MLP
(MC Dropout)

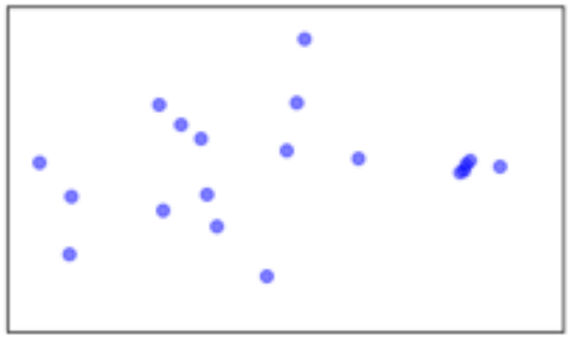
p=0.25
#sampling=30



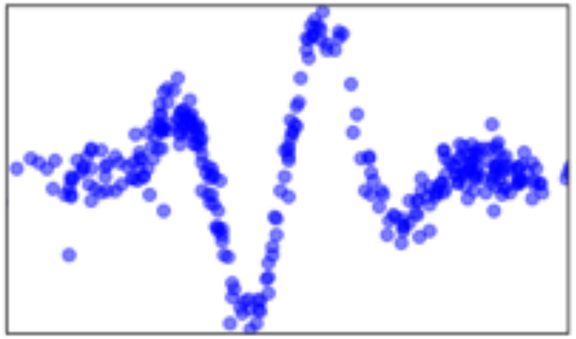
MLP
(MC Dropout)

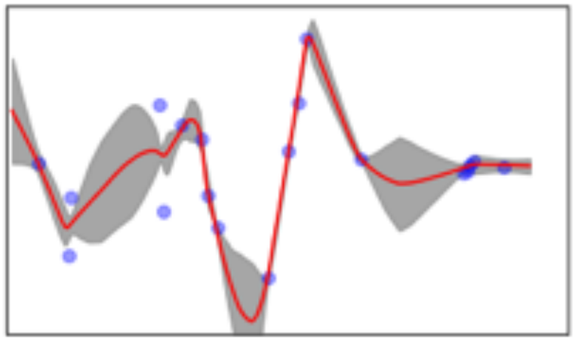
p=0.25
#sampling=30

Neural Networks (MLP) + 信頼区間推定



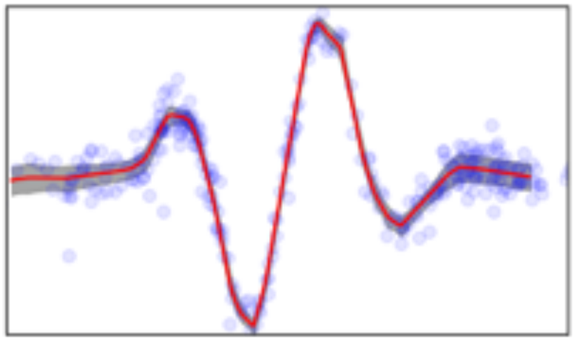
arch:
(Linear(1, 100), ReLU,
Dropout(p),
Linear(100, 100), ReLU,
Dropout(p),
Linear(100, 1))





MLP
(30 Ensemble)

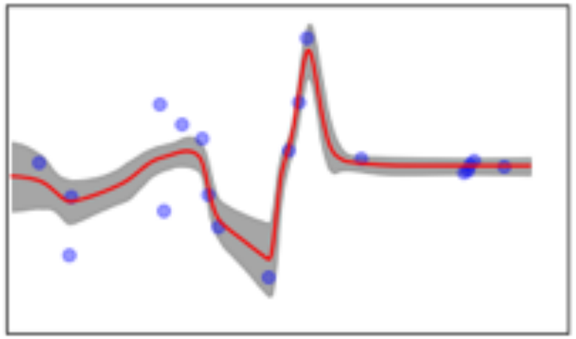
p=0.0



MLP
(30 Ensemble)

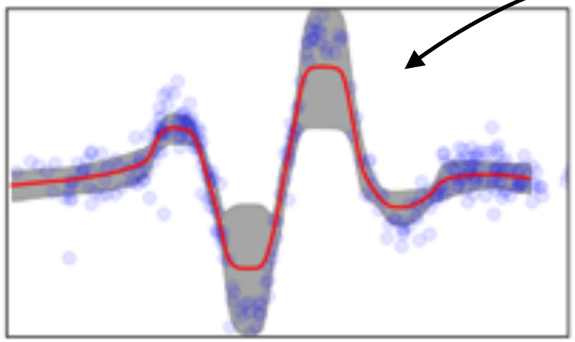
p=0.0

Dropoutのせいで
端っこの値はどう
しても分散大きめに?



MLP
(30 Ensemble)

p=0.25



MLP
(30 Ensemble)

p=0.25

MLPはちゃんと使えば異様にパワフル+奥が深い…

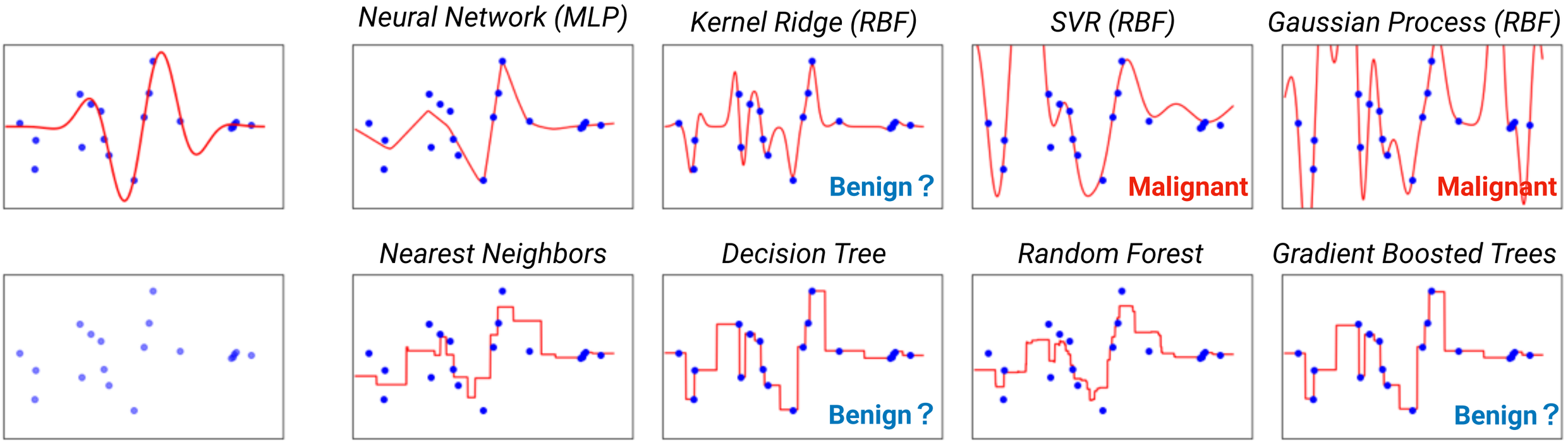
ちゃんとしたデータで学習も適切にコントロールされていればCNNもTransformerもAttentionすらも**要らない**んかも… (MLPをうまく使えばよいだけなのかも)?

- Melas-Kyriazi L.
  Do You Even Need Attention? A Stack of Feed-Forward Layers Does Surprisingly Well on ImageNet. 2021. http://arxiv.org/abs/2105.02723
- Tolstikhin IO, Houlsby N, Kolesnikov A, Beyer L, Zhai X, Unterthiner T, et al.
  MLP-Mixer: An all-MLP Architecture for Vision. *NeurIPS* 2021.
- Kadra A, Lindauer M, Hutter F, Grabocka J.
  Well-Tuned Simple Nets Excel on Tabular Datasets. *NeurIPS* 2021.
- Liu H, Dai Z, So D, Le Q.
  Pay Attention to MLPs. *NeurIPS* 2021.

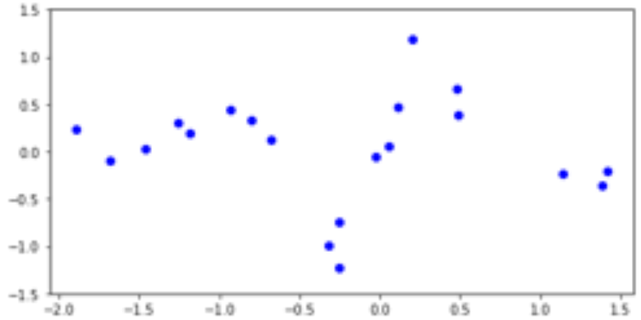
考えたい点②

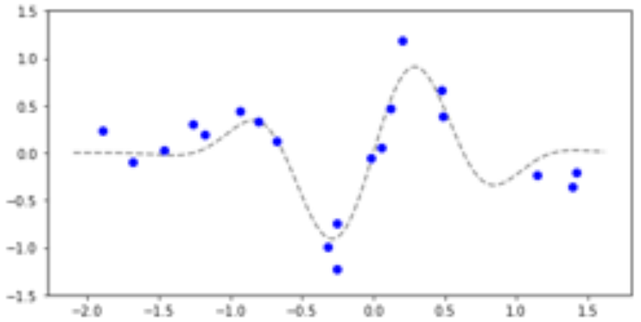
「決定木は**Overfitting**しやすい」はマズい特性か？

→ アンサンブルを取る主動機になっている。ただし、この「Overfitting」は必ずしも有害とは言えない (Benign overfitting)。**ノイズあり事例で訓練誤差0**でもそれなりに役に立つ！

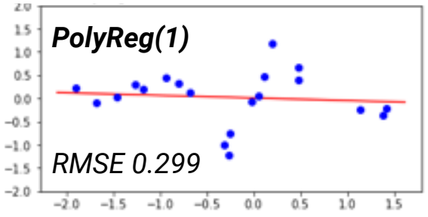


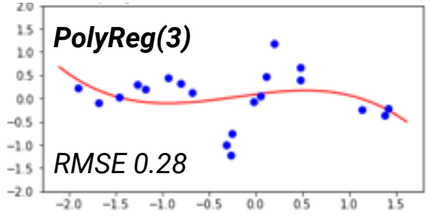
Benign Overfitting：ノイズありデータで訓練誤差0でも無害

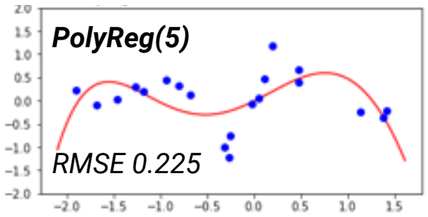


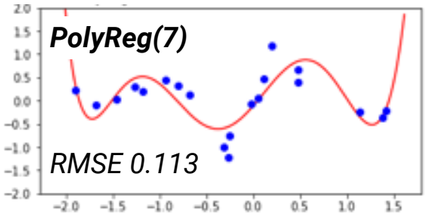


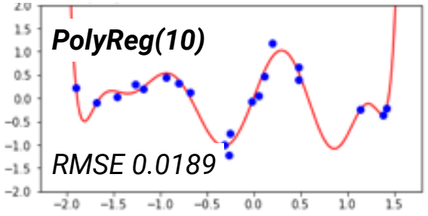
Problematic overfitting by polynomial regression of order k

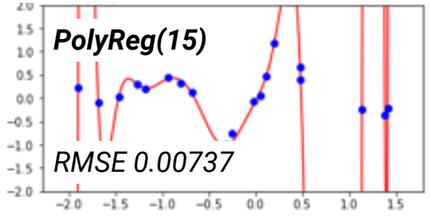


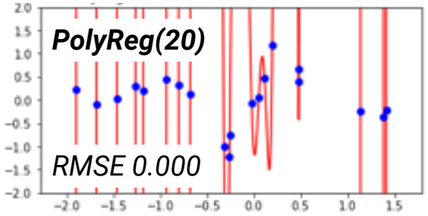


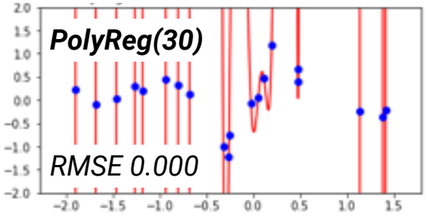




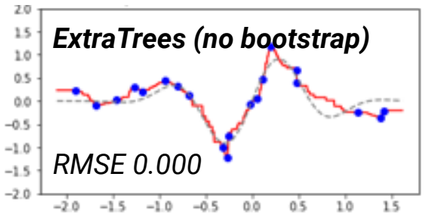


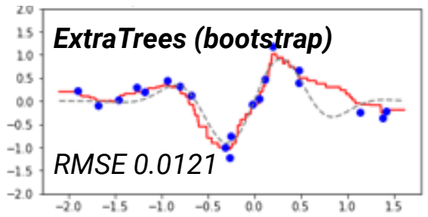


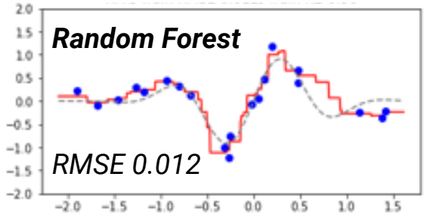


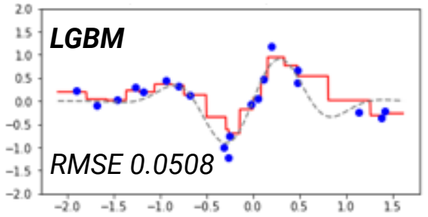


clearly overfitted but harmless (still informative)

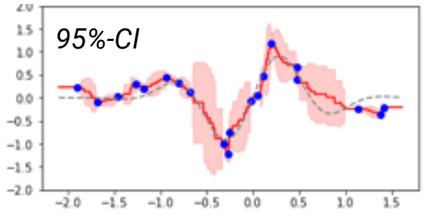


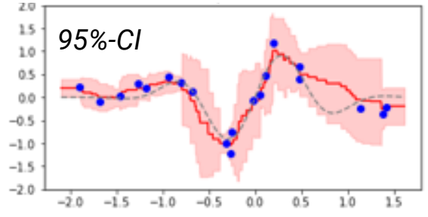


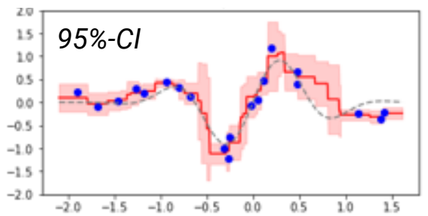


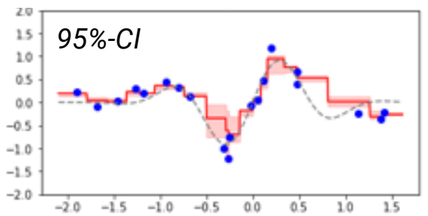


also we can assess
the uncertainty



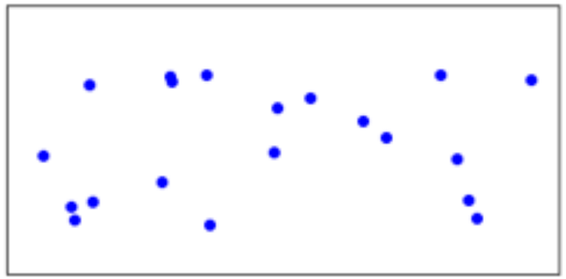




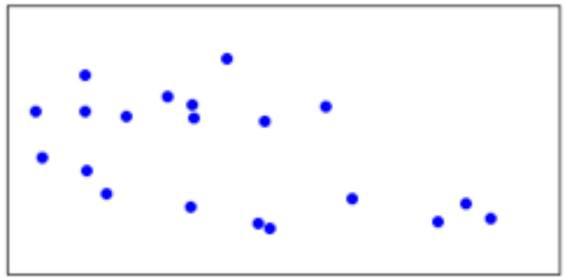


ExtraTreesやGradient Boosted Treesは訓練誤差0は余裕

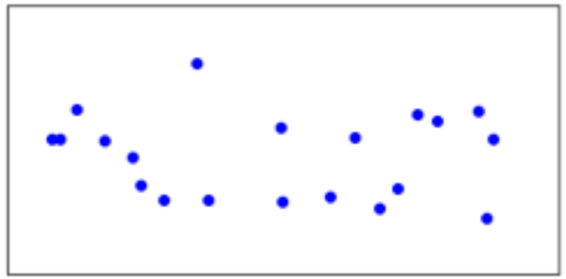
全くランダムなラベルでも訓練誤差0を達成できる



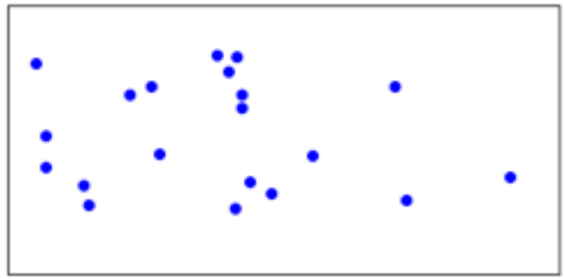
ExtraTrees (no bootstrap)



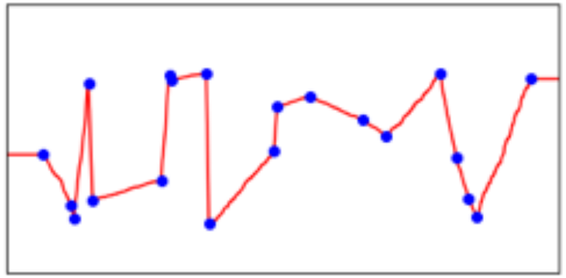
ExtraTrees (no bootstrap)



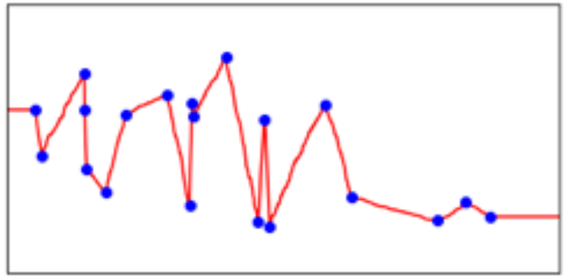
ExtraTrees (no bootstrap)



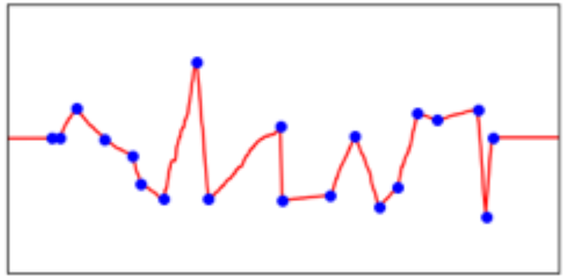
ExtraTrees (no bootstrap)



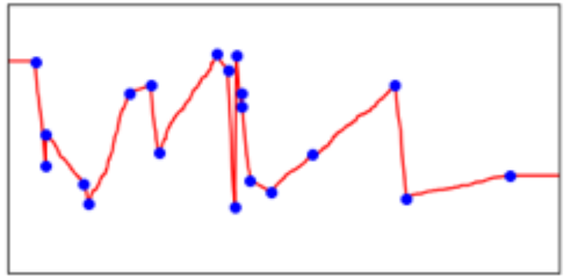
Gradient Boosted Trees



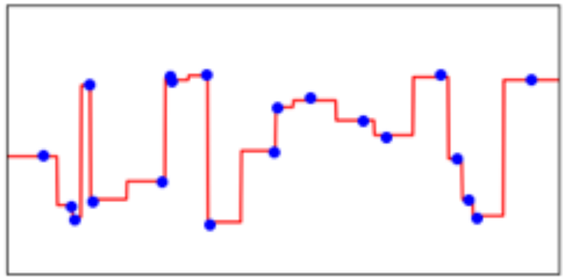
Gradient Boosted Trees

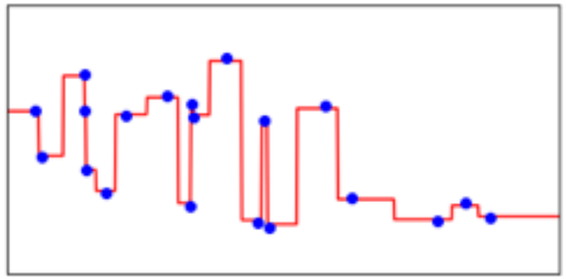


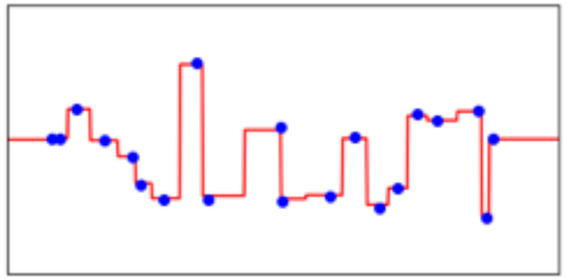
Gradient Boosted Trees

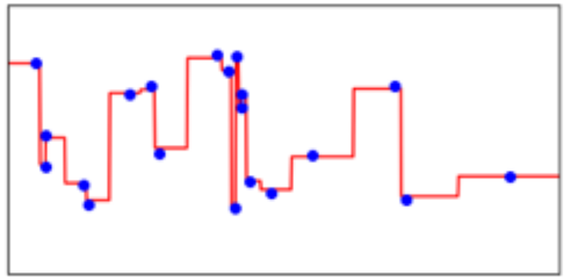


Gradient Boosted Trees



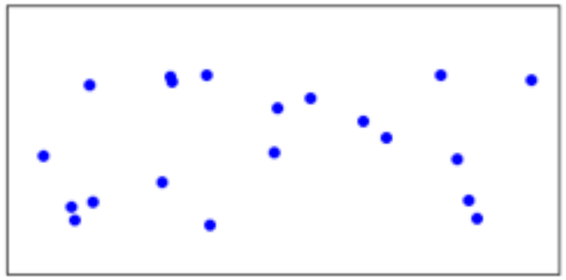




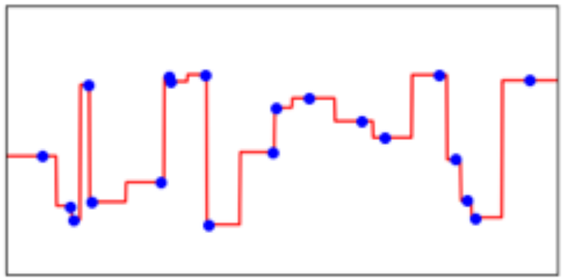


ちなみに最近隣法や決定木も持つ性質でわりと当たり前

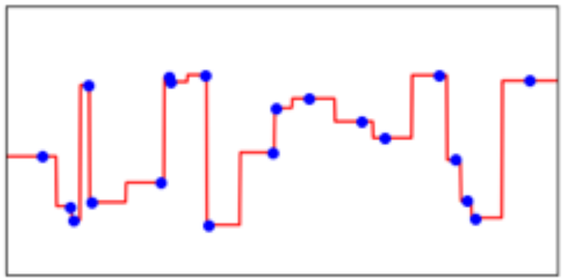
全くランダムなラベルでも訓練誤差0を達成できる (この設定ではGBDT, 1-NN, DTはほぼ同じ)



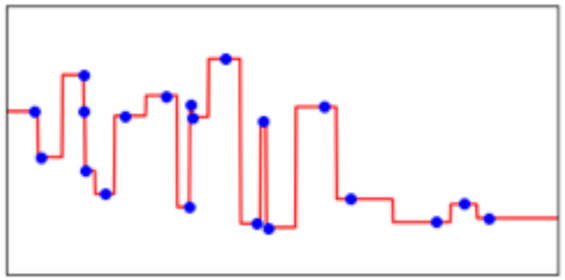
Nearest Neighbor (k=1)



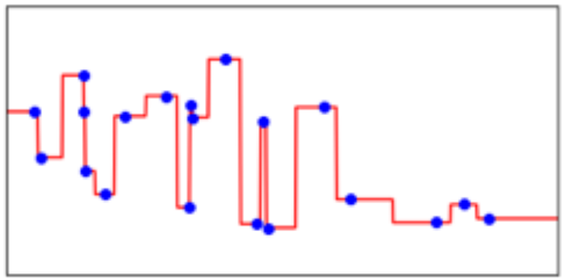
Decision Tree

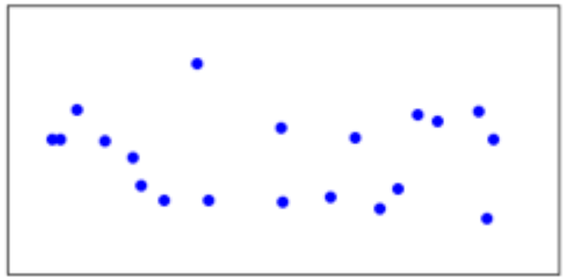


Nearest Neighbor (k=1)

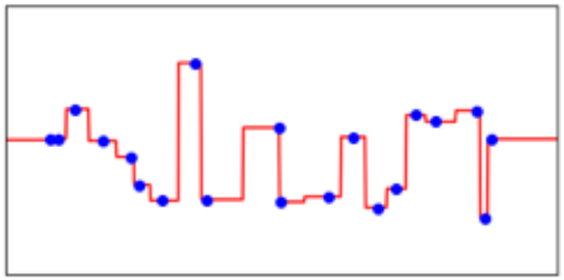


Decision Tree

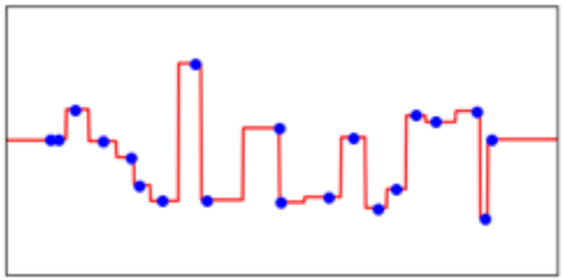


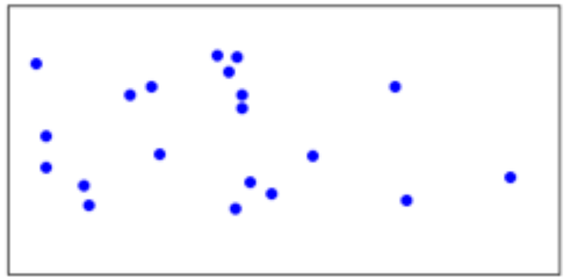


Nearest Neighbor (k=1)

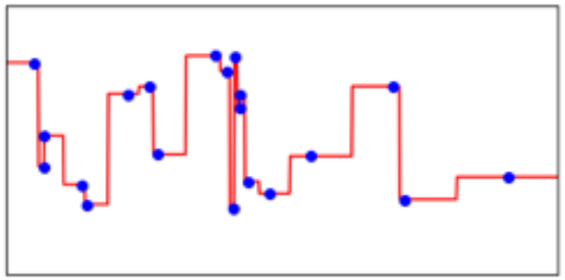


Decision Tree

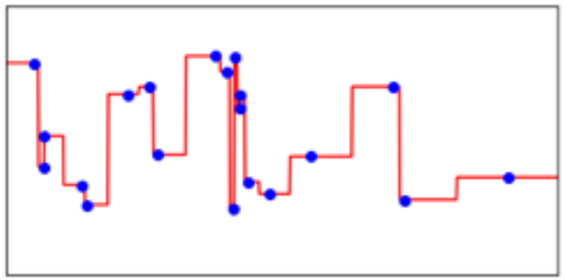




Nearest Neighbor (k=1)

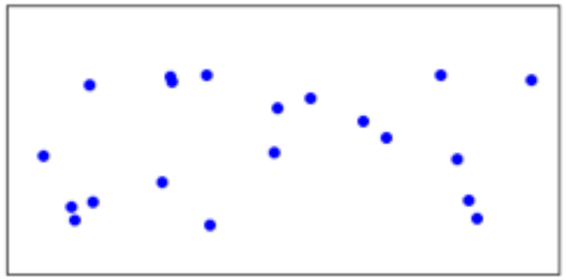


Decision Tree



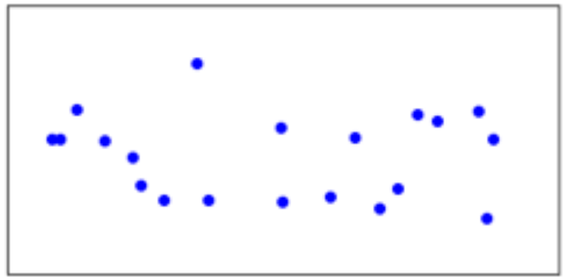
Random Forestや3-最近隣法はこの性質を持たない

Bootstrapを伴うRandom ForestやExtraTrees、k-NN (k>1)などはoverfitできない

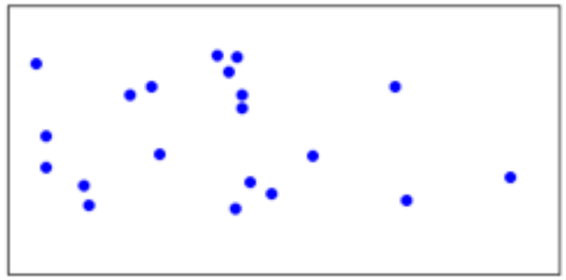


Random Forest

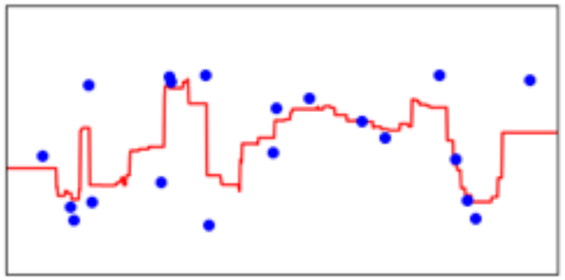
Random Forest



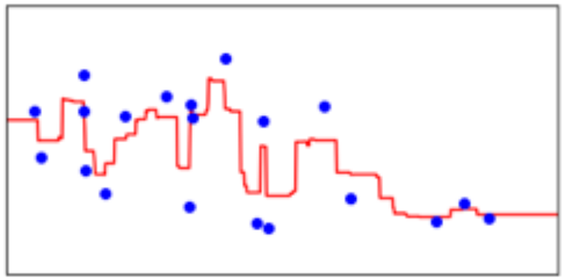
Random Forest



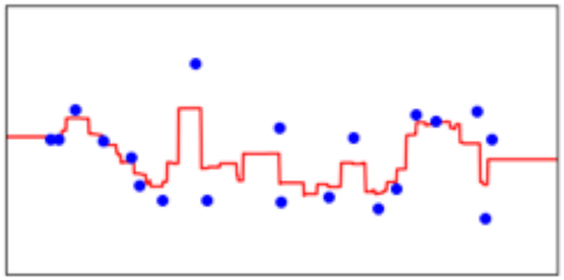
Random Forest



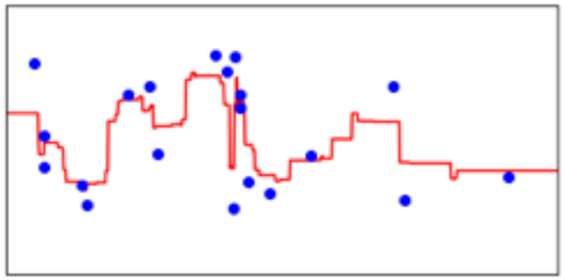
Nearest Neighbor (k=3)



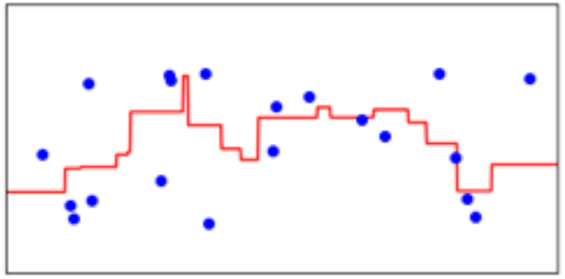
Nearest Neighbor (k=3)

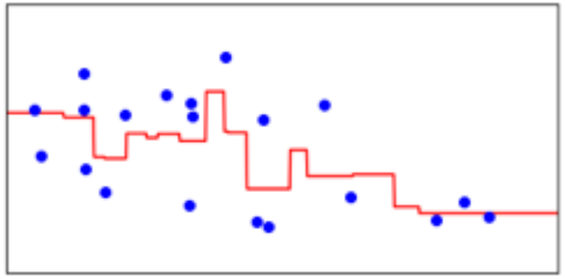


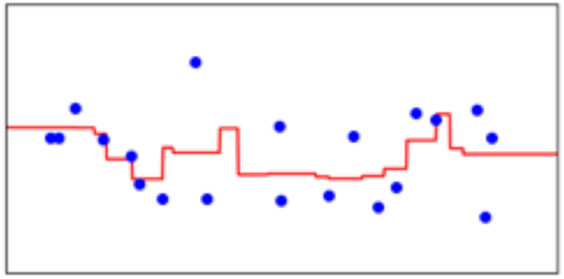
Nearest Neighbor (k=3)

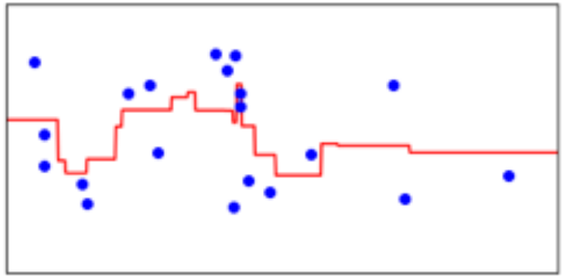


Nearest Neighbor (k=3)









深層学習モデルがこの性質を持つことは注目されている

"our experiments establish that state-of-the-art convolutional networks for image classification trained with stochastic gradient methods easily fit a random labeling of the training data. This phenomenon is qualitatively unaffected by explicit regularization and occurs even if we replace the true images by completely unstructured random noise. "

Zhang C, Bengio S, Hardt M, Recht B, Vinyals O.
Understanding Deep Learning (Still) Requires Rethinking Generalization. *Commun ACM*. 2021;64: 107–115.

1. *Zero training error on random labels:* Zero empirical risk can also be achieved for random labels using the same architecture and training scheme with only slightly increased training time: This suggests that the considered hypothesis set of NNs $\mathcal{F}$ can fit arbitrary binary labels, which would imply that $\mathrm{VCdim}(\mathcal{F}) \approx m$ or $\mathfrak{R}_m(\mathcal{F}) \approx 1$ rendering our uniform generalization bounds in Theorem 1.19 and in (1.12) vacuous.

4. *Interpolation of noisy training data:* One still observes low test error when training up to approximately zero empirical risk using a regression (or surrogate) loss on noisy training data. This is particularly interesting, as the noise is captured by the model but seems not to hurt generalization performance.

Berner J, Grohs P, Kutyniok G, Petersen P.
The Modern Mathematics of Deep Learning. *arXiv [cs.LG]*. 2021. http://arxiv.org/abs/2105.04026

Benign Overfitting

経験的事実として、実際の深層学習の現場では**教師ラベルにノイズがあろうがなかろうが**、
テスト誤差が小さいモデルは「訓練誤差も小さい(ほぼゼロ誤差！)」な場合がとても多い。

2.4 Limits of classical theory and double descent

There is ample evidence that classical tools from statistical learning theory alone, such as Rademacher averages, uniform convergence, or algorithmic stability may be unable to explain the full generalization capabilities of NNs [ZBH+17, NK19]. It is especially hard to reconcile the classical bias-variance trade-off with the observation of good generalization performance when achieving zero empirical risk on noisy data using a regression loss. On top of that, this behavior of overparametrized models in the interpolation regime turns out not to be unique to NNs. Empirically, one observes for various methods (decision trees, random features, linear models) that the test error decreases even below the sweet-spot in the u-shaped bias-variance curve when further increasing the number of parameters [BHMM19, GJS+20, NKB+20]. This is often referred to as the *double descent curve* or *benign overfitting*, see Figure 2.1. For special cases, e.g., linear regression or random feature regression, such behavior can even be proven, see [HMRT19, MM19, BLLT20, BHX20, MVSS20].

Berner J, Grohs P, Kutyniok G, Petersen P.
The Modern Mathematics of Deep Learning. arXiv [cs.LG]. 2021. http://arxiv.org/abs/2105.04026

or Double Descent?

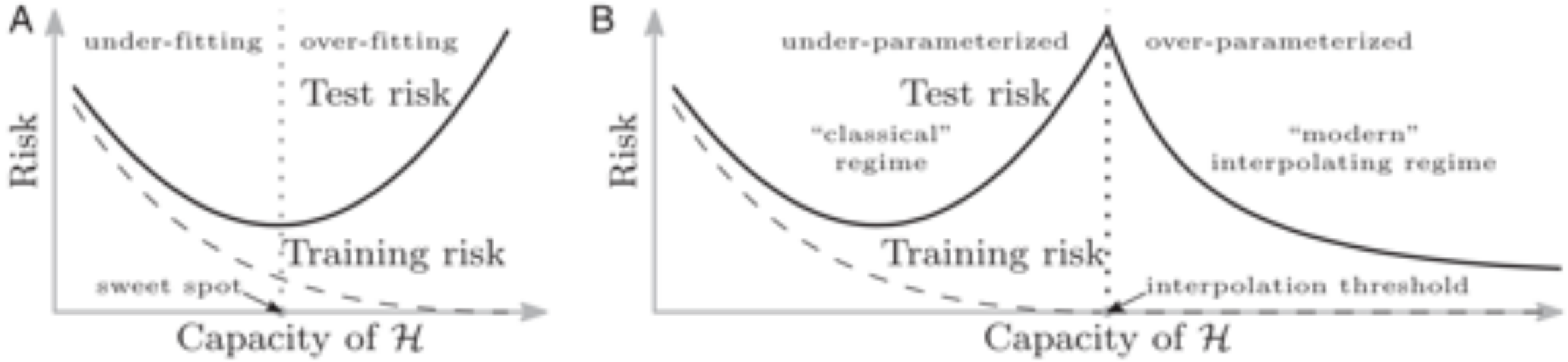
PNAS

Reconciling modern machine-learning practice and the classical bias–variance trade-off

Mikhail Belkin[a,b,1], Daniel Hsu[c], Siyuan Ma[a], and Soumik Mandal[a]

[a]Department of Computer Science and Engineering, The Ohio State University, Columbus, OH 43210; [b]Department of Statistics, The Ohio State University, Columbus, OH 43210; and [c]Computer Science Department and Data Science Institute, Columbia University, New York, NY 10027

Edited by Peter J. Bickel, University of California, Berkeley, CA, and approved July 2, 2019 (received for review February 21, 2019)



Interpolation, Perfect Fitting, and Double Descent

PNAS (2020)

Benign overfitting in linear regression

Peter L. Bartlett[a,b,1], Philip M. Long[c], Gábor Lugosi[d,e,f], and Alexander Tsigler[a]

[a]Department of Statistics, University of California, Berkeley, CA 94720-3860; [b]Computer Science Division, University of California, Berkeley, [c]Google Brain, Mountain View, CA 94043; [d]Economics and Business, Pompeu Fabra University, 08005 Barcelona, Spain; [e]Institució Catalana Estudis Avançats, Passeig, Lluis Companys 23, 08010 Barcelona, Spain; and [f]Barcelona Graduate School of Economics, 08005 Barcelona, Spai

Edited by Richard Baraniuk, Rice University, Houston, TX, and accepted by Editorial Board Member David L. Donoho March 4, 2020 (receive June 2, 2019)

The phenomenon of benign overfitting is one of the key mysteries uncovered by deep learning methodology: deep neural networks seem to predict well, even with a perfect fit to noisy training data. Motivated by this phenomenon, we consider when a perfect fit to training data in linear regression is compatible with accurate prediction. We give a characterization of linear

enough that a perfect fit is guaranteed. We consid
infinite-dimensional space (a separable Hilbert sp
results apply to a finite-dimensional subspace as a
There is an ideal value of the parameters, θ^*, cor
the linear prediction rule that minimizes the expec
loss. We ask when it is possible to fit the data ex

Ann. Statist. (2020)

JUST INTERPOLATE: KERNEL "RIDGELESS" REGRESSION CAN GENERALIZE

BY TENGYUAN LIANG[1] AND ALEXANDER RAKHLIN[2]

[1]*Econometrics and Statistics, Booth School of Business, University of Chicago, tengyuan.liang@chicagobooth.edu*
[2]*Center for Statistics & IDSS, Massachusetts Institute of Technology, rakhlin@mit.edu*

In the absence of explicit regularization, Kernel "Ridgeless" Regression with nonlinear kernels has the potential to fit the training data perfectly. It has been observed empirically, however, that such interpolated solutions can still

NeurIPS (2018)

Overfitting or perfect fitting? Risk bounds for classification and regression rules that interpolate

Mikhail Belkin
The Ohio State University

Daniel Hsu
Columbia University

Partha P. Mitra
Cold Spring Harbor Laboratory

Abstract

arXiv (2019)

Surprises in High-Dimensional Ridgeless Least Squares Interpolation

Trevor Hastie Andrea Montanari* Saharon Rosset Ryan J. Tibshirani*

Abstract

Interpolators—estimators that achieve zero training error—have attracted growing attention in machine learning, mainly because state-of-the art neural networks appear to be models of this type. In this paper, we study minimum ℓ_2 norm ("ridgeless") interpolation in high-dimensional least squares regression. We consider two different models for the feature distribution: a linear model, where the feature vectors $x_i \in \mathbb{R}^p$ are obtained by applying a linear transform

Benign Overfitting (私感)

参考) 今泉允聡, 深層学習の原理解析:汎化誤差の側面から. 日本統計学会誌, 50(2):257-283, 2021.

Benign Overfitting (私感)

- ニューラルネットや決定木では**「低次元でも」ノイズありデータで訓練誤差ほぼ0の場合が最もテスト誤差も少ない実例に結構出会う**ので、同質で扱って良いのかは大いに疑問。

参考) 今泉允聡, 深層学習の原理解析:汎化誤差の側面から. 日本統計学会誌, 50(2):257-283, 2021.

Benign Overfitting (私感)

- ニューラルネットや決定木では**「低次元でも」ノイズありデータで訓練誤差ほぼ0の場合が最もテスト誤差も少ない実例に結構出会う**ので、同質で扱って良いのかは大いに疑問。
- **というか「Benign Overfitting」=「Double descent」なのかはかなり疑問。**そもそもDouble descentでは一回テスト誤差が悪くなるバンプがあることになるが、Adaboostの初期議論を筆頭に訓練誤差0の後も**(悪くなる揺り戻しは特に無しに)**テスト誤差が減りつづける事例は色々報告もある。

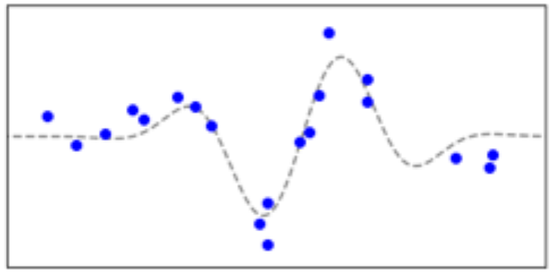
参考) 今泉允聡, 深層学習の原理解析:汎化誤差の側面から. 日本統計学会誌, 50(2):257-283, 2021.

Benign Overfitting (私感)

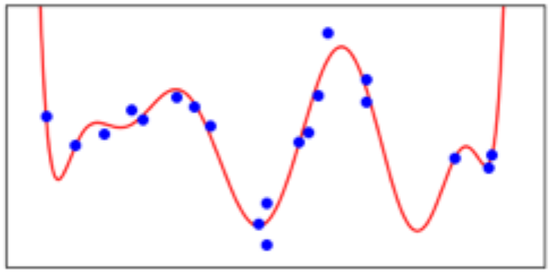
- ニューラルネットや決定木では**「低次元でも」ノイズありデータで訓練誤差ほぼ0の場合が最もテスト誤差も少ない実例に結構出会う**ので、同質で扱って良いのかは大いに疑問。
- **というか「Benign Overfitting」=「Double descent」なのかはかなり疑問。**そもそもDouble descentでは一回テスト誤差が悪くなるバンプがあることになるが、Adaboostの初期議論を筆頭に訓練誤差0の後も**(悪くなる揺り戻しは特に無しに)**テスト誤差が減りつづける事例は色々報告もある。
- Linear Regression('Ridgeless' Least squares)でも起こるのはサンプル数より大きい高次元では**全点を通る関数が(たくさん)引けるから**。解を一意にするのに、Bartlett+(2020)でもHastie+(2019)でも**2-norm最小解(MP一般逆解)**を仮定しているがoverparametrize(高次元射影)して一般逆で線形回帰すればOKとは思えない (c.f. Extreme Learning Machine(ELM), Reservoir Computing)。むしろ**「全点通る関数が引けちゃう」性質のほうが本質的？**

参考) 今泉允聡, 深層学習の原理解析:汎化誤差の側面から. 日本統計学会誌, 50(2):257-283, 2021.

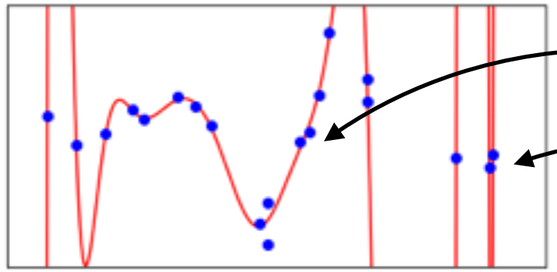
有害なoverfitting再考



PolyReg(10)
Train RMSE 0.0189



PolyReg(15)
Train RMSE 0.00737

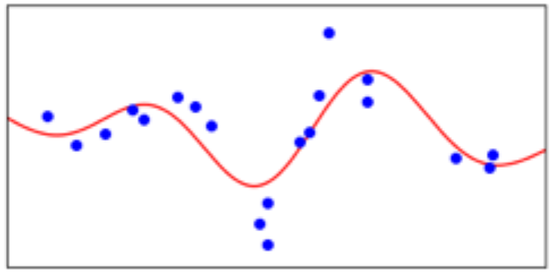


ここに合わせようとすると

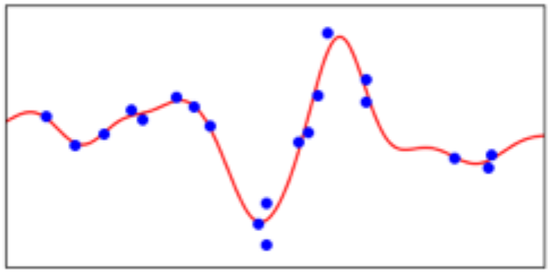
トレードオフでこのへんに
ひどい影響がでてしまう

フィッティングが「local」じゃない

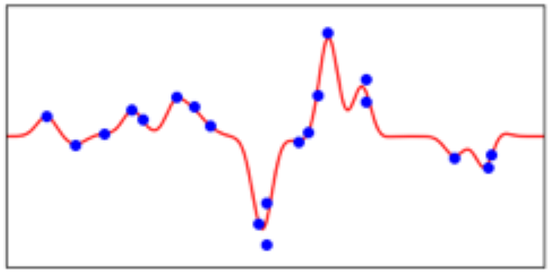
KernelRidge RBF(γ=1)
Train RMSE 0.103



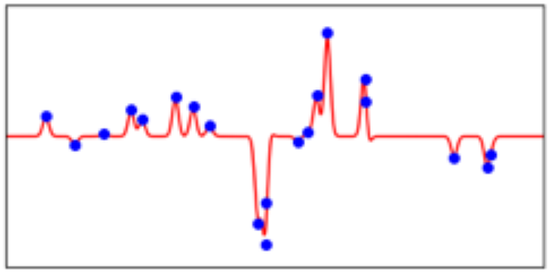
KernelRidge RBF(γ=10)
Train RMSE 0.0139



KernelRidge RBF(γ=100)
Train RMSE 0.00726

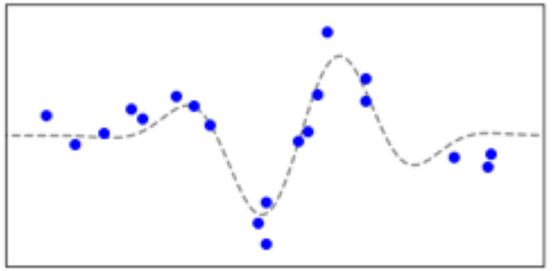


KernelRidge RBF(γ=1000)
Train RMSE 0.00726



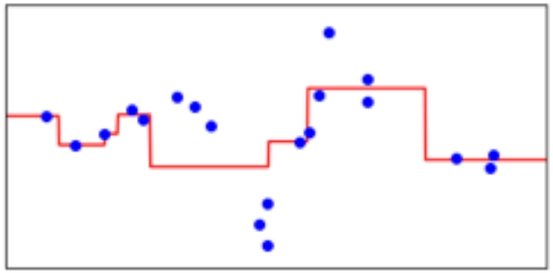
フィッティングは「local」だけど各点の近傍はすべて同じスケールで扱われてしまう

決定木のfitting (葉数固定でもoverfitできる)

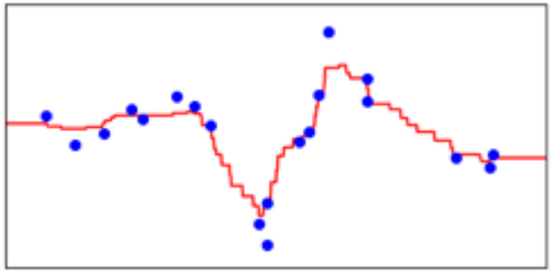


Note: GBDT w/ ExtraTreesはsklearnにGBを
1行だけ書き換えれば簡単に試せる

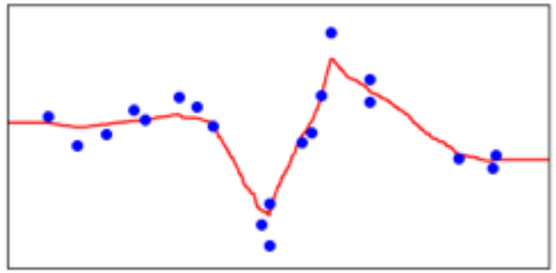
*ExtraTrees (**#trees=1**, #leaves=8, bootstrap=off), Train RMSE **0.0189***



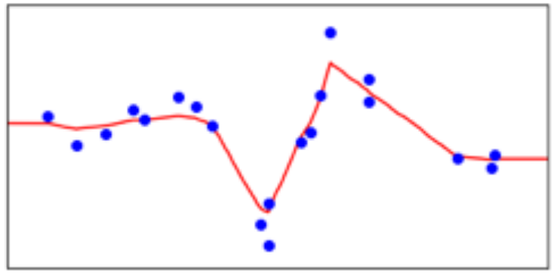
*ExtraTrees (**#trees=10**, #leaves=8, bootstrap=off), Train RMSE **0.0274***



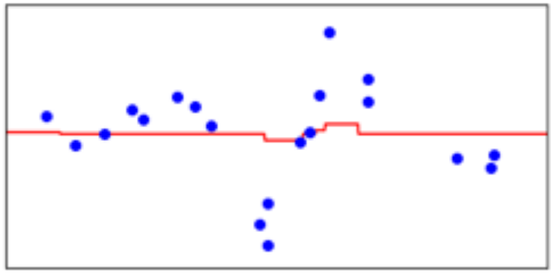
*ExtraTrees (**#trees=10^2**, #leaves=8, bootstrap=off), Train RMSE **0.0279***



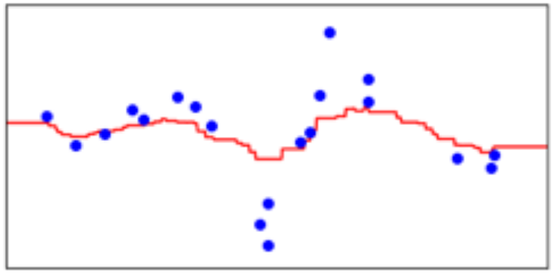
*ExtraTrees (**#trees=10^3**, #leaves=8, bootstrap=off), Train RMSE **0.0243***



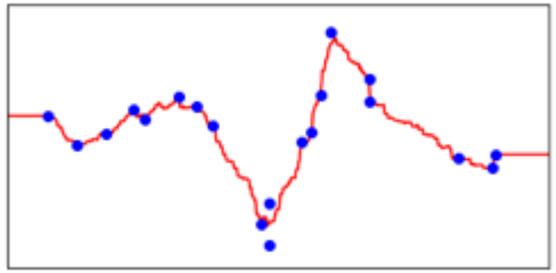
*GBDT w/ **1** ExtraTree (#leaves=8) Train RMSE **0.29***



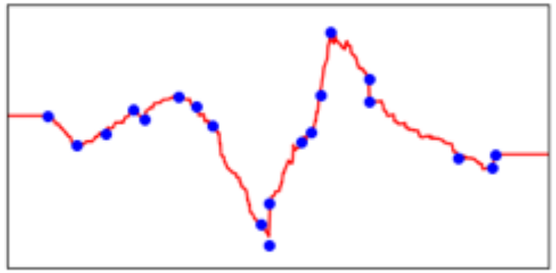
*GBDT w/ **10** ExtraTree (#leaves=8) Train RMSE **0.17***



*GBDT w/ **10^2** ExtraTree (#leaves=8) Train RMSE **0.00597***

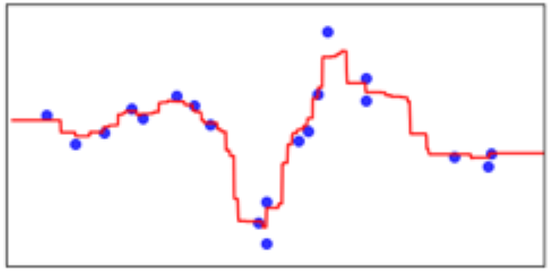


*GBDT w/ **10^3** ExtraTree (#leaves=8) Train RMSE **0.000997***

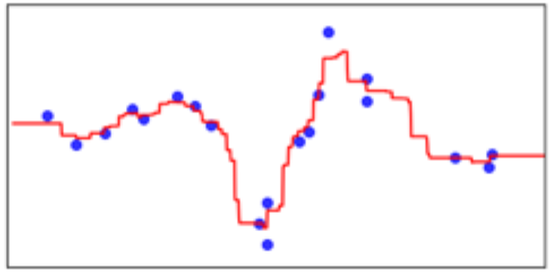


決定木のfitting (RF vs ET, RFF)

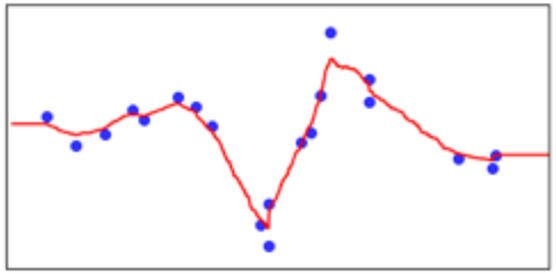
Random Forest *(**#trees=100**)*
*Train RMSE **0.0109***



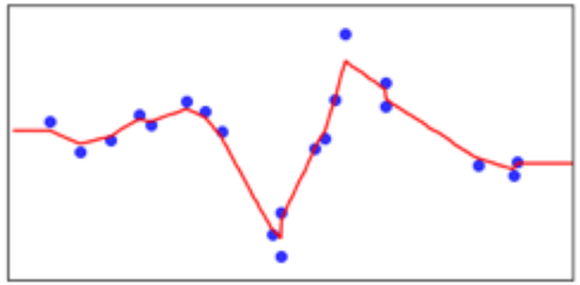
Random Forest *(**#trees=1000**)*
*Train RMSE **0.0274***



ExtraTrees *(**#trees=100**, bootstrap)*
*Train RMSE **0.0109***

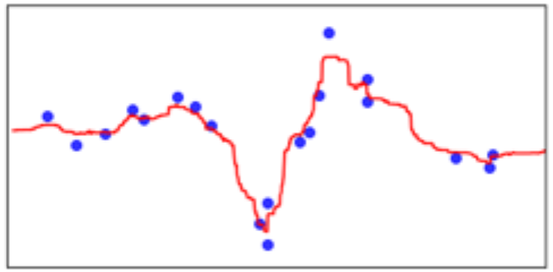


ExtraTrees *(**#trees=1000**, bootstrap)*
*Train RMSE **0.0243***

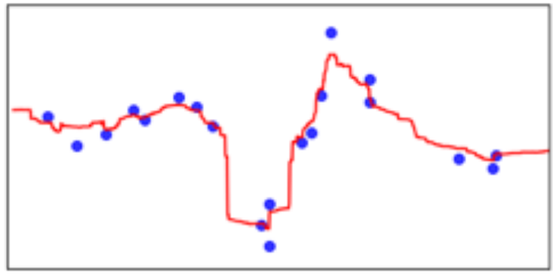


With Random Fourier Features (100-dim Random Kitchen Sinks)

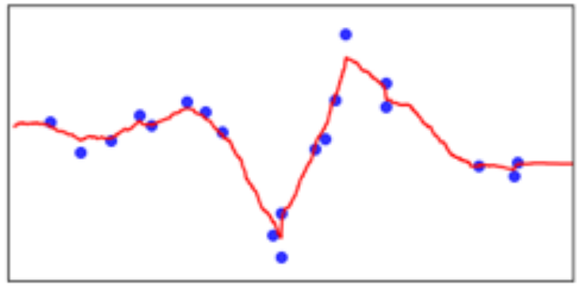
Random Forest *(**#trees=100**)*
*Train RMSE **0.0112***



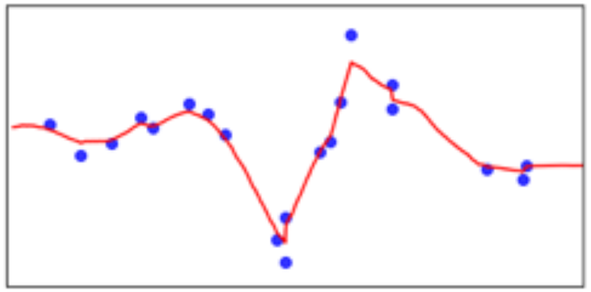
Random Forest *(**#trees=100**)*
*Train RMSE **0.0114***



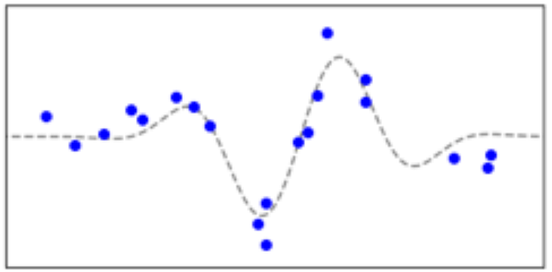
ExtraTrees *(**#trees=100**, bootstrap)*
*Train RMSE **0.0099***

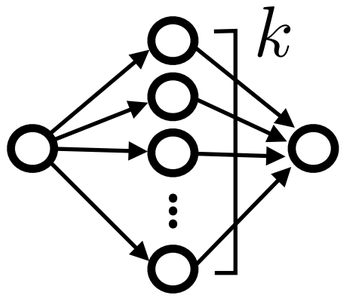


ExtraTrees *(**#trees=1000**, bootstrap)*
*Train RMSE **0.00977***



1-k-1 ReLU MLPのfitting



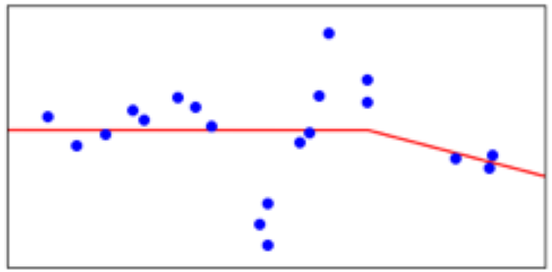


arch:
(Linear(1, k),
ReLU,
Linear(k, 1))

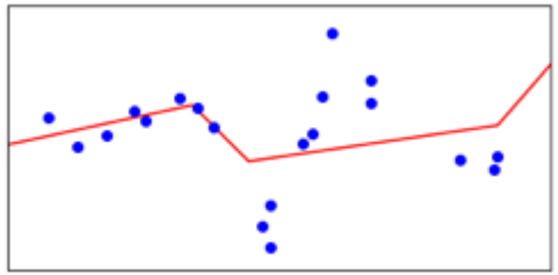
Optimized With L-BFGS

※初期値に依存するばらつきは結構ある

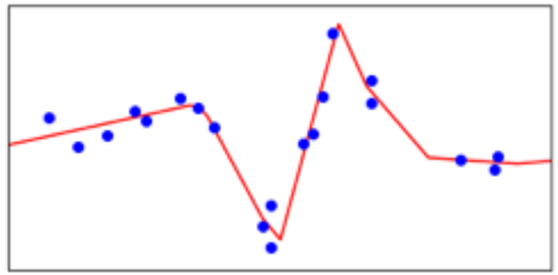
*1-**1**-1 ReLU MLP*
*Train RMSE **0.287***



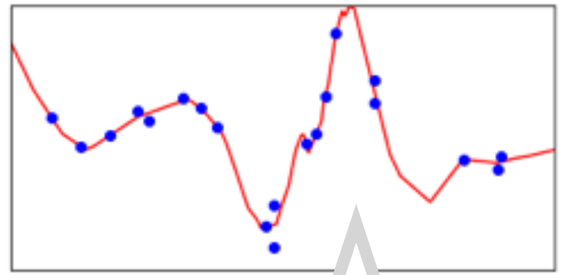
*1-**5**-1 ReLU MLP*
*Train RMSE **0.287***



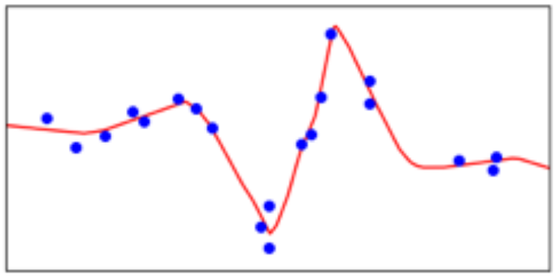
*1-**10**-1 ReLU MLP*
*Train RMSE **0.0191***



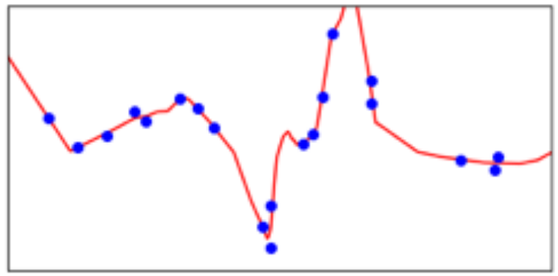
*1-**100**-1 ReLU MLP*
*Train RMSE **0.00816***



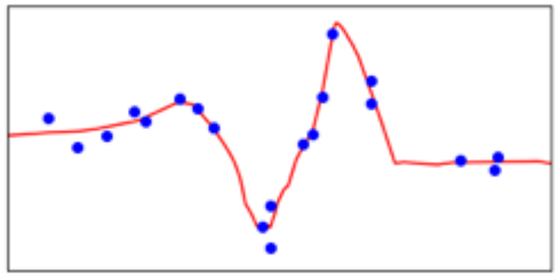
*Train RMSE **0.0139***



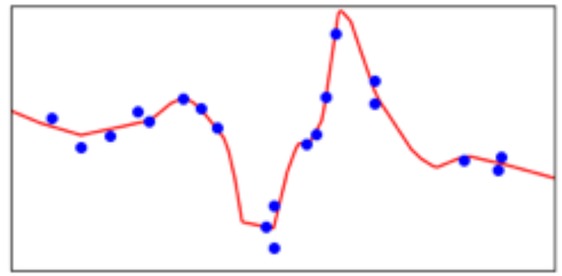
*Train RMSE **0.0066***



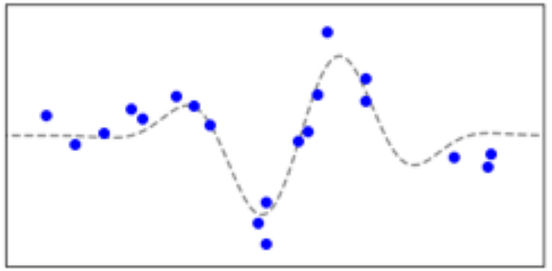
*Train RMSE **0.0121***

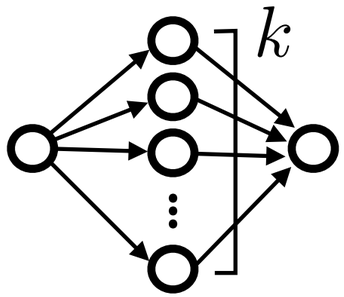


*Train RMSE **0.0139***



1-k-1 ReLU MLPのfitting

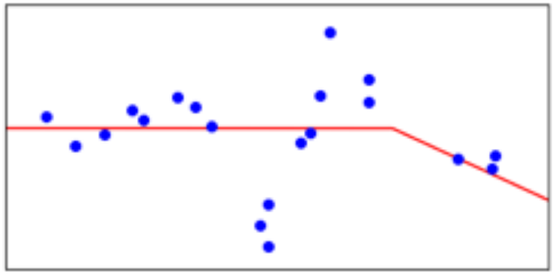




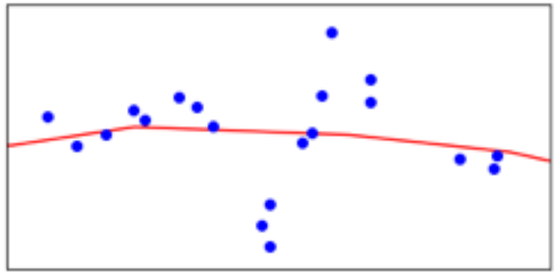
arch:
(Linear(1, k),
ReLU,
Linear(k, 1))

*Optimized With **Adam***

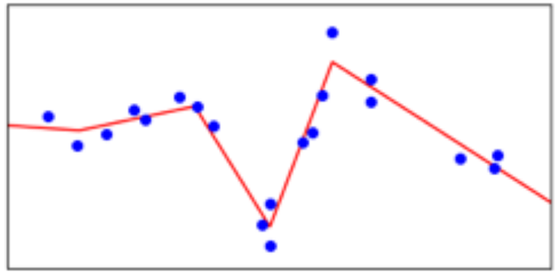
*1-**1**-1 ReLU MLP*
*Train RMSE **0.289***



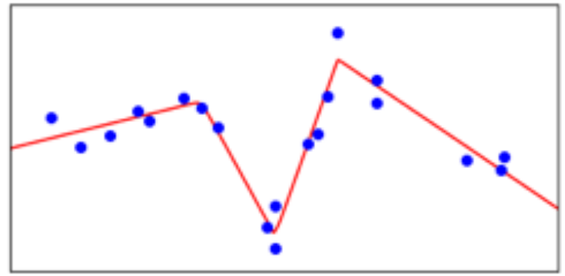
*1-**5**-1 ReLU MLP*
*Train RMSE **0.297***



*1-**10**-1 ReLU MLP*
*Train RMSE **0.0237***

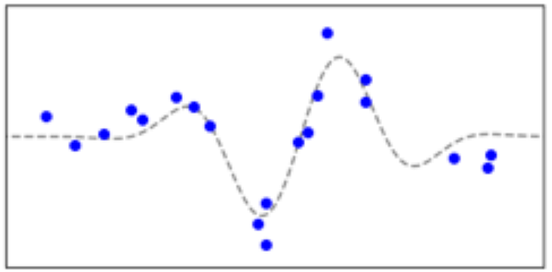


*1-**100**-1 ReLU MLP*
*Train RMSE **0.0243***



初期値に依存するばらつきは少なく
かなりの確率でコレになる

1-(k x 9)-1 ReLU MLPのfitting

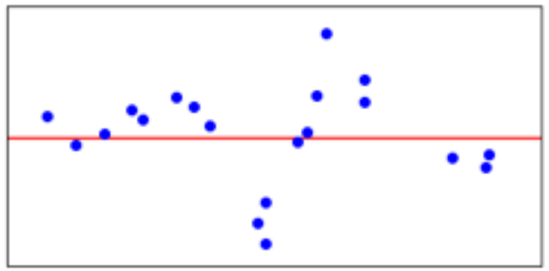


arch:
(Linear(1, k),
ReLU,
Linear(k, k),
ReLU,
Linear(k, 1))
x 9

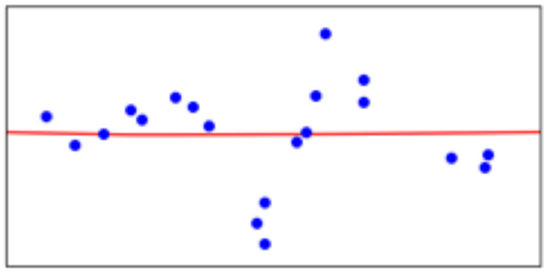
*Optimized With **Adam***

※初期値に依存するばらつきは結構ある

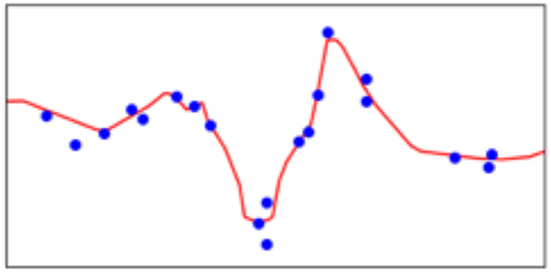
***k=1** ReLU MLP*
*Train RMSE **0.305***



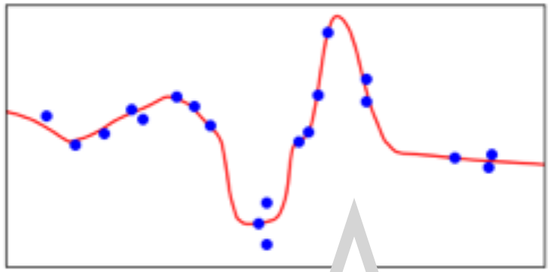
***k=5** ReLU MLP*
*Train RMSE **0.303***



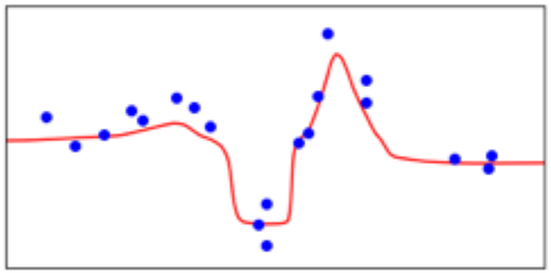
***k=10** ReLU MLP*
*Train RMSE **0.0131***



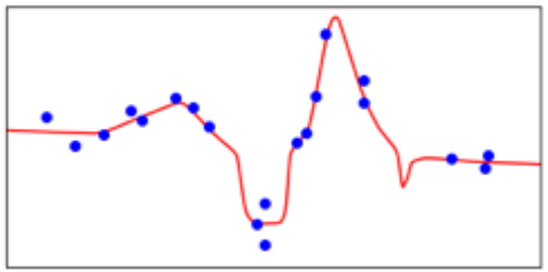
***k=100** ReLU MLP*
*Train RMSE **0.00979***



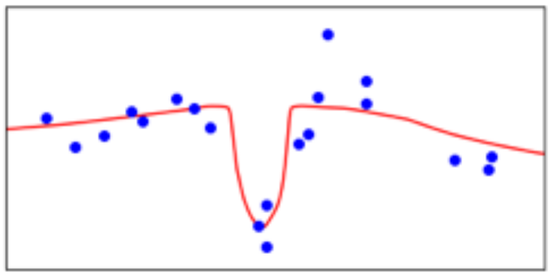
*Train RMSE **0.0443***



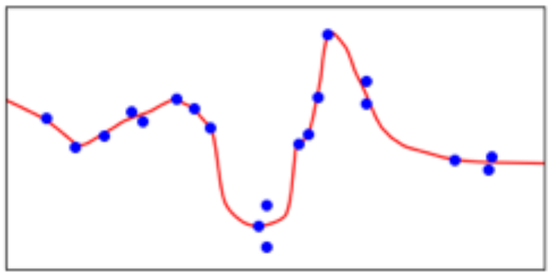
*Train RMSE **0.0124***



*Train RMSE **0.0811***



*Train RMSE **0.00864***



今日の話題提供

業務(自然科学での機械学習利活用)でユーザとして**決定木アンサンブル(とニューラルネット)**を使っていて出会った現象と問題の紹介

- 決定森回帰の**信頼区間**推定・Benign Overfitting
- 多変量木とReLUネットの**入力空間分割**

ReLU Network

Daubechies, I., DeVore, R., Foucart, S. *et al.* **Nonlinear Approximation and (Deep) ReLU Networks.** *Constr Approx* **55,** 127–172 (2022). https://doi.org/10.1007/s00365-021-09548-z

Nonlinear Approximation and (Deep) ReLU Networks

I. Daubechies, R. DeVore, S. Foucart, B. Hanin, and G. Petrova [1]

1D(単変量)の問題で詳しく解析していて分かりやすい

↳ ちなみにただのアルファベット順、責任著者はDeVore

Ingrid Daubechies

	すべて	2017 年以来
引用	101573	19776
h 指標	81	43
i10 指標	189	122

Ronald DeVore

	すべて	2017 年以来
引用	29682	7711
h 指標	70	36
i10 指標	157	91

ReLU Network

To set some notation, recall the definition of the ReLU function applied to $x = (x_1, \ldots, x_d) \in \mathbb{R}^d$:

$$\mathrm{ReLU}(x_1, \ldots, x_d) = (\mathrm{ReLU}(x_1), \ldots, \mathrm{ReLU}(x_d)) = (\max\{0, x_1\}, \ldots, \max\{0, x_d\}).$$

ReLU
= 負値を0置換

Definition 2.1. *A fully connected feed-forward* ReLU *network* $\mathcal{N}$ *with width* W *and depth* L *is a collection of weight matrices* $M^{(0)}, \ldots, M^{(L)}$ *and bias vectors* $b^{(0)}, \ldots, b^{(L)}$. *The matrices* $M^{(\ell)}$, $\ell = 1, \ldots, L-1$, *are of size* $W \times W$, *whereas* $M^{(0)}$ *has size* $W \times 1$, *and* $M^{(L)}$ *has size* $1 \times W$. *The biases* $b^{(\ell)}$ *are vectors of size* W *if* $\ell = 0, \ldots, L-1$ *and a scalar if* $\ell = L$. *Each such network* $\mathcal{N}$ *produces a univariate real-valued function*

$$A^{(L)} \circ \mathrm{ReLU} \circ A^{(L-1)} \circ \cdots \circ \mathrm{ReLU} \circ A^{(0)}(x), \quad x \in \mathbb{R},$$

where

$$A^{(\ell)}(y) = M^{(\ell)} y + b^{(\ell)}, \quad \ell = 0, \ldots, L.$$

Daubechies, I., DeVore, R., Foucart, S. *et al.* **Nonlinear Approximation and (Deep) ReLU Networks.** *Constr Approx* **55,** 127–172 (2022). https://doi.org/10.1007/s00365-021-09548-z

ReLU Network

- ReLU Networkが表現する関数は「Continuous Piecewise Linear (CPwL) function」

$\Upsilon^{W,L} := \{S : \mathbb{R} \to \mathbb{R},\ S \text{ is produced by a ReLU network of width } W \text{ and depth } L\}$

$n(W, L)$ the number of its parameters

$\Sigma_n := \{S : \mathbb{R} \to \mathbb{R},\ S \text{ is a CPwL function with at most } n \text{ distinct breakpoints in } (0, 1)\}$

The elements of Σ_n are also called free knot linear splines.

Several interesting results [6, 21, 29] show that, for arbitrarily large $k \geq 1$ and $n = n(W, L)$ sufficiently large,

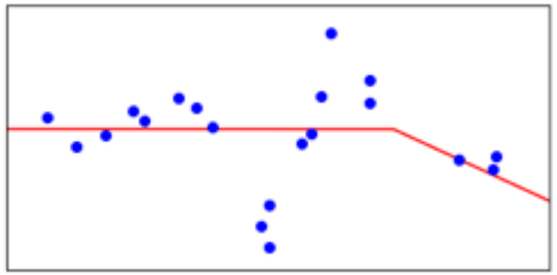
$$\Upsilon^{W,L} \setminus \Sigma_{n^k} \neq \emptyset, \tag{2}$$

Daubechies, I., DeVore, R., Foucart, S. *et al.* **Nonlinear Approximation and (Deep) ReLU Networks.** *Constr Approx* **55,** 127–172 (2022). https://doi.org/10.1007/s00365-021-09548-z

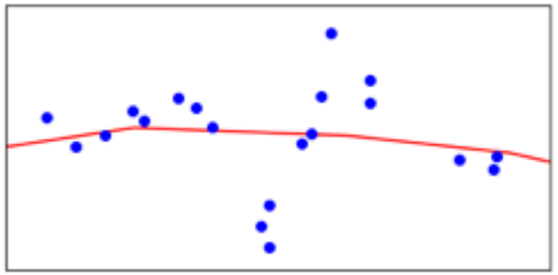
ReLU Network vs 決定木・決定森

- ReLU Networkが表現する関数は「Continuous Piecewise Linear (CPwL) function」

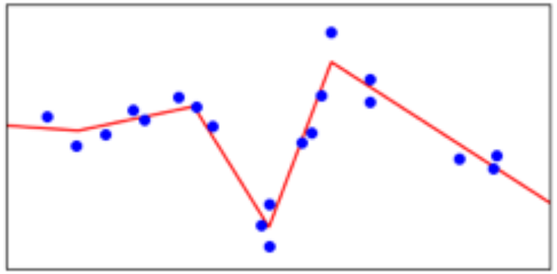
*1-**1**-1 ReLU MLP*
*Train RMSE **0.289***



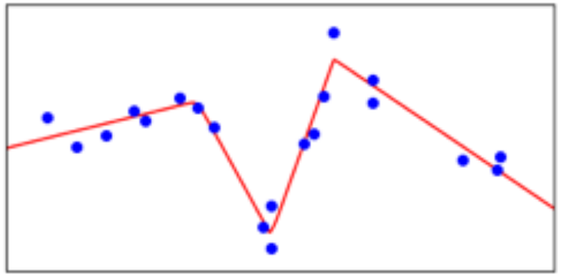
*1-**5**-1 ReLU MLP*
*Train RMSE **0.297***



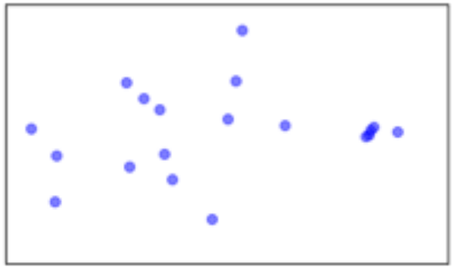
*1-**10**-1 ReLU MLP*
*Train RMSE **0.0237***



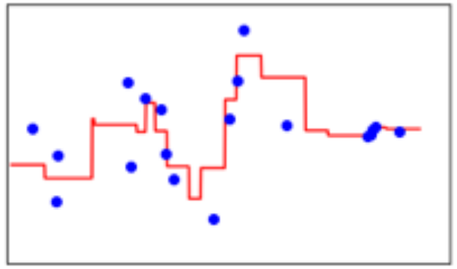
*1-**100**-1 ReLU MLP*
*Train RMSE **0.0243***



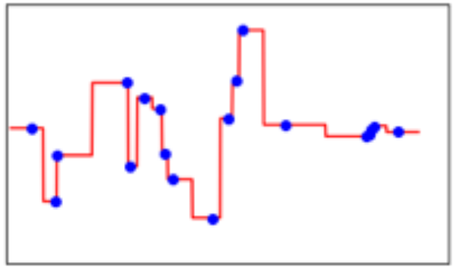
- 決定木・決定森(や最近隣法)が表現する関数は「Piecewise Constant function」



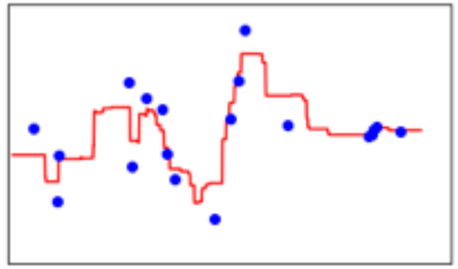
Nearest Neighbors



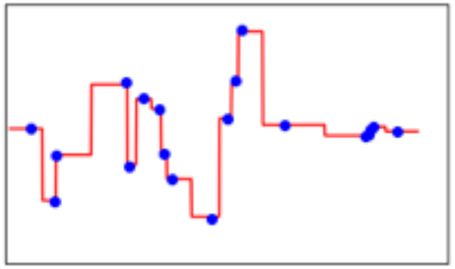
Decision Tree



Random Forest



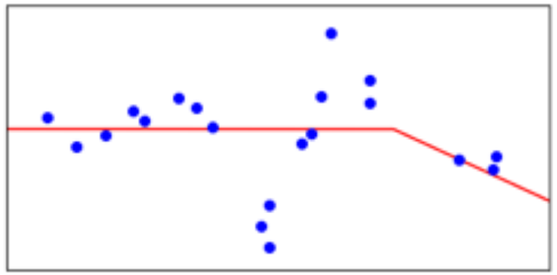
Gradient Boosted Trees



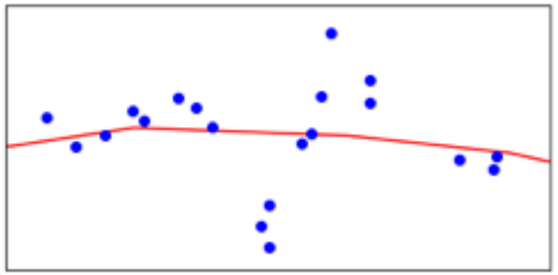
ReLU Network

- ReLU Networkが表現する関数は「Continuous Piecewise Linear (CPwL) function」

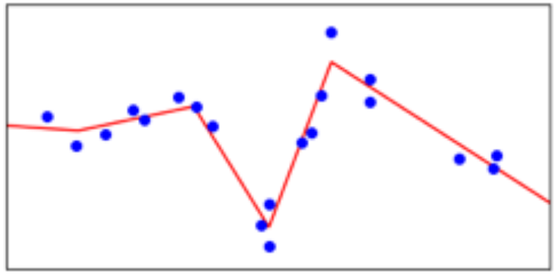
*1-**1**-1 ReLU MLP*
*Train RMSE **0.289***



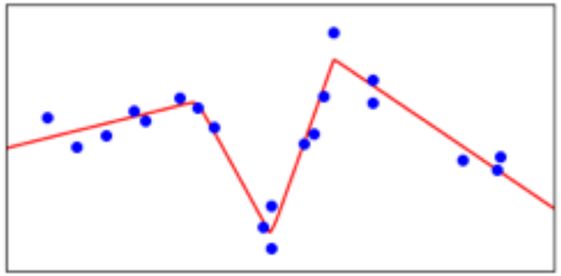
*1-**5**-1 ReLU MLP*
*Train RMSE **0.297***



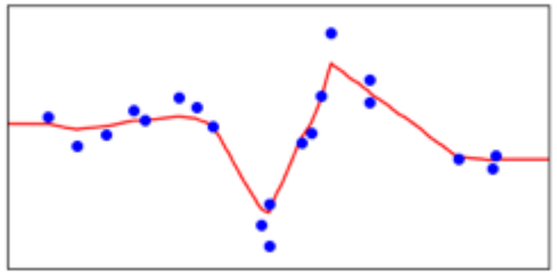
*1-**10**-1 ReLU MLP*
*Train RMSE **0.0237***



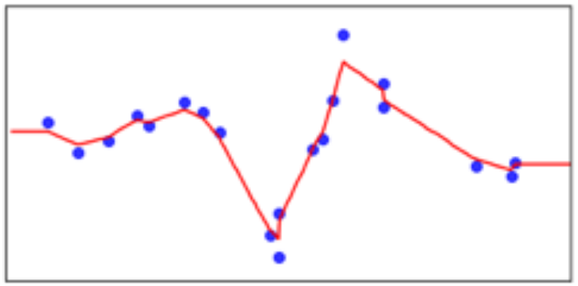
*1-**100**-1 ReLU MLP*
*Train RMSE **0.0243***



*ExtraTrees (**#trees=10³**, #leaves=8, bootstrap=off), Train RMSE **0.0243***



***ExtraTrees** (**#trees=1000**, bootstrap)*
*Train RMSE **0.0243***



ExtraTreesも区分的定数なのだが
木数を増やしていくと非常に
興味深い挙動を示す…

この性質はExtraTreesの元論文でももちろん議論されている

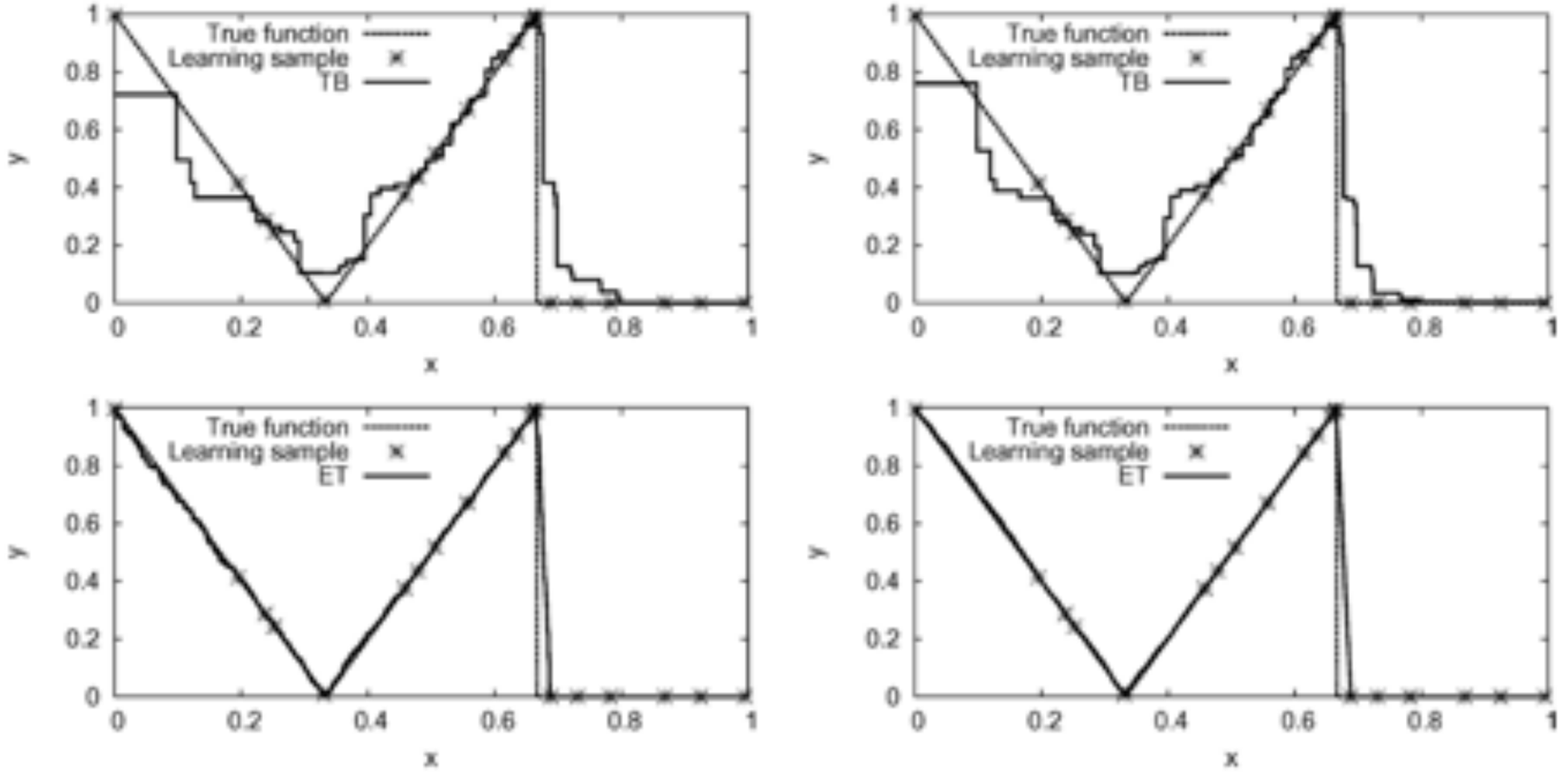
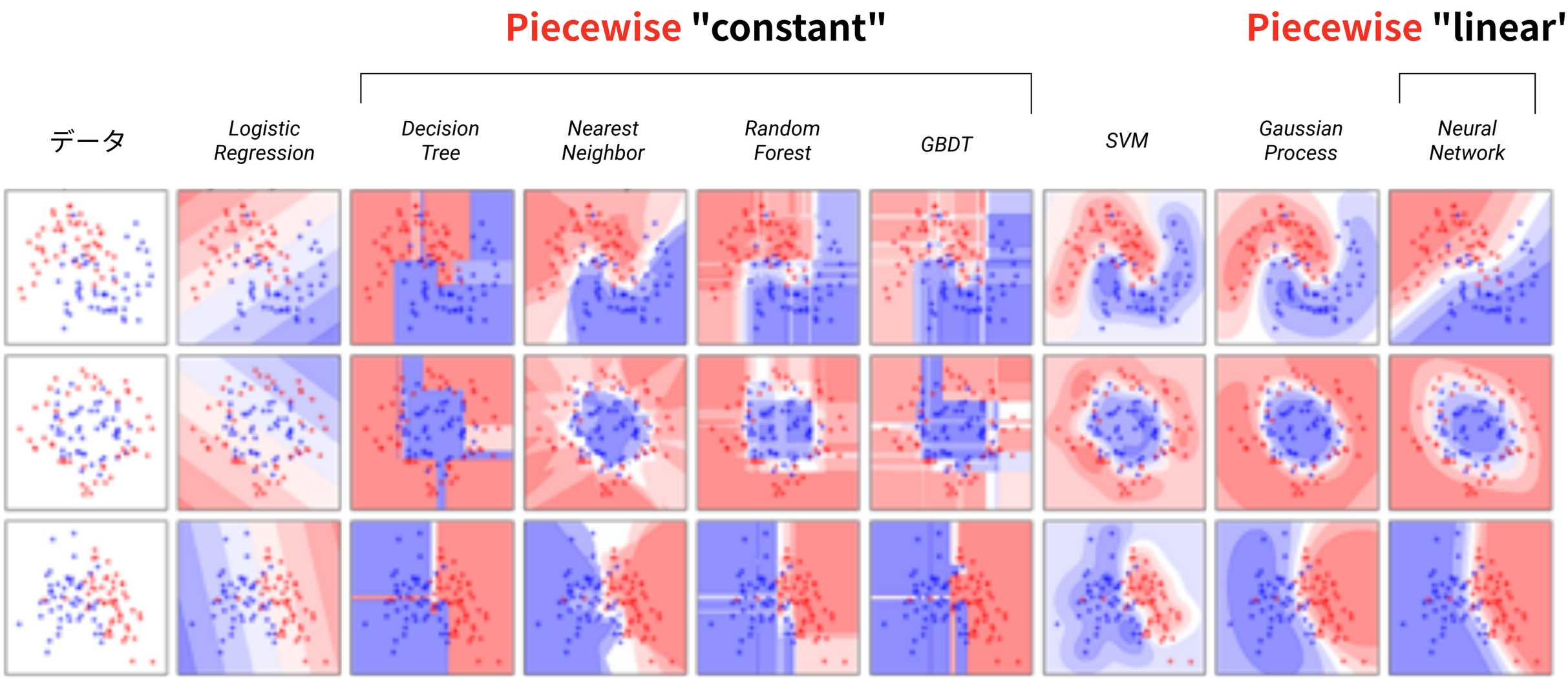


Fig. 10 Tree Bagging, and fully developed Extra-Trees ($n_{\min} = 2$) on a one-dimensional piecewise linear problem ($N = 20$). Left with $M = 100$ trees, right with $M = 1000$ trees.

Geurts, P., Ernst, D. & Wehenkel, L. Extremely randomized trees. Mach Learn 63, 3–42 (2006). https://doi.org/10.1007/s10994-006-6226-1

いずれにせよ領域ごとのlocalityが理解の鍵になる？



ReLU Networkの入力空間分割を考える

ICML 2018

A Spline Theory of Deep Networks

Randall Balestriero [1] **Richard G. Baraniuk** [1]

Abstract

We build a rigorous bridge between deep networks (DNs) and approximation theory via spline functions and operators. Our key result is that a large class of DNs can be written as a composition of *max-affine spline operators* (MASOs), which provide a powerful portal through which to view and analyze their inner workings. For instance, conditioned on the input signal, the output of a MASO DN can be written as a simple affine transformation of the input. This implies that a DN constructs a set of signal-dependent, class-specific templates against which the signal is compared via a simple inner product; we explore the links to the classical theory of optimal classification via matched filters and the effects of data memorization. Going further, we propose a simple penalty term that can be added to the cost function of any DN learning algorithm to force the templates to be orthogonal with each other; this leads to significantly improved classification performance and reduced overfitting with no change to the DN architecture. The spline partition of the input signal

and then significantly improving performance over classical approaches.

Despite this empirical progress, the precise mechanisms by which deep learning works so well remain relatively poorly understood, adding an air of mystery to the entire field. Ongoing attempts to build a rigorous mathematical framework fall roughly into five camps: (i) probing and measuring DNs to visualize their inner workings (Zeiler & Fergus, 2014); (ii) analyzing their properties such as expressive power (Cohen et al., 2016), loss surface geometry (Lu & Kawaguchi, 2017; Soudry & Hoffer, 2017), nuisance management (Soatto & Chiuso, 2016), sparsification (Papyan et al., 2017), and generalization abilities; (iii) new mathematical frameworks that share some (but not all) common features with DNs (Bruna & Mallat, 2013); (iv) probabilistic generative models from which specific DNs can be derived (Arora et al., 2013; Patel et al., 2016); and (v) information theoretic bounds (Tishby & Zaslavsky, 2015).

In this paper, we build a rigorous bridge between DNs and *approximation theory* via *spline functions and operators*. We prove that a large class of DNs — including convolutional neural networks (CNNs) (LeCun, 1998), residual

NeurIPS 2019

The Geometry of Deep Networks: Power Diagram Subdivision

Randall Balestriero ECE Department Rice University

Romain Cosentino ECE Department Rice University

Behnaam Aazhang ECE Department Rice University

Richard Baraniuk ECE Department Rice University

Abstract

We study the geometry of deep (neural) networks (DNs) with piecewise affine and convex nonlinearities. The layers of such DNs have been shown to be *max-affine spline operators* (MASOs) that partition their input space and apply a region-dependent affine mapping to their input to produce their output. We demonstrate that each MASO layer's input space partitioning corresponds to a *power diagram* (an extension of the classical Voronoi tiling) with a number of regions that grows exponentially with respect to the number of units (neurons). We further show that a composition of MASO layers (e.g., the entire DN) produces a progressively subdivided power diagram and provide its analytical form. The subdivision process constrains the affine maps on the (exponentially many) power diagram regions to greatly reduce their complexity. For classification problems, we obtain a formula for a MASO DN's decision boundary in the input space plus a measure of its curvature that depends on the DN's nonlinearities, weights, and architecture. Numerous numerical experiments support and extend our theoretical results.

線形スプライン(多変量アフィンスプライン)

"A large class of DNs can be written as a **composition** of ***max-affine spline operators (MASOs)***"

Multivariate Affine Splines. Consider a *partition* of a domain $\mathbb{R}^D$ into a set of regions $\Omega = \{\omega_1, \ldots, \omega_R\}$ and a set of local mappings $\Phi = \{\phi_1, \ldots, \phi_R\}$ that map each region in the partition to $\mathbb{R}$ via $\phi_r(\boldsymbol{x}) := \langle [\alpha]_{r,:}, \boldsymbol{x} \rangle + [\beta]_r$ for $\boldsymbol{x} \in \omega_r$.

$$P(\boldsymbol{x}; \boldsymbol{a}_{:}, b_{:}) = \sum_{r=1}^{|\Omega|} \left(\langle \boldsymbol{a}_r, \boldsymbol{x} \rangle + b_r \right) \mathbf{1}_{\{\boldsymbol{x} \in \omega_r\}}$$

Disjointな領域のpartitionがあり
各領域ごとに関数(ふつう多項式)を
applyするけど「境界で連続になる
ように」なっているのが**スプライン**

その中でも写像が「1次+定数項」の
Affineスプラインを考える

スプラインのフィッティングは一般には
「領域ごとの写像」と「領域分割」の同時最適化になり非常に難しい

Max-Affine Spline (MAS)

"A large class of DNs can be written as a **composition** of ***max-affine spline operators (MASOs)***"

Multivariate Affine Splines. Consider a *partition* of a domain $\mathbb{R}^D$ into a set of regions $\Omega = \{\omega_1, \ldots, \omega_R\}$ and a set of local mappings $\Phi = \{\phi_1, \ldots, \phi_R\}$ that map each region in the partition to $\mathbb{R}$ via $\phi_r(\boldsymbol{x}) := \langle [\alpha]_{r,:}, \boldsymbol{x} \rangle + [\beta]_r$ for $\boldsymbol{x} \in \omega_r$.

$$P(\boldsymbol{x}; \boldsymbol{a}_{:}, b_{:}) = \sum_{r=1}^{|\Omega|} \left(\langle \boldsymbol{a}_r, \boldsymbol{x} \rangle + b_r \right) \mathbf{1}_{\{\boldsymbol{x} \in \omega_r\}}$$

Max-Affine Spline Functions.

$$P(\boldsymbol{x}; \boldsymbol{a}_{:}, b_{:}) = \max_{r=1,\ldots,R} \langle \boldsymbol{a}_r, \boldsymbol{x} \rangle + b_r$$

Disjointな領域のpartitionがあり
各領域ごとに関数(ふつう多項式)を
applyするけど「境界で連続になる
ように」なっているのが**スプライン**

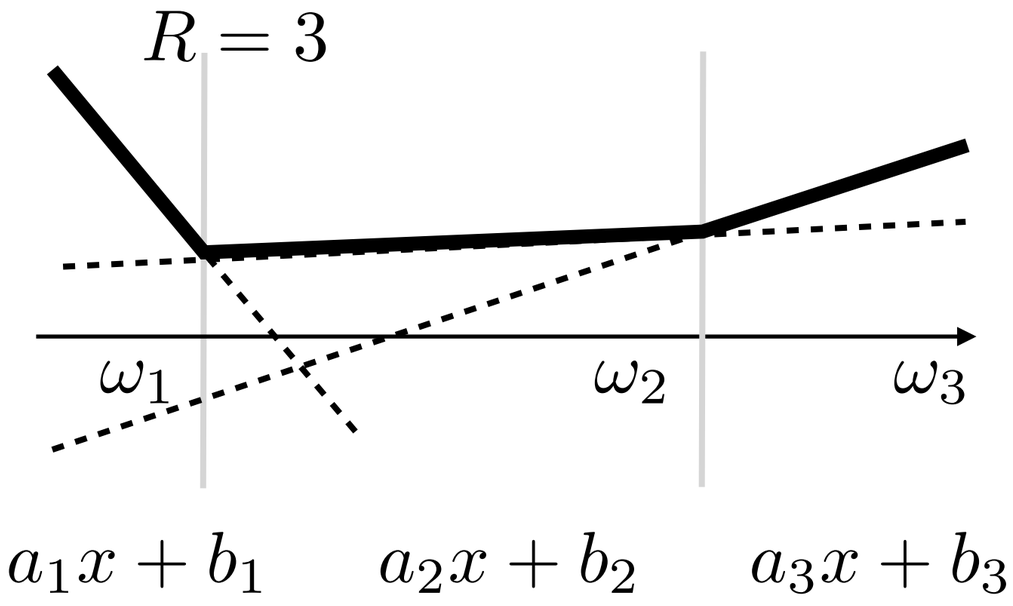
その中でも写像が「1次+定数項」の
Affineスプラインを考える

Max Affineスプラインでは明示的に
「領域分割」を考える必要がない！

Max-Affine Spline (MAS)

Max-Affine Spline Functions.

$$P(\boldsymbol{x}; \boldsymbol{a}_{:}, b_{:}) = \max_{r=1,\ldots,R} \langle \boldsymbol{a}_r, \boldsymbol{x} \rangle + b_r$$



Max Affineスプラインでは明示的に
「領域分割」を考える必要がない！

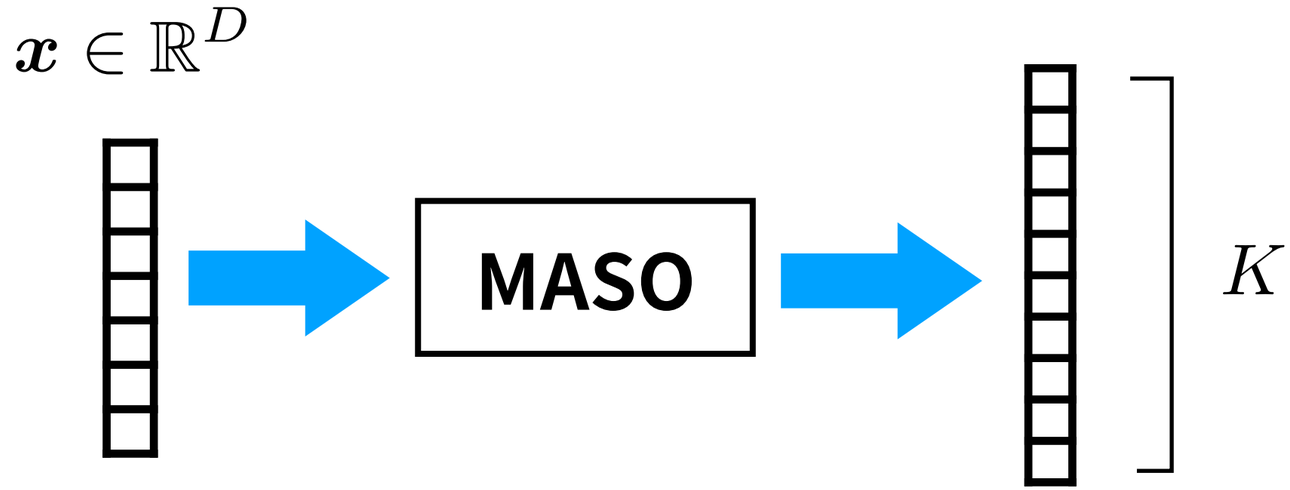
- **Globally convex**な任意のAffineスプラインはMASで書ける
- MASは常にpiecewise affineかつglobally convexなので、どんなパラメタでも常にcontinuous！
- 逆に任意のpiecewise affine, **globally convex**かつcontinuousな関数はMASとして書ける。

Max-Affine Spline Operator (MASO)

"A large class of DNs can be written as a **composition** of ***max-affine spline operators (MASOs)***"

$$M(\boldsymbol{x}; \boldsymbol{A}_{\cdot}, \boldsymbol{b}_{\cdot}) = \max_{r=1,\dots,R} (\boldsymbol{A}_r \boldsymbol{x} + \boldsymbol{b}_r) = \begin{bmatrix} \max_{r=1,\dots,R} \langle [\boldsymbol{A}_r]_{1,:}, \boldsymbol{x} \rangle + [\boldsymbol{b}_r]_1 \\ \vdots \\ \max_{r=1,\dots,R} \langle [\boldsymbol{A}_r]_{K,:}, \boldsymbol{x} \rangle + [\boldsymbol{b}_r]_K \end{bmatrix}$$

MASO
= K個のMASを使った
Operator



ReLU Network = MASOの合成関数

"A large class of DNs can be written as a **composition** of ***max-affine spline operators (MASOs)***"

$$f_\Theta(\boldsymbol{x}) = \left(f^{(L)}_{\theta^{(L)}} \circ \cdots \circ f^{(1)}_{\theta^{(1)}} \right)(\boldsymbol{x}) \qquad \Theta = \left\{ \theta^{(1)}, \ldots, \theta^{(L)} \right\}$$

$f^{(i)}_{\theta^{(i)}}$

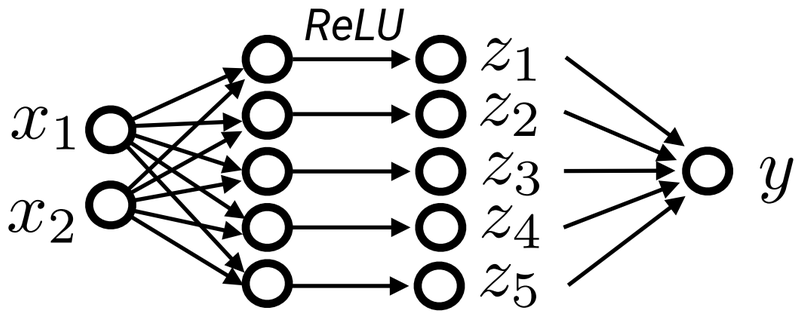
- fully-connected
- convolution
- activation
  - ReLU
  - leaky ReLU
  - absolute value
- pooling (max, average, channel, etc)
- recurrent
- skip connection

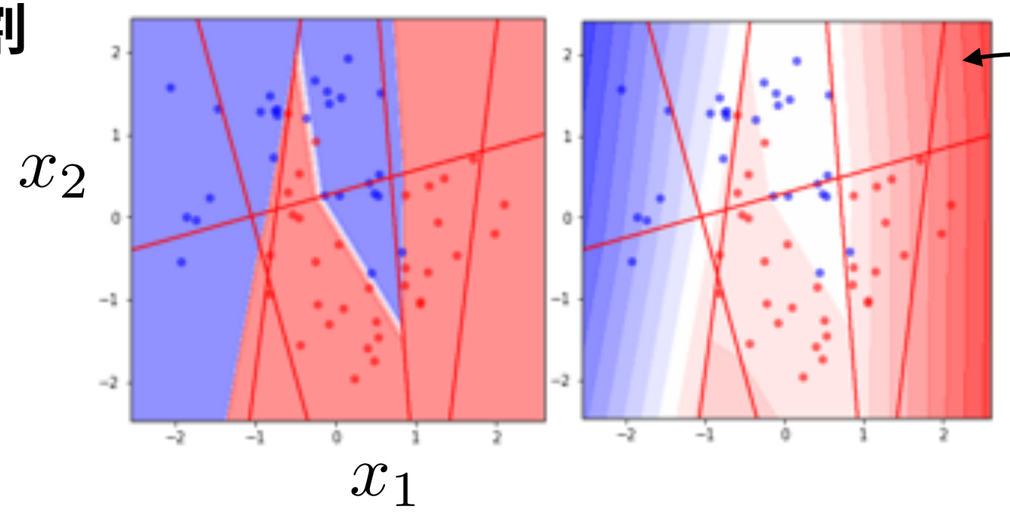
→ ***MASO***

DNs are signal-dependent affine transformations. *The particular affine mapping applied to x depends on which parrtition of the spline it falls in* R^d.

ReLU Networkの入力空間分割

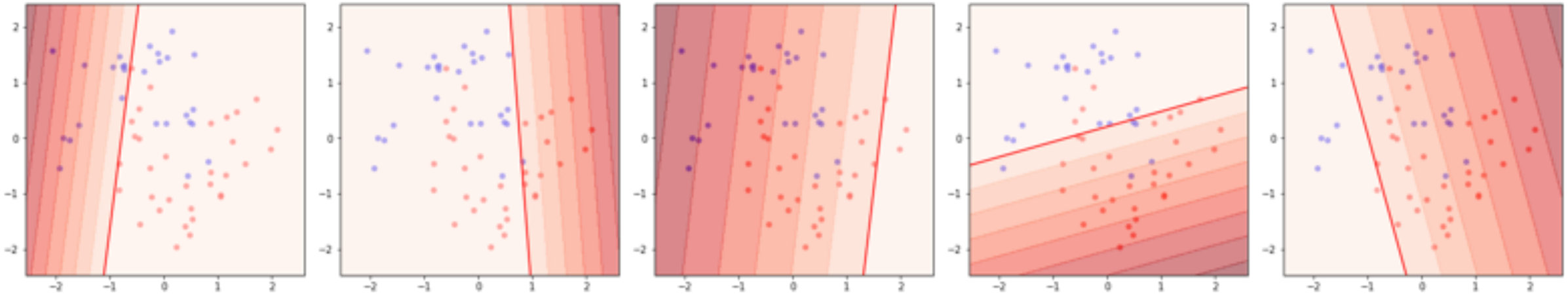
2-5-1 ReLU Networkの入力空間分割



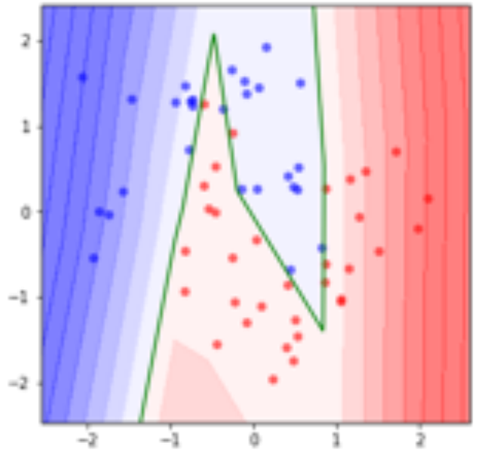


各領域では下記を
組み合わせた
Affine変換が適用
される
(境界で連続になる)

z_1 z_2 z_3 z_4 z_5

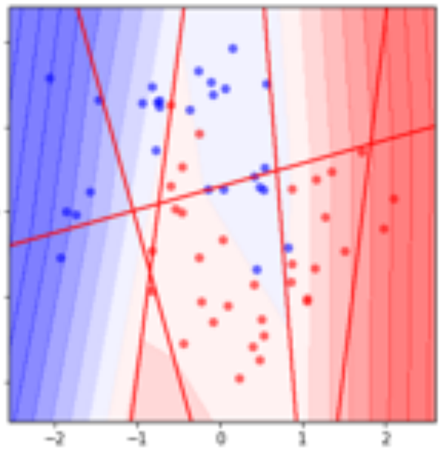


ReLU Networkの入力空間分割



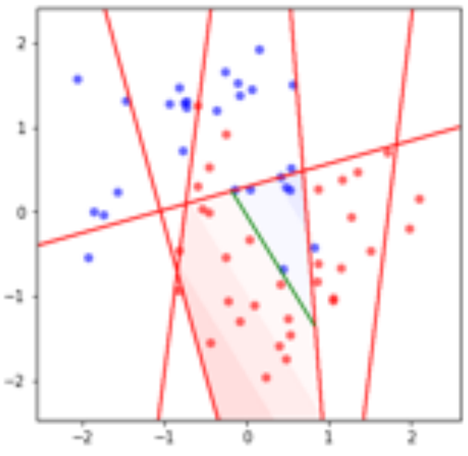
-12.43

z_1



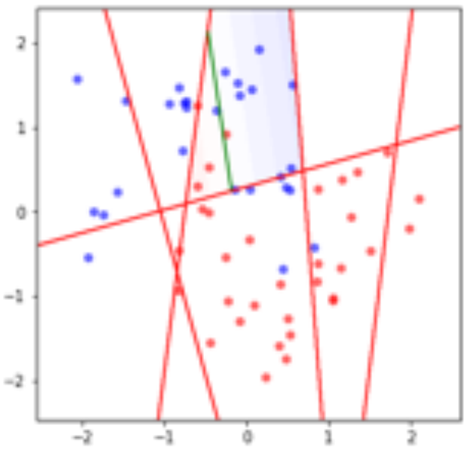
15.48

z_2



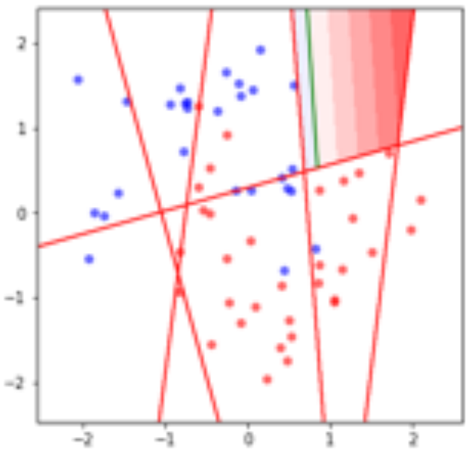
2.04

z_3



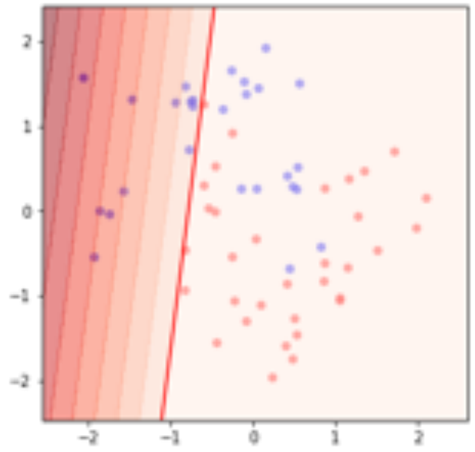
2.05

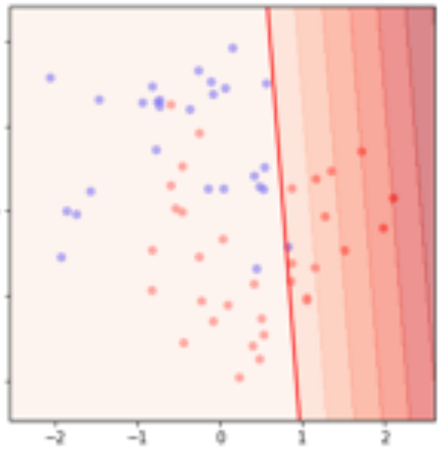
z_4

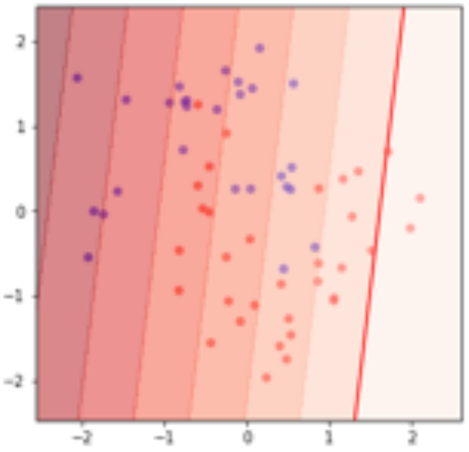


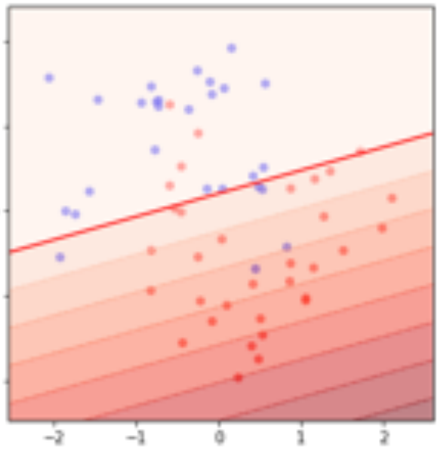
-2.48

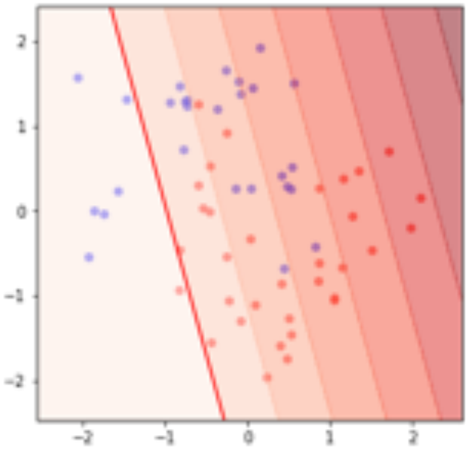
z_5







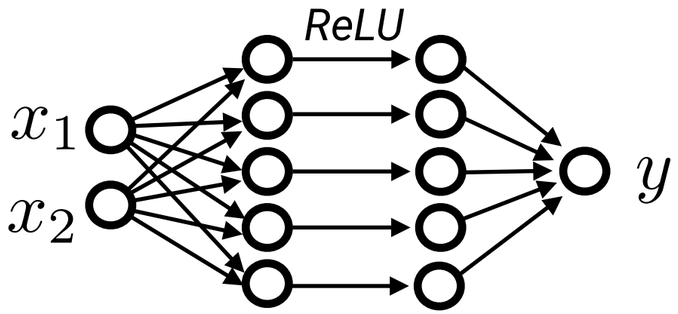


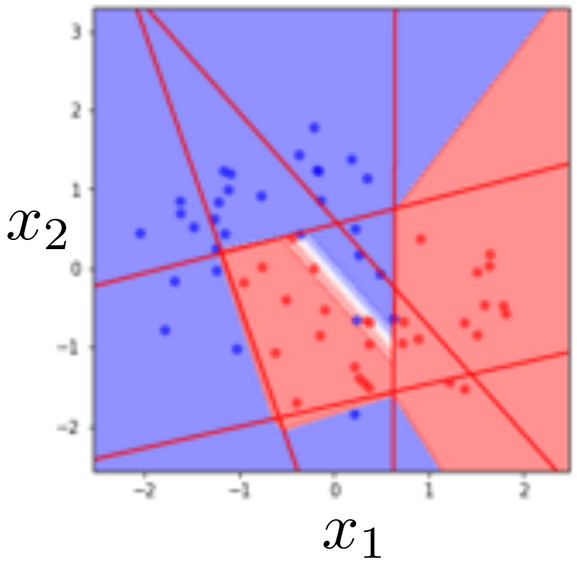


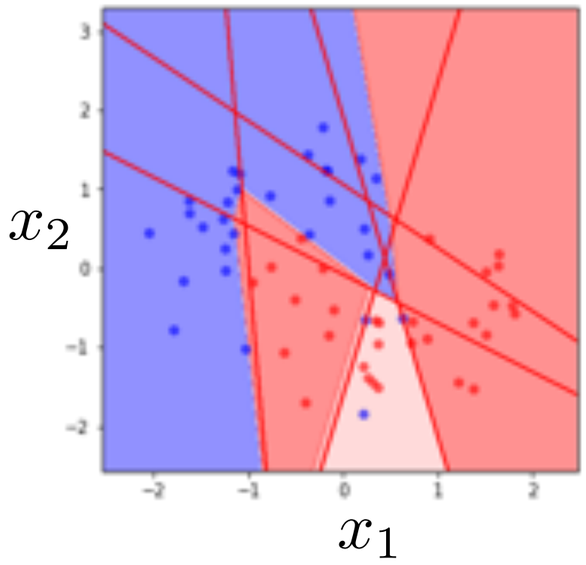
ReLU Networkの入力空間分割

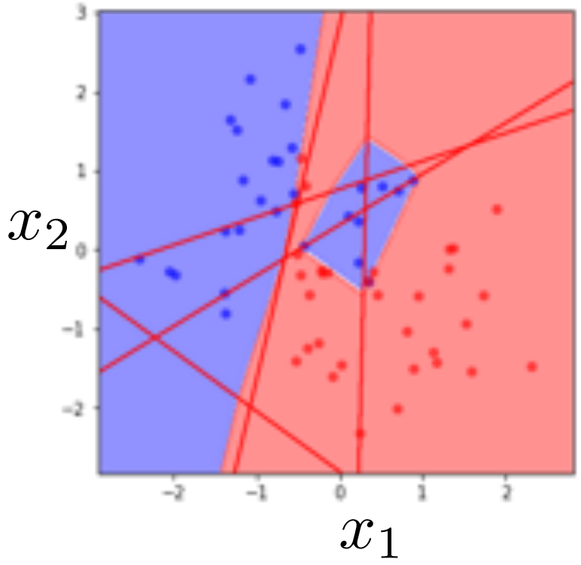
2-5-1 ReLU Networkの入力空間分割

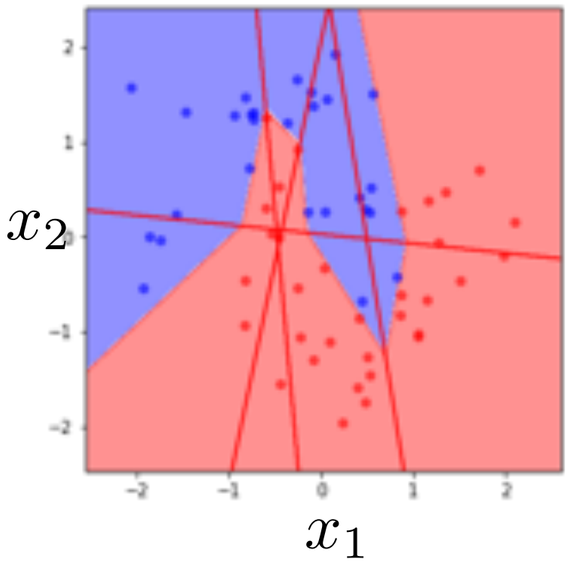
arch:
(Linear(2, 5),
ReLU,
Linear(5, 1))







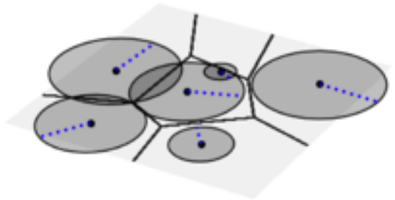


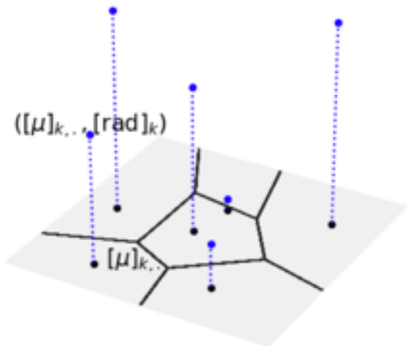


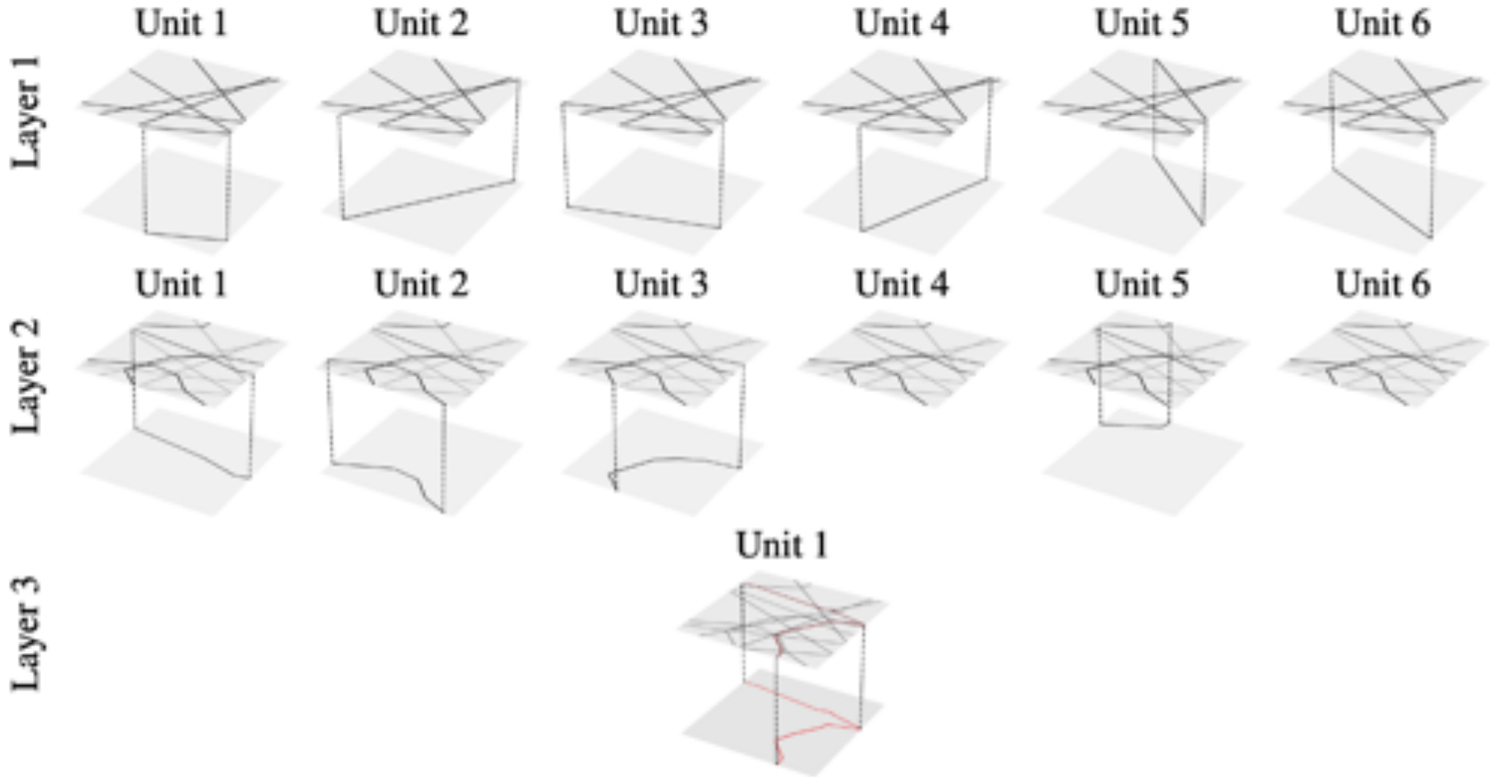
ReLU Networkの入力空間分割

Theorem 3. *DN layer partitions its input space according to a PD with* $\{1, \dots, R\}^K$ *cells, centroids* $\mu_{\boldsymbol{r}} = \sum_{k=1}^{K} [A]_{k,[\boldsymbol{r}]_k,}$. *and radii* $\mathrm{rad}_{\boldsymbol{r}} = 2\langle \boldsymbol{1}, B_{\boldsymbol{r}} \rangle + \|\mu_{\boldsymbol{r}}\|^2$ *(recall (2)).*
Corollary 2. *The input space partitioning of a DN layer is composed of convex polytopes.*

Power diagram (PD)
aka Laguerre–Voronoi diagram

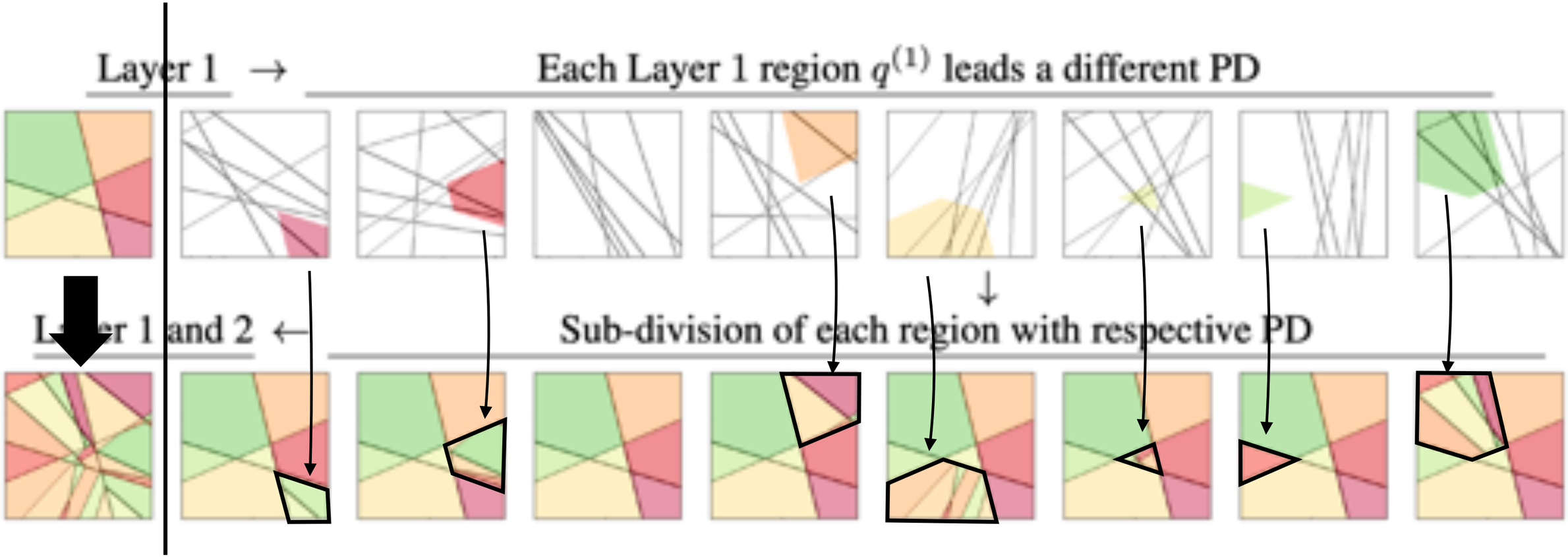






ReLU Networkの入力空間分割

多層になる際には**領域ごとに異なる領域分割で細分**されていく



ReLU Networkの入力空間分割

Piecewise Linearで「Spline」(境界で連続)ということは超平面をクシャクシャにした状態？

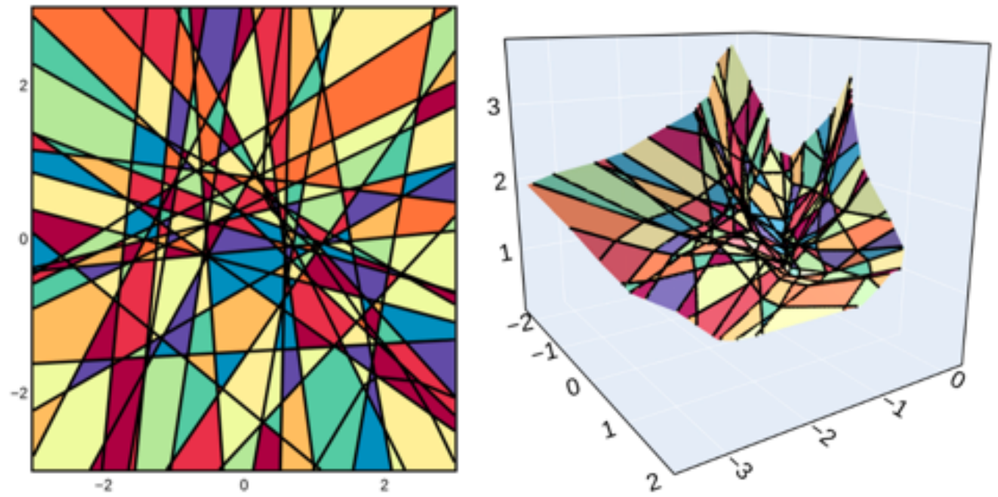


Figure 4.1 : Visual depiction of Thm. 4.1 with a (random) generator $G : \mathbb{R}^2 \mapsto \mathbb{R}^3$. **Left:** generator input space partition Ω made of polytopal regions. **Right:** generator image $Im(G)$ which is a continuous piecewise affine surface composed of the polytopes obtained by affinely transforming the polytopes from the input space partition (left) the colors are per-region and correspond between left and right plots. *This input-space-partition / generator-image / per-region-affine-mapping relation holds for any architecture employing piecewise affine activation functions. Understanding each of the three brings insights into the others, as we demonstrate in this paper.*

Theorem 4.1 (Per-region affine subspace)

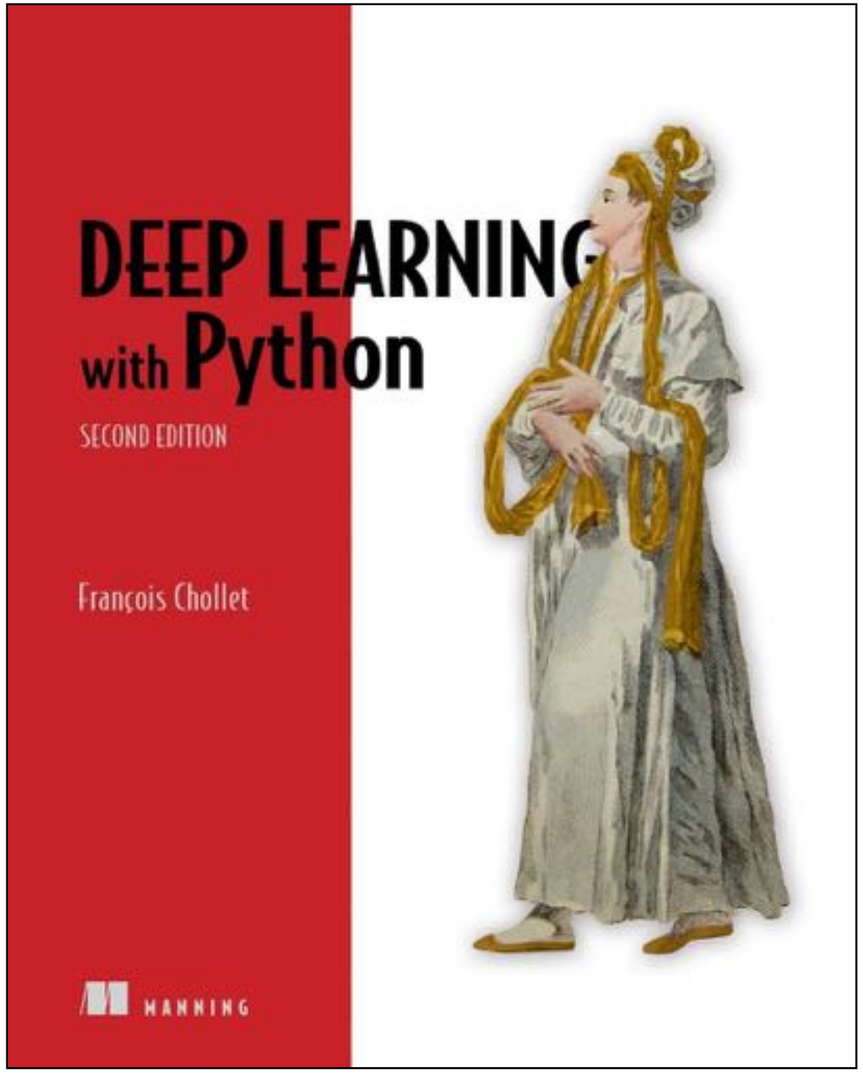
The image of a generator G employing MASO layers is a continuous piecewise affine surface made of connected polytopes obtained by affine transformations of the polytopes of the input space partition Ω as in

$$Im(G) \triangleq \{G(\boldsymbol{z}) : \boldsymbol{z} \in \mathbb{R}^S\} = \bigcup_{\omega \in \Omega} \text{Aff}(\omega; \boldsymbol{A}_\omega, \boldsymbol{b}_\omega) \tag{4.4}$$

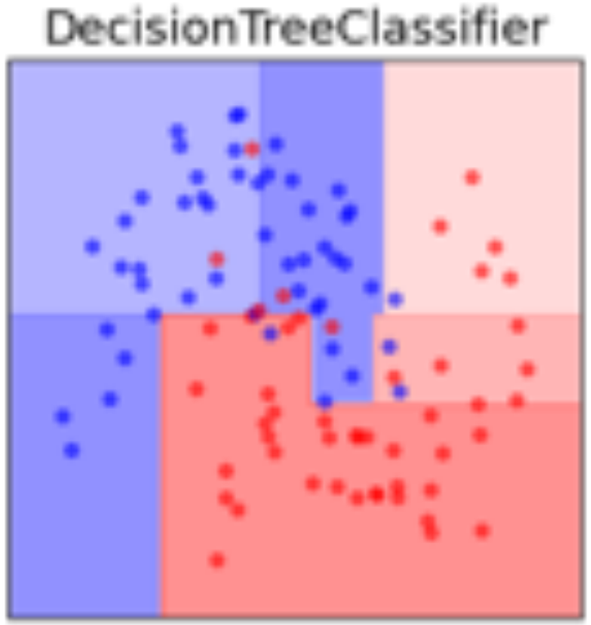
with $\text{Aff}(\omega; \boldsymbol{A}_\omega, \boldsymbol{b}_\omega) = \{\boldsymbol{A}_\omega \boldsymbol{z} + \boldsymbol{b}_\omega : \boldsymbol{z} \in \omega\}$; we will denote for conciseness $G(\omega) \triangleq \text{Aff}(\omega; \boldsymbol{A}_\omega, \boldsymbol{b}_\omega)$ and the volume of a region $\omega \in \Omega$ denoted by $\mu(\omega)$ is related to the volume of $G(\omega)$ as per $\mu(G(\omega)) = \sqrt{\det(\boldsymbol{A}_\omega^T \boldsymbol{A}_\omega)}\mu(\omega)$ with $\boldsymbol{A}_\omega$ being full-rank.

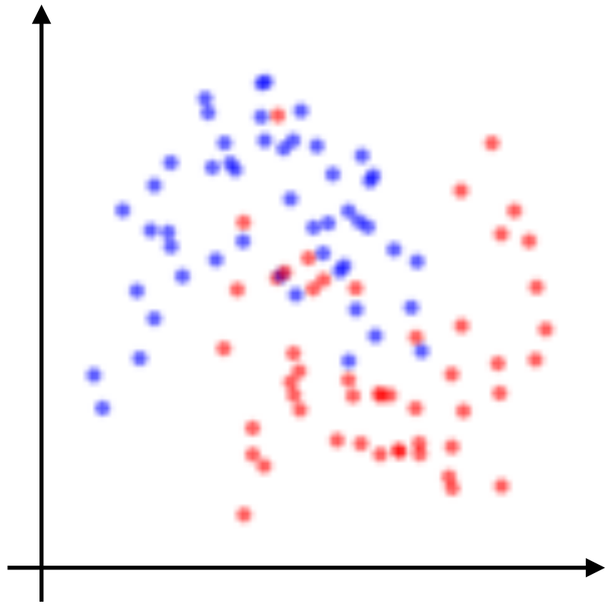
Balestriero, Randall. "Max-Affine Splines Insights Into Deep Learning." (2021) Diss., Rice University. https://hdl.handle.net/1911/110439.

Neural network as locality-sensitive hashing

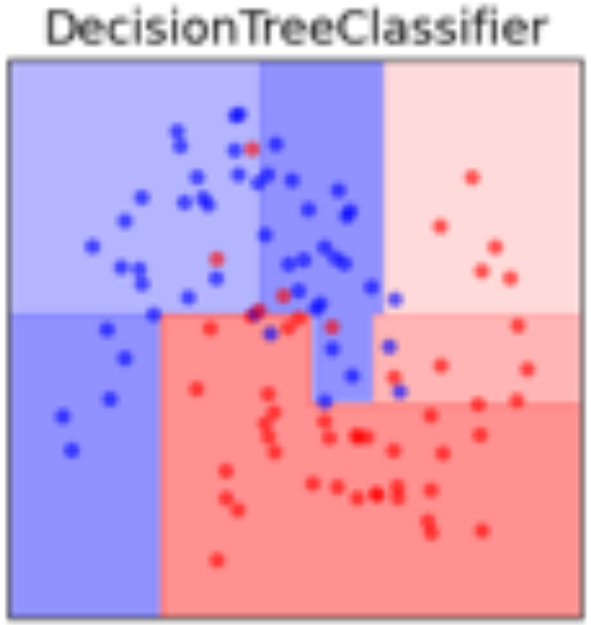


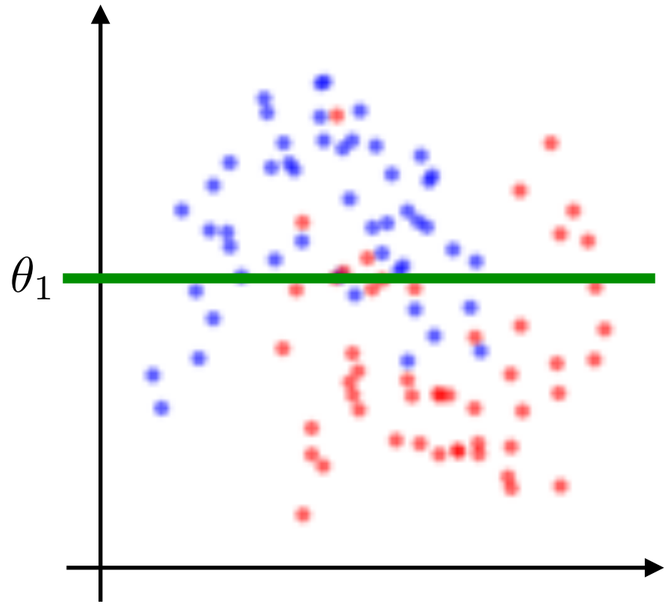
決定木の入力空間分割

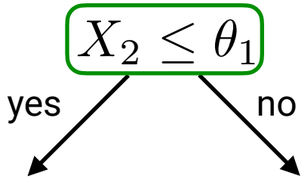




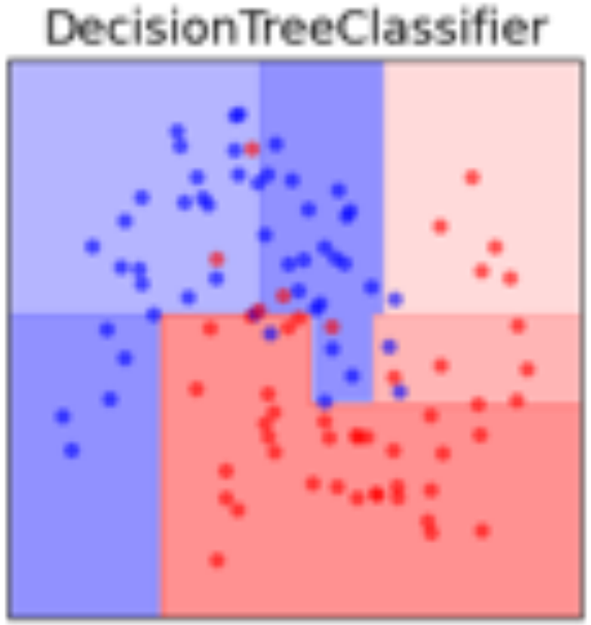
決定木の入力空間分割

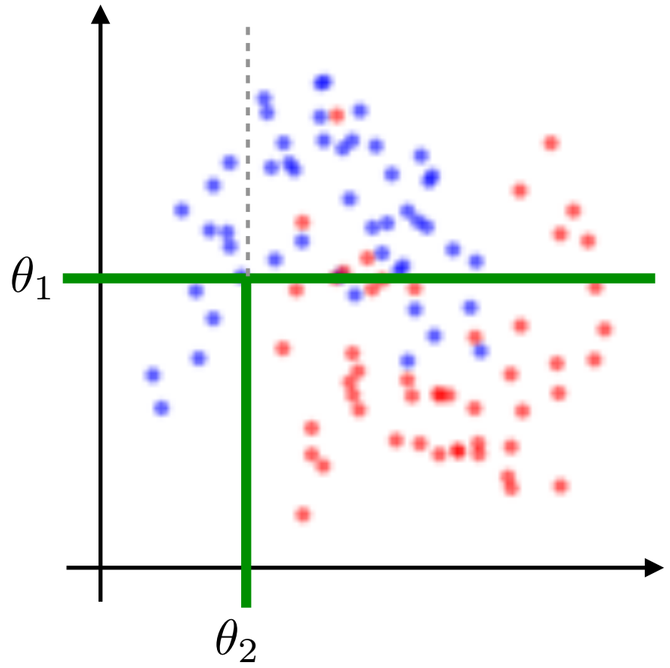


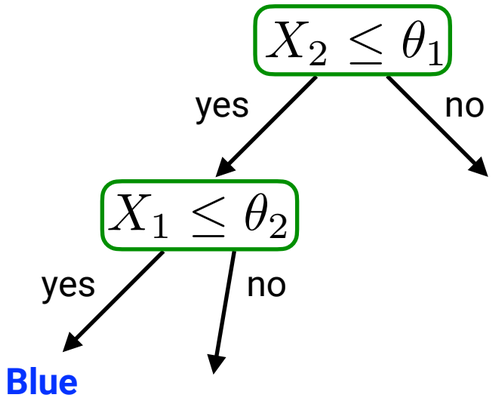




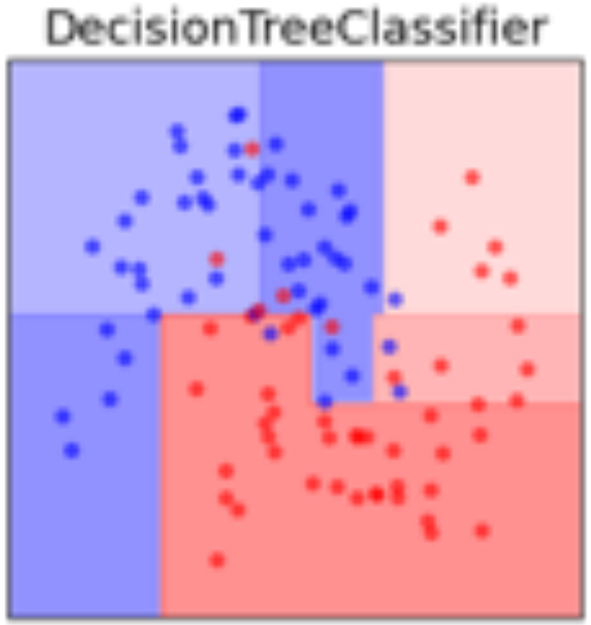
決定木の入力空間分割

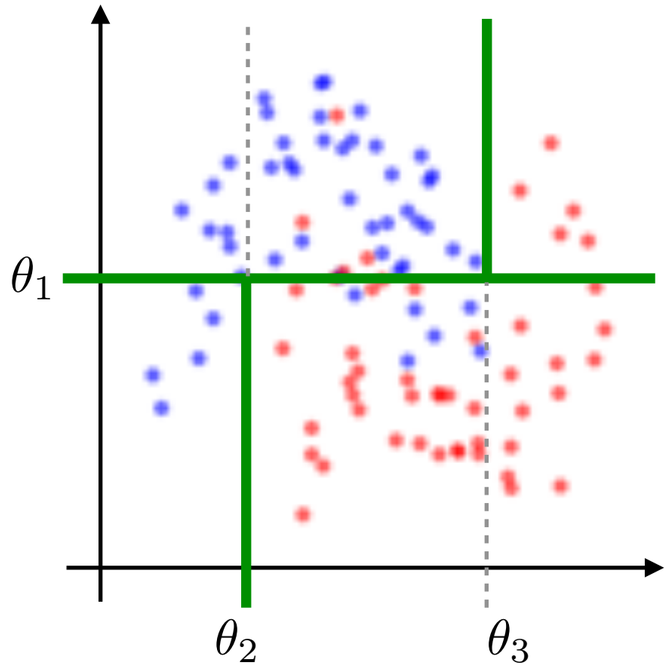

DecisionTreeClassifier

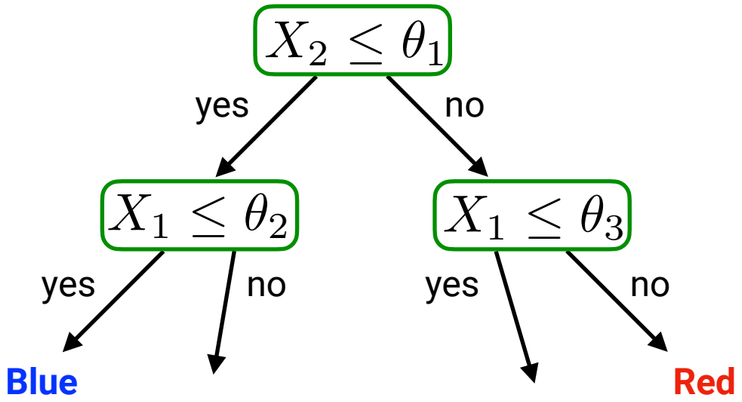

θ_1
θ_2


$X_2 \leq \theta_1$
yes
no
$X_1 \leq \theta_2$
yes
no
Blue

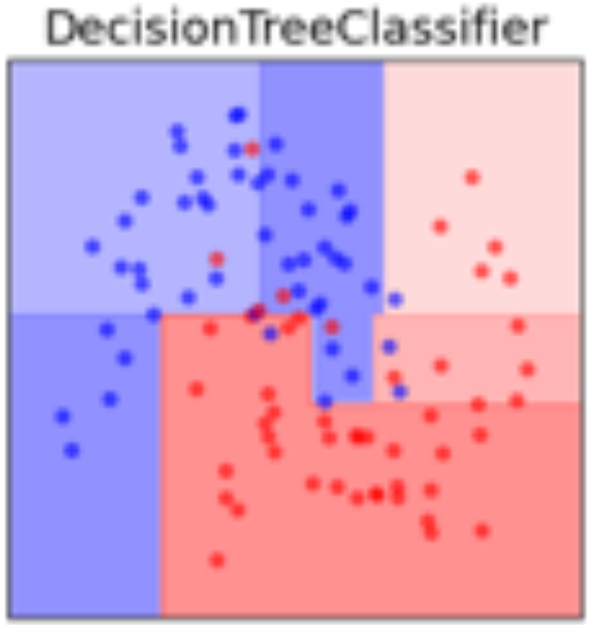
決定木の入力空間分割

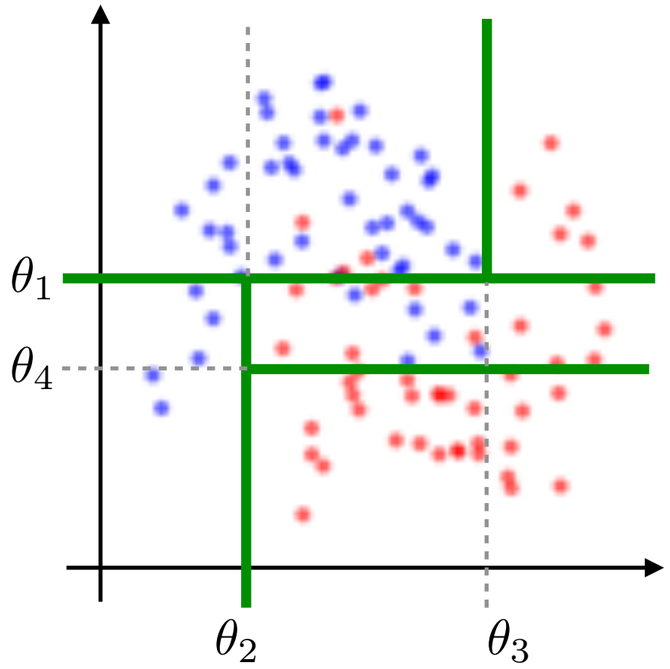

DecisionTreeClassifier

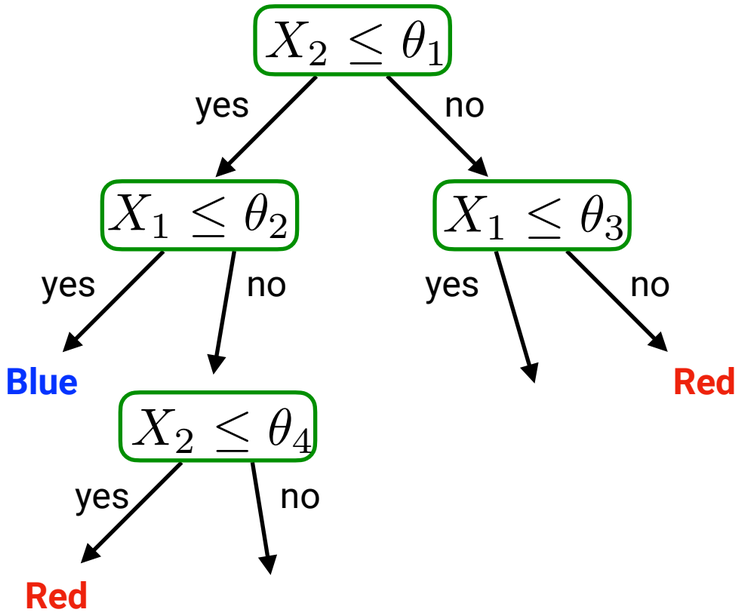

θ_1
θ_2
θ_3


$X_2 \leq \theta_1$
yes
no
$X_1 \leq \theta_2$
$X_1 \leq \theta_3$
yes
no
yes
no
Blue
Red

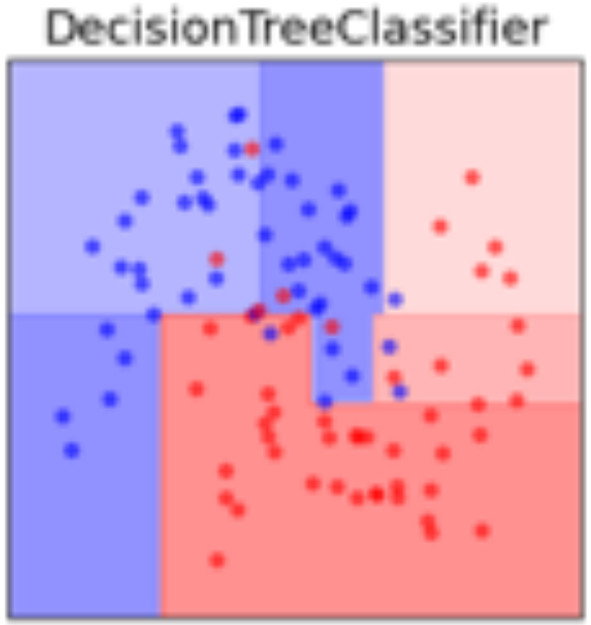
決定木の入力空間分割

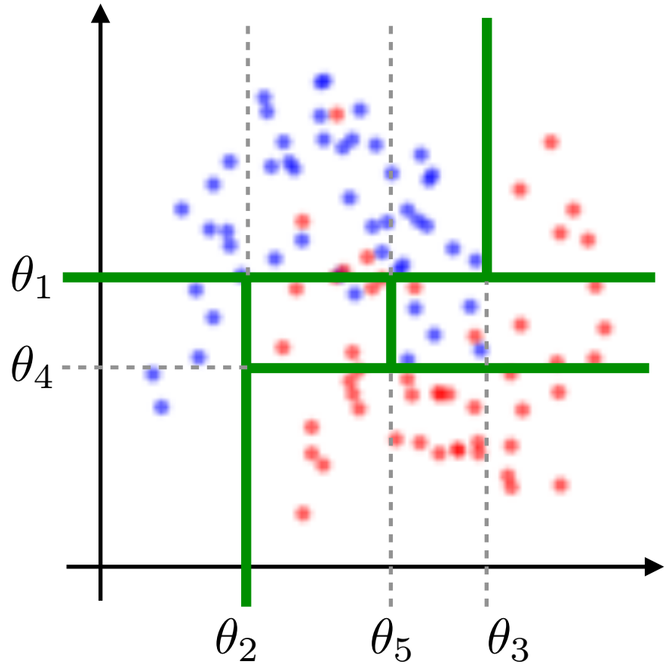

DecisionTreeClassifier

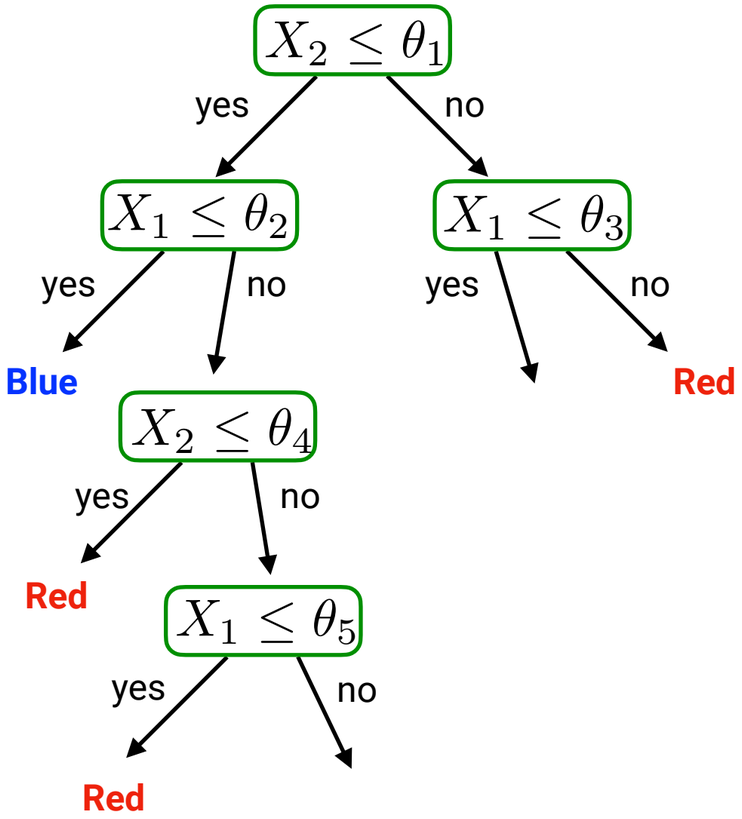

θ_1
θ_4
θ_2
θ_3


$X_2 \leq \theta_1$
yes
no
$X_1 \leq \theta_2$
$X_1 \leq \theta_3$
yes
no
yes
no
Blue
$X_2 \leq \theta_4$
Red
yes
no
Red

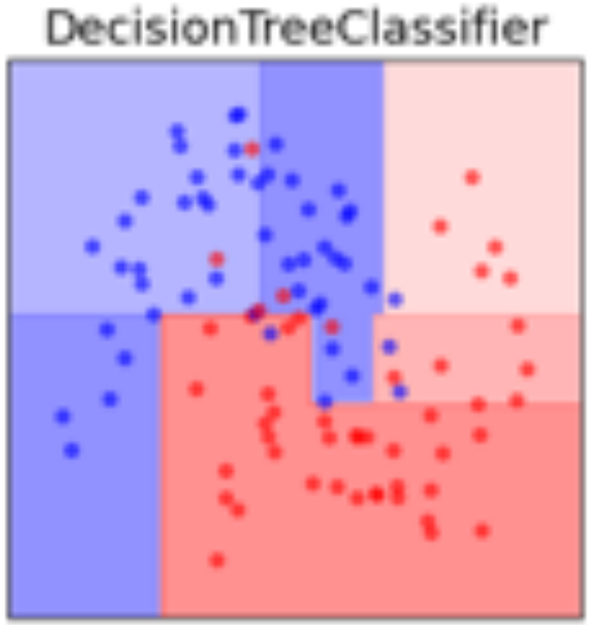
決定木の入力空間分割

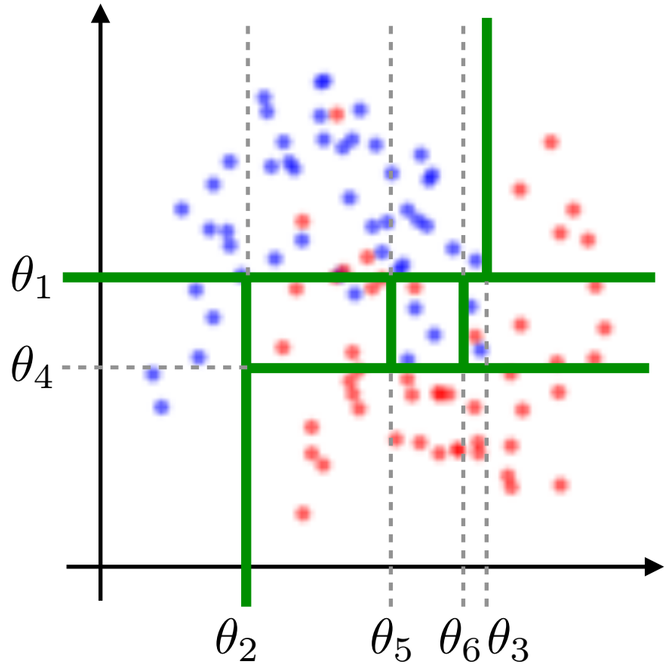

DecisionTreeClassifier

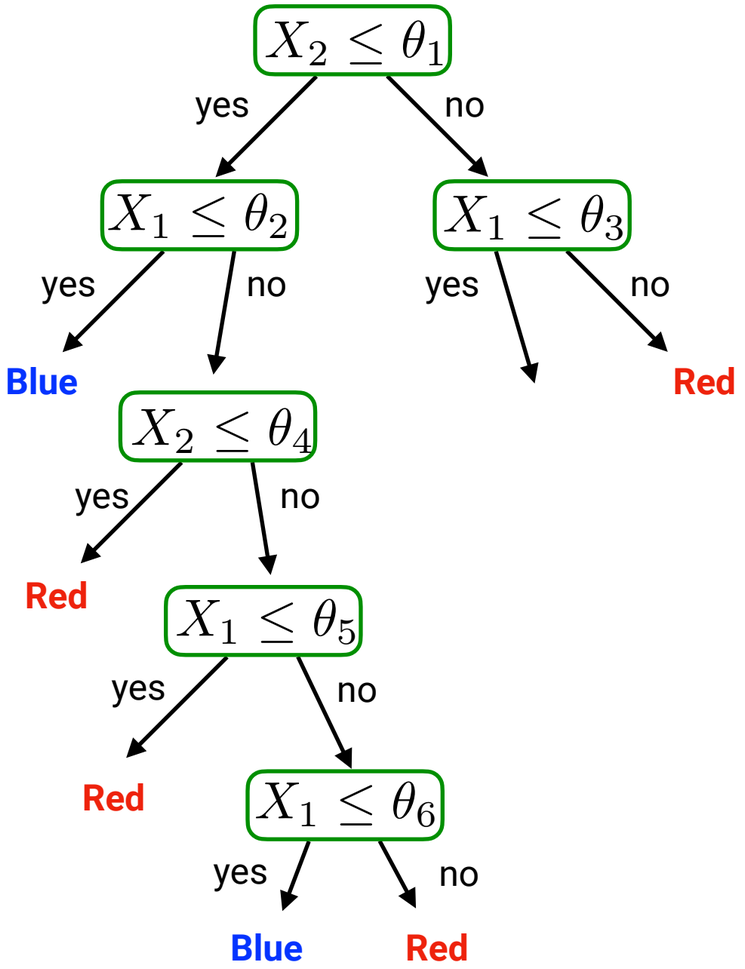

θ_1
θ_4
θ_2
θ_5
θ_3


$X_2 \leq \theta_1$
yes
no
$X_1 \leq \theta_2$
$X_1 \leq \theta_3$
yes
no
yes
no
Blue
Red
$X_2 \leq \theta_4$
yes
no
Red
$X_1 \leq \theta_5$
yes
no
Red

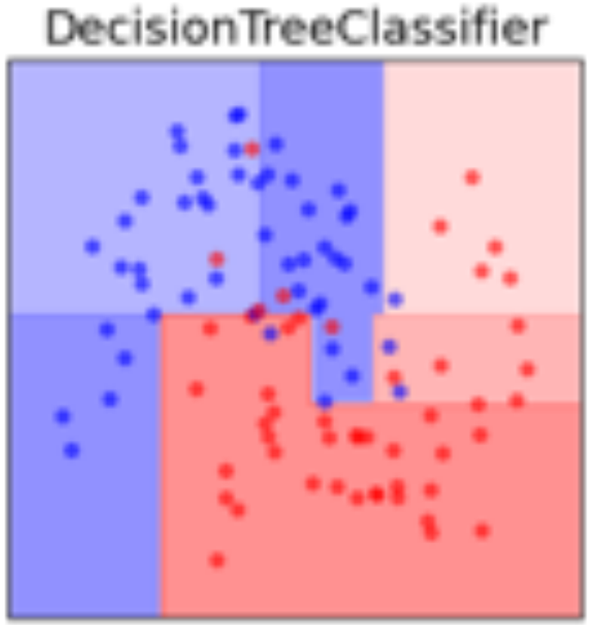
決定木の入力空間分割

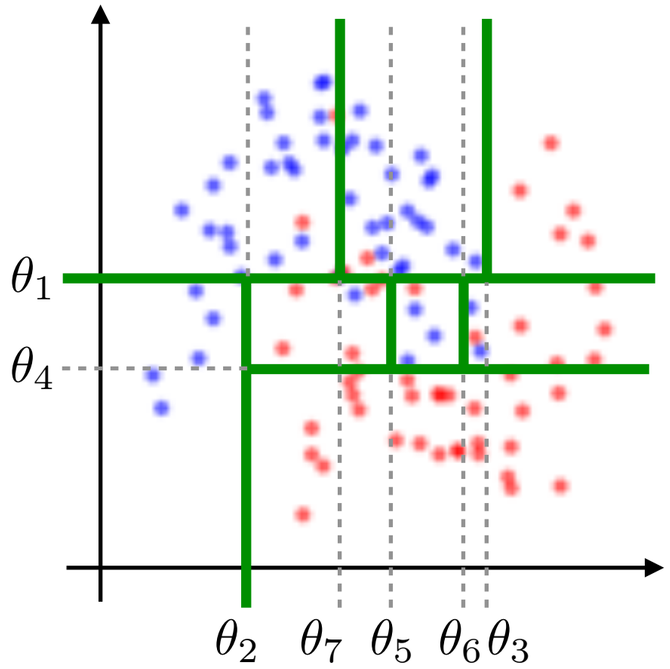

DecisionTreeClassifier

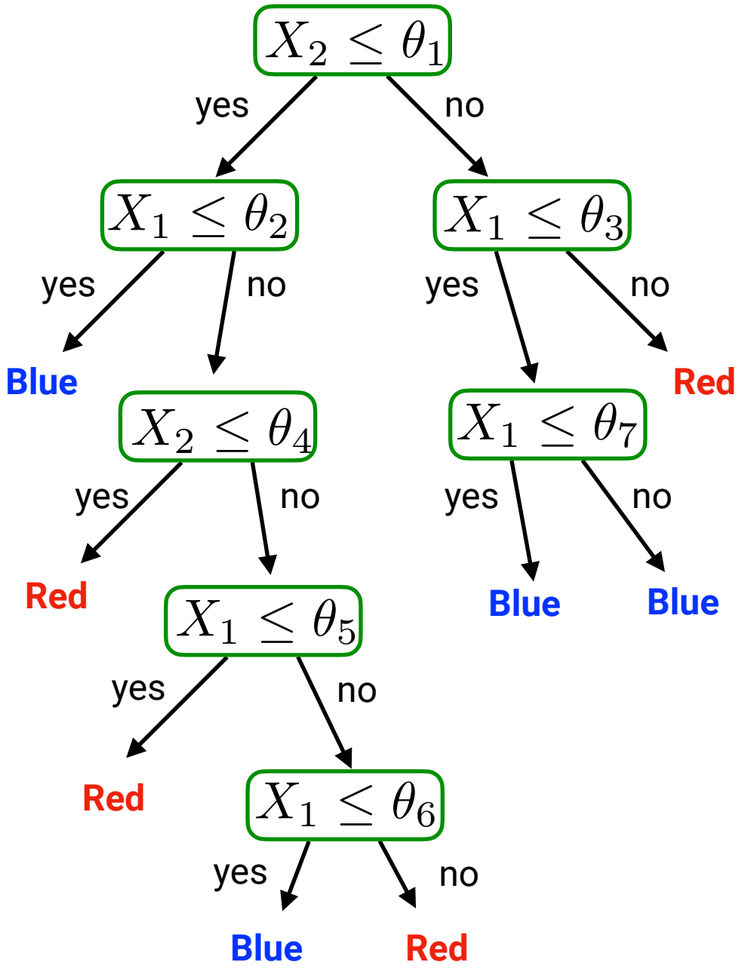

θ_1
θ_4
θ_2
θ_5
θ_6
θ_3


$X_2 \leq \theta_1$
yes
no
$X_1 \leq \theta_2$
$X_1 \leq \theta_3$
yes
no
yes
no
Blue
Red
$X_2 \leq \theta_4$
yes
no
Red
$X_1 \leq \theta_5$
yes
no
Red
$X_1 \leq \theta_6$
yes
no
Blue
Red

決定木の入力空間分割


DecisionTreeClassifier


θ_1
θ_4
θ_2
θ_7
θ_5
θ_6
θ_3

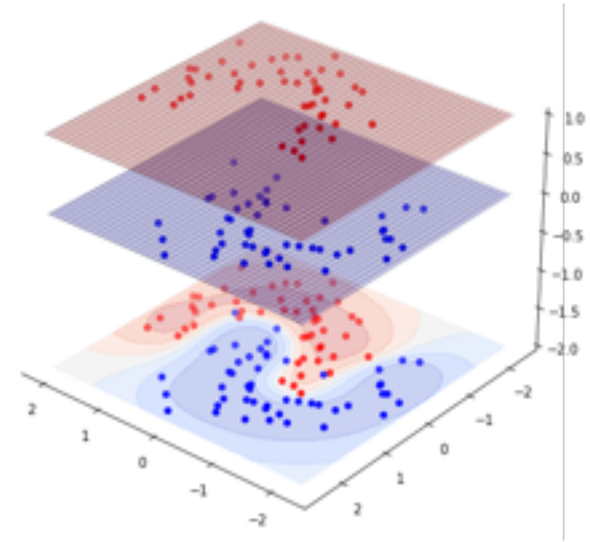

$X_2 \leq \theta_1$
yes
no
$X_1 \leq \theta_2$
$X_1 \leq \theta_3$
yes
no
yes
no
Blue
Red
$X_2 \leq \theta_4$
$X_1 \leq \theta_7$
yes
no
yes
no
Red
Blue
Blue
$X_1 \leq \theta_5$
yes
no
Red
$X_1 \leq \theta_6$
yes
no
Blue
Red

決定木の入力空間分割

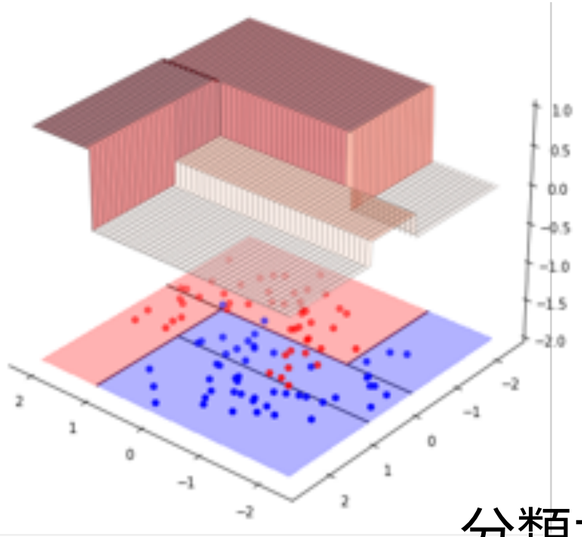
決定木は領域分割上の**多次元ヒストグラム (区分的定数予測)**

→ 統計的には**ほとんど経験分布**に近い良い性質を持つ (**悪いことが非常に起きづらい**)

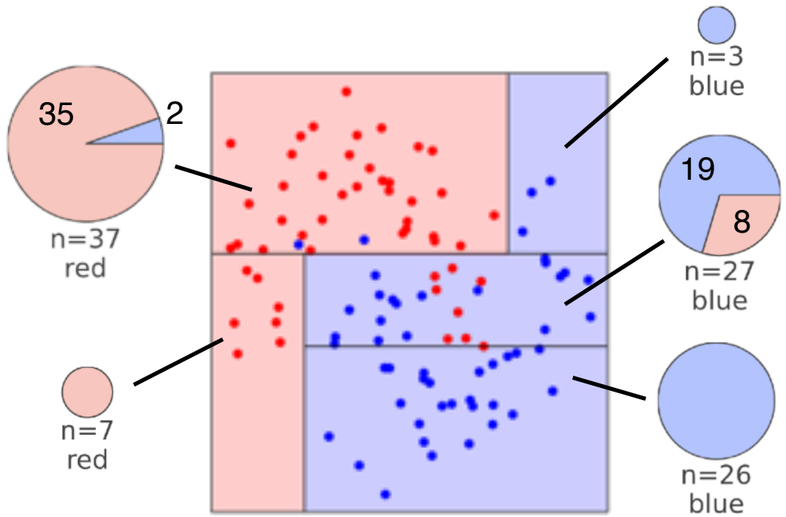
→ 「Greedyで妥当性**もおぼつかない**テキトーな領域分割」に「**超コンサバ**な予測」を抱合せ



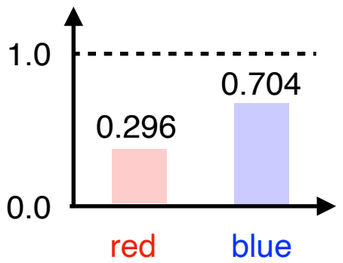
Piecewise "constant"



分類木



`predict_proba`(x)



8/27 = 0.296

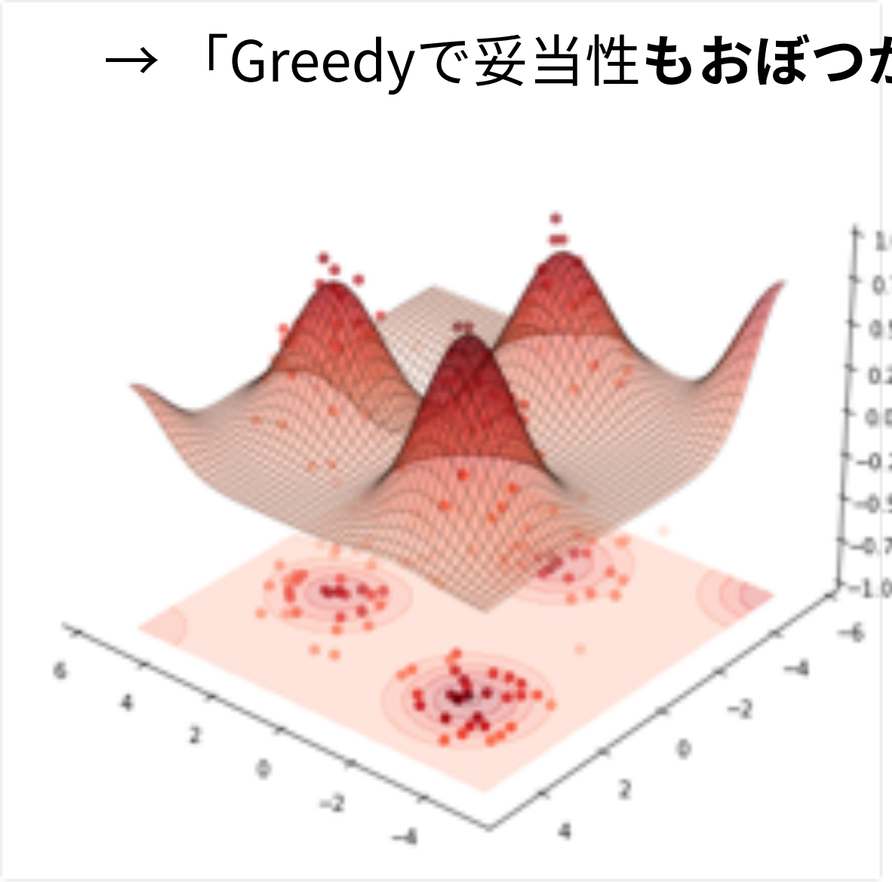
19/27 = 0.704

決定木の入力空間分割

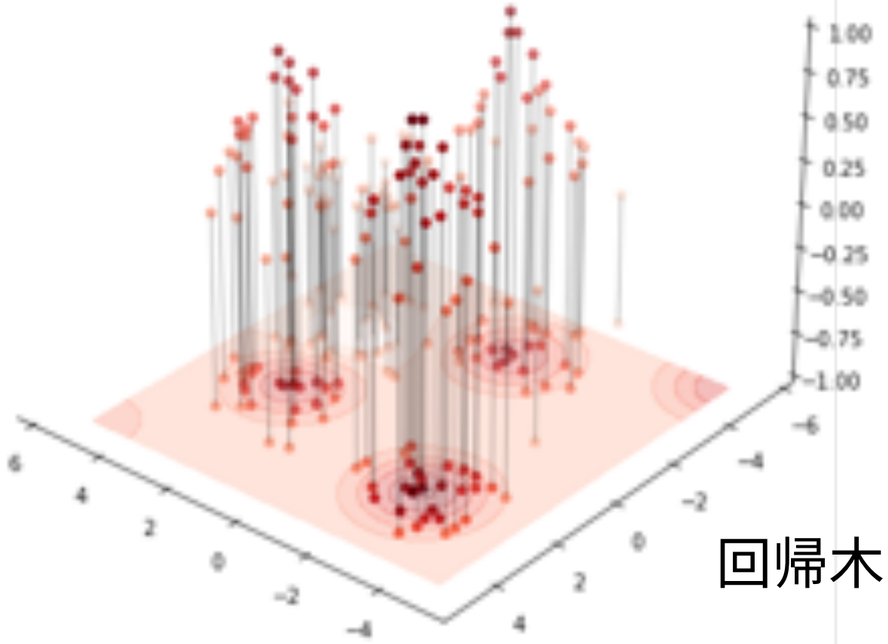
決定木は領域分割上の**多次元ヒストグラム (区分的定数予測)**

→ 統計的には**ほとんど経験分布**に近い良い性質を持つ (**悪いことが非常に起きづらい**)

→ 「Greedyで妥当性**もおぼつかない**テキトーな領域分割」に「**超コンサバ**な予測」を抱合せ



Piecewise "constant"



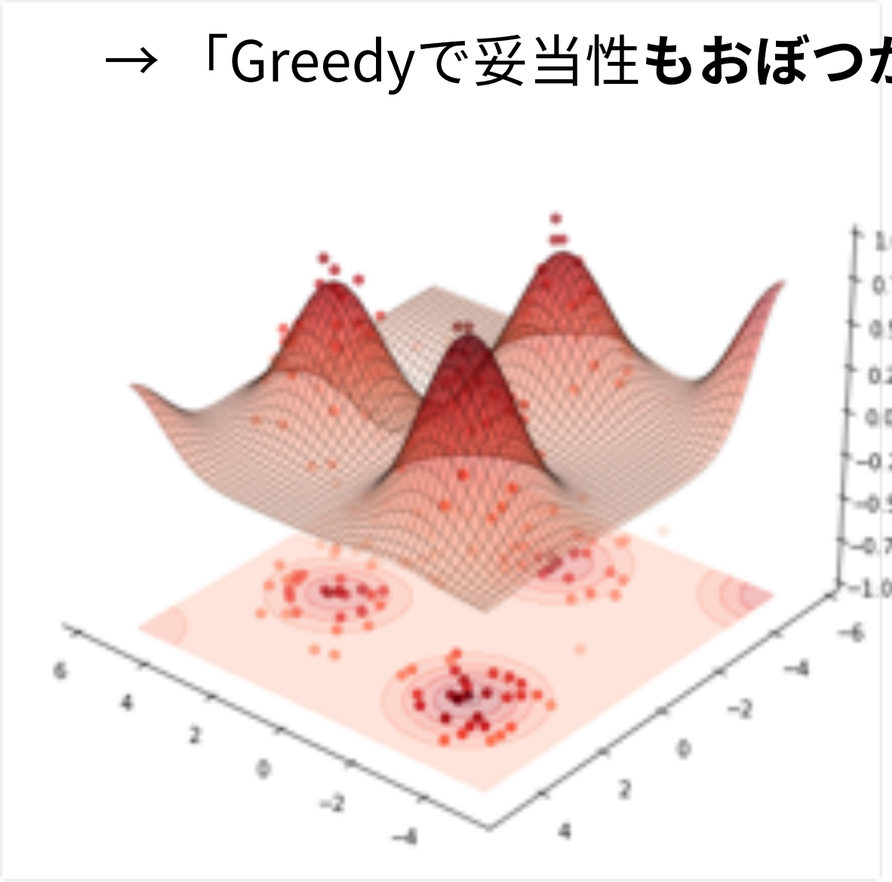
回帰木

決定木の入力空間分割

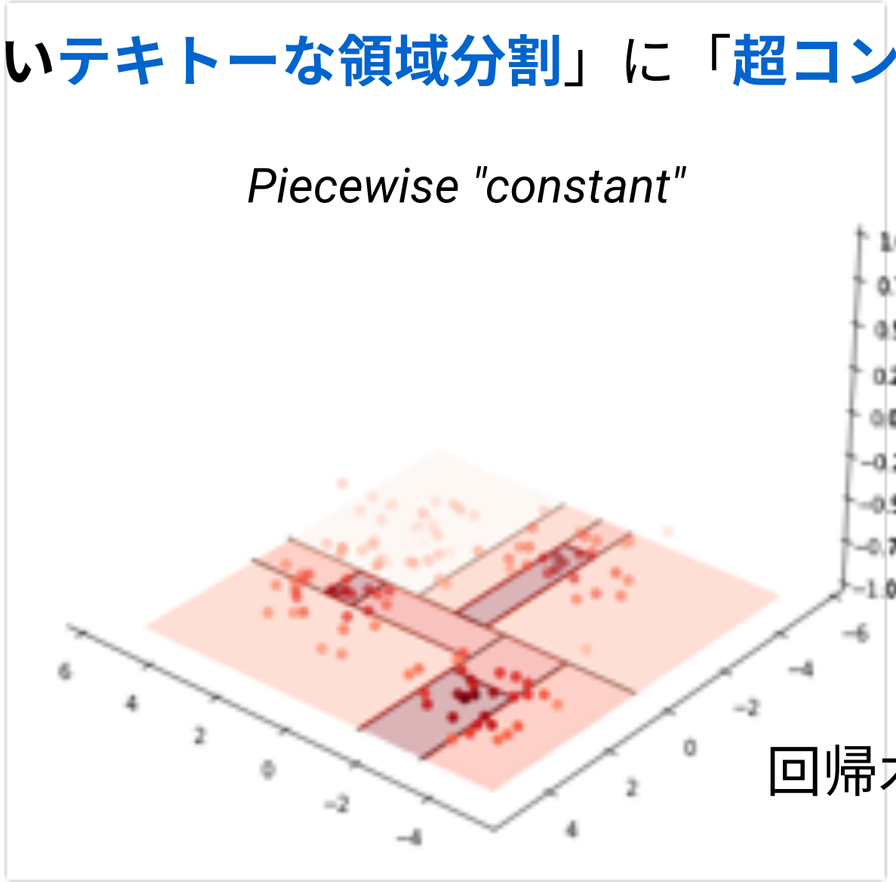
決定木は領域分割上の**多次元ヒストグラム (区分的定数予測)**

→ 統計的には**ほとんど経験分布**に近い良い性質を持つ (**悪いことが非常に起きづらい**)

→ 「Greedyで妥当性**もおぼつかない**テキトーな領域分割」に「**超コンサバ**な予測」を抱合せ



Piecewise "constant"



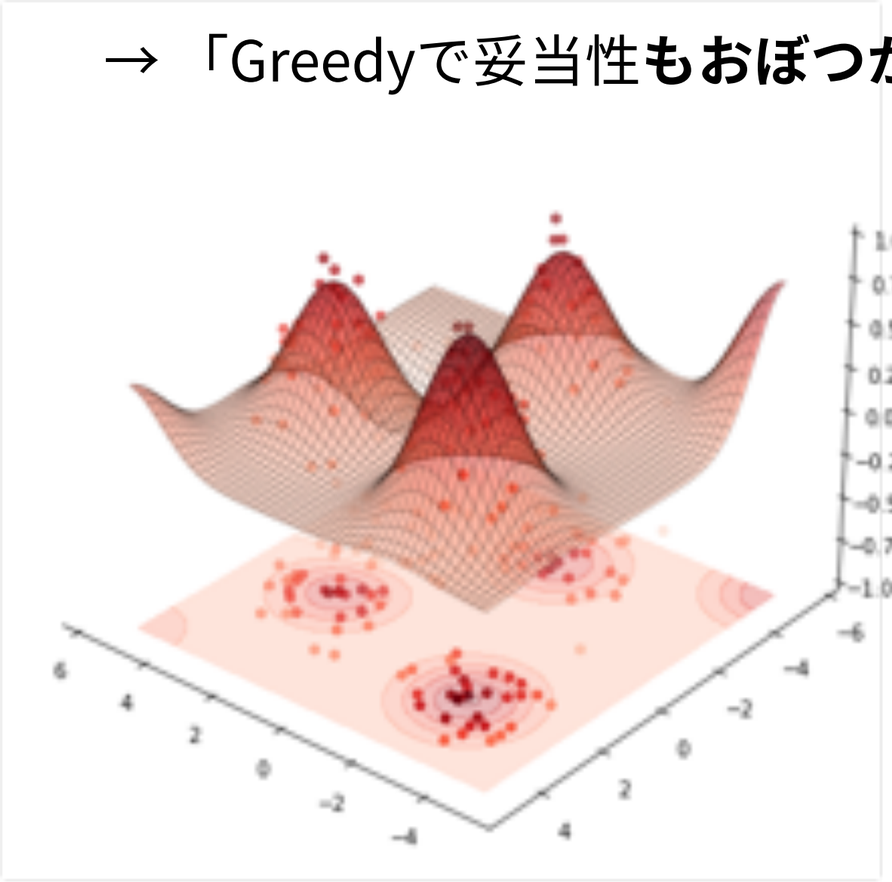
回帰木

決定木の入力空間分割

決定木は領域分割上の**多次元ヒストグラム (区分的定数予測)**

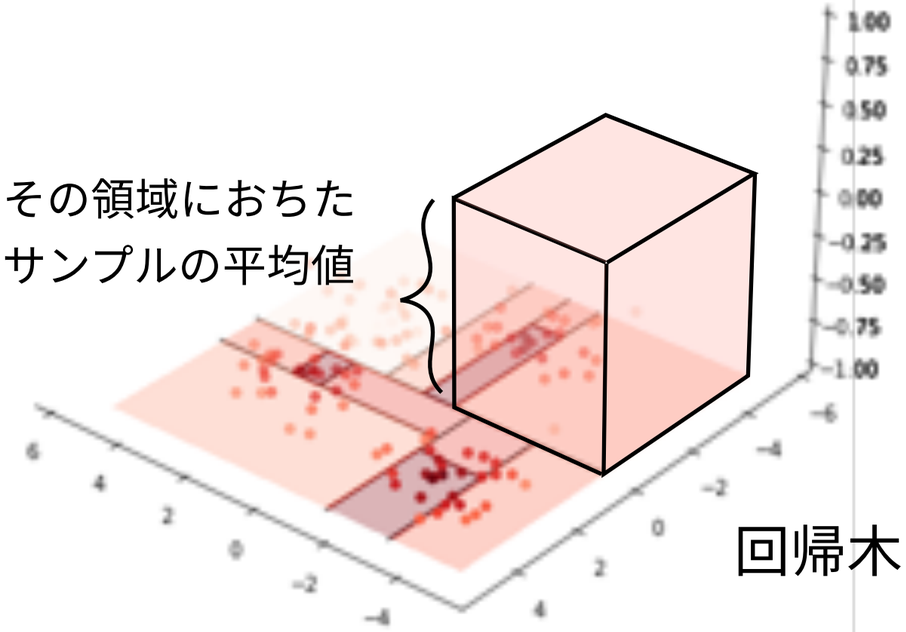
→ 統計的には**ほとんど経験分布**に近い良い性質を持つ (**悪いことが非常に起きづらい**)

→ 「Greedyで妥当性**もおぼつかない**テキトーな領域分割」に「**超コンサバ**な予測」を抱合せ



Piecewise "constant"

その領域におちた
サンプルの平均値



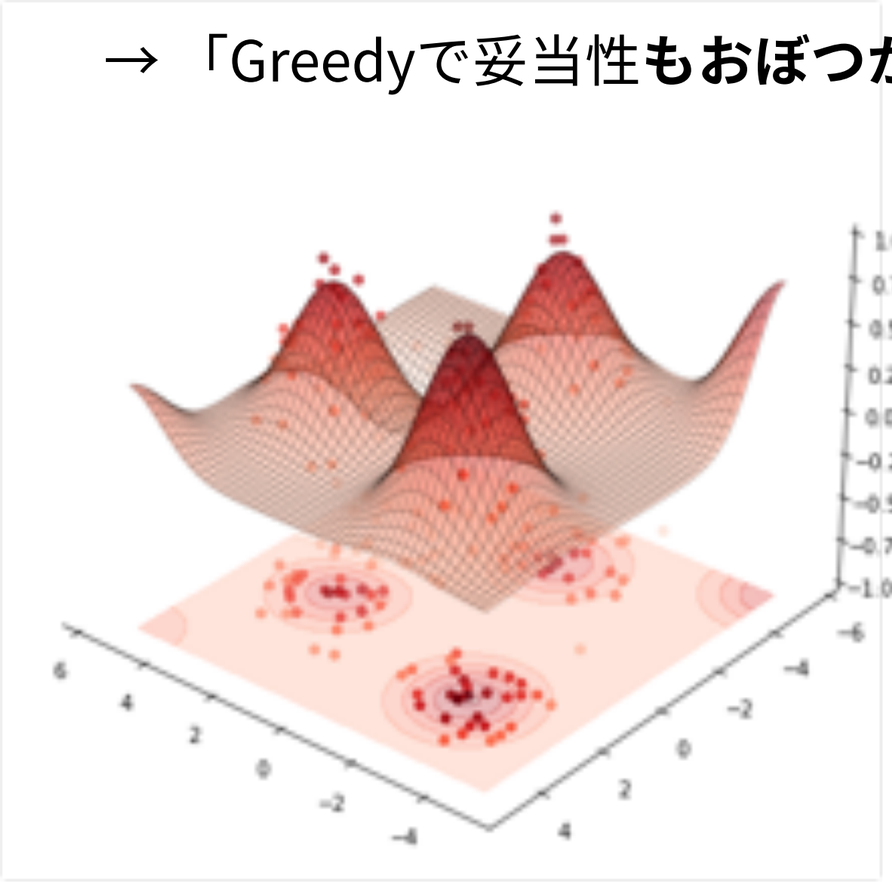
回帰木

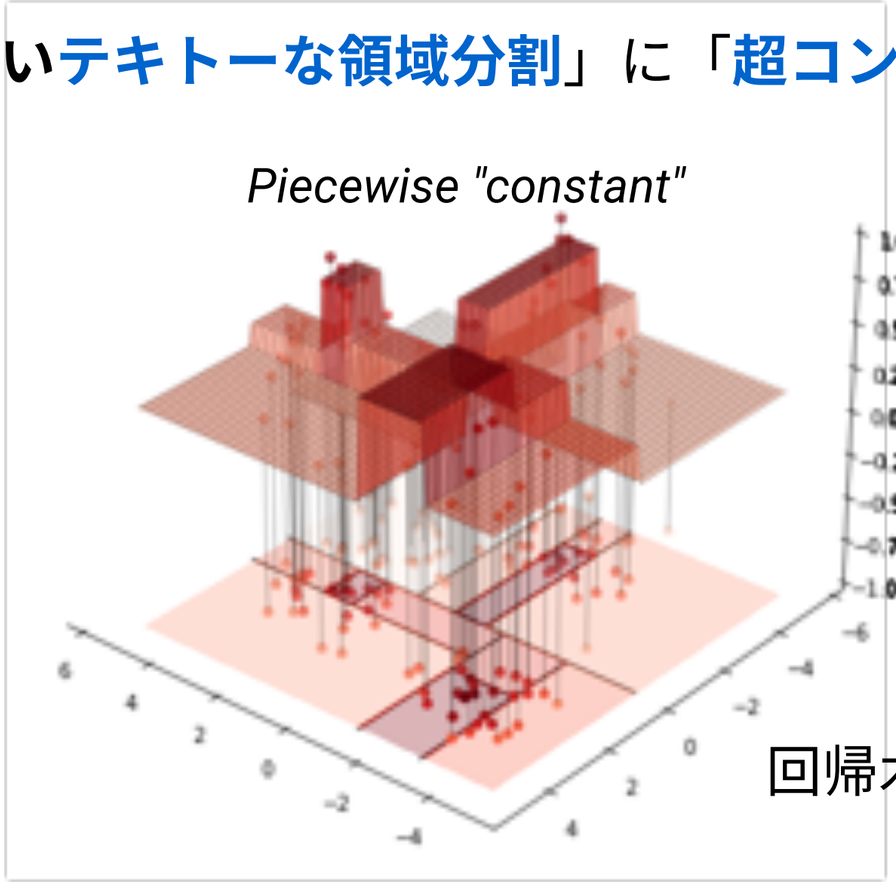
決定木の入力空間分割

決定木は領域分割上の**多次元ヒストグラム (区分的定数予測)**

→ 統計的には**ほとんど経験分布**に近い良い性質を持つ (**悪いことが非常に起きづらい**)

→ 「Greedyで妥当性**もおぼつかない**テキトーな領域分割」に「**超コンサバ**な予測」を抱合せ



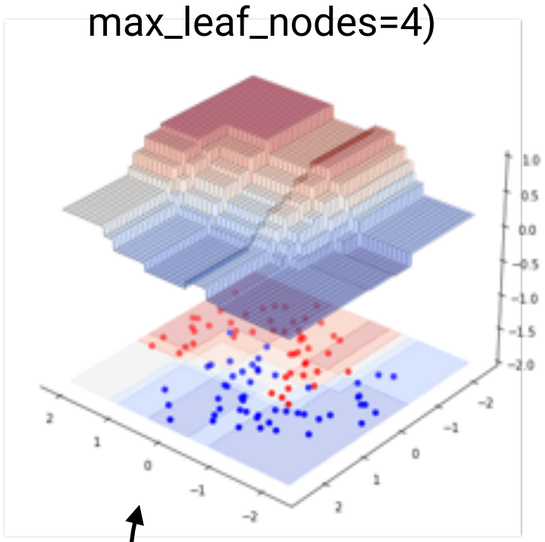


回帰木

決定木の入力空間分割

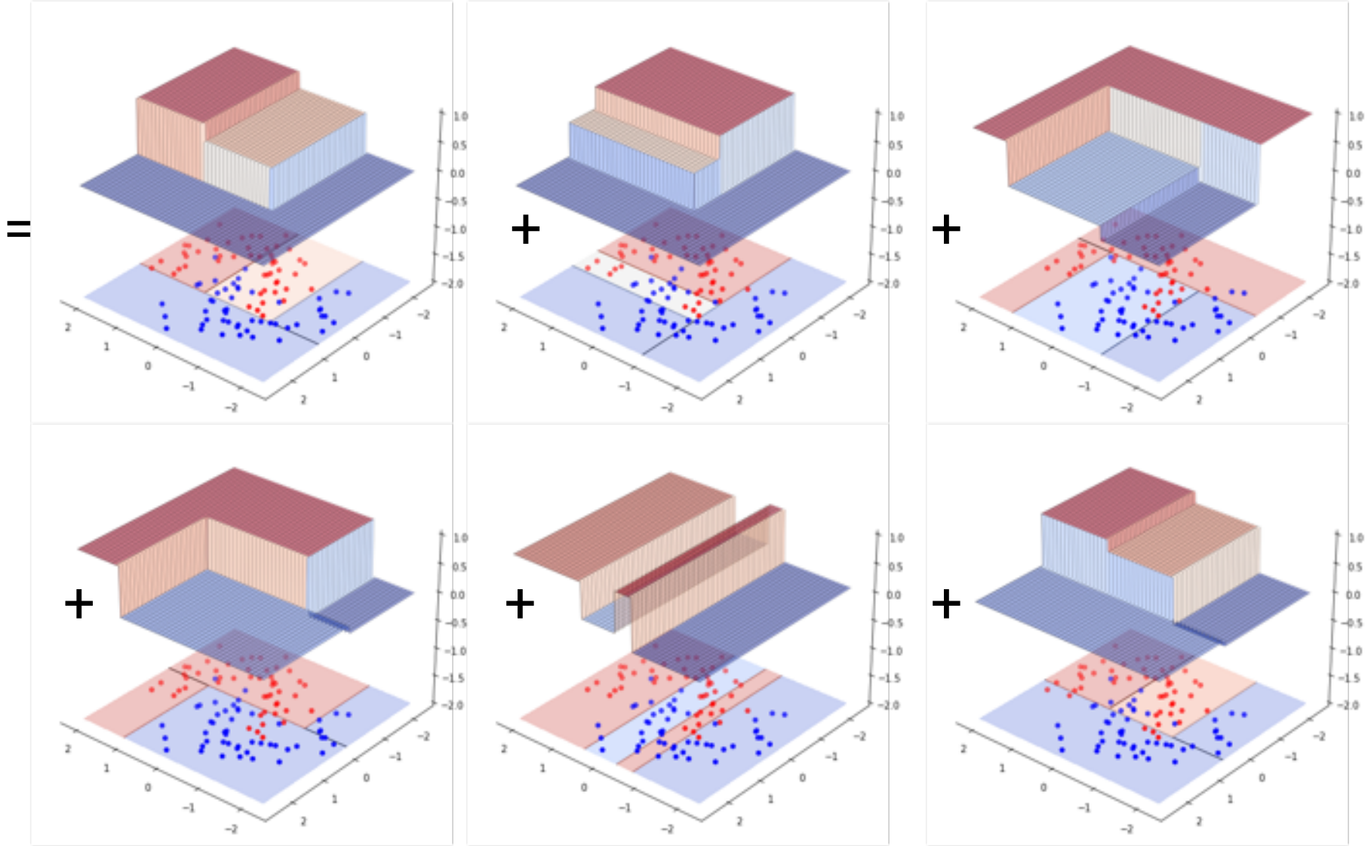
決定木アンサンブルはその(重み付き)和

RandomForestClassifier
(n_estimators=6,
max_leaf_nodes=4)



アンサンブル後の
領域数は組合せで
4より大幅に増える！

6 × DecisionTreeClassifier(max_leaf_nodes=4)



加法モデル(和)
による領域の細分

Piecewise "constant"

Splineと異なり
境界での**連続性は**
全く担保されない

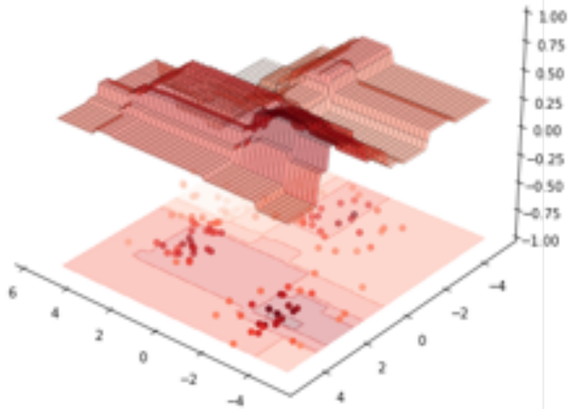
↓

ただし**アンサンブル**
による平滑化効果で
めちゃめちゃ不連続
にはなりづらい！

決定木の入力空間分割

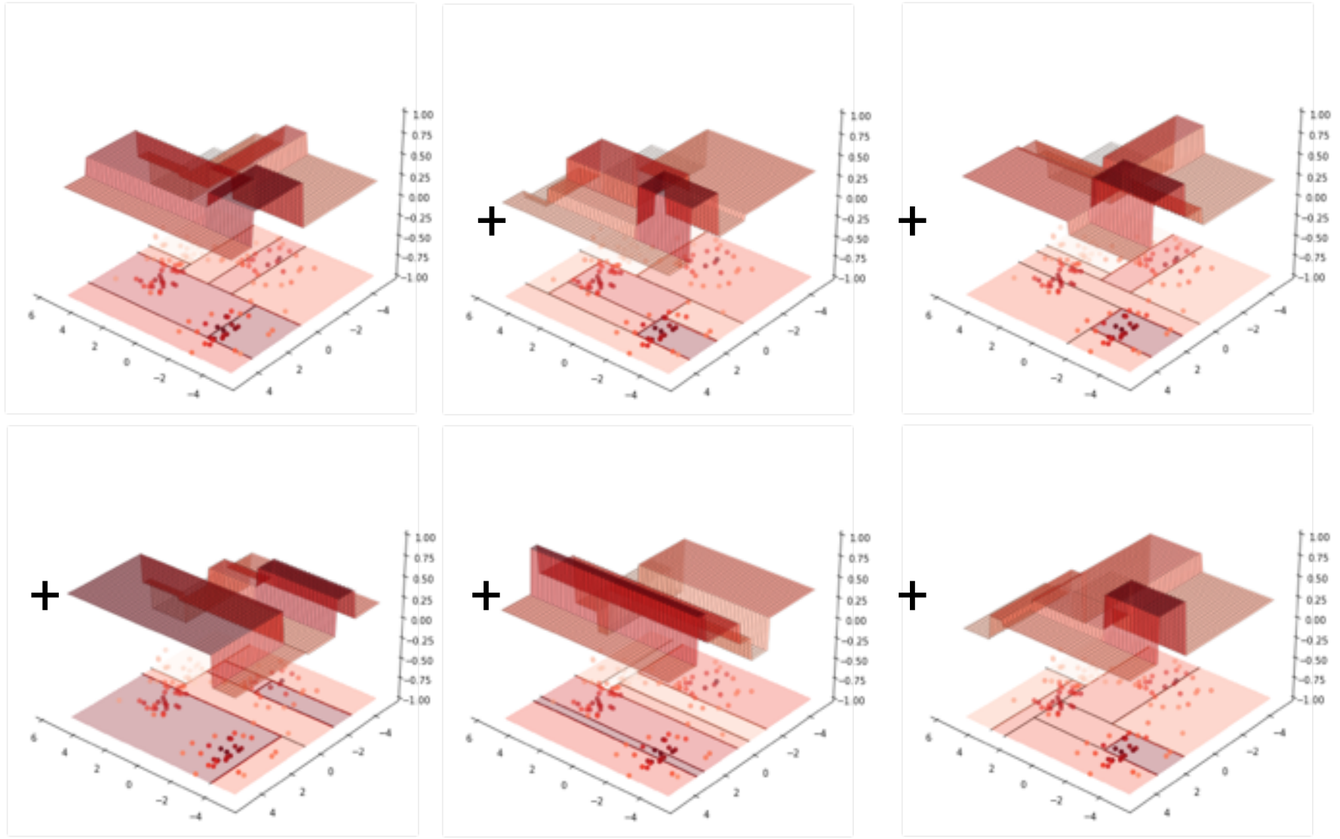
決定木アンサンブルはその(重み付き)和

RandomForestRegressor
(n_estimators=6,
max_leaf_nodes=8)



=

6 × DecisionTreeRegressor(max_leaf_nodes=8)



加法モデル(和)
による領域の細分

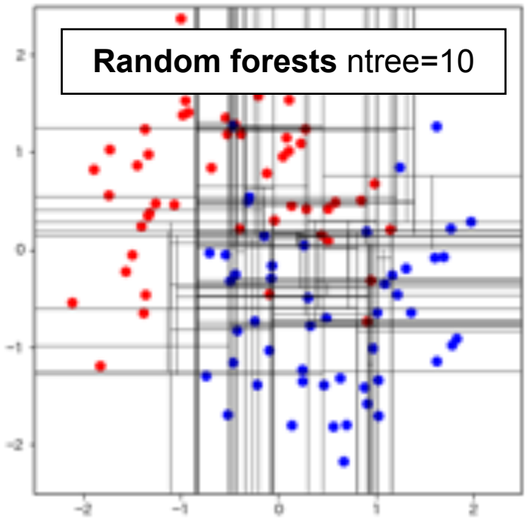
Piecewise "constant"

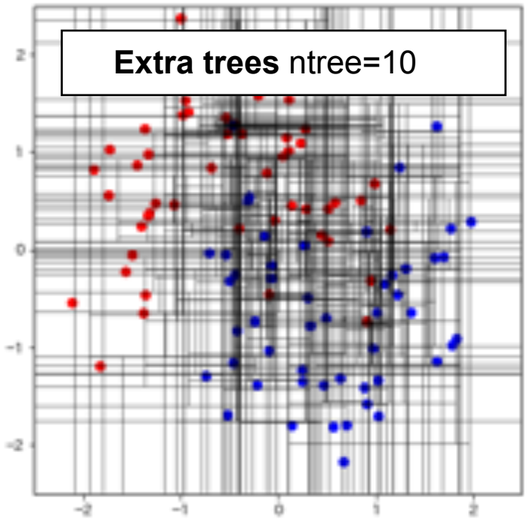
Splineと異なり
境界での**連続性は**
全く担保されない

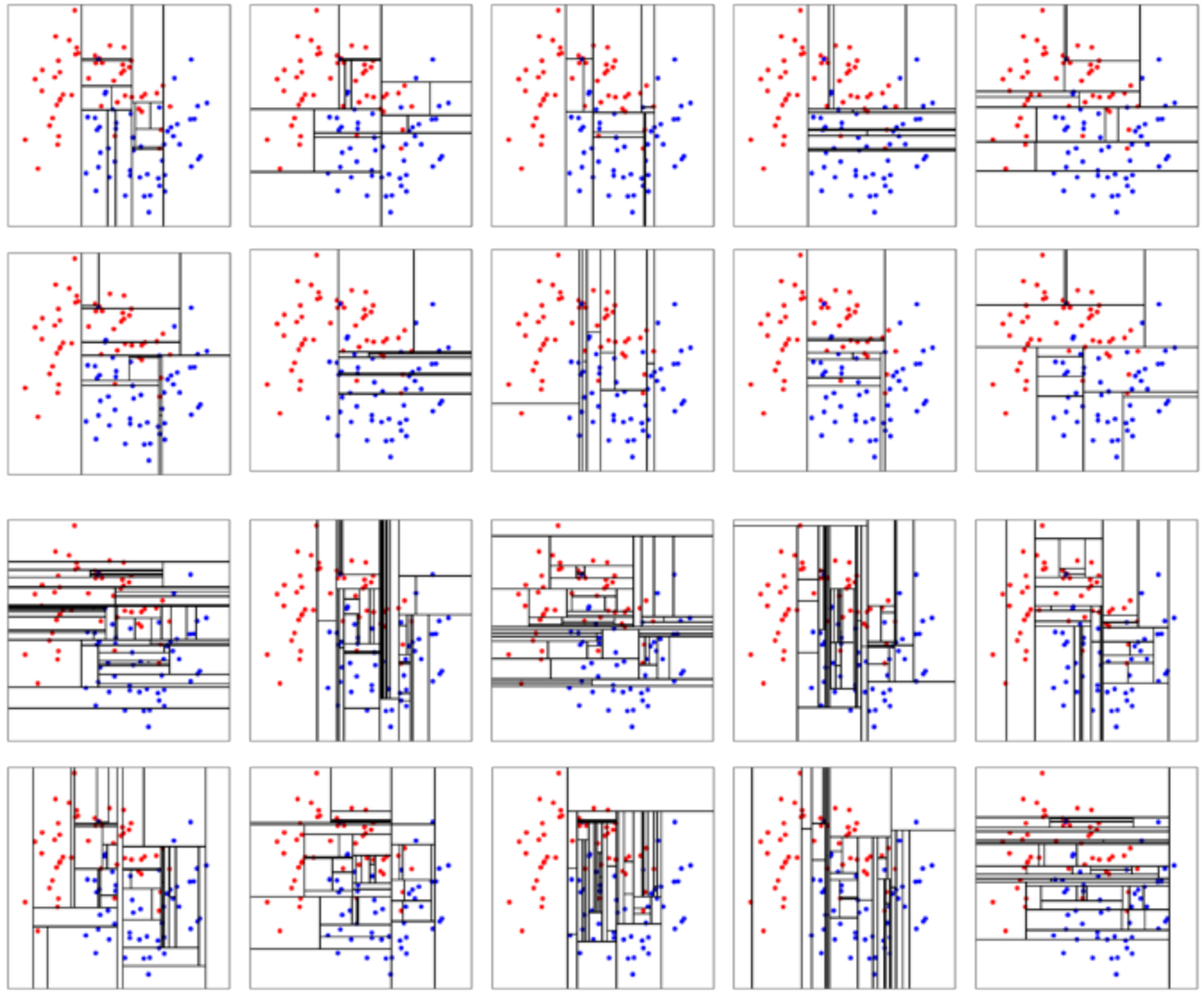
↓

ただし**アンサンブル**
による平滑化効果で
めちゃめちゃ不連続
にはなりづらい！

決定木の入力空間分割







加法モデル(和)
による領域の細分

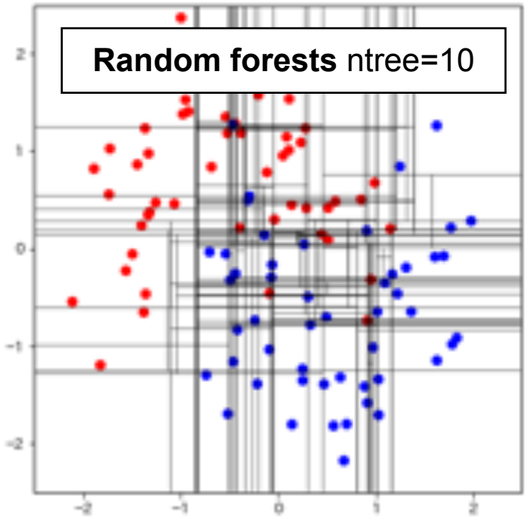
Piecewise "constant"

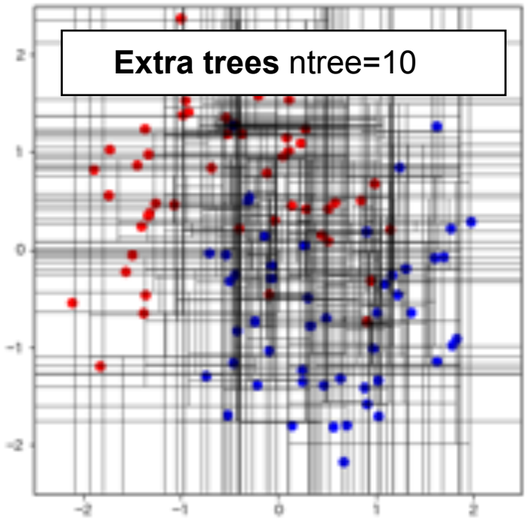
Splineと異なり
境界での**連続性は**
全く担保されない

↓

ただし**アンサンブル**
による平滑化効果で
めちゃめちゃ不連続
にはなりづらい！

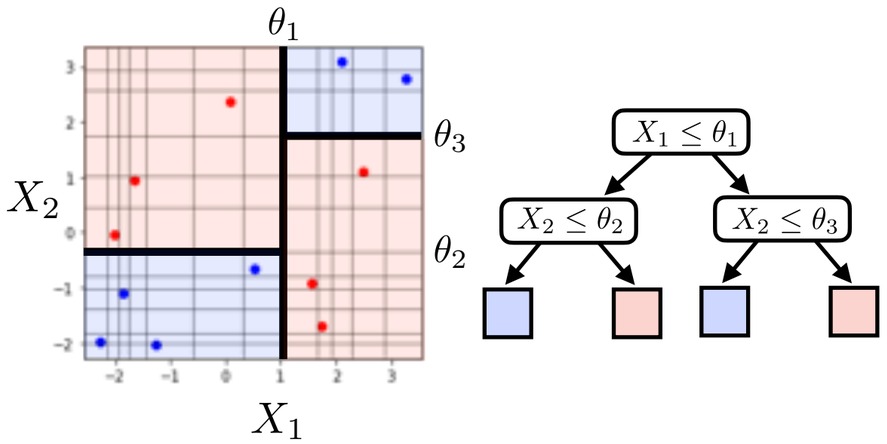
参考：Oblivious Treesの分割



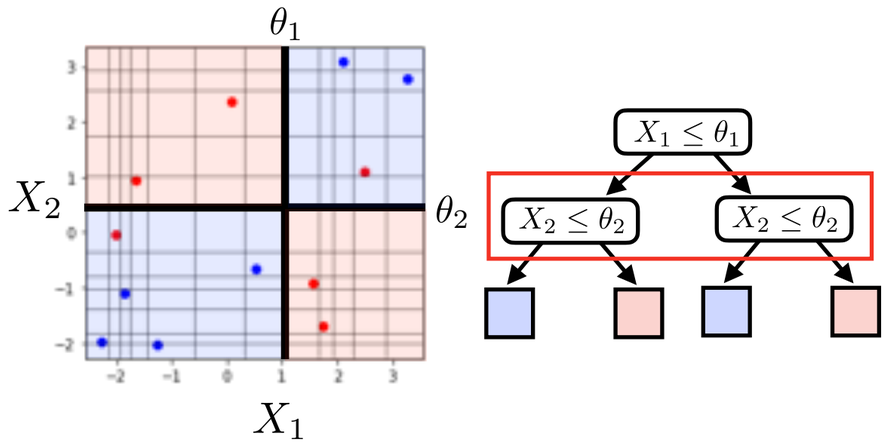


決定木をハード化するとかSoft Treesなど計算がネックになる場合、
同じレベルのsplitterの共有(Oblivious Trees)が有効 (NODE, CatBoost, etc)

通常の再帰的二分割



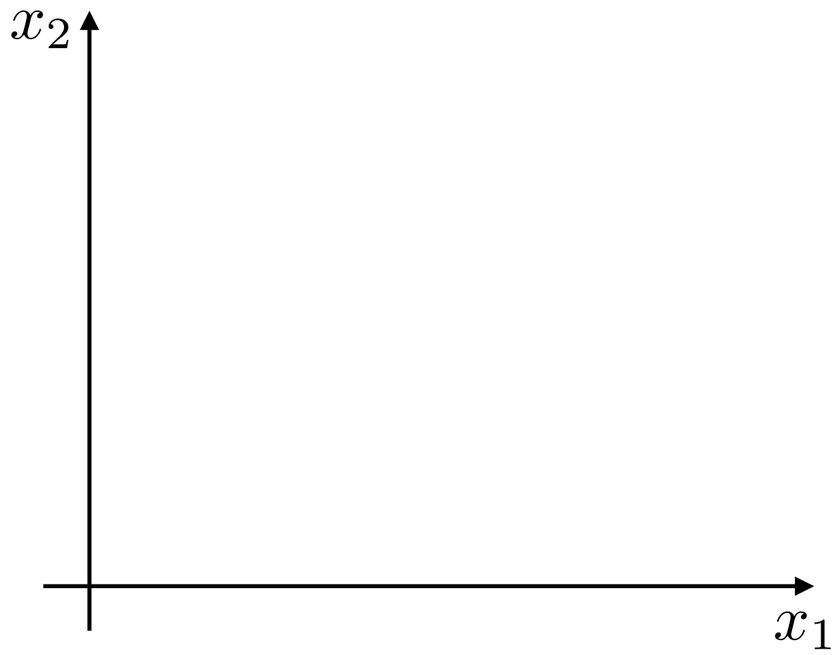
Oblivious Trees = メッシュ的構造になる



Obliviousにしようがしまいが、もし木数が非常に大きくできるならどうせほぼ**空間メッシュ的構造**になるので近似能力上ほぼ差はなく、計算効率化の恩恵だけをうまく享受できるのかも！？

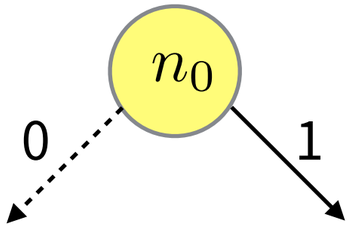
決定木の入力空間分割

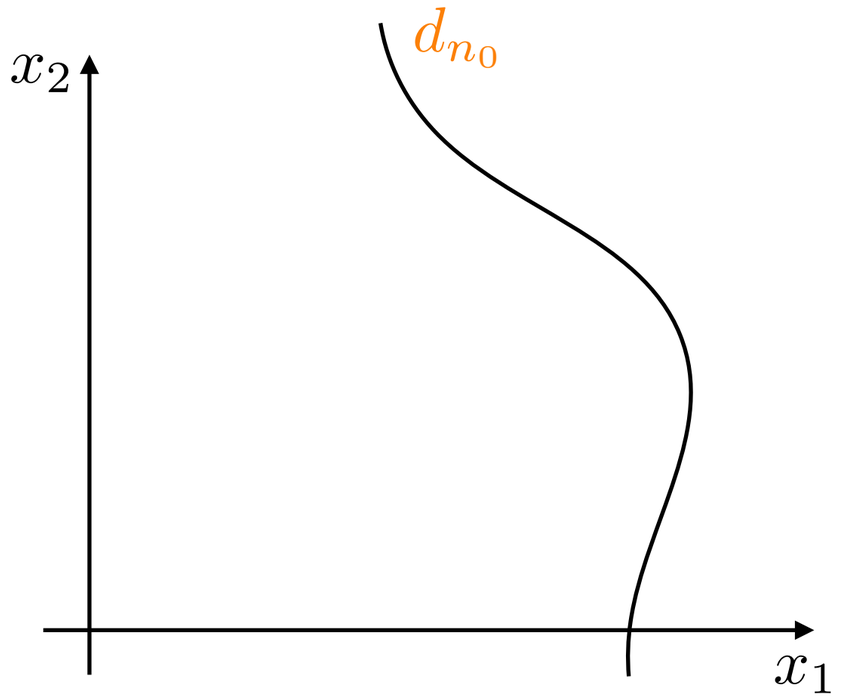
決定木の一般化



決定木の入力空間分割

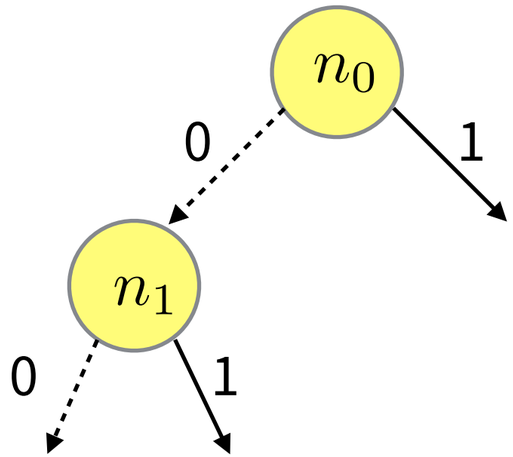
決定木の一般化

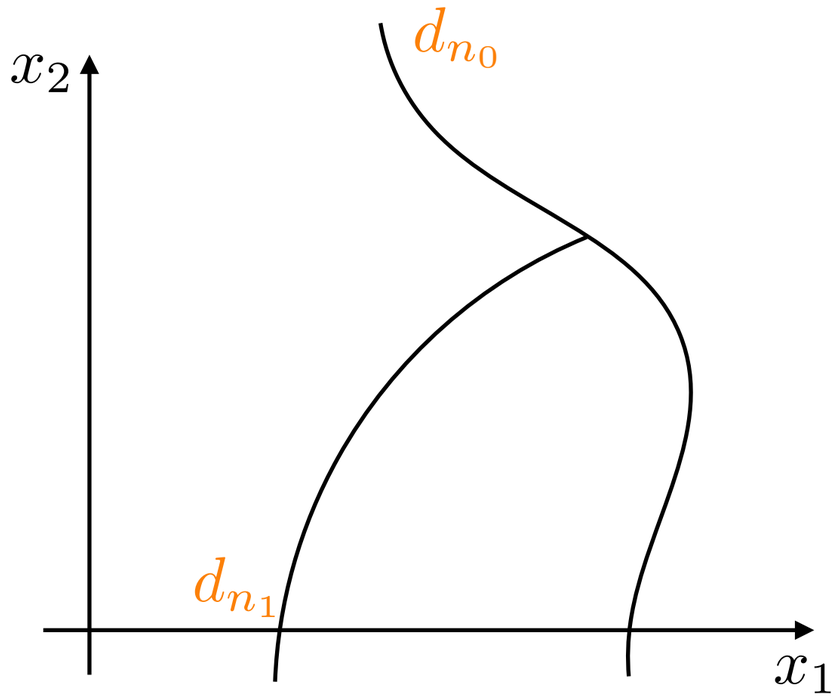




決定木の入力空間分割

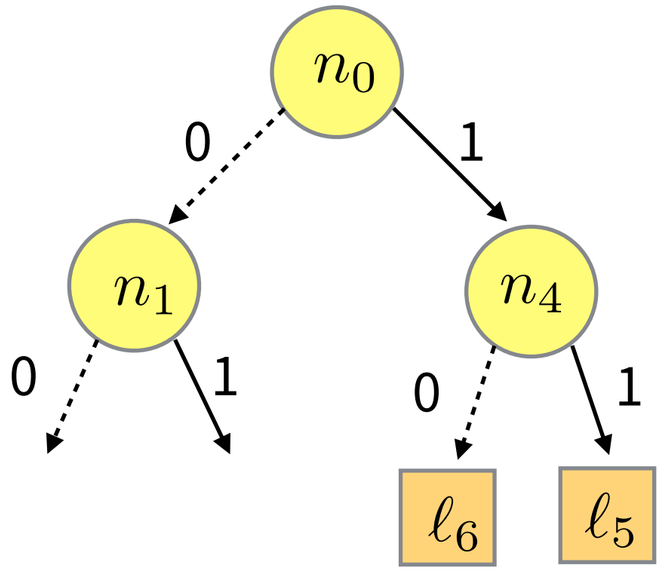
決定木の一般化

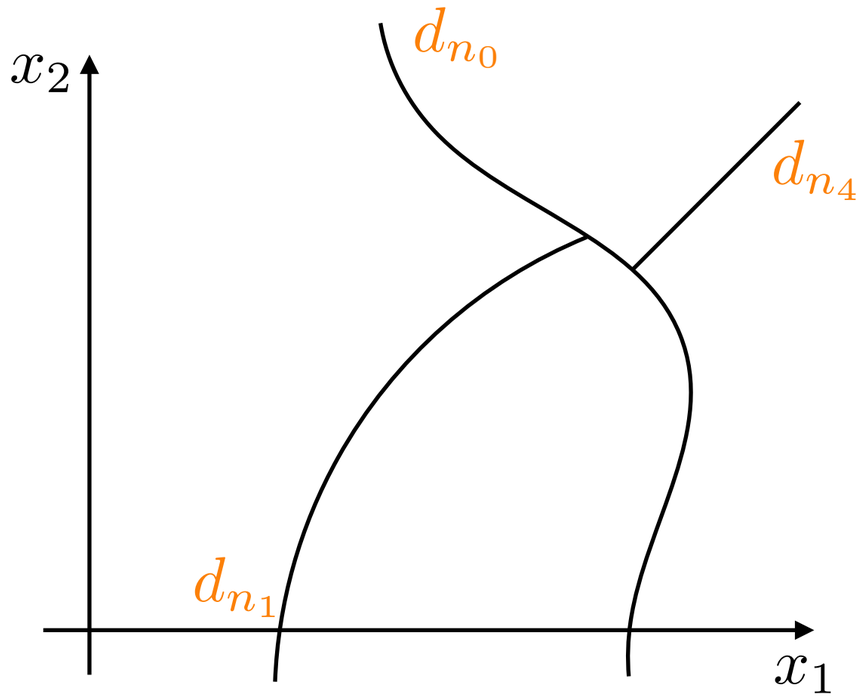




決定木の入力空間分割

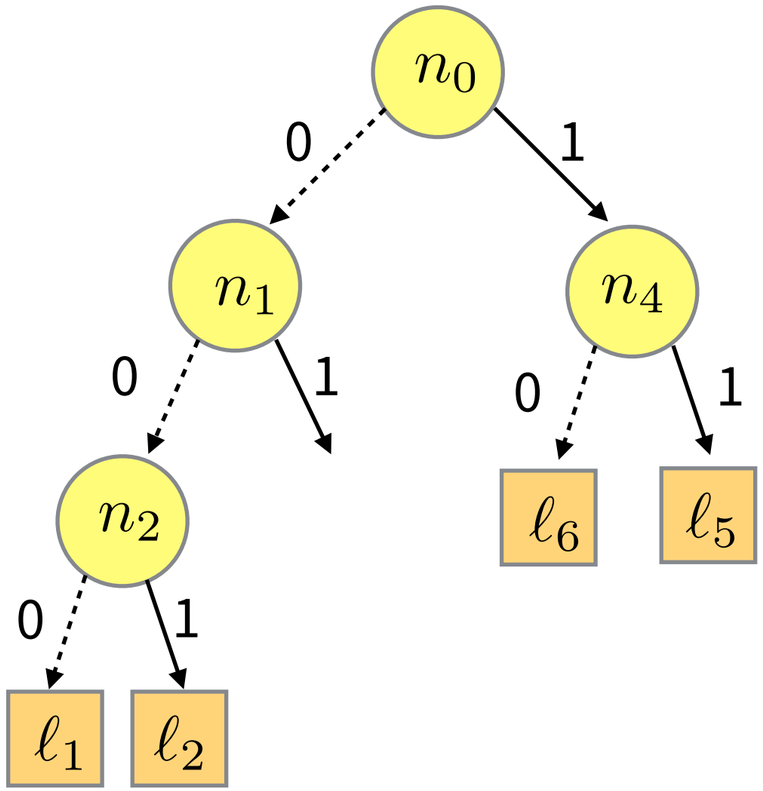
決定木の一般化

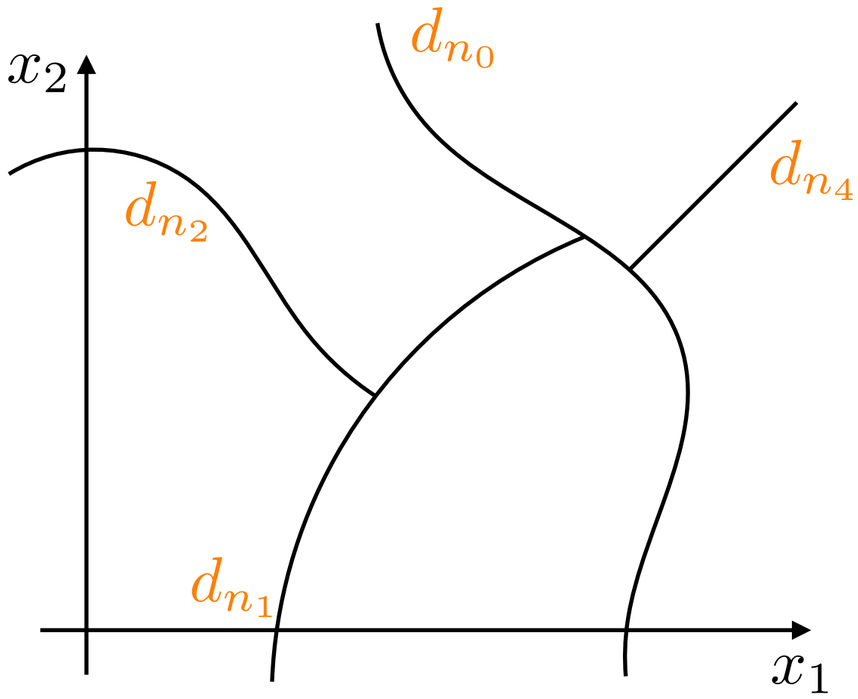




決定木の入力空間分割

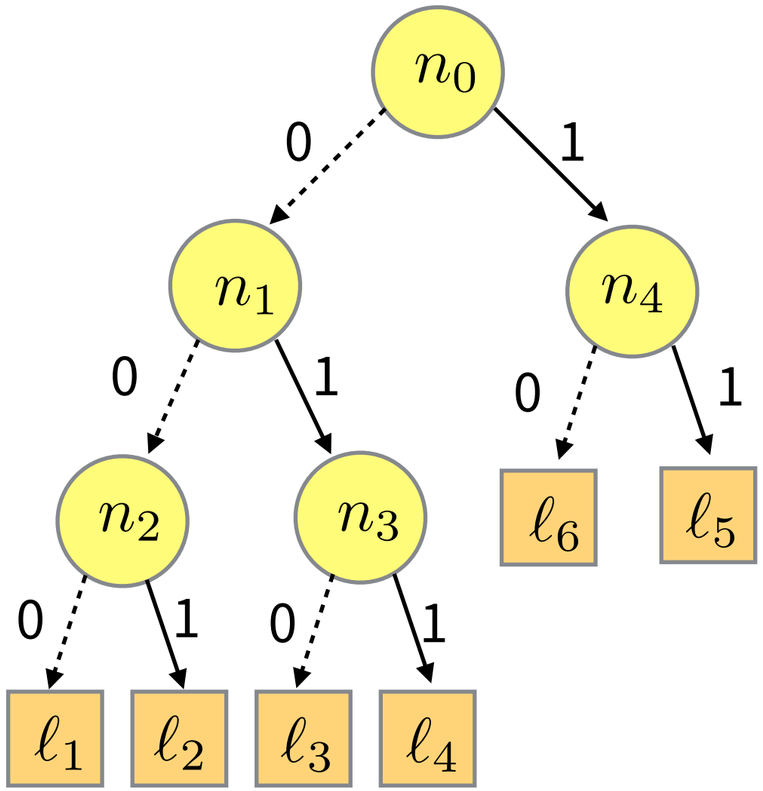
決定木の一般化

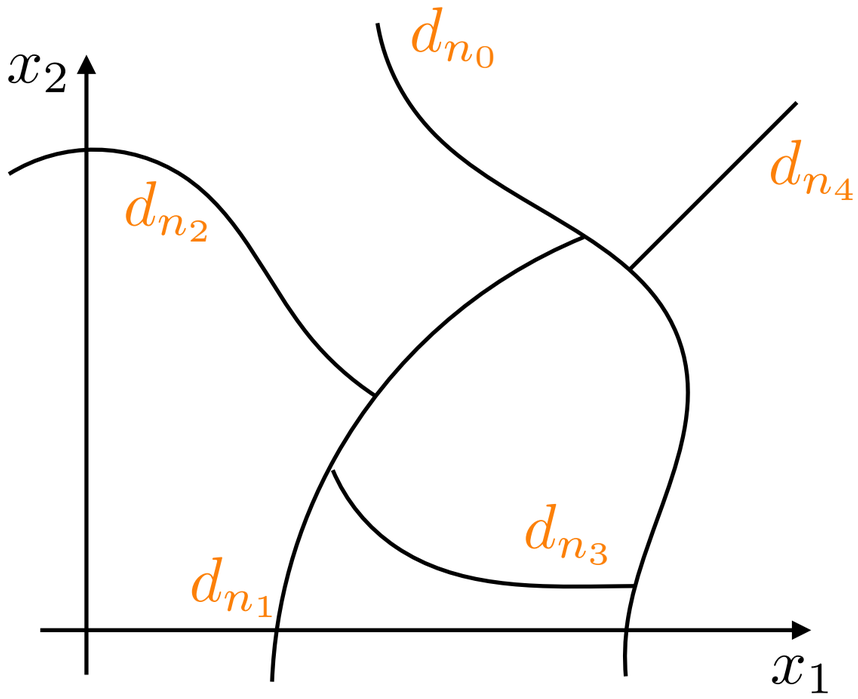




決定木の入力空間分割

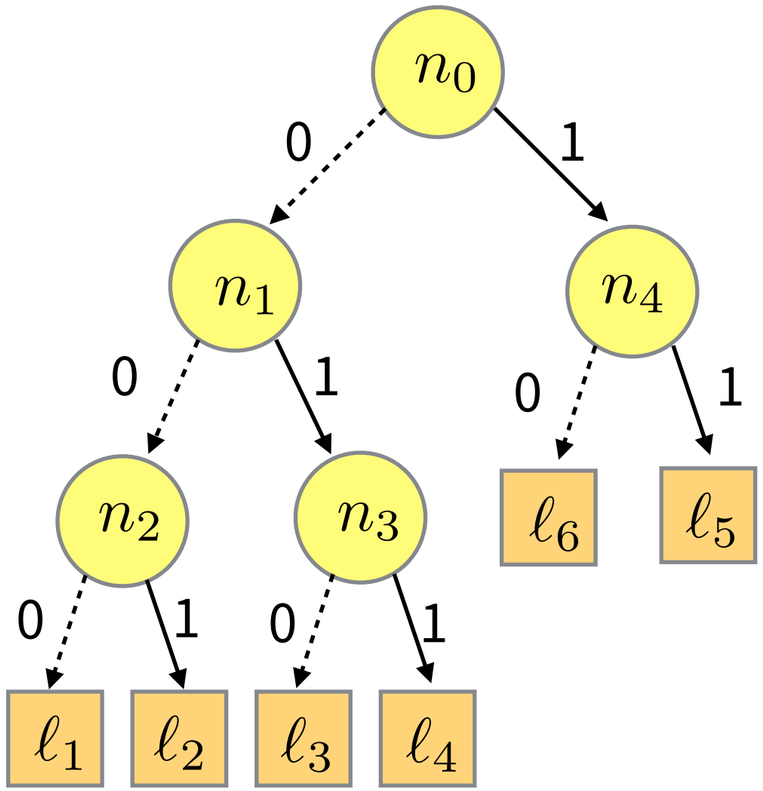
決定木の一般化

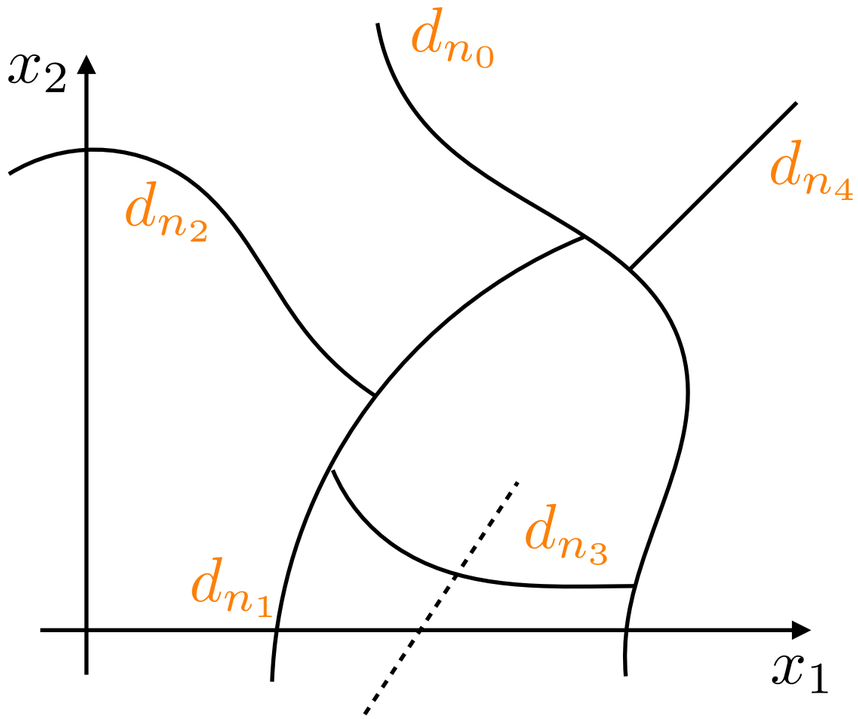




決定木の入力空間分割

決定木の一般化



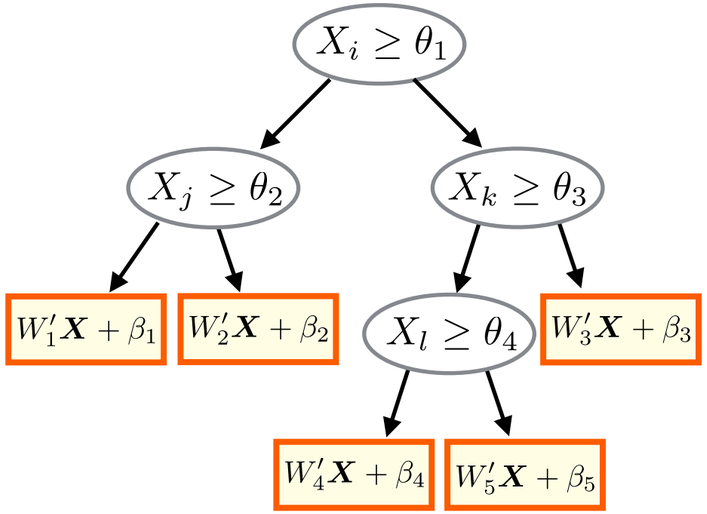


各領域で自由に $P(y|x)$ を学習

決定木の入力空間分割

モデル木(Model tree)

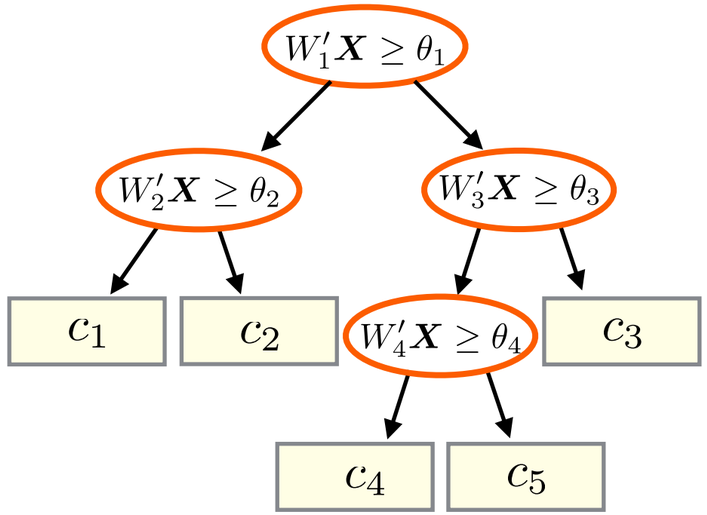
葉ノードが定数予測→**線形予測**



- M5/M5’ (Quinlan, 1992; Wang & Witten 1997)
- RETIS (Karalic & Cestnik 1991)
- SECRET (Dobra & Gehrke, 2002)
- SMOTI (Malerba+ 2004)
- MAUVE (Vens & Blockeel, 2006)
- LLRT (Vogel, Asparouhov, Scheffer, 2007)
- Cubist (Quinlan, 2011)

多変量木(Multivariate tree)

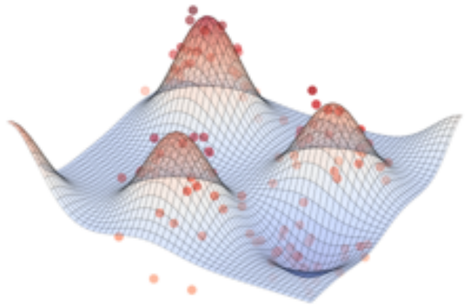
分割が単変量→**多変量線形**



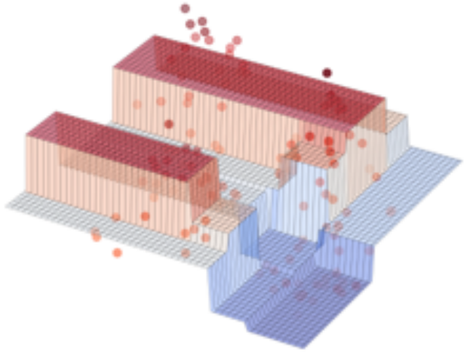
- OC1/Oblique Tree (Murthy+ 1994)
- Perceptron Tree (Utgoff , 1988)
- Large Margin Tree (Wu+ 1999; Bennett+ 2000)
- Margin Tree (Tibshirani & Hastie 2007)
- Geometric Decision Tree (Manwani & Sastry, 2012)
- HHCART (Wickramarachchi+, 2016)

決定木の入力空間分割

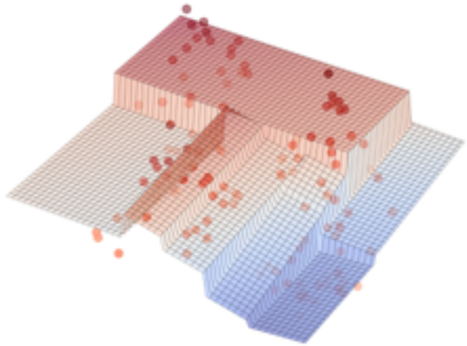
モデル木 (Model Trees)：葉ノードが定数予測→**線形予測**



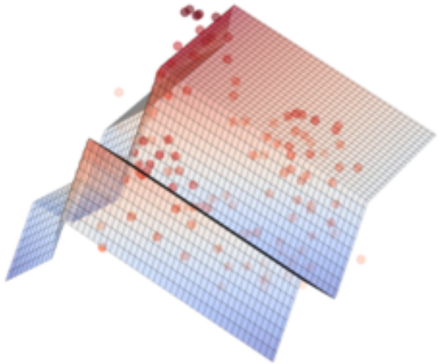
データ (回帰)



Regression Trees



M5

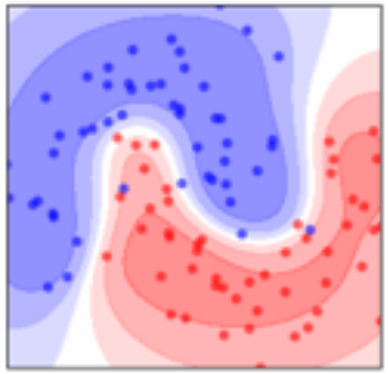


MARS

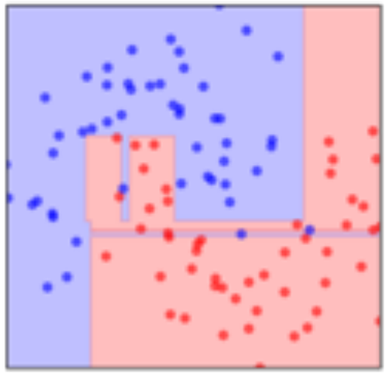
多変量適応的回帰スプライン(MARS)

決定木で領域を二分割する代わりに領域を折り曲げる。
(Greedyで学習)

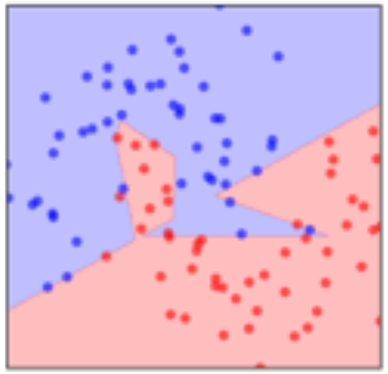
多変量木 (Multivariate Trees)：分割が単変量→**多変量線形**



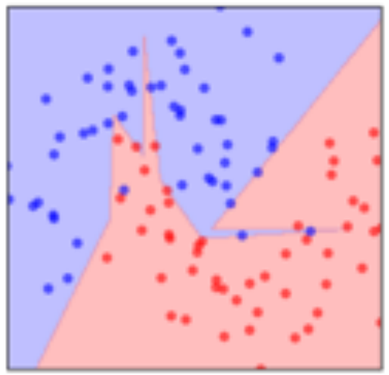
データ (分類)



Classification Trees



OC1



Multivariate CART

あまり実用化していない…

✓ **モデル木：葉に(定数予測より)複雑なモデルを持たせる**

■ 領域境界が著しく不安定・非連続になる

対処
- M5/M5'の平滑化ヒューリスティクス
- 多変量適応型回帰スプライン(MARS)による折れ線型区分線形学習
- 確率的決定木

⟷ **スプライン**

領域境界で**連続**

■ 最良分割点探索に葉モデルのフィットが必要となり
ちゃんと同時に最適化すると計算コストが非常に大きくなる

✓ **多変量木：二分割に(単変量二分割より)複雑な二値分類器を用いる**

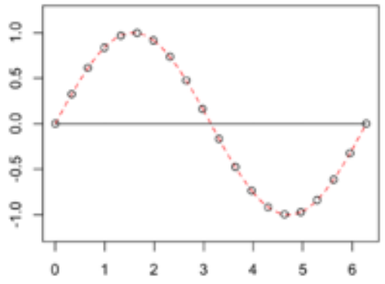
■ データ断片化(各領域のサンプル数減少)を起こし過剰適合しやすい
→ 多分岐ではなくて二分岐ベースが主流なのもこの理由による

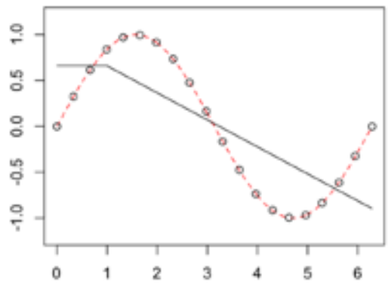
※これは学習方式がGreedyなせい？

■ 分割により各領域に落ちるサンプル数はどんどん減るので次元を
減らさない限り、小サンプル設定で良い分割の学習が困難になる

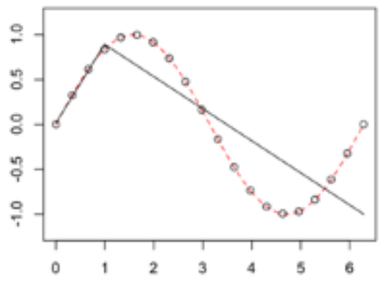
本当に区分的定数・軸並行二分割で良いのかは最大の関心

Friedman JH. **Multivariate Adaptive Regression Splines**. *The Annals of Statistics*, 1991;19: 1–67.

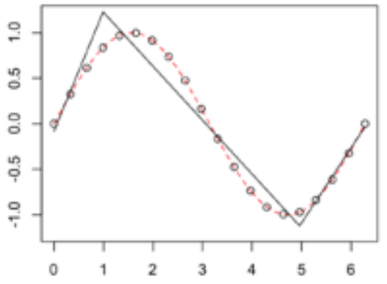




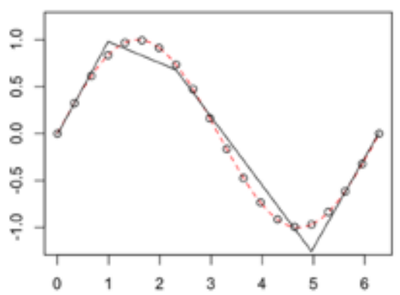
y=0.66 + -0.29 max(0,x-0.99)



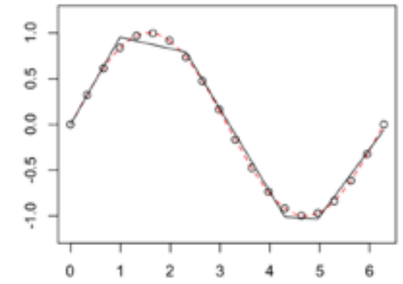
y=0.89 + -0.36 max(0,x-0.99)
+ -0.88 max(0,0.99-x)



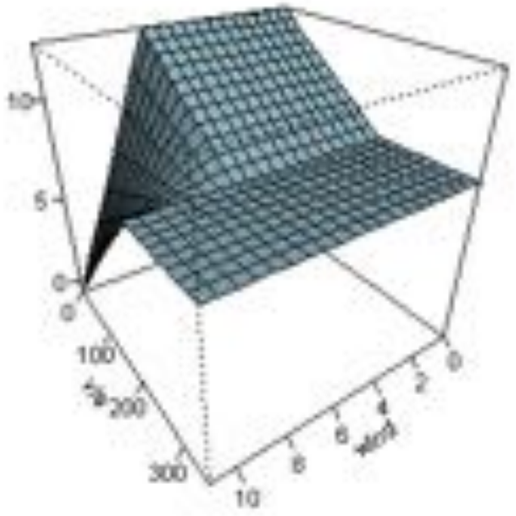
y=1.23 + -0.59 max(0,x-0.99)
+ -1.32 max(0,0.99-x)
+ 1.41 max(0,x-4.96)



y=0.98 + -0.23 max(0,x-0.99)
+ -1.0 max(0,0.99-x)
+ 1.69 max(0,x-4.96)
+ -0.51 max(0,x-2.31)



y=0.95 + -0.12 max(0,x-0.99)
+ -0.96 max(0,0.99-x)
+ 0.77 max(0,x-4.96)
+ -0.79 max(0,x-2.31)
+ 0.87 max(0,x-2.31)



$$\hat{f}(\boldsymbol{X}) = \sum_{i=1}^{k} c_i B_i(\boldsymbol{X})$$

$$B_i(\boldsymbol{X}) = \begin{cases} 1 & \text{constant} \\ h_{x_{i_j}}(X_j) & \text{hinge at a sample} \\ h_{x_{i_j}}(X_j) h_{x_{i_k}}(X_k) & \text{product of hinges} \end{cases}$$

愚直にやると**計算時間がかかる**ので実際の学習には
Fast MARS (Friedman, 1993)がよく用いられる

$$h_c(x) = \begin{cases} \max(0, x - c) \\ \max(0, c - x) \end{cases}$$

本当に区分的定数・軸並行二分割で良いのかは最大の関心

決定木は領域分割上の**多次元ヒストグラム (区分的定数予測)**

→ 統計的には**ほとんど経験分布**に近い良い性質を持つ (**悪いことが非常に起きづらい**)

→ 「Greedyで妥当性**もおぼつかない**テキトーな領域分割」に「**超コンサバ**な予測」を抱合せ

軸並行な再帰的二分割
Greedy探索

区分的「定数」
(ただのビン可変なヒストグラム)

- MARSのような"スプライン(折れ面)"の直接のフィッティングは**実際に使うと難がある**
  (領域分割と領域wiseな写像の同時最適化はやはり一般にはとても難しい)
- 深層学習がMASOの合成で書けるpiecewise affineな予測と考えれば、パラメタ最適化だけすれば**「領域分割のほうは勝手に定まる」MAS**になっている点は非常に興味深い

ランダム射影木: Random Projection Trees

空間分割データ構造としては決定木はk-d treeだが高次元ではRP木のほうが性質が良いはず

STOC 2008

Google scholar citations: 419

Random projection trees and low dimensional manifolds

Sanjoy Dasgupta
UC San Diego
dasgupta@cs.ucsd.edu

Yoav Freund
UC San Diego
yfreund@cs.ucsd.edu

ABSTRACT

We present a simple variant of the k-d tree which automatically adapts to intrinsic low dimensional structure in data without having to explicitly learn this structure.

1. INTRODUCTION

A k-d tree [4] is a spatial data structure that partitions $\mathbb{R}^D$ into hyperrectangular cells. It is built in a recursive manner, splitting along one coordinate direction at a time (Figure 1, left). The succession of splits corresponds to a binary tree whose leaves contain the individual cells in $\mathbb{R}^D$.

These trees are among the most widely-used spatial partitionings in machine learning and statistics. To understand

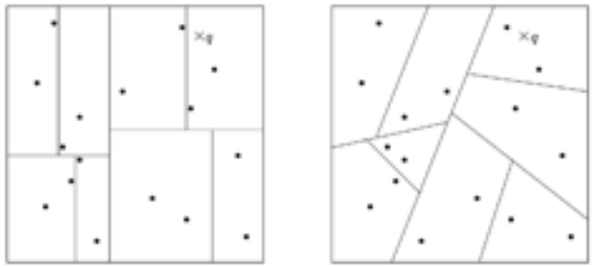


Figure 1: Left: A spatial partitioning of $\mathbb{R}^2$ induced by a k-d tree with three levels. The dots are data points; the cross marks a query point q. Right: Par-

NIPS 2010

Google scholar citations: 10

Random Projection Trees Revisited

Aman Dhesi*
Department of Computer Science
Princeton University
Princeton, New Jersey, USA.
adhesi@princeton.edu

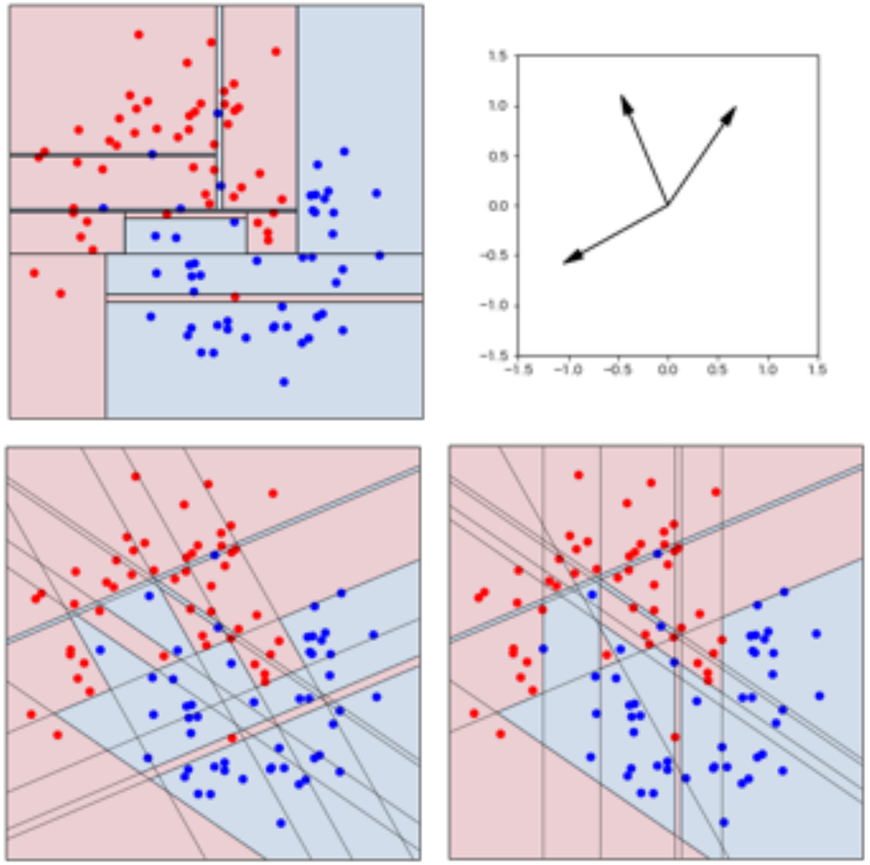
Purushottam Kar
Department of Computer Science and Engineering
Indian Institute of Technology
Kanpur, Uttar Pradesh, INDIA.
purushot@cse.iitk.ac.in

Abstract

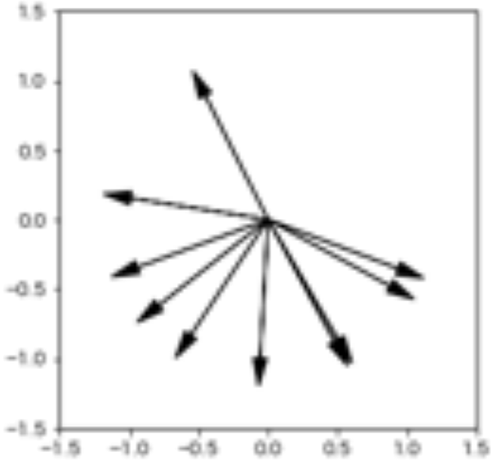
The Random Projection Tree (RPTREE) structures proposed in [1] are space partitioning data structures that automatically adapt to various notions of intrinsic dimensionality of data. We prove new results for both the RPTREE-MAX and the RPTREE-MEAN data structures. Our result for RPTREE-MAX gives a near-optimal bound on the number of levels required by this data structure to reduce the size of its cells by a factor $s \geq 2$. We also prove a packing lemma for this data structure. Our final result shows that low-dimensional manifolds have bounded

区分的定数だと(Splineと異なり)領域境界が不連続すぎる…?

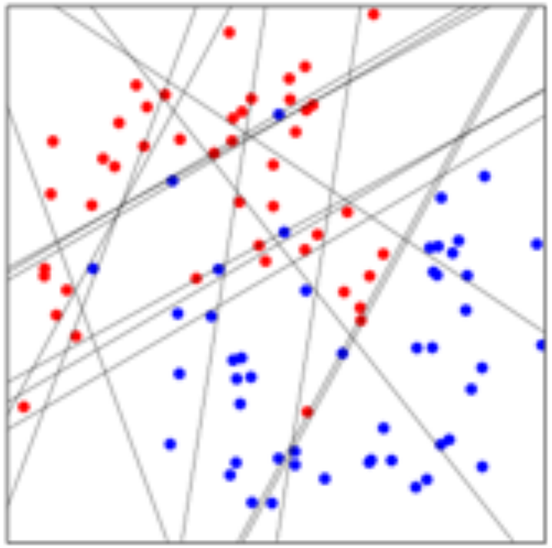
あるいは不連続性の問題は「アンサンブル」で解消するから気にしなくて良いのかダメなのか



10 RP vectors



RP決定木の領域分割



Affineスプラインに似た**感じ**の領域分割は
容易に可能

Open questions

- 現在非常によく使われる決定木アンサンブルはpiecewise 'constat'で領域構築はGreedy軸並行二分割
  → これが現実的な最も良い妥協解なのか、改善するアイデアがあり得るのか
- RP Tree + ExtraTrees + Gradient Boosting?
- 各領域ごとにAffineな写像を定義するのは要検討?
- 区分的定数のままが経験分布に沿っていて良い？
  ExtraTreesで擬似的な連続性を表現する?

これはNN-likeな決定木構築や決定木のNN-likeな学習に関係

- **Differentiableな決定木 / Neural Networksからの決定木構築**

Zantedeschi V, Kusner MJ, Niculae V. **Learning Binary Decision Trees by Argmin Differentiation.** *ICML 2021*, http://arxiv.org/abs/2010.04627

Hazimeh H, Ponomareva N, Mol P, Tan Z, Mazumder R. **The Tree Ensemble Layer: Differentiability meets Conditional Computation**. *ICML 2020*, https://arxiv.org/abs/2002.07772

Lee G-H, Jaakkola TS. **Oblique Decision Trees from Derivatives of ReLU Networks.** *ICLR 2020*, https://openreview.net/pdf?id=Bke8UR4FPB

Popov S, Morozov S, Babenko A. **Neural Oblivious Decision Ensembles for Deep Learning on Tabular Data.** *ICLR 2020*, http://arxiv.org/abs/1909.06312

Lay N, Harrison AP, Schreiber S, Dawer G, Barbu A. **Random Hinge Forest for Differentiable Learning.** *ICML 2018*, http://arxiv.org/abs/1802.03882

Kontschieder P, Fiterau M, Criminisi A, Bulo SR. **Deep Neural Decision Forests.** *IJCAI 2016*, https://www.ijcai.org/Proceedings/16/Papers/628.pdf

今日の話題提供

業務(自然科学での機械学習利活用)でユーザとして**決定木アンサンブル(とニューラルネット)**を使っていて出会った現象と問題の紹介

- 決定森回帰の**信頼区間**推定・Benign Overfitting
- 多変量木とReLUネットの**入力空間分割**

会議後の自己Follow-up

Thanks to　加納龍一さん (DNA/NII)

1. NGBoost and Prediction Intervals
2. Uncertainty Quantification (UQ)
3. Aleatoric uncertaintyとEpistemic uncertainty
4. Distribution-Free UQ in ML
5. Conformal Prediction

NGBoost

https://stanfordmlgroup.github.io/projects/ngboost/

Stanford ML Group

NGBoost: Natural Gradient Boosting for Probabilistic Prediction

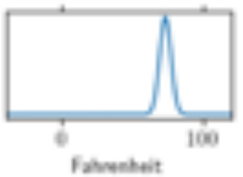
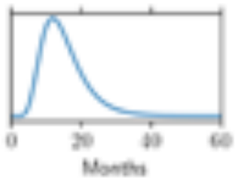
Tony Duan*, Anand Avati*, Daisy Yi Ding, Sanjay Basu, Andrew Ng, Alejandro Schuler

Predictive Uncertainty Estimation in the real world.

Estimating the uncertainty in the predictions of a machine learning model is crucial for production deployments in the real world. Not only do we want our models to make accurate predictions, but we also want a correct estimate of uncertainty along with each prediction. When model predictions are part of an automated decision-making workflow or production line, predictive uncertainty estimates are important for determining manual fallback alternatives or for human inspection and intervenion.

Probabilistic prediction (or probabilistic forecasting), which is the approach where the model outputs a full probability distribution over the entire outcome space, is a natural way to quantify those uncertainties.

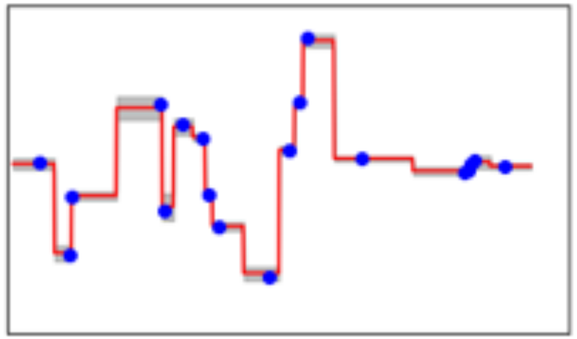
Compare the point predictions vs probabilistic predictions in the following examples.

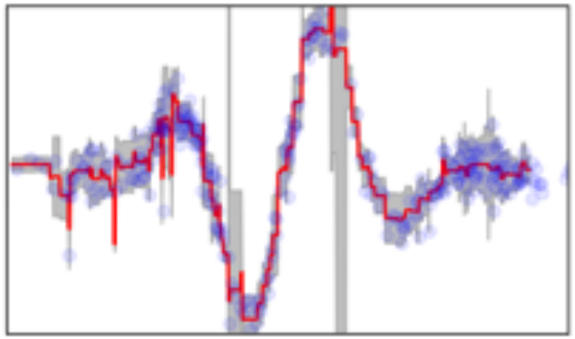
Question	Point Prediction (No uncertainty estimate)	Probabilistic Prediction (Uncertainty is implicit)
What will be the temperature at noon tomorrow?	73.4 Fahrenheit	
How long will this patient live?	11.3 months	

NGBoost and Prediction Intervals

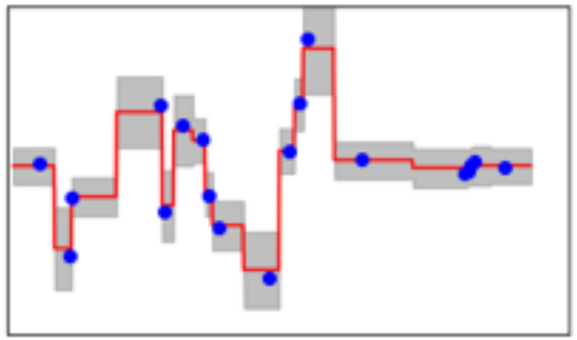
https://towardsdatascience.com/interpreting-the-probabilistic-predictions-from-ngboost-868d6f3770b2

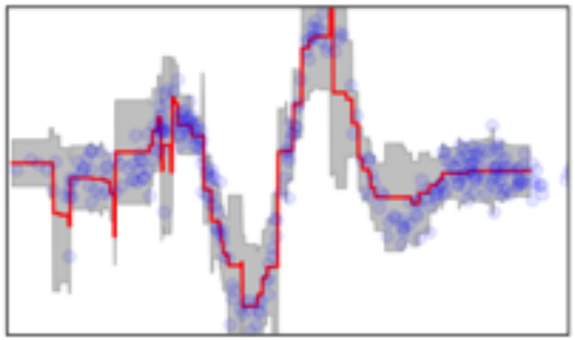
NGBRegressor





```
model = NGBRegressor()
```

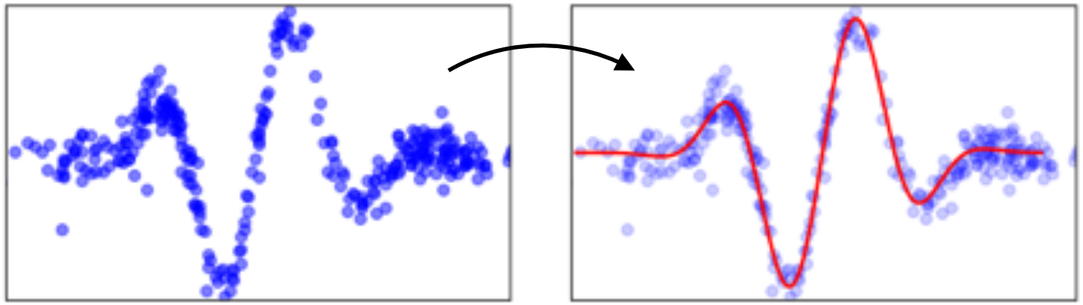




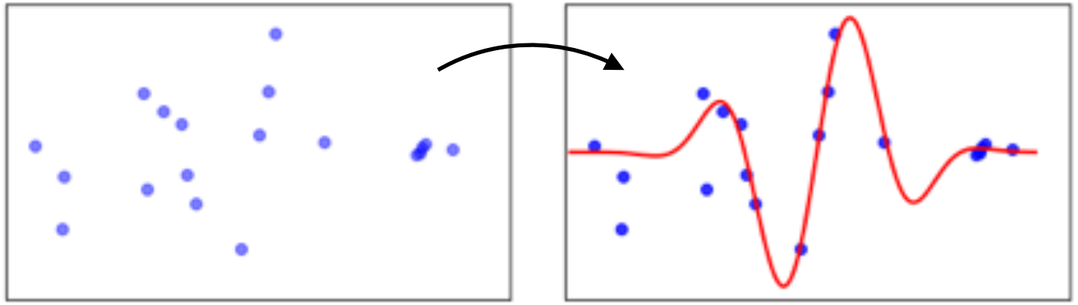
```
base = DecisionTreeRegressor(max_leaf_nodes=8)
model = NGBRegressor(Base=base, n_estimators=200)
```

ちなみに↓のようなそもそも論も大いにある…

これならともかく…



そもそも これは出来て良いのか？



そもそも論として、統計学的にはサンプル不足であるUnderspecified な状況で何か建設的な議論は可能なのか？？

機械学習モデルのdeploy後に出会うデータ(テストデータ)は**無限**、訓練データ・検証データは**有限**、という無理設定では結局**「モデルの帰納バイアス(inductive bias)」が目前の問題にマッチするか**だけが重要？

⟶ この意味でもモデルの挙動の深い理解とそれを実用ツールと実問題に還元する実践のiterationは大事！

Uncertainty Quantification (UQ)

機械学習の「予測」そのものはデータのアヤや問題設定そのものの限界によって100%正確には**決してならない**。むしろ高次元性などの困難さを踏まえると常に一定の確度の限界があると考えるべき。

機械学習予測の「不確実性」評価(不確かさの定量化)は実用機械学習の重要な問題！

Uncertainty Quantification (UQ)

https://en.wikipedia.org/wiki/Uncertainty_quantification

↳ 統計学だけではなく計測や数理モデリングなど工学領域もふくんで様々に議論され、さまざまなUQ手法が提案されてきた。

Aleatoric uncertaintyとEpistemic uncertainty

Uncertainty(不確実性)には「**Aleatoric**」なものと「**Epistemic**」なものがあり、様々な分野で議論されてきた。二種類の不確実性が混合されて予測モデリングに影響する。

https://en.wikipedia.org/wiki/Uncertainty_quantification#Aleatoric_and_epistemic

- **Aleatoric Uncertainty (Statistical Uncertainty)**　　**偶然誤差 or 偶然的な不確実さ**

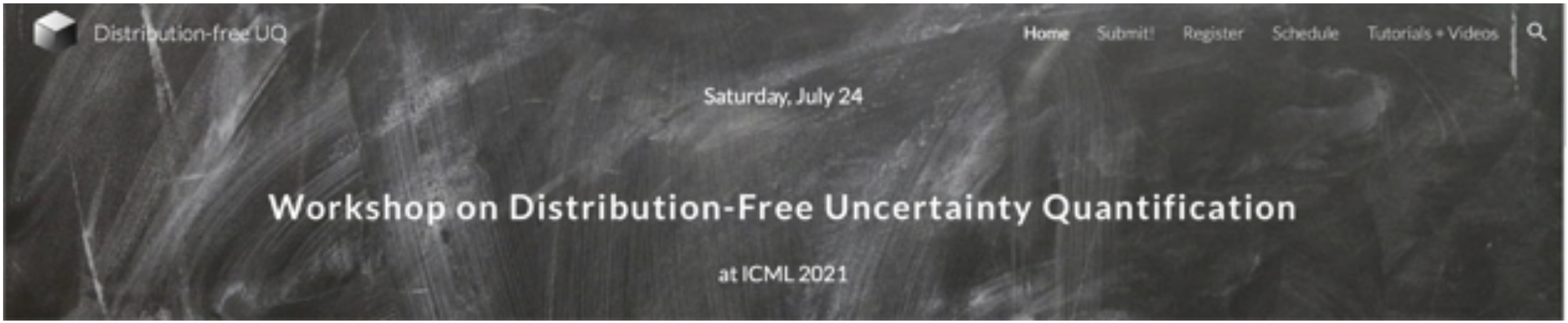
  対象現象にそもそも含まれる本質的な偶然性による「不確実性」。同じ実験や計測を二回やると値がまったく同じにはならない。人手で付与する教師ラベルのinconsistencyなども含む。より多くの情報を集めたとしても低減することができない不確かさ。

- **Epistemic Uncertainty (Systematic uncertainty)**　**系統誤差 or 認識論的な不確実さ**

  知識が不足あるいはモデルが不正確・不完全であることに起因する「不確実性」。現時点では入手できない情報があるために生じる不確かさ。

Distribution-Free Uncertainty Quantification (UQ)

- Accuracy alone does not suffice for reliable, consequential decision-making; we also need uncertainty.
- **Distribution-free UQ** gives finite-sample statistical guarantees for any predictive model, no matter how bad/misspecified, and any data distribution, even if unknown.
- DF techniques such as conformal prediction represent a new, principled approach to UQ for complex prediction systems, such as deep learning.



Distribution-Free Uncertainty Quantification (UQ)

What is distribution-free uncertainty quantification?

Distribution-free methods make minimal assumptions about the data distribution or model, yet still provide uncertainty quantification. Examples of DF methods include conformal prediction, tolerance regions, risk-controlling prediction sets, calibration by binning, and more. We take a broad outlook on DF methods, so any assumption-light uncertainty quantification approaches are welcome.

- *conformal prediction*
- *tolerance regions*
- *risk-controlling prediction sets*
- *calibration by binning*

and more

Conformal prediction (CP)

https://en.wikipedia.org/wiki/Conformal_prediction

Journal of Machine Learning Research 9 (2008) 371-421

A Tutorial on Conformal Prediction

Glenn Shafer* GSHAFER@RUTGERS.EDU
Department of Accounting and Information Systems
Rutgers Business School
180 University Avenue
Newark, NJ 07102, USA

Vladimir Vovk VOVK@CS.RHUL.AC.UK
Computer Learning Research Centre
Department of Computer Science
Royal Holloway, University of London
Egham, Surrey TW20 0EX, UK

Conformal prediction (CP)

Conformal prediction can be used with any method of point prediction for classification or regression, including support-vector machines, decision trees, boosting, neural networks, and Bayesian prediction. Starting from the method for point prediction, we construct a *nonconformity measure*, which measures how unusual an example looks relative to previous examples, and the *conformal algorithm* turns this nonconformity measure into prediction regions.

"Conformity" 整合性

↳ ***Non-conformity measure*** 不整合性基準

ある例がそれまでの例と比べてどれくらいunusualかを測り、CPアルゴリズムは
この不整合性を元に予測区間を構築する

Conformal prediction (CP)

Awesome Conformal Prediction

https://github.com/valeman/awesome-conformal-prediction

Why Conformal Prediction?

One of the most influential and celebrated machine learning researchers - Professor Michael I. Jordan:

'**Conformal Prediction ideas are THE answer to UQ (uncertainty quantification), I think it's the best I have seen - its simple, generalisable etc.**' (ICML 2021 UQ workshop). 🔥🔥🔥🔥🔥

One the most influential statistics Professors - Larry Wasserman (Carnegie Mellon):

'**So the beauty of the conformal thing is how simple it is to do it and how general it is. So I think you know ideas that catch on, general ideas that are pretty general and easy to implement that you can picture yourself using in real applications are the reason that people using conformal prediction.**' 🚀🚀🚀🚀🚀

https://slideslive.com/icml-2021/workshop-on-distributionfree-uncertainty-quantification

Conformal prediction (CP)

Awesome Conformal Prediction

https://github.com/valeman/awesome-conformal-prediction

In 2022 conformal prediction research experienced exponential growth in academia and with the availabity of open-source libaries the industry is positioned to replicate this growth in the industry.

🚂🚂Industry take notice. The revolution in Uncertainty Quantification / Probabilistic Prediction / Forecasting is already here 🚂🚂 A big one 🔥🔥🔥🔥🔥

🌟🌟🌟🌟🌟

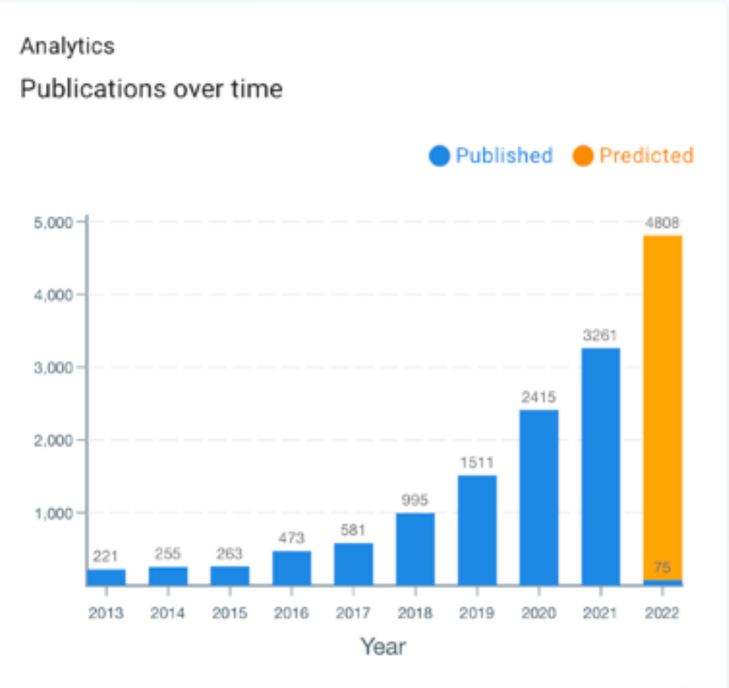
Featured resources:

A Gentle Introduction to Conformal Prediction and Distribution-Free Uncertainty Quantification

This is newest version of the super-popular tutorial on Conformal Prediction (over 3,500 stars on YouTube) now significantly expanded (2x), including advanced techniques such as covariate shift conformal, as well as a super fun history and literature review in Section 7.

A Tutorial on Conformal Prediction by Anastasios Angelopoulos and Stephen Bates (2021) 🔥🔥🔥🔥🔥

⭐⭐⭐⭐⭐



Conformal prediction (CP)

https://www.stat.cmu.edu/~aramdas/conformal.html

Distribution-free uncertainty quantification (conformal, calibration) (package 1) (package 2) (tutorial)

- **Tutorial talk (U Toulouse)**
- **Tutorial talk (IAS Princeton)**
- **Top-label calibration**
  C. Gupta, A. Ramdas `ICLR, 2022` arxiv
- **Distribution-free calibration guarantees for histogram binning without sample splitting**
  C. Gupta, A. Ramdas `ICML, 2021` arxiv proc
- **Distribution-free uncertainty quantification for classification under label shift**
  A. Podkopaev, A. Ramdas `UAI, 2021` arxiv
- **Distribution-free binary classification: prediction sets, confidence intervals and calibration**
  C. Gupta, A. Podkopaev, A. Ramdas `NeurIPS, 2020` arxiv proc talk
- **Nested conformal prediction and quantile out-of-bag ensemble methods**
  C. Gupta, A. Kuchibhotla, A. Ramdas `Pattern Recognition, Special Issue on Conformal Prediction` arxiv code talk
- **Predictive inference with the jackknife+**
  R. Barber, E. Candes, A. Ramdas, R. Tibshirani `Annals of Stat., 2020` arxiv code proc
- **The limits of distribution-free conditional predictive inference**
  R. Barber, E. Candes, A. Ramdas, R. Tibshirani `Information and Inference, 2020` arxiv proc
- **Conformal prediction under covariate shift**
  R. Tibshirani, R. Barber, E. Candes, A. Ramdas `NeurIPS, 2019` arxiv proc
- **Distribution-free prediction sets with random effects**
  R. Dunn, L. Wasserman, A. Ramdas (JASA, revision) arxiv

MAPIE - Model Agnostic Prediction Interval Estimator

https://github.com/scikit-learn-contrib/MAPIE

MAPIE methods belong to the field of conformal inference.

[1] Rina Foygel Barber, Emmanuel J. Candès, Aaditya Ramdas, and Ryan J. Tibshirani. "Predictive inference with the jackknife+." Ann. Statist., 49(1):486–507, February 2021.

[2] Byol Kim, Chen Xu, and Rina Foygel Barber. "Predictive Inference Is Free with the Jackknife+-after-Bootstrap." 34th Conference on Neural Information Processing Systems (NeurIPS 2020).

[3] Mauricio Sadinle, Jing Lei, and Larry Wasserman. "Least Ambiguous Set-Valued Classifiers With Bounded Error Levels." Journal of the American Statistical Association, 114:525, 223-234, 2019.

[4] Yaniv Romano, Matteo Sesia and Emmanuel J. Candès. "Classification with Valid and Adaptive Coverage." NeurIPS 202 (spotlight).

[5] Anastasios Nikolas Angelopoulos, Stephen Bates, Michael Jordan and Jitendra Malik. "Uncertainty Sets for Image Classifiers using Conformal Prediction." International Conference on Learning Representations 2021.

MAPIE - Model Agnostic Prediction Interval Estimator

With MAPIE, uncertainties are back in machine learning

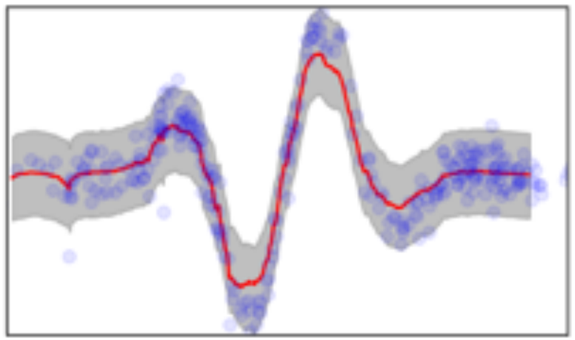
https://towardsdatascience.com/with-mapie-uncertainties-are-back-in-machine-learning-882d5c17fdc3

Family	Method	Theoretical guarantees	Model agnostic	Open-source implementation	Language	Compatible with big datasets
QR	Quantile Regression	✗	✓	✓	R/Python	✓
Data perturbation	Bootstrap	✗	✓	✓	R/Python	✓
	Jackknife	✗	✓	✓	R/Python	✗
	Jackknife+	✓	✓	✗	-	✗
	CV+	✓	✓	✗	-	✓
Model perturbation	Random seed	✗	✗	✓	R/Python	✓
	MC Dropout	✗	✗	✓	R/Python	✓
Bayesian	Bayesian inference	✓	✗	✓	R/Python	✗

MAPIE - Model Agnostic Prediction Interval Estimator

https://github.com/scikit-learn-contrib/MAPIE

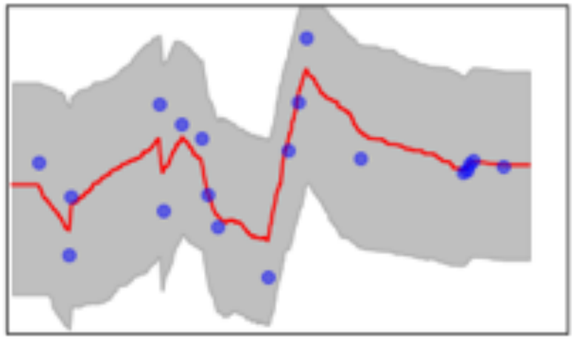
regressor = ExtraTreesRegressor(max_leaf_nodes=32, bootstrap=True)



MapieRegressor(regressor, method="plus", cv=-1)

Jacknife+

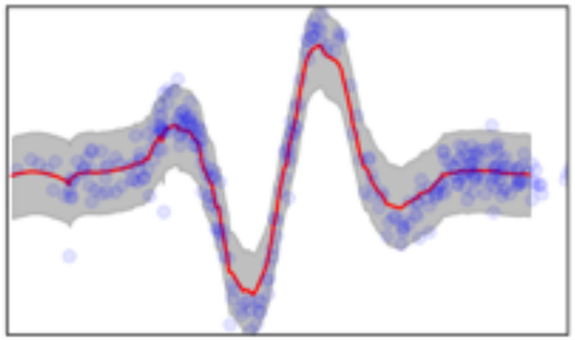
95% prediction intervals



MapieRegressor(regressor, method="plus", cv=-1)

Jacknife+

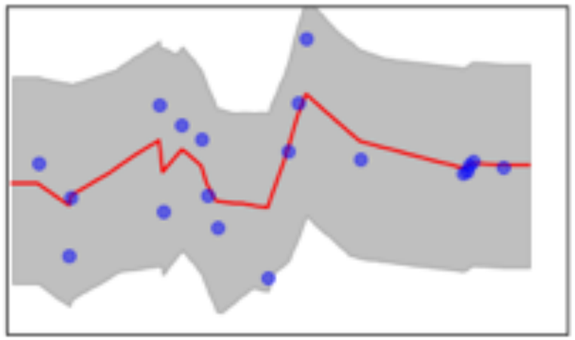
90% prediction intervals



MapieRegressor(regressor, method="plus", cv=Subsample(n_resamplings=50))

Jacknife+ after bootstrap

95% prediction intervals



MapieRegressor(regressor, method="plus", cv=Subsample(n_resamplings=50))

Jacknife+ after bootstrap

90% prediction intervals