ソフトウェアエンジニアのための

機械学習による データ分析 実践編

@canard0328

本日の目的

実際にデータを触りながら
機械学習によるデータ分析について
一連のプロセスを体験

タスク：教師あり学習の分類タスク

分析環境：  python™ or 

本日の資料

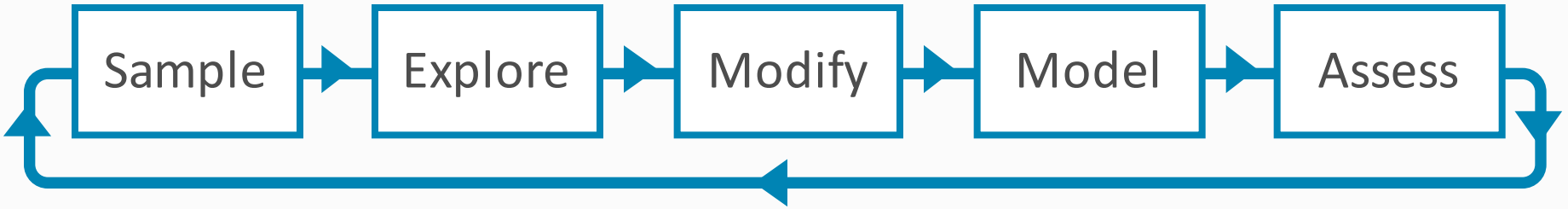
演習資料・スクリプト

http://nbviewer.ipython.org/gist/canard0328/a5911ee5b4bf1a07fbcb/
https://gist.github.com/canard0328/07a65584c134a2700725

http://nbviewer.ipython.org/gist/canard0328/6f44229365f53b7bd30f/
https://gist.github.com/canard0328/b2f8aec2b9c286f53400

分析プロセス

SEMMA



Sample　データの取得
Explore　データの探索（可視化など）
Modify　データの作成・選択・変換（前処理）
Model　モデリング（機械学習）
Assess　評価

その他の分析プロセス

CRISP-DM（CRoss-Industry Standard Process for Data Mining）

Business Understanding
Data Understanding
Data Preparation
Modeling
Evaluation
Deployment

KDD（Knowledge Discovery in Databases）

Selection
Preprocessing
Transformation
Data Mining
Interpretation/Evaluation

KKD（Keiken, Kan and Dokyo）

データの入手

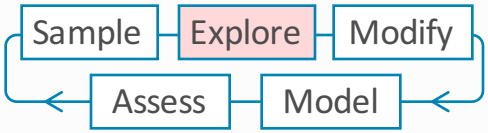
タスク：タイタニック号乗客の生存予測

http://biostat.mc.vanderbilt.edu/wiki/pub/Main/DataSets/titanic3.csv
(Data obtained from http://biostat.mc.vanderbilt.edu/DataSets)


```
>>> import pandas as pd
>>> data = pd.read_csv('titanic3.csv')
```

```
> data = read.csv("titanic3.csv",
+ stringsAsFactors=F, na.strings=c("","NA"))
```

データの探索



データの確認

- データのサイズ・種類
- 欠損値の有無

データの可視化

- 分布：ヒストグラム，箱ひげ図
- 割合：帯グラフ，積み上げ棒グラフ
- データ間の関係：散布図，クロス集計
- 変化：折れ線グラフ

一般的なデータ形式

説明変数，特徴量　　目的変数

年齢	性別	加入日	加入プラン	地区	解約
23	男	2012/03/03	スタンダード	東京	0
34	女	2014/11/23	スタンダード	埼玉	1
49	男	2000/05/11	プレミアム	千葉	0
19	男	2013/12/05	ライト	大阪	0
60	女	2011/03/28	シニア	東京	0
		・			
		・			
		・			

特徴量の分類

名義尺度

名前，電話番号など

順序尺度

レースの着順など

間隔尺度

摂氏，華氏など（乗除不可）

比例尺度

質量，長さなど

特徴量の分類

数値データ（量的変数）

比例尺度，（間隔尺度）

カテゴリデータ（質的変数）

名義尺度，順序尺度，（間隔尺度）

演習

1. データの入手
2. データの確認
3. 欠損値の確認
4. データの可視化

データの前処理

欠損値の処理

カテゴリ変数の処理

データの標準化

特徴量の作成・選択

欠損値の処理

捨てる

欠損値が少数，データが大量

置換する

最頻値，中央値，平均値

補間する

時系列データ

欠損値の生じ方が完全にランダムでない限り
分析に影響を与える
⇒完全情報最尤推定法，多重代入法

特徴量の分類

数値データ（量的変数）
比例尺度，（間隔尺度）

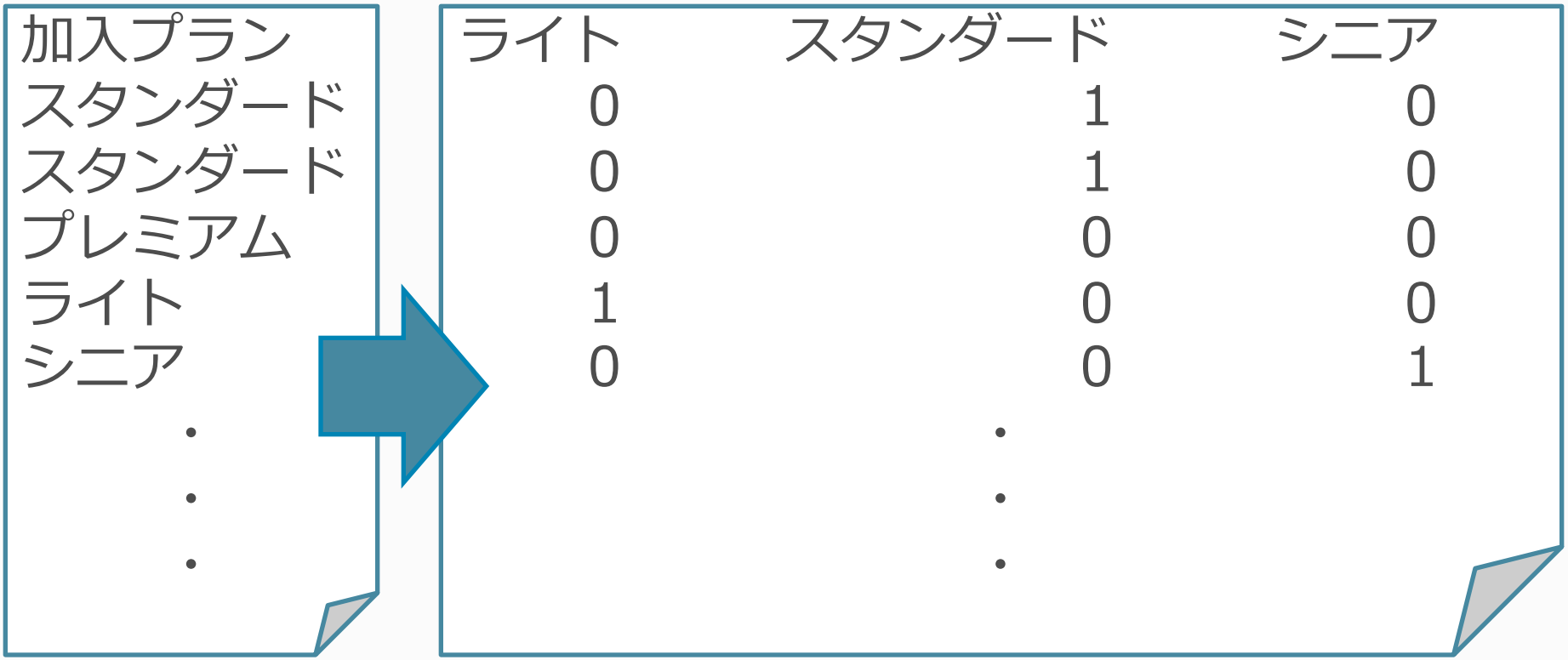
カテゴリデータ（質的変数）
名義尺度，順序尺度，（間隔尺度）

機械学習アルゴリズムは数値データを前提としているものが多い.

カテゴリデータを数値データへ変換

ダミー変数（1-of-K表現）

カテゴリデータを数値データへ変換



N種類の変数をN-1個の特徴量で表現可能

Feature hashing / Hashing trick

ダミー変数はカテゴリの種類が多いと
特徴量の次元数が大きくなりすぎる

Feature hashingにより任意の次元に削減

```
x := new vector[N]
for f in features:
    h := hash(f)
    x[h mod N] += 1
```

http://en.wikipedia.org/wiki/Feature_hashing

Nの値がある程度大きければ精度への影響小

次元の呪い (Curse of dimensionality)

特徴量（説明変数）の数が増えると汎化性能※を向上させることが難しくなる

使えそうなデータはなんでも特徴量に加えてしまえ，は危険

特徴選択や**次元削減**により特徴量の数を減らす

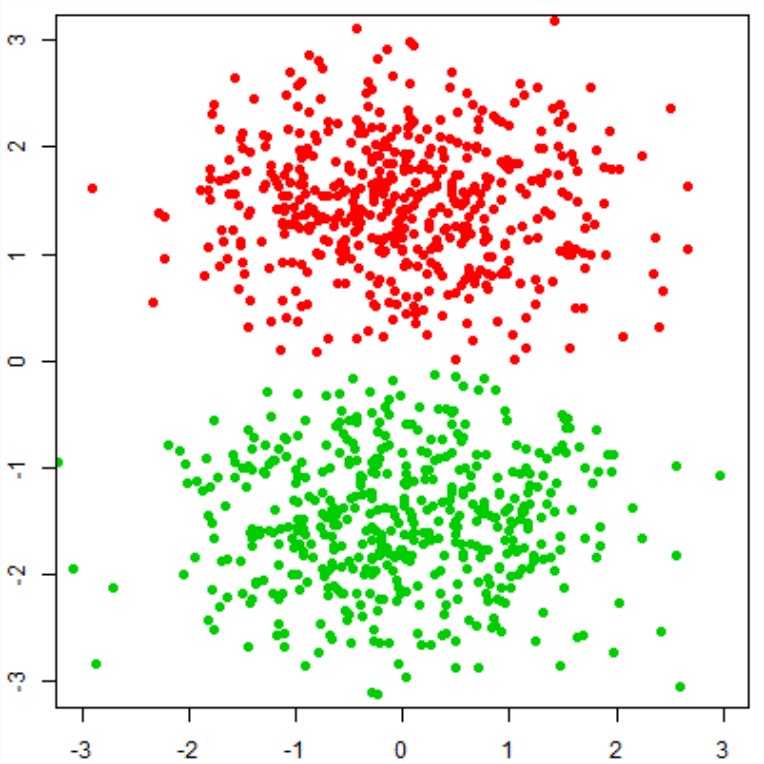
データを用意する段階で特徴量を吟味することが非常に重要

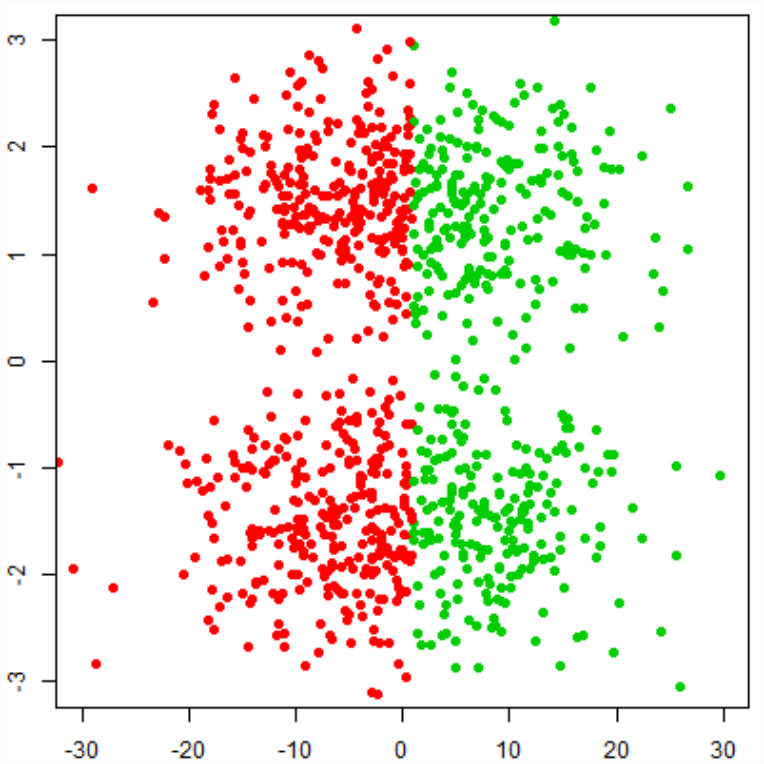
次元の呪いについて，詳しくは「球面集中現象」を検索

※未知のデータを予測する性能

標準化 (Standardization)

xの値を10倍しただけでクラスタリングの結果が変わってしまう





標準化 (Standardization)

必要であれば特徴量ごとに標準化(Standardization)を行う

$$z = \frac{x - \mu}{\sigma}$$

μ ：xの平均

σ ：xの標準偏差

平均０，標準偏差1にする変換が一般的

特徴量の作成・選択

特徴量の作成

特徴量同士の積を新たな特徴量に

特徴選択(Feature selection)

特徴量の中から有用なものを選び出す

前向き法(Forward stepwise selection)

後ろ向き法(Backward stepwise selection)

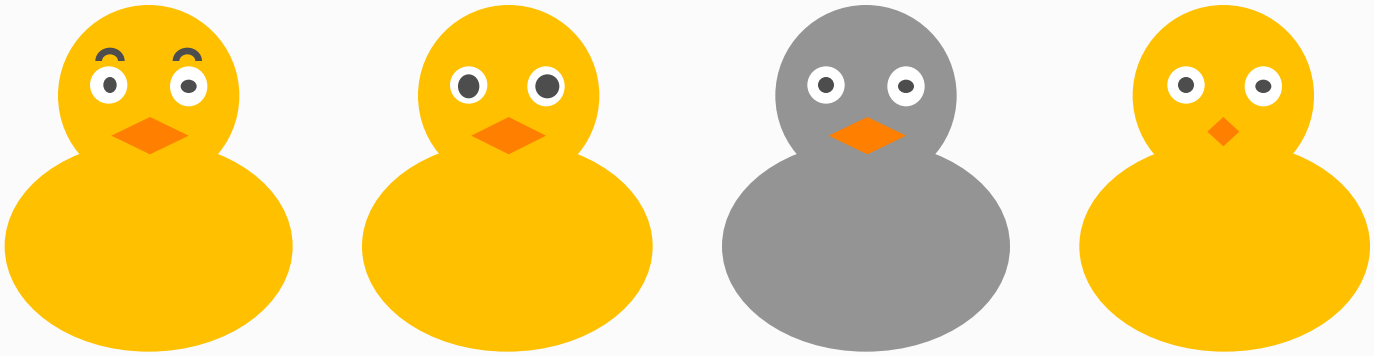
醜いアヒルの子定理

Ugly duckling theorem

醜いアヒルの子と普通のアヒルの子の類似性は
２羽の普通のアヒルの子の類似性と等しい

問題から独立した**万能な特徴量は存在しない**

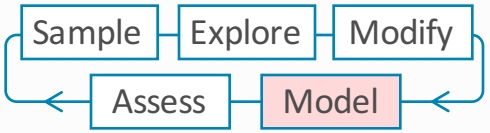
特徴量の設計が重要



演習

4. 欠損値の処理
5. カテゴリ変数の処理
6. データの標準化

モデリング



機械学習とは

“Machine learning is the science of getting computers to act without being explicitly programmed.”
Andrew Ng

一般的にはコンピュータの振る舞い方（モデル）を（大量の）データから**学習**することにより獲得する.

機械学習の分類

教師あり学習（supervised learning）

データが入力と出力のペアから成る

- 分類（識別）（classification）：出力がラベル
- 回帰（regression）：出力が数値

教師なし学習（unsupervised learning）

データは入力のみ

- クラスタリング
- 頻出パタンマイニング
- 外れ値検出（outlier detection）

機械学習の分類

その他の分類

- 半教師あり学習（semi-supervised learning）
- 強化学習（reinforcement learning）
- 能動学習（active learning）
- 逐次学習（online learning）
- 転移学習（transfer learning）

・・・

機械学習のアルゴリズム

教師あり学習

- 線形モデル（単／重回帰）
- ロジスティック回帰
- 判別分析
- k近傍法
- 決定木
- サポートベクターマシン
- ニューラルネットワーク
- ナイーブベイズ
- ランダムフォレスト

機械学習のアルゴリズム

教師なし学習

- K-means クラスタリング
- 階層的クラスタリング
- Apriori
- One-class SVM

アルゴリズム利用時の注意点

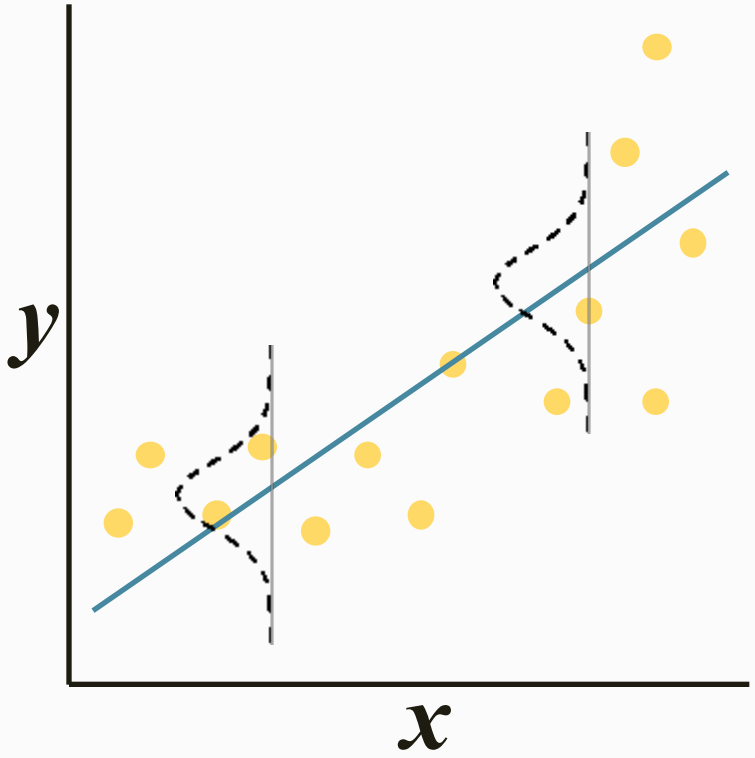
アルゴリズムによっては
データの分布などに仮定を
おいているものがある.

仮定に合わないデータを分析した場合
適切な結果が得られないことも

アルゴリズム利用時の注意点

線形モデル（単／重回帰分析）

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_i x_i + \varepsilon$$



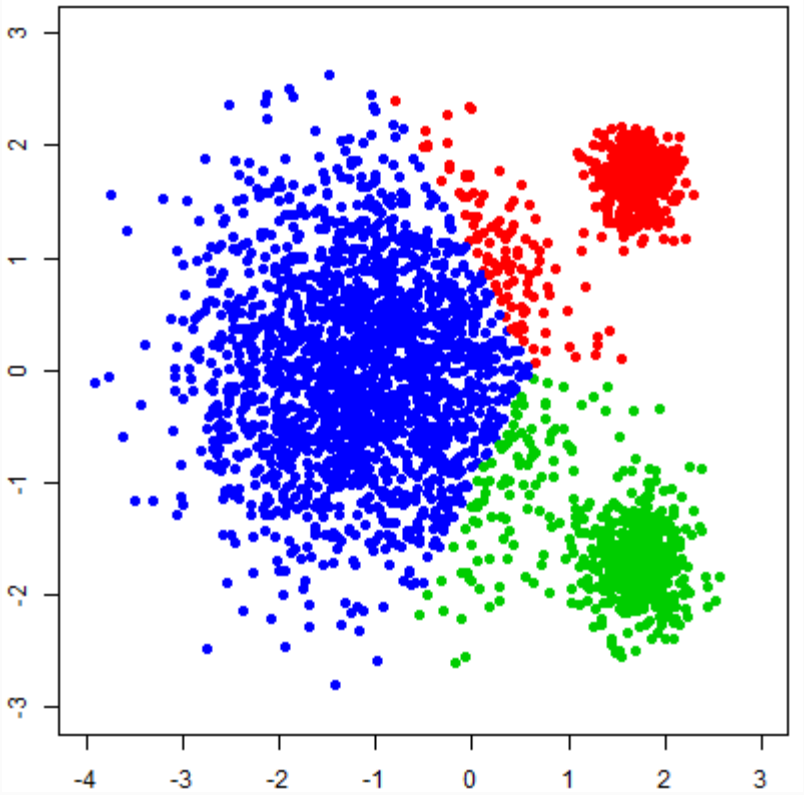
線形モデルは誤差が等分散
正規分布であることを仮定

一般化
（ポアソン分布，二項分布，
ガンマ分布，・・・）

一般化線形モデル
（generalized linear model）
※ロジスティック回帰はこの一種

アルゴリズム利用時の注意点

K-meansクラスタリング



K-meansクラスタリングは各クラスタが同じ大きさの超球であることを仮定している

クラスタの大きさに差がある場合

混合正規分布
（Gaussian mixture model）

アルゴリズム利用時の注意点

ノーフリーランチ定理

あらゆる問題で性能の良い

万能な学習アルゴリズムは存在しない

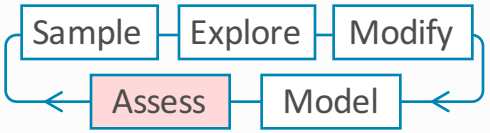
目的に適したアルゴリズムを選択しましょう

とは言っても，実用上，上手くいくことの多いアルゴリズムがあるのも事実

演習

7. モデリング

評価



評価基準

グリッドサーチ

交差検証

バイアス・バリアンス

学習曲線

回帰モデルの評価基準

平均絶対誤差(Mean absolute error)

小さいほど良い

平均二乗誤差(Mean square(d) error)

小さいほど良い

Root Mean Square(d) Errorもよく使われる

決定係数R^2(Coefficient of determination)

説明変数が目的変数をどれくらい説明するか

0(悪い)～1(良い)

特徴量が多いほど大きな値に⇒自由度調整済み決定係数

分類（識別）モデルの評価基準

精度(Accuracy)

正解数÷データ数

誤差率(Error rate)

1－精度

1万人のデータの内100人が陽性の場合，
常に陰性と判定するモデルの精度は**99%**
これはよいモデルといえるだろうか？

分類（識別）モデルの評価基準

混同行列(Confusion matrix)

		予測値	
		陽性 (Positive)	陰性 (Negative)
正解	陽性	真陽性 (True positive : TP)	偽陰性 (False negative : FN)
	陰性	偽陽性 (False positive : FP)	真陰性 (True negative : TN)

※予測したい事象が生じている状態が「陽性」
病気を判別したければ，病気の状態が「陽性」で健康な状態が「陰性」

分類（識別）モデルの評価基準

適合率(Precision)

TP/(TP + FP)
陽性と予測したものの正解率

再現率(Recall)

TP/(TP + FN)
陽性のうち正しく予測できた率

F値(F1 score, F-measure)

2・(適合率・再現率) / (適合率＋再現率)

		予測値	
		P	N
正解	P	TP	FN
	N	FP	TN

分類（識別）モデルの評価基準

真陽性率(True Positive Rate)

TP/(TP + FN)
陽性のうち正しく予測できた率（ヒット率）

偽陽性率(False Positive Rate)

FP/(FP + TN)
陰性のうち誤って陽性と予測した率（誤報率）

		予測値	
		P	N
正解	P	TP	FN
	N	FP	TN

分類（識別）モデルの評価基準

1万人のデータの内100人が陽性のとき
常に陰性と判定するモデル

		予測値	
		陽性(Positive)	陰性(Negative)
正解	陽性	0	100
	陰性	0	9900

精度：0.99

適合率：0

再現率：0

F値：0

脱線：不均衡データの分析

ラベルに偏りのあるデータは予測が困難

重みづけ

ライブラリを利用する場合，簡単に重みづけ可能な場合が多い

サンプル数の調整

少ない方を増やす，多い方を減らす，両方

SMOTEアルゴリズム

実際にはどちらも決め手とならないことも多い…

分類（識別）モデルの評価基準

真陽性率と偽陽性率はトレードオフ

陽性の取りこぼしが無いよう閾値を設定すると，
真陽性率は高くなるが，偽陽性率も高くなる．

ROC曲線

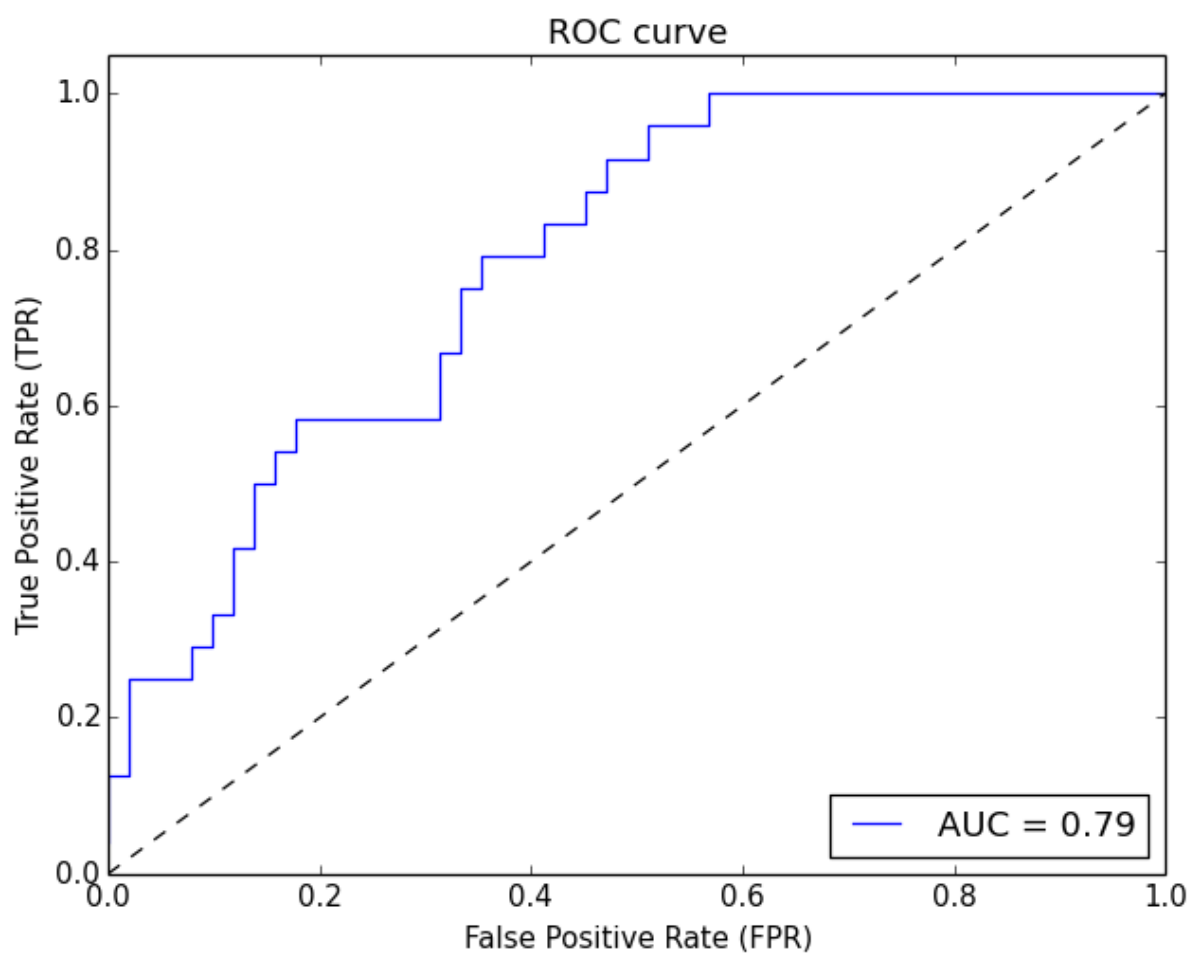
モデルのパラメータを変化させながら，偽陽性率と真陽性率をプロットしたもの

AUC

ROC曲線の下側の面積．1.0が最良

分類（識別）モデルの評価基準

ROC曲線とAUC



分析時の注意点

適切にデータを前処理して，
適切なアルゴリズムを選んで分析した．

```
> clf = SVC().fit(X, y)
```

誤差が大きい，このアルゴリズムは
使えない！

本当ですか？

ハイパーパラメータ

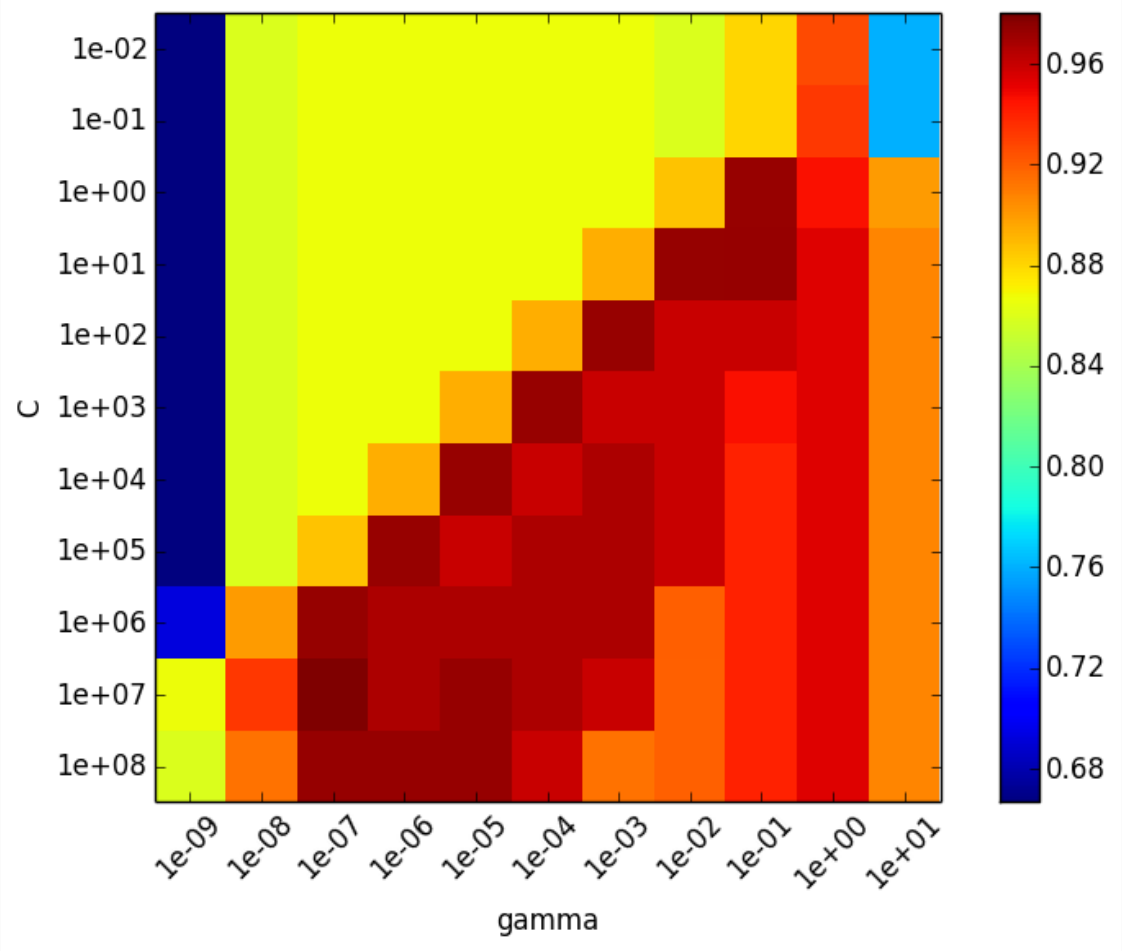
アルゴリズムはハイパーパラメータを調整することで性能が大きく変化

```
> clf = SVC(kernel='rbf', C=1.0 gamma=0.1).fit(X, y)
```

ハイパーパラメータの調整法は？

グリッドサーチ

各パラメータを適当な範囲で変化させ，最も性能のよいパラメータの組み合わせを選択する



グリッドサーチ

パラメータの変化幅，刻み幅

経験に依るところ大

物理量的なもの(例:決定木の深さ)は常識的な範囲で

そうでないものは桁を変えて(10^{-2},10^{-1},10^{0},10^{1},10^{2})

２段（多段）グリッドサーチ

初めは広く，荒く

範囲を絞って狭く，細かく

分析時の注意点

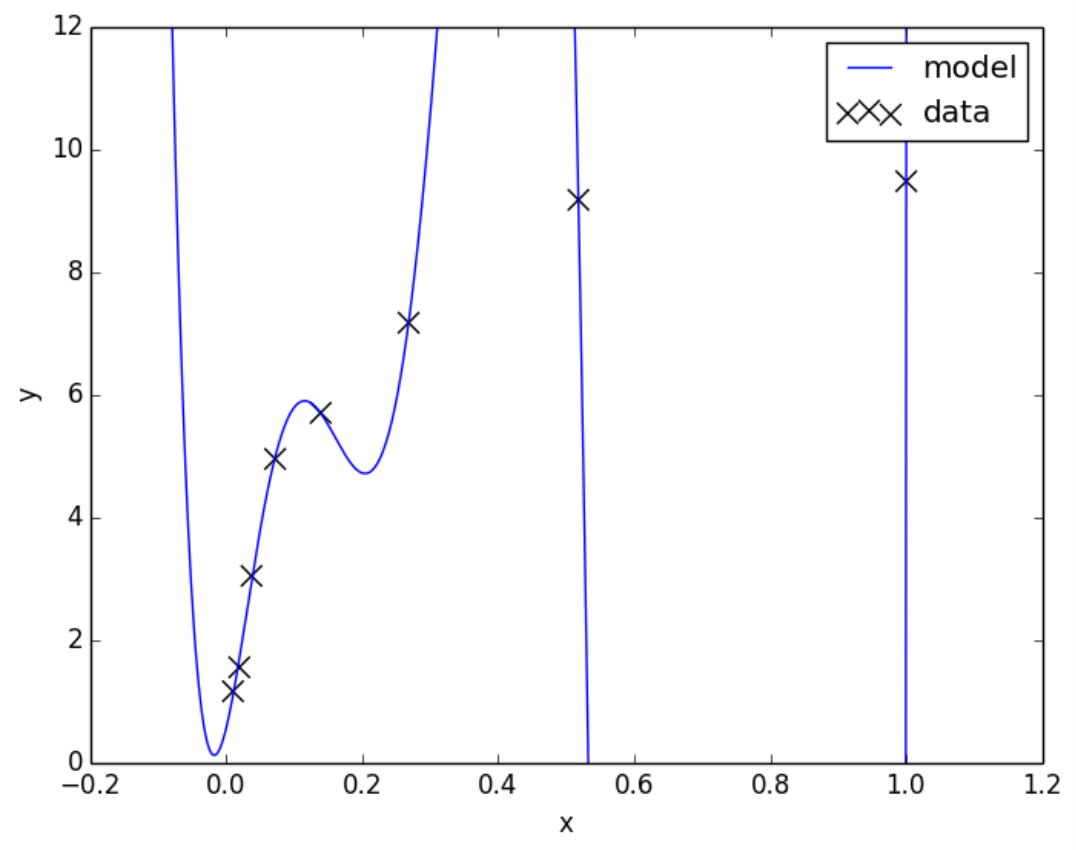
適切にデータを前処理して,
適切なアルゴリズムを選んで分析した.

誤差0.0（回帰）／F値1.0（分類）だ！
完璧なモデルができた！

本当ですか？

分析時の注意点

このモデル（誤差0.0）は未知のデータを正しく予測できるでしょうか？



過学習

過学習(Over fitting)

与えられたデータに（ノイズも含めて）過度に適合してしまい，**訓練誤差**は小さいが，未知データに対する性能が低下してしまう状態.

汎化性能

未知のデータに対する性能（汎化性能）を定量化した**汎化誤差**を小さくすることが重要

表現力の高いアルゴリズム使用時，特徴量が多いとき，与えられたデータが少ないときに過学習しやすい.

過学習対策

モデルを学習する際に，複雑になりすぎないようにパラメータを制御し，過学習を防ぐ

正則化(Regularization)パラメータの調整

リッジ回帰，Lasso，SVMなど

決定木の深さの制御

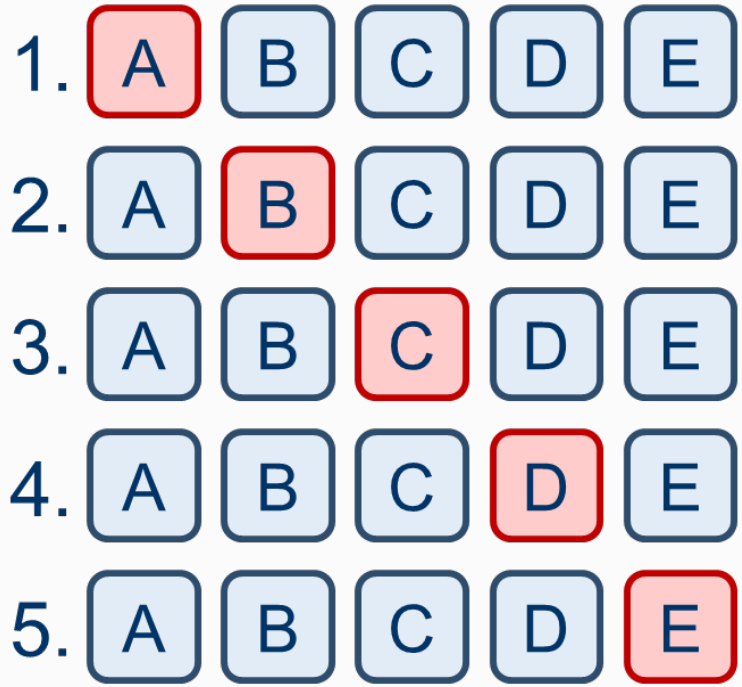
決定木，ランダムフォレストなど

正則化しすぎても性能がでない(Under fitting)

過学習対策

交差検証(Cross validation)

データを学習用と評価用に分割する



1. B～Eで学習，Aで評価
2. A,C～Eで学習，Bで評価
3. A,B,D,Eで学習，Cで評価
4. A～C,Eで学習，Dで評価
5. A～Dで学習，Eで評価
6. １～5の平均を算出

5分割交差検証（5-fold cross validation）

交差検証

未知データの性質を考慮し分割手法を選択

ランダムにK分割

1サンプルとそれ以外に分割 (Leave-one-out cross validation)

ラベルの比率を保ったまま分割 (Stratified cross validation)

ラベルの比率に偏りのある場合に有効

先頭から順にK分割

近傍のデータに関連がある場合

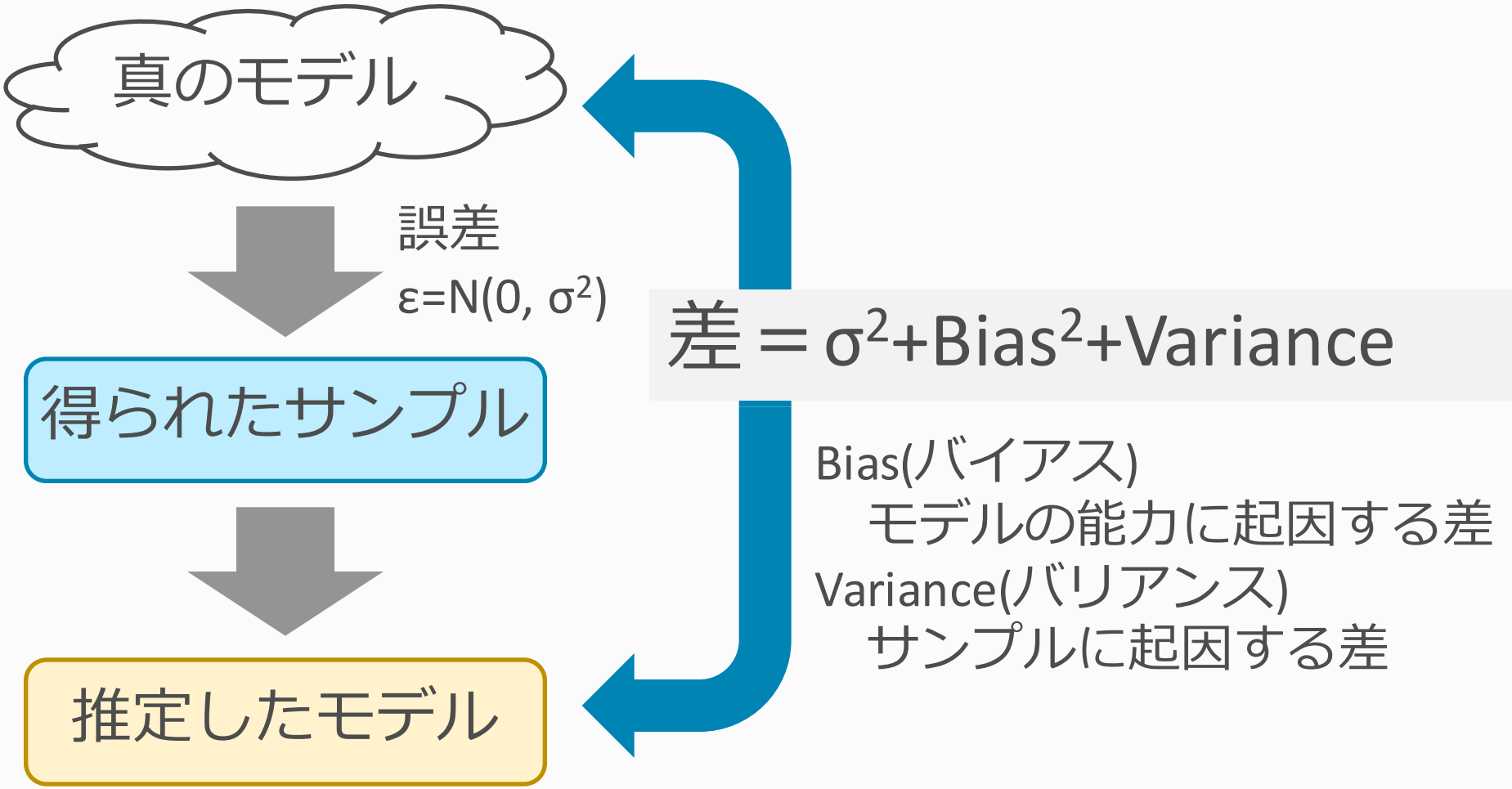
何らかの属性に応じて分割

被験者ごとなど（未知の被験者に対するモデルの性能を評価）

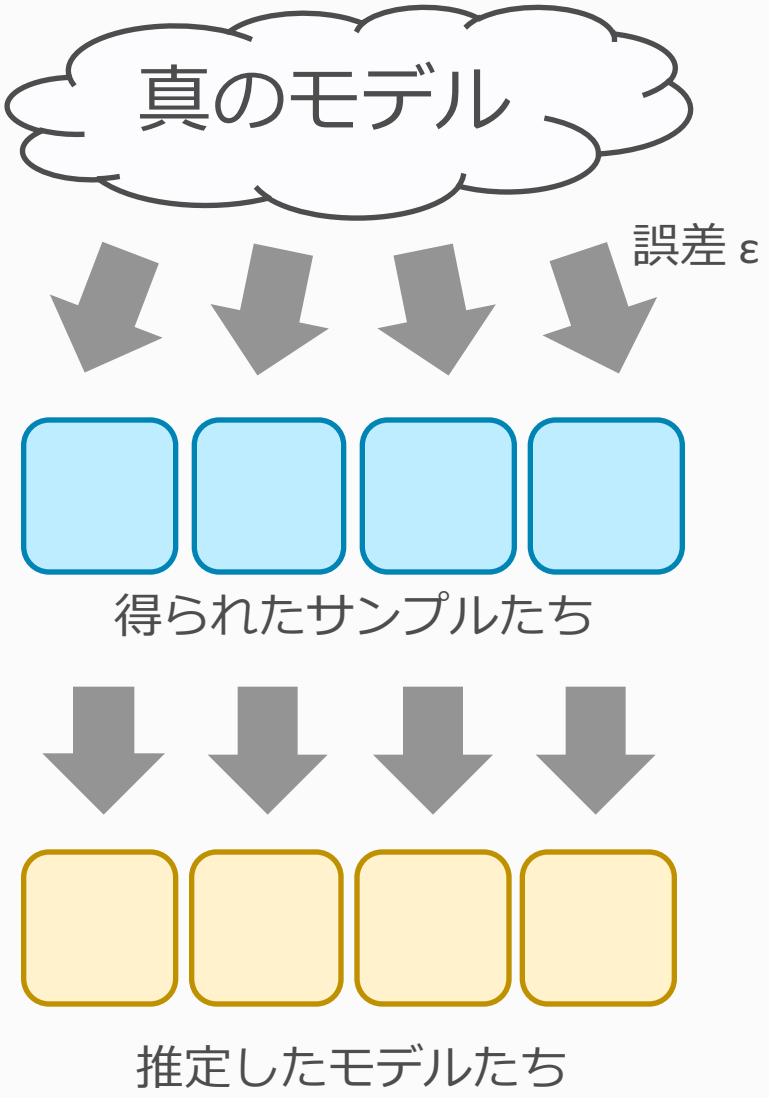
演習

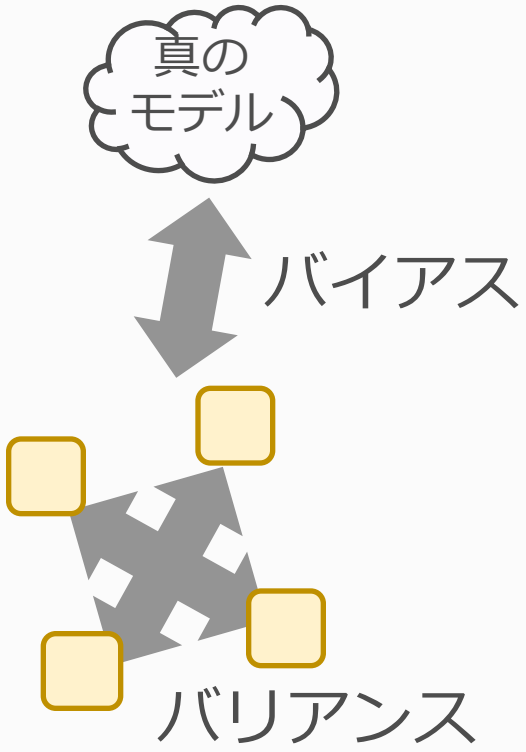
8. グリッドサーチ
9. 交差検証

誤差について


真のモデル
誤差
ε=N(0, σ²)
得られたサンプル
推定したモデル
差＝σ²+Bias²+Variance
Bias(バイアス)
モデルの能力に起因する差
Variance(バリアンス)
サンプルに起因する差

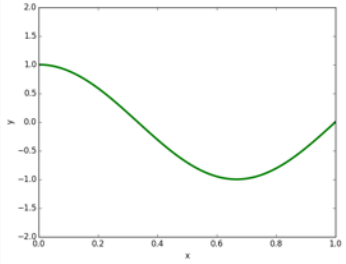
バイアスとバリアンス





バイアスとバリアンス

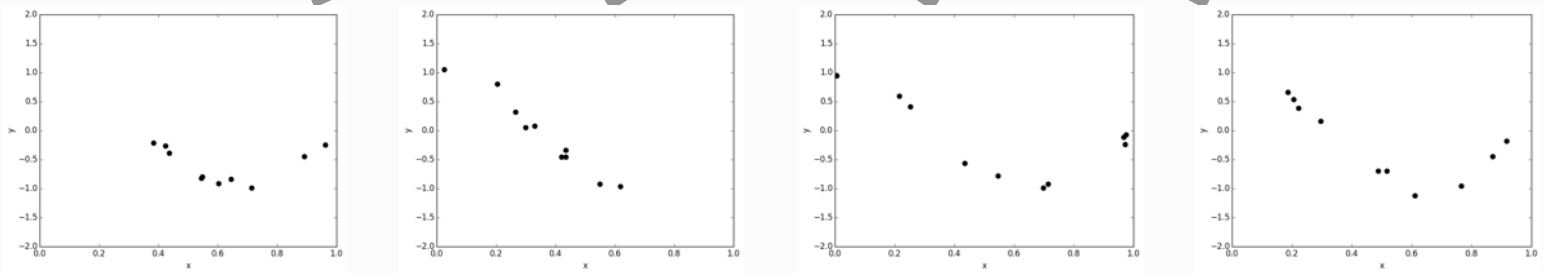
1次式でモデリング



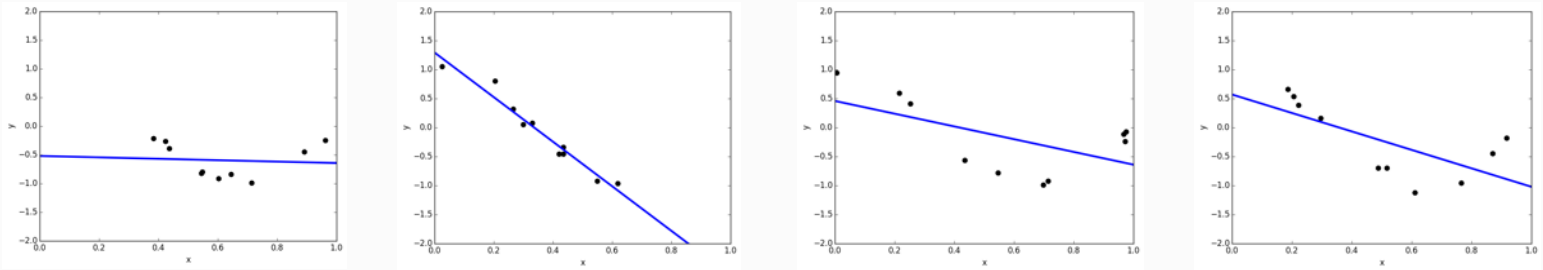
真のモデル

誤差 ε

得られた
サンプルたち



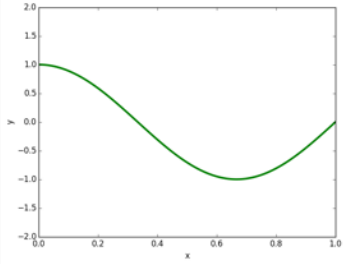
推定した
モデルたち



差は大きいが，差のばらつきは小さい　→　**ハイバイアス／ローバリアンス**

バイアスとバリアンス

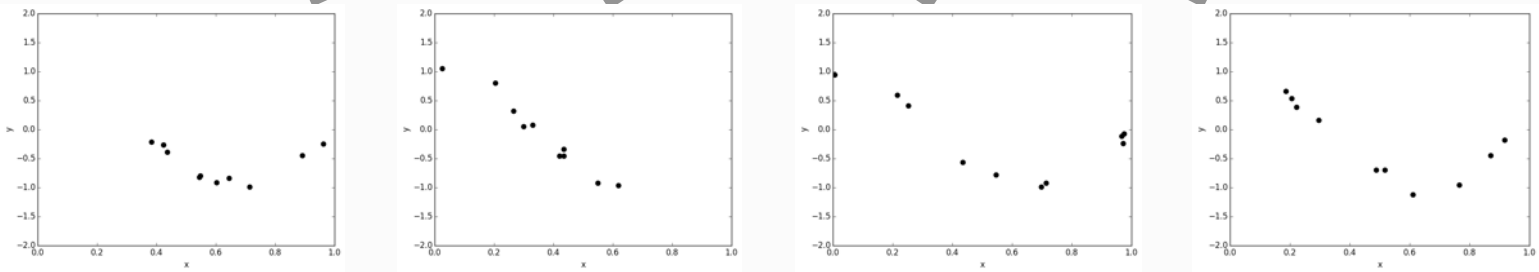
多項式でモデリング



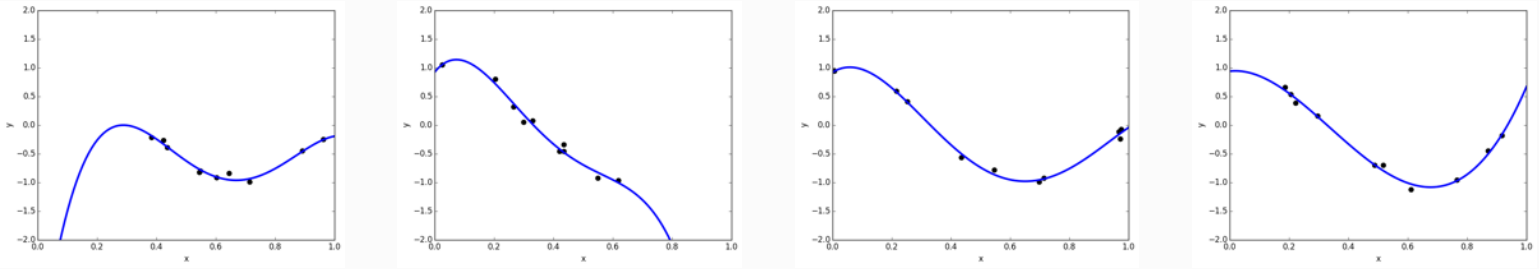
真のモデル

誤差 ε

得られた
サンプルたち



推定した
モデルたち



サンプルによる差が大きい　→　**ローバイアス／ハイバリアンス**

バイアスとバリアンス

バイアスとバリアンスは**トレードオフ**の関係

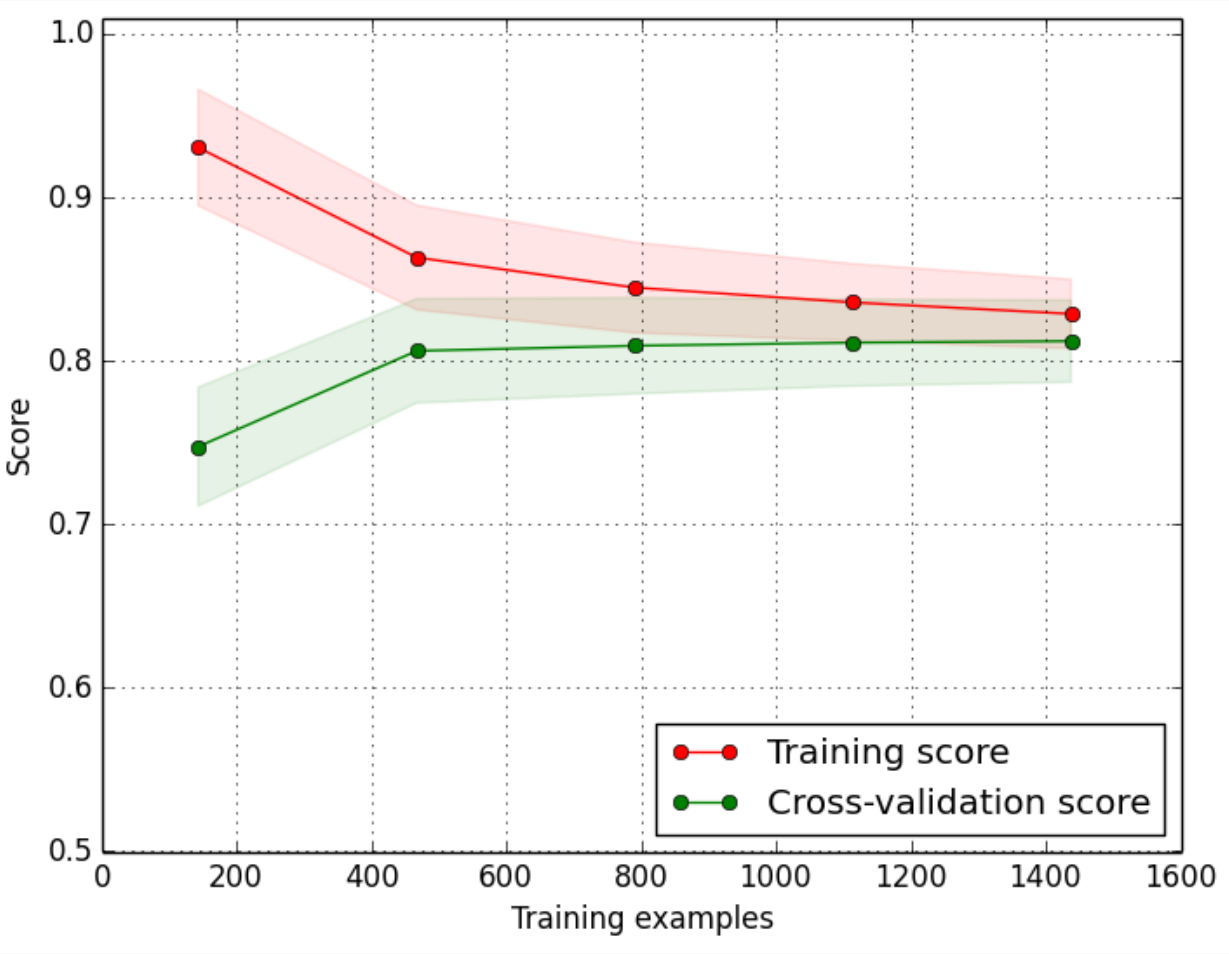
柔軟性の高いモデル（アルゴリズム）
バイアス小，バリアンス大⇒**ハイバリアンス**
過学習(Over fitting)

柔軟性の低いモデル（アルゴリズム）
バイアス大，バリアンス小⇒**ハイバイアス**
Under fitting

現在のモデルの状態を確認するには？

学習曲線

データサイズを変えながら訓練スコア(誤差)
汎化スコア(誤差)をプロット



学習曲線

ハイバイアスの目安

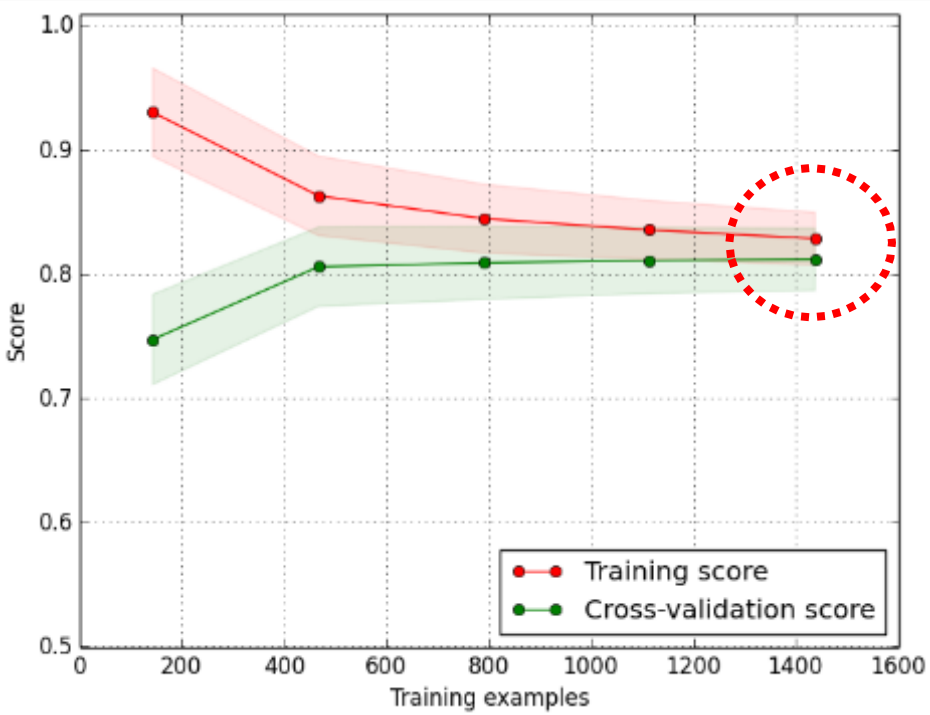
訓練スコア(誤差)が低い(大きい)
訓練スコアと汎化スコアの差が小さい

ハイバリアンスの目安

訓練スコアと汎化スコアの差が大きい
汎化スコアの改善がサチっていない

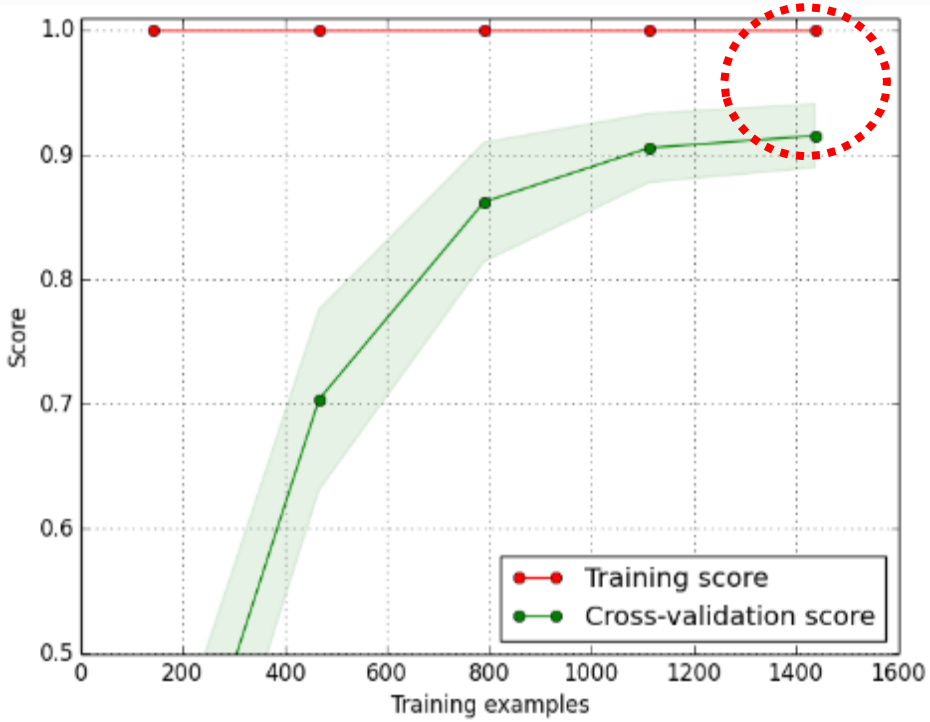
学習曲線

ハイバイアス



スコアが低い
スコアの差が小さい

ハイバリアンス



スコアの差が大きい

学習曲線

ハイバイアスの場合

（有効な）特徴量を増やす

アルゴリズムを（柔軟性の高いものに）変更する

ハイバリアンスの場合

データを増やす

（不要な）特徴量を削除する

演習

10. 学習曲線
11. モデルの選択
12. 特徴量の作成

さらなる性能向上を目指して

アンサンブル学習 (Ensemble learning)

- 複数のモデルの結果を統合
- Stacking／Bagging／Boosting
- データ分析コンペでは必須

Deep learning

- Neural networksの発展
- 特徴量設計が不要なアルゴリズム**ではない**

でも...

The Inconvenient Truth About Data Science

2015年4月26日 | 4,967 | 179 | 21

1. Data is never clean.
2. You will spend most of your time cleaning and preparing data.
3. 95% of tasks do not require deep learning.
4. In 90% of cases generalized linear regression will do the trick.
5. Big Data is just a tool.
6. You should embrace the Bayesian approach.
7. No one cares how you did it.
8. Academia and business are two different worlds.
9. Presentation is key - be a master of Power Point.
10. All models are false, but some are useful.
11. There is no fully automated Data Science. You need to get your hands dirty.

https://www.linkedin.com/pulse/inconvenient-truth-data-science-kamil-bartocha

宣伝

機械学習支援システム MALSS

(Machine Learning Support System)

機械学習によるデータ分析の一部を自動化する
Pythonライブラリ

機能

- ダミー変数生成，欠損値補間，正規化
- アルゴリズム自動選択
- 交差検証，グリッドサーチ
- 分析結果レポート
- サンプルコード生成

宣伝

機械学習支援システム MALSS

インストール

```
> pip install -U malss
```

利用方法

```
> from malss import MALSS
> clf = MALSS('classification‘, lang=‘jp’)
> clf.fit(X, y, ‘report_output_dir')
> clf.make_sample_code('sample_code.py')
```

機械学習支援システム MALSS

レポート

アルゴリズム	交差検証のスコア(f1)
Support Vector Machine (RBF Kernel)	0.849
Random Forest	0.836
Support Vector Machine (Linear Kernel)	0.856
Logistic Regression	0.859
Decision Tree	0.752
k-Nearest Neighbors	0.842

※交差検証のスコア:

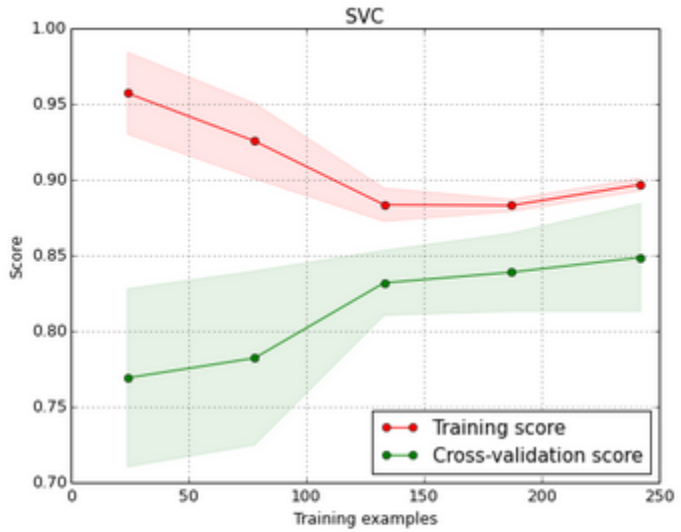
- 機械学習では，学習データに含まれない未知のデータに対して良い結果を出す能力，**汎化能力**が重要となります．
- モデルの学習と評価に同じデータを使うと学習データに過度に適応(過学習)してしまい，汎化能力が低下してしまいます．
- 過学習を防ぐためには交差検証を行い汎化能力を評価します．
  代表的な交差検証法であるK-fold cross validationでは，まずデータセットをK個(default: 5)に分割します．そして，そのうちの1つをテスト用とし，残るK-1個でモデルを学習します．交差検証はK個に分割されたデータそれぞれをテストデータとしてK回検証を行い，得られた結果を平均して1つのスコアを得ます．
- 交差検証は様々な手法が提案されているので，目的に応じて適切な手法を選択してください．
  (デフォルトでは，回帰(regression)タスクでは5-fold cross validationが，分類(classification)タスクではStratified 5-fold cross validationが選択されています．)

※評価基準:

- ラベルに偏りがあり，1%のデータのみが陽性の場合，常に陰性と予測するモデルの精度(accuracy)は99%ですが，このモデルは実用的ではありません．
- モデルの評価基準(scoringオプション)はsklearn.metricsモジュールから適切なものを選択してください．
  (デフォルトでは，回帰(regression)タスクでは平均二乗誤差(mean squared error)が，分類(classification)タスクではF値(f1 score)が選択されています．)

機械学習支援システム MALSS

レポート



学習曲線(Learning curve)

- 学習曲線はデータサイズを変えた時の訓練データでのスコア，交差検証のスコアをプロットしたものです．
- 学習曲線が以下のような場合，モデルは**ハイバリアンス**(オーバーフィッティング(過学習))であると言えます：
  - 学習データ増加に伴う交差検証のスコアの改善が飽和していない(改善し続けている)．
  - 訓練データのスコアと交差検証のスコアの差が大きい．
- 学習曲線が以下のような場合，モデルは**ハイバイアス**(アンダーフィッティング)であると言えます：
  - 訓練データのスコアでさえも悪い．
  - 訓練データのスコアと交差検証のスコアの差が小さい．

参考文献

戦略的データサイエンス入門
F. Provost他／オライリー・ジャパン

Coursera: Machine Learning
Andrew Ng／https://www.coursera.org/course/ml

scikit-learn Tutorials
http://scikit-learn.org/stable/tutorial/

Tutorial: Machine Learning for Astronomy with Scikit-learn
http://www.astroml.org/sklearn_tutorial/

データ解析のための統計モデリング入門
久保 拓弥／岩波書店

データ解析の実務プロセス入門
あんちべ／森北出版

参考文献

MALSS (Machine Learning Support System)

https://pypi.python.org/pypi/malss/

https://github.com/canard0328/malss

Pythonでの機械学習を支援するツール MALSS（導入）

Qiita／http://qiita.com/canard0328/items/fe1ccd5721d59d76cc77

Pythonでの機械学習を支援するツール MALSS（基本）

Qiita／http://qiita.com/canard0328/items/5da95ff4f2e1611f87e1

Pythonでの機械学習を支援するツール MALSS（応用）

Qiita／ http://qiita.com/canard0328/items/3713d6758fe9c045a19d

本日お話したこと

1. 分析プロセス
   SEMMA，CRISP-DM，KDD，KKD
2. データの探索・前処理
   ダミー変数，次元の呪い，標準化，醜いアヒルの子定理
3. モデリング
   教師あり学習，ノーフリーランチ定理
4. 評価
   混同行列，過学習，交差検証，学習曲線，バイアス・バリアンス