第36回 CMDワークショップ
マテリアルズ・インフォマティクス・コース
初級コース

国立研究開発法人 物質・材料研究機構
小山 幸典

実習について

- 実習用のテキストおよびデータは下記のサイトからダウンロードしてください
  - https://bitbucket.org/koyama_y/mi_tutorial-2020feb/downloads/ (“リポジトリをダウンロードする” からダウンロード)
  - CMDワークショップ終了後はダウンロードできなくなります
- 実習は, 実習用プログラムをデモンストレーションし,
  重要なポイントを解説する方式で行います
- レベルアップのためには自習が重要ですので, 例題を改造してみましょう

マテリアルズ・
インフォマティクス概論

マテリアルズ・インフォマティクスとは？

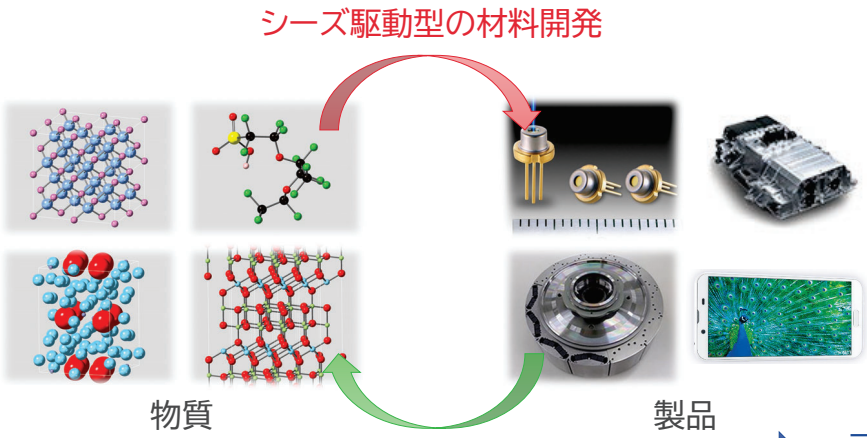
マテリアルズ・インフォマティクス ＝ **材料研究** ✖ **データ科学**

統計, 機械学習, 最適化,
ベイズ推論, データ同化, …

- 組成・構造 ➔ 材料インフォマティクス
- 合成 ➔ プロセスインフォマティクス
- 分析 ➔ 計測インフォマティクス

…

材料開発の方向



研究者の思考パターンを考えてみる

論理的推論

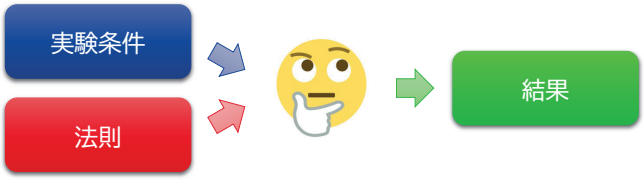
『前提条件』,『規則』,『結論』のうち
2つが与えられた場合に, 残りの1つを推定する

『演繹』,『帰納』,『アブダクション』の3つのパターンがある

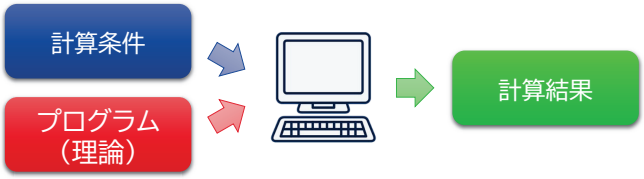
演繹

「前提条件」と「規則」から「結論」を導く

思考実験



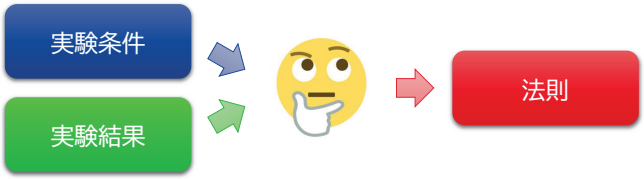
理論計算
（シミュレーション）



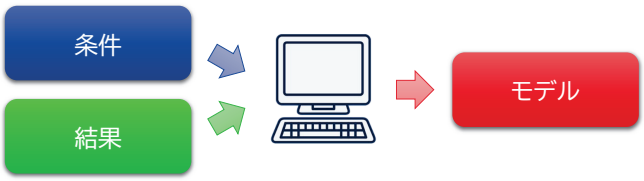
帰納

「前提条件」と「結論」から「規則」を推定する

考察

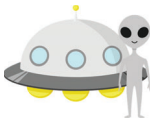


機械学習

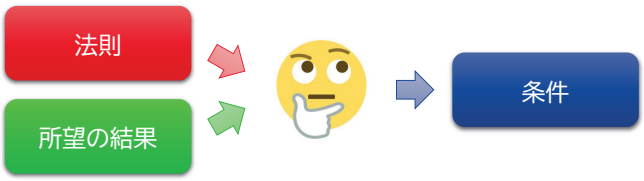


アブダクション

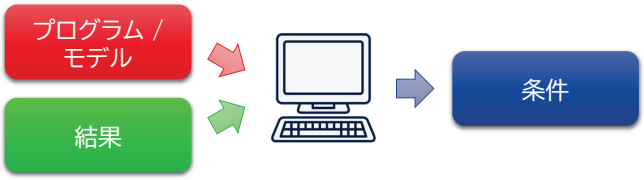
「規則」と「結論」から「前提条件」を推定する



設計



探索・最適化

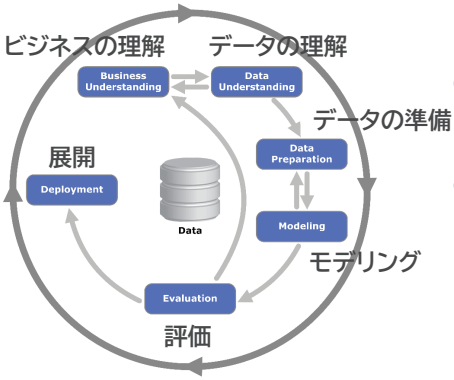


推論の注意点

- 演繹は「正しい答え」を導くことができる
  - ただし，直面している問題に対して演繹が可能かどうかは別問題
- 帰納とアブダクションは推定であり，「正しい答え」であるという保証はない
  - 確からしさは議論できるが，絶対に正しいとは言えない
  - データ科学は，主に帰納やアブダクションの文脈で用いられる
- 帰納とアブダクションの違いでわかるように
  モデルを「作る」と「使う」は全く別の行為である

データマイニングにおける分析手順

Cross-industry standard process for data mining（CRISP-DM）



https://en.wikipedia.org/wiki/cross-industry_standard_process_for_data_mining

- 単純に一方向に進むことはほとんどなく
  必要に応じて行ったり来たりを繰り返す
- 分析を一度実施すれば終了ということはなく
  運用で得られた結果を次の分析につなげる

  処理の分け方は様々なので，適宜読み替える

ビジネスの理解（Business Understanding）

『ビジネス』＝ あなたの仕事

「問題」を明確にする

- 解決されるべき問題は何か？
- 問題解決の目標水準は？
- 解決策の案は？（仮説）
- 問題解決に許されるコストは？

ワークフローの中で最も重要であり
これが不十分だと収集すべきデータや
作成すべきモデルを決定できない

問題を当事者間で共有することも重要

データの理解（Data Understanding）

ここでいう『データ』は，個々の項目ではなく **データ全体（データセット）**

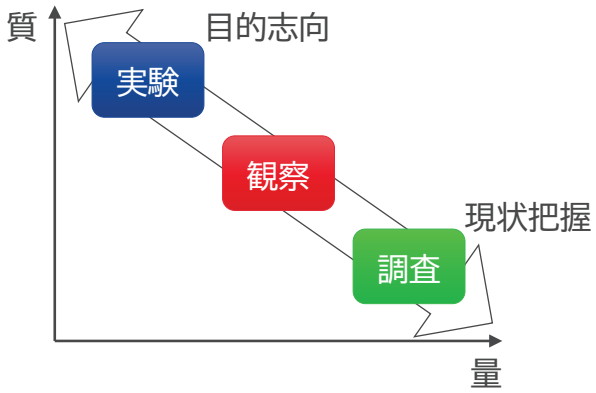
データセットの特徴を理解する

- データの母集団は？
- データは(再)取得可能か？ コストは？
- データの信頼性は？
- データは整っているか？
- データはラベル付きか？
- データの依存性は？

データ分析やモデリングでは「過去のデータ」を使用するため
実際にモデルを使用する状況を考慮することが重要

データが既にある場合とこれから取得する場合でとりうる手段は異なる

データ収集の目的



広範な領域をカバーする(汎用の)データベースと
特定目的の(領域が限られた)データベースは本質的に別物

データの準備（Data Preparation）

データをモデリングに利用可能な状態にする

- データの収集
- 形式の変換
- 欠損の処理
- 異常値の処理
- 属性の加工・選択

⇔

データの関係把握 **『探索的データ分析』**
集計，可視化

問題によっては，モデリングせずとも
データ分析だけで解決する場合もある

モデリング（Modeling）

モデル(関係を表すパターン)を作成する（機械学習）

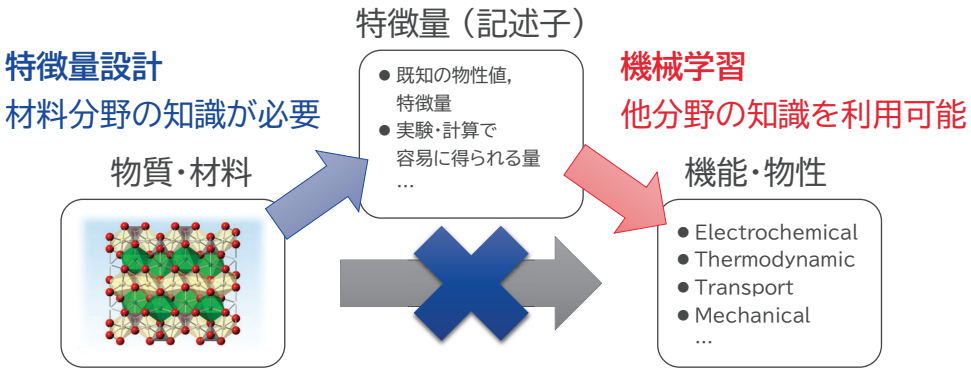
- モデルの構築
- モデルの評価
- モデルの最適化・選択

⇔ 結果の把握
モデリング結果の集計，可視化

目的属性が明示的で，「過去のデータ」でも得られている

➡『教師あり学習』

特徴量（記述子）



「物質・材料」を数学的に取り扱うことは困難

特徴量(記述子)と呼ばれる数値表現に変換する

評価（Evaluation）

ビジネス問題を解決する視点で検証する

- 実際の状況でのモデルの予測性能は十分か？
- 予測のための評価時間は妥当か？
- 実際の状況でのデータの取得コストは妥当か？
- モデルは実際の状況でのデータ量に対応可能か？
- 異常なデータへの対応は適切か？

『ビジネスの理解』,『データの理解』が適切に行なわれていれば
大半の項目は問題ないはずだが,『展開』する前に検証しておくべきである

展開（Deployment）

モデルを『ビジネス』で利用するときの運用上の課題

- モデルの性能を追跡可能か？
- モデルの更新は可能か？（アルゴリズム, 学習結果）
- 想定外の結果に対する対応策は？

- 「過去のデータ」≒「将来のデータ」とは限らない
- 「異常値」の想定は適切か？
- 運用時にモデルを更新する場合, 不適切なモデルとなる可能性がある

モデルに対する考え方

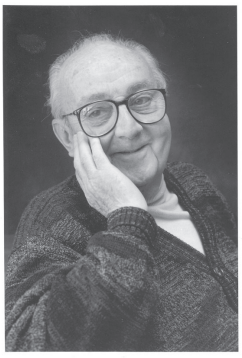
Q. どこまで検討すれば, 正しい答えと判断してよいか？

A. All models are wrong, but some are useful.

「全てのモデルは正しくないが,
中には役に立つものもある」

例えば, 理想気体の状態方程式（PV = RT）は
実在気体に対しては厳密には正しくないが, 有用な近似を与える

現実の問題では「正しい答え」はないので,
「役に立つ」と判断できる水準を定めておく

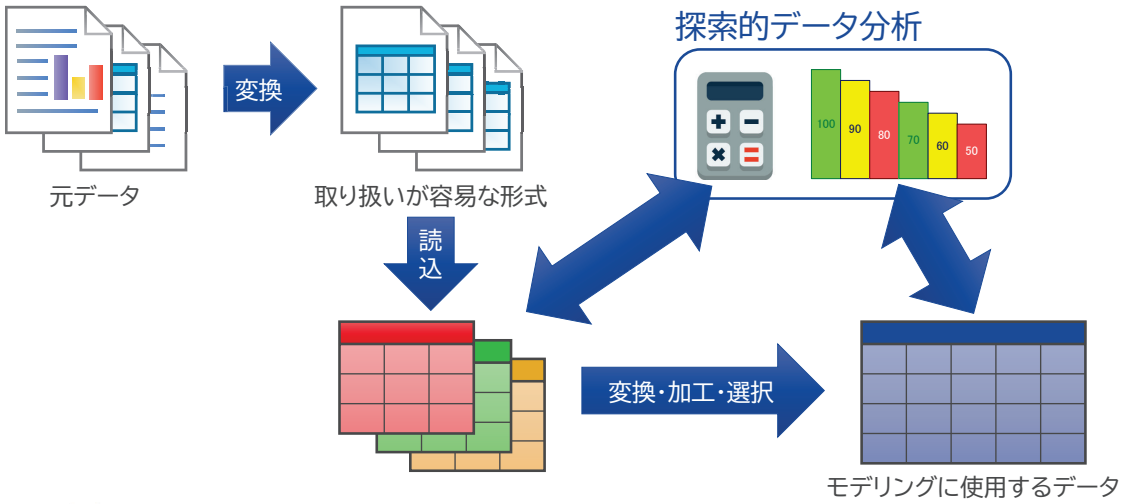


George Box

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

データの前処理

データ前処理の流れ



データの性質

- データの尺度水準［使用可能な比較・演算］
  - 名義尺度【最下位】: 単なるラベル［=, ≠］
  - 順序尺度: ラベルに順序がある［<, >］
    - 順位, モース硬度, 震度
  - 間隔尺度: 順序の差が等しい［+, −］
    - 時刻, 摂氏温度
  - 比率尺度【最上位】: 乗除演算が可能（0が絶対的）［×, ÷］
- 上位水準の尺度は, より下位水準の尺度の性質を含む
- 実際のデータ処理では, カテゴリ変数か連続変数かのみの区別が多い

データの形式

- データ分析やモデリングでは, 基本的に数値(ベクトル)を用いる
- 文字列など非数値データは数値表現に変換することが定石
  - 非数値データを直接取り扱うモデリング手法もあるが,
    実質的には数値表現を経由している場合が大半である
- 時刻は数値で表現することは容易であるが,
  日付や曜日などを考慮する必要があると取り扱いは難しくなる
- カテゴリ変数は, カテゴリに「含まれる」「含まれない」を意味する
  {1, 0} の二値で表してモデリングすることが多い（ダミー変数）
  - モデリングに入る直前に変換する

ファイル形式の変換

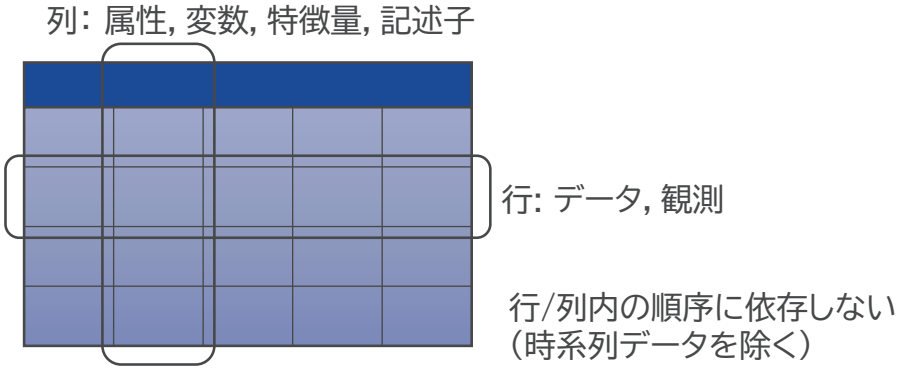
- 元データのファイルは様々
  - Excel, PDF, データベース, 実験装置の独自形式, …
- 元データを取り扱いが容易な形式に変換し,
  以後は「変換された元データ」を元データに準ずるデータとして使用する
  - 変換は自動化できることが望ましいが,
    人力でしなければならないことも多い
  - 元データが変更されない限り,「変換された元データ」は変更しない
  - 形式はCSVなどが汎用的でよいが,
    Excelや分析ソフトウェアの独自形式でもよい場合はある

表形式の変換

- 基本的に「整理されたデータ (tidy data)」となるようにする
- 表の形式としてワイド形式とロング形式がある
  - 何を一つの「観測」とするかは問題やモデリング次第
- 最初は探索的データ分析に主眼を置いて整理された形式に変換し,
  最後にモデリングに適した形式に変換する

表形式のデータ

基本的なデータ分析・モデリングでは, 表形式のデータを使用する

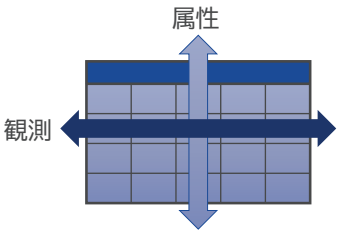


整理されたデータ（tidy data）

『**整理されたデータ（tidy data）**』データ分析に適したデータ表現形式の考え方

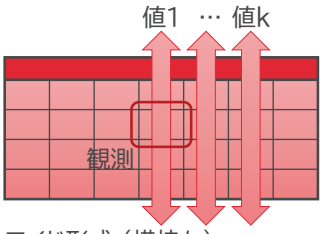
Hadley Wickham, “Tidy Data,” J. Statistical Software, 59, 10 (2014)

整理されたデータ

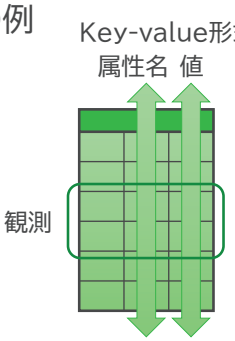


- 各属性に専用の列がある
- 各観測に専用の行がある

整理されていないデータの例



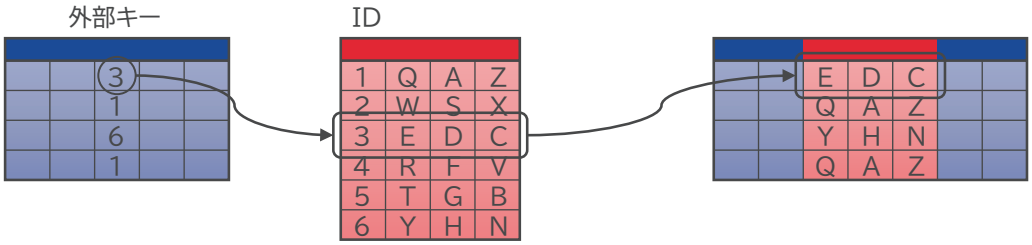
ワイド形式（横持ち）



ロング形式（縦持ち）

表の集約

- データセットが複数の表から構成される場合，集約して一つの表にする
- 関係データベース(RDB)におけるJOIN



データの選択

- データ分析やモデリングに不要なデータを削除する
  - 観測(行)の選択
  - 属性(列)の選択
- 当該分野の知識から無意味と判断される属性は除外する方がよいが，関係が不確かな属性は残す方がよい

欠損データ

- 欠損データの取り扱い
  - 観測を削除
  - 属性を削除
  - データを推定（平均値などの代表値, 多重代入法, など）
  - 「欠損」を明示化
- 何が「欠損」か注意が必要
  - 項目が空白, 空文字列との区別が必要な場合もある
  - 欠損を表す数値（例: 非負データに対して−1, 非常に大きな値）
  - 時系列データの場合, 観測そのものが欠落している場合もある

異常値

- 異常値(外れ値)を含むとモデルの性能が低下することがある
  - 通常のデータに興味がある場合は異常値を除外することが選択肢となる
  - 極端なデータに興味がある場合は異常値そのものが議論となる
  - 異常値がデータ欠損など他の意味を持つ場合もある
- 何が「異常値」かは問題による
  - 異常値(外れ値)を機械的に判定することは困難

データの加工

- 1属性の変換
  - スケールの変換（線形変換）
  - 非線形変換（対数, 逆数, …）
  - 離散化
- 2属性(以上)の変換
  - 合成（和, 差, 積, 商, …）
  - 高次項, 交互作用
  - 多変量分析（主成分分析, …）

探索的データ分析

- データがもつ特徴を理解する
  - 要約統計量
    - 平均, 分散, 範囲, …
  - 分布
    - 棒グラフ, ヒストグラム, 密度図, 箱ひげ図, …
  - 属性同士の関係
    - 相関係数, 分布図, ヒートマップ, …
- モデルによっては変数の分布に暗黙の仮定がある場合がある
  - データの特徴を理解してデータを加工する

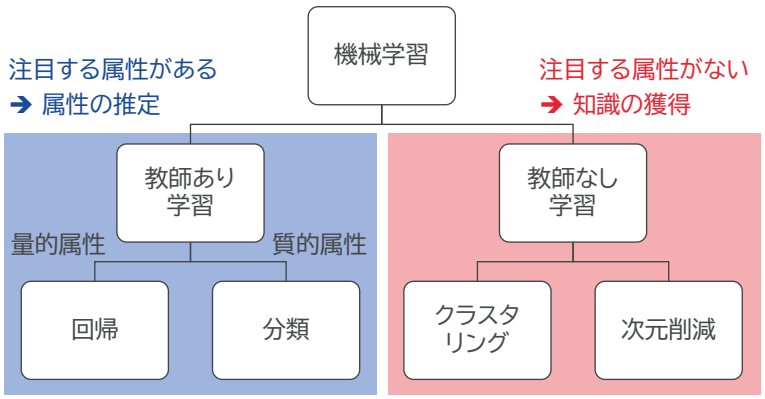
可視化のための次元削減

- データの属性が多い場合に,「上手く」可視化するために次元を削減する
  (多くの場合は2次元, ないし, 3次元)
  - 可視化による理解を優先しているため,
    得られた値(座標)に意味が無い場合がある
  - 学習するごとに結果が変わることがしばしばある
- 代表的な手法
  - 主成分分析(PCA)
  - Isomap
  - t-SNE
  - 多様体学習(manifold learning)

モデリング・機械学習

機械学習

機械学習: データからパターンを発見する統計的手法

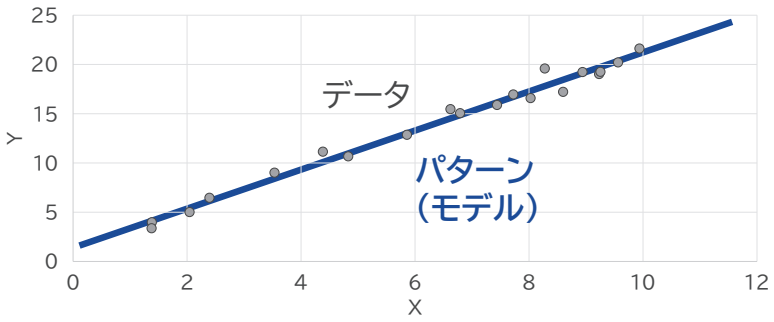


本コースでは「教師あり学習」に限定して解説

最小二乗法

データ

X	Y
8.027	16.587
2.392	6.471
4.383	11.151
6.790	15.058
8.277	19.592
1.381	3.985
9.228	19.029
4.827	10.674
…	…



線形関数を用いて
誤差の二乗和が最小となるように
その係数を決定する

モデル
損失関数（性能）
学習

用語

- 目的属性が量的変数(数値)の場合は回帰,
  質的変数(カテゴリ)の場合は分類という
- 回帰や分類の入力に使用する変数・属性を
  説明変数, 独立変数などという
- 回帰や分類で出力される変数・属性を
  目的変数, 従属変数, 応答変数などという
- 記述子や特徴量という言葉もよく使われるが,
  人によってニュアンスが違うことがあるので注意が必要
  (入力変数のみに使う, 入力・出力に関係なく使う, など)

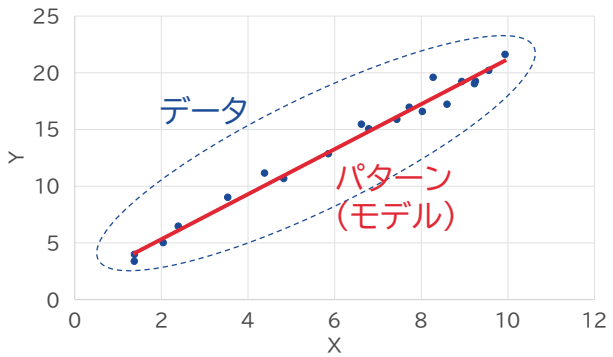
機械学習の注意点

- 機械学習の性能は，特に教師あり学習の場合，
  「既知のデータへの当てはまりの良さ」ではなく
  「未知のデータにどれくらい良く当てはまるか」（『汎化性能』）
  で評価すべきである　　**適切な評価手順がある**
- 機械学習で取り扱うのはデータに表れている相関であり
  自然法則の因果ではない
- 機械学習の性能と理解のしやすさの両立は，一般に困難である

モデル

- 「モデル」という言葉は曖昧である
  - パターンの大まかな構造（線形関係）
  - 具体的な数字を含むパターンの構造（$y = 1.425 + 1.983 \times x$）
- モデルには2通りのアプローチがある
  - データを再現するような関数フィッティング
    - $y = f(x;\ \theta)$ に対して，データに合うように θ を決定する
    - 中には関数形が自明でないものもある（『ノンパラメトリックモデル』）
    - フィッティングの目的関数（損失関数）は手法ごとに決められている
  - 確率論に基づくデータの生成モデル
    - y は，平均 $a + bx$ ，分散 σ^2 の正規分布に従う

線形モデル

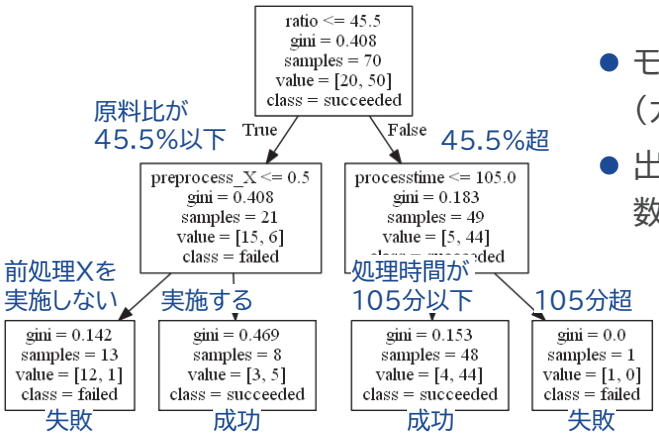


- $y = a + bx$（単回帰）
- $y = a + bx_1 + cx_2$（重回帰）
- $y = a + bx + cx^2$（多項式）
  $(x_1, x_2) = (x, x^2)$ と見なせば
  重回帰と同じ
- $y = a + b\sin x + c\cos x$
  $(x_1, x_2) = (\sin x\,, \cos x)$

パラメータ（係数 $a, b, c, \cdots$）に対して線形である

$y = a + b\sin(\kappa x)$ は κ に対して線形でないため，線形モデルではない

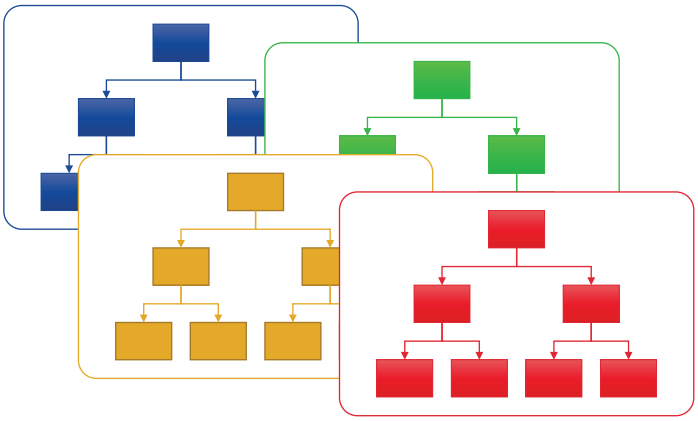
木モデル



- モデルの柔軟性が高い
  （カテゴリ変数の取り扱いも容易）
- 出力がカテゴリ変数の場合は決定木，
  数値の場合は回帰木と呼ばれる

If-then-else ルールで出力を決定

アンサンブルモデル

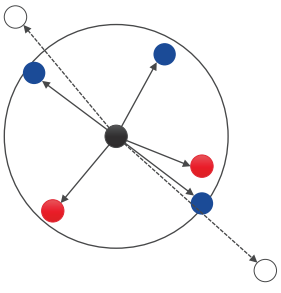


- 基本学習器には
  木モデルがよく使用される
- 基本学習器をそれぞれ
  どのように学習させるかで
  様々な手法がある

複数の学習器を組み合わせて一つの機械学習モデルとする

k近傍法

- 近傍の測定点の多数決より，未測定の値を予測する手法
  - 分類問題において用いられる

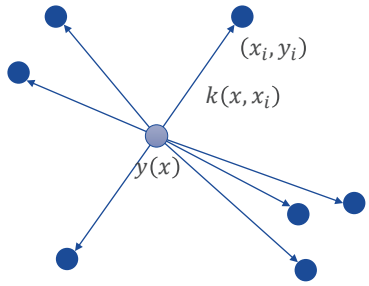


●の近傍5点（k = 5）は，●が3点，●が2点

●は●であると予測

カーネル法

- 数学的には，高次元空間へ射影した上で線型モデルを適用することに相当する
- 直感的には，測定点の加重平均により，未測定の値を推定する
  - 測定点同士の類似度を，カーネルという関数で与える

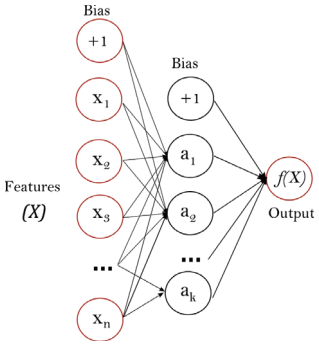


$$y(x) = \sum_i w(x, x_i) y_i$$

$w(x, x_i)$ は $k(x, x_i)$ と $k(x_i, x_j)$ から計算

ニューラルネットワーク

- 比較的単純な非線形関数を積み重ねることで非線形な関数を表現
  - 積層数が多い深層学習は最近のホットな分野



- $y = f(z)$
- $z = g(v)$
- …
- $u = h(x)$

$$y = f\left(g\left(\cdots\left(h(x)\right)\cdots\right)\right) = F(x)$$

http://scikit-learn.org/stable/modules/neural_networks_supervised.html より引用

未学習・過学習

- 機械学習モデルの性能は
  「既知のデータへの当てはまりの良さ」ではなく
  「未知のデータにどれくらい良く当てはまるか」で評価すべきである
- 機械学習モデルの柔軟性が低い場合
  - データの本質的なパターンを高い精度で表現することが困難
    ➔『バイアス』が大きい
  - 観測誤差やデータの偶然的な偏りに対して頑強
    ➔『バリアンス』が小さい
  - データを増やしても性能を向上させることが難しい
    ➔『未学習（underfitting）』を起こしやすい

未学習・過学習

- 機械学習モデルの柔軟性が高い場合
  - データの本質的なパターンを高い精度で表現することが可能
    ➔『バイアス』が小さい
  - データ数が少ないと観測誤差やデータの偶然的な偏りに適合してしまう
    ➔『バリアンス』が大きい
  - 観測誤差や偶然的な偏りへの適合は，未知のデータに対しては意味がない
    ➔『過学習（overfitting）』を起こしやすい
- 過学習を避けて妥当なモデルを得るために
  モデルに対して制約（複雑度に対するペナルティ）を加える ➔『正則化』

過学習の抑制・正則化

- 損失関数にパラメータに対するペナルティ項を追加する
  - リッジ回帰（L2正則化）
  - LASSO（L1正則化）
- モデルの大きさ（柔軟性）を制限する
  - 木モデルにおける木の深さ
  - ニューラルネットワークの構造
- 増分的な学習において，繰り返しを制限する
  - ニューラルネットワークの早期打ち切り

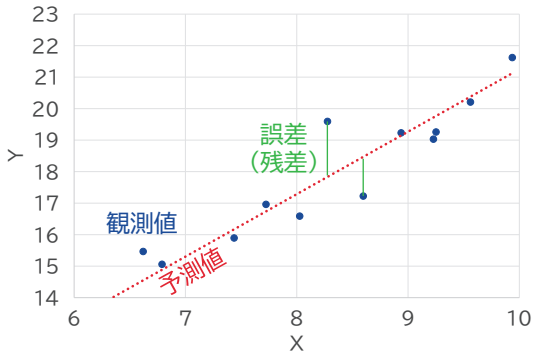
評価指標

評価指標

- 「良い」モデルを選択するためには
  モデルの「良さ」を定量的に表す必要がある ➔ 評価指標
- 複数の評価指標を用いてよいが,
  指標によって最良のモデルが食い違うことがある
  - 最終判断に用いる評価指標は目的に合わせる ➔「ビジネスの理解」
- 評価指標は, 同じデータセットについてモデルを比較するものであり,
  異なるデータセット間で比較するものではない
- 観測誤差など制御不能な因子が含まれる場合,
  それ以上の精度を得ることはできない

回帰の評価指標: (R)MSE

- (Root) Mean Squared Error
  - 誤差の2乗の平均値(の平方根)
- 0以上の値であり, 小さいほどよい
- 誤差の程度を表す代表的な指標
- 統計学的に好ましい性質を持つ
  (微分可能, 分散)
- データ毎の重要度を考慮した
  重み付き(R)MSEというものもある
- 外れ値に弱い



回帰の評価指標: MAE

- Mean Absolute Error
  - 誤差の絶対値の平均値
- 0以上の値であり, 小さいほどよい
- MAEの方がRMSEよりも数値が小さくなる
- RMSEより外れ値に強い
- Median Absolute Error (誤差絶対値の中央値)
  と略語が同じなので注意する
  - 中央値は外れ値に強い

回帰の評価指標: R^2値（決定係数）

- $R^2 = 1 - MSE \div Y$の分散（scikit-learnでの定義）
  - 負の値もとりうる
- スケールを調整した無次元の指標である
- 1以下の値であり，大きいほど(1に近いほど)よい

分類の評価指標: 混同行列と精度

- 混同行列（confusion matrix）
  - 予測したクラスと実際のクラスを表にしたもの（指標ではない）
- 精度（accuracy）
  - 全体の中で予測が正しかった割合（正答率）
  - (17 + 15 + 20 + 18) / 100 = 0.70

予測クラス

真のクラス	A	B	C	D	計
A	17	3	1	4	25
B	5	15	2	3	25
C	2	1	20	2	25
D	3	2	2	18	25
計	27	21	25	27	100

分類の評価指標: 適合率，再現率

- 2値分類の多くの問題では，2つのクラスは同等でないことが多い
  - 「興味がある」クラスを正例（positive），そうでないクラスを負例（negative）とすることが多い
  - クラスが同等でない場合，単純な精度は評価指標として適切ではない
  - 正例・負例の数のバランスがとれていないデータでは特に重要
- 適合率（precision）
  - 正例と予測した中で，実際に正例の割合
- 再現率（recall）
  - 実際に正例の中で，正例と予測できた割合

分類の評価指標: 適合率, 再現率

- 精度: A = (TP + TN) / (TP + FP + FN + TN)
  = (28 + 63) / 100 = 0.91
- 適合率: P = TP / (TP + FP)
  = 28 / 35 = 0.80
- 再現率: R = TP / (TP + FN)
  = 28 / 30 = 0.93

FP (偽陽性): 誤報
FN (偽陰性): 見落とし　　に相当する

「FP」と「FN」のラベルは人によって違う場合があるので注意が必要

予測クラス

真のクラス	正 (P)	負 (N)	計
正 (P)	TP 28	FN 2	30
負 (N)	FP 7	TN 63	70
計	35	65	100

T : True (正答)
F : False (誤答)

分類の評価指標: F値

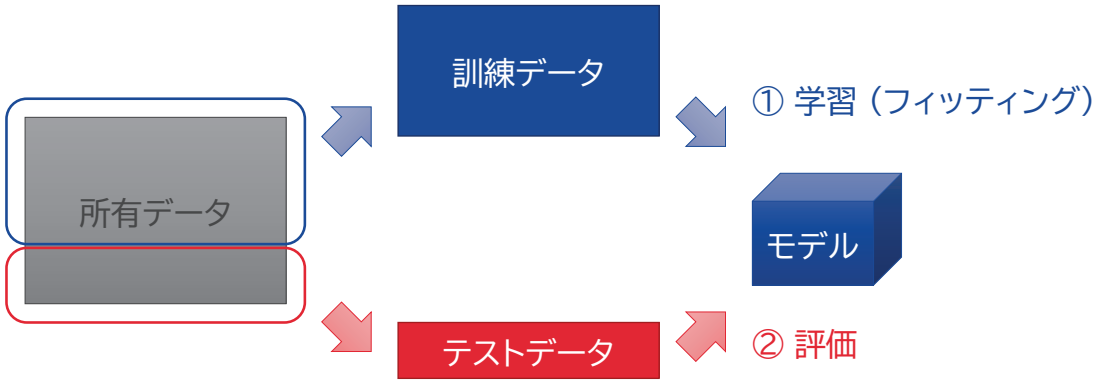
- 適合率と再現率はトレードオフの関係にある
  - クラスを判定する閾値を調整可能なモデルは多い
  - 適合率と再現率のどちらが重要かは『ビジネス』による
- F値
  - 適合率と再現率の調和平均:
    F値 = 2×適合率×再現率 / (適合率+再現率)
  - F = 2×0.80×0.93 / (0.80 + 0.93) = 0.86
  - 適合率と再現率を総合した評価指標の代表例

モデリングの手順

訓練データとテストデータ

- 機械学習モデルの性能は
  「既知のデータへの当てはまりの良さ」ではなく
  「未知のデータにどれくらい良く当てはまるか」で評価すべきである
- 現実には未知データを評価することは不可能であるため,
  モデリングの手順を工夫して推定する
- 所有するデータを
  「機械学習モデルを構築するためのデータ」(訓練データ, 開発データなど) と
  「モデルの良さを評価するためのデータ」(テストデータ, 検証データなど)
  に分割することが原則

訓練データとテストデータ

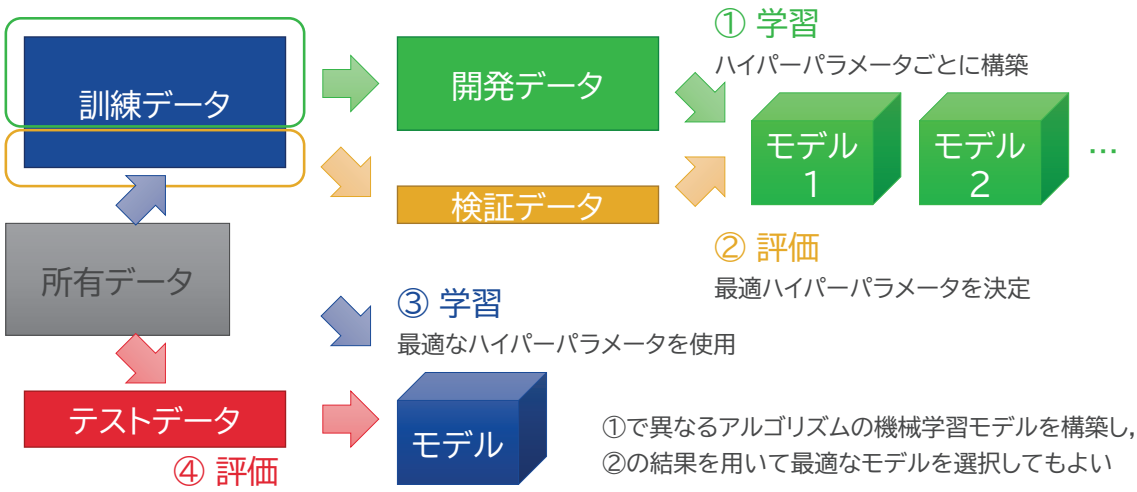


テストデータは最終評価まで使用しない

パラメータの決定

- 機械学習のパラメータには, 学習(フィッティング)で求めることができるものとできないものがある
- 学習で求めることができるパラメータ
  - 線形モデルの係数, 木モデルの分割基準, …
    (モデルを具体化するためのパラメータ)
- 学習で求めることができないパラメータ (『ハイパーパラメータ』)
  - 正則化項の係数, 木の深さ, 多項式の次数, …
    (モデルの形を決めるパラメータや学習に関するパラメータ)
  - 汎化性能を最適化するように決定する必要があるが
    テストデータを使用してはいけない

ハイパーパラメータ決定を含むモデリング手順



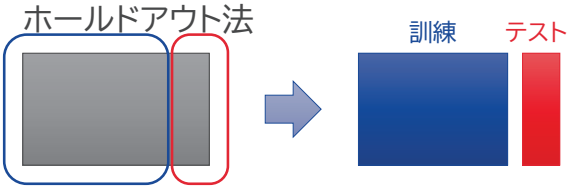
①で異なるアルゴリズムの機械学習モデルを構築し，
②の結果を用いて最適なモデルを選択してもよい

データの分割方法

- ホールドアウト（hold-out）法
  - 訓練データとテストデータへの分割を1回だけ行なう
  - 訓練データとテストデータは
    それぞれがデータ全体の特徴をカバーしていることが望ましい
- 交差検証（cross validation）法
  - データをkセットに分割し，
    各々のデータセットをテストデータとするk回の学習とテストを行なう
  - 分割数がデータ数と同一の場合(テストデータが1個)，
    1個抜き交差検証（leave-one-out CV, LOOCV）という

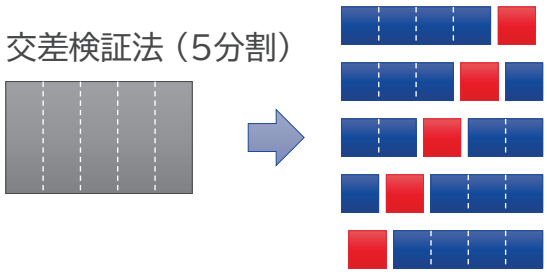
データの分割方法

ホールドアウト法



データ数が十分でないと
評価された指標にバラツキが生じる

交差検証法（5分割）



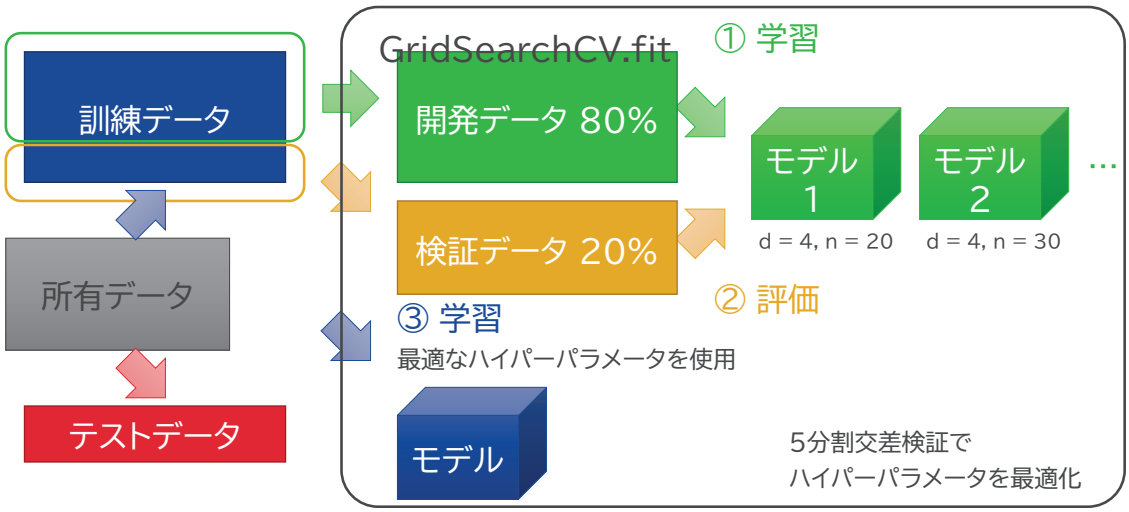
各分割ごとに指標を評価し，
その平均値で汎化性能を推定する

ホールドアウト法ほどではないが，
データ分割によるバラツキがある

学習と汎化性能の評価

- データ数が十分に多くない場合，
  ハイパーパラメータを決定するために交差検証法を用いることが多い
  - 交差検証法では学習を繰り返すために時間がかかるため，
    データ数が十分に多ければホールドアウト法を用いることもある
  - ハイパーパラメータの探索手法はいろいろあるが，
    パラメータ数が少なければ網羅探索(グリッド探索)が簡便
- 最終評価するためのテストデータはホールドアウト法で隔離しておき，
  モデル構築中は手を付けない
  - 検証データで評価した結果で汎化性能を推定することもできるが，
    モデル選択により推定にバイアスが生じていることに注意が必要である

Scikit-learn GridSearchCV



モデリングのための前処理

- モデリングを適切に行うために，説明変数を変換(前処理)することが
  しばしばある
- 線形変換（値域のシフト，スケールの変換）
  - 標準化（Zスコア化）: 平均 0，標準偏差 1
  - 正規化: 最小値 0，最大値 1
- 非線形変換
  - 対数変換
  - Box-Cox変換
- 高次項，交互作用

線形モデルの前処理

- 単純な線形回帰(OLS)であれば, 解析的には線形変換は不要なはず
  - 数値計算の観点ではスケールが揃っている方が好ましい
- 非線形相互作用や属性間の影響(交互作用)を考慮するために
  高次項を考慮することがある
- 正則化を加える場合(リッジ回帰, LASSO回帰, Elastic Net回帰),
  正則化の効果は係数のスケールによって変わる
  - 説明変数のスケールを揃えることが一般的

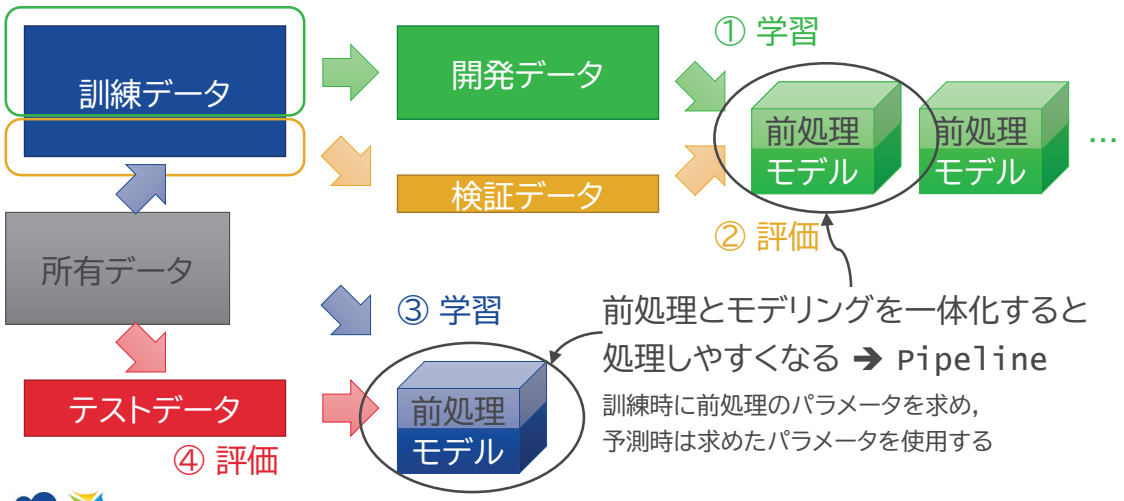
その他のモデルの前処理

- 木モデル
  - 線形変換や単調増加(減少)な非線形変換の影響を受けない
- カーネルリッジ回帰
  - カーネルの計算方法に依存するが, 変数のスケールに依存することがある
  - ガウス過程回帰では, 目的変数の平均値を暗黙的に0とすることが多い
- ニューラルネットワーク
  - 基本的には線形モデルと同じであり,
    L2正則化を加える場合はスケールの違いの影響を受ける
  - 係数の初期値の設定で, 変数のスケールに暗黙の前提がある場合がある

前処理の注意点

- 前処理にパラメータが必要となる場合,
  原則として, 訓練データのみからパラメータを定める
  - スケールやゼロに意味がある場合は,
    テストデータ(検証データ)を含めた全データからパラメータを求めると
    テストデータ(検証データ)の情報を使うことになる
- Hold-out分割の場合はテストデータを使用しないようにすることは容易
- k分割交差検証の場合は処理の手順に注意が必要
  - Scikit-learn では Pipeline を使用するとよい

前処理を含むモデリング手順



ベイズ最適化

最適化問題

- 最適化問題には大域的最適化と局所的最適化がある
  - このセミナーでは大域的最適化のみを取り上げる
- 工学的な問題では
  - 多次元の組み合わせ問題が多い
  - 入力から結果に至る過程が複雑である（『ブラックボックス関数』）
  - 評価コスト（お金, 時間, 手間）が高い

  その結果, 様々な問題が生じる
    - 網羅的な探索が困難, 解析的な解法が困難, 局所解が多い, …
- 真の最適解を求めるというよりは

  「適切な近似解をより少ない評価回数で求める」という問題となる

大域的最適化の手法

- メタヒューリスティクス
  - 自然に見られる最適化の現象をモデル化した手法
  - 正しい解が得られる保証はないが，概ね上手くいくことが知られている
    - 遺伝的アルゴリズム
    - 粒子群最適化法
    - 蟻コロニー最適化法
    - 模擬焼き鈍し法
    - …
- ベイス最適化
  - 確率論（特にベイズの定理）を活用した最適化手法

ベイズ最適化の基本的な考え方

実験を積み重ねた後に，次に実験する候補（条件）を考える

- これまでに良い結果が得られた条件に似た条件　『**活用**』
- これまでの試行が少なく，結果が不確かな条件　『**探索**』

このバランスを考えて，次の実験候補を決める

⇒『**活用と探索のトレードオフ**』

ベイズの定理の事後分布

- これまでの結果から，目的変数を**確率分布として予測**するモデルを構築する
- 予測値とその不確かさを考慮して，各候補の「実験する価値」を評価する
- 「実験する価値」が高い候補を実験する　『**獲得関数**』

ガウス過程

- 確率変数とは，値が確率的に決まる変数 Y のことをいう
- 確率過程とは，$Y_1, Y_2, \cdots$ が全て確率変数である数列 $(Y_1, Y_2, \cdots)$ のことをいう
  - 測定点 $X_1, X_2, \cdots$ に対する条件付き確率 $P(Y_1, Y_2, \cdots | X_1, X_2, \cdots)$ は関数の確率分布と見なすことができる
- $Y_1, Y_2, \cdots$ が多変量正規分布にしたがう確率過程をガウス過程という
  - ベイズ最適化でよく用いられる
  - 正規分布の事後分布は正規分布となるため，数学的な取り扱いが容易

ガウス過程: 多変量正規分布

$$P(Y_1,\cdots,Y_n|X_1,\cdots,X_n) = (2\pi)^{-\frac{n}{2}}|K|^{-\frac{1}{2}}\exp\left(-\frac{1}{2}(y-m)^T K^{-1}(y-m)\right)$$

Y_i　　$y = (Y_1,\cdots,Y_n)^T$

平均値　　$m = \big(m(X_1),\cdots,m(X_n)\big)^T$

共分散　　$K = \begin{pmatrix} k(X_1,X_1) & \cdots & k(X_1,X_n) \\ \vdots & \ddots & \vdots \\ k(X_n,X_1) & \cdots & k(X_n,X_n) \end{pmatrix}$　　$k(X,X')$：カーネル

ガウス過程: 事後確率

データ $D = \{(X_1,Y_1),\cdots,(X_n,Y_n)\}$ が得られている場合, $Y|X$ の事後確率は

$$P(Y|X,D) = N(m(X) + k^T K^{-1}(y-m), k(X,X) - k^T K^{-1}k)$$

$$k = \big(k(X,X_1),\cdots,k(X,X_n)\big)^T$$

新しい測定点 X と
過去の測定点 $X_1,\cdots,X_n$ の共分散

データ D を取得することで

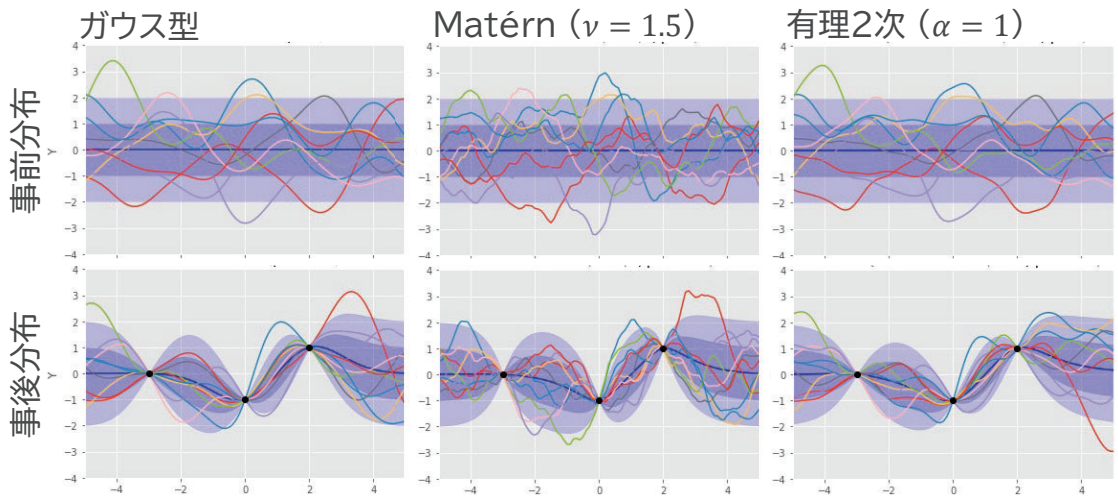
- 平均値が $k^T K^{-1}(y-m)$ だけ変化
- 分散が $k^T K^{-1}k$ だけ減少

機械学習では, 平均 $m(X) = 0$ とすることが多い

ガウス過程: カーネル（共分散）

- 共分散を定義するカーネル $k(X,X')$ が Y の振る舞いを特徴付ける
  - X と X' における Y の相関の程度（類似度）を表す
- 代表的なカーネル
  - 線形カーネル
    - 線形回帰と同等になるため, 数理解析で用いられることがある
  - ガウス型カーネル
  - Matérnカーネル
  - 有理2次カーネル
    - Matérnカーネルと有理2次カーネルには形状を表すパラメータがあり, 無限大の極限ではガウス型カーネルに収束する

各カーネルから導かれる関数



獲得関数

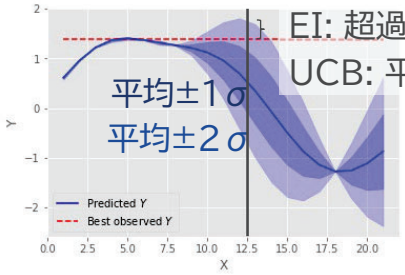
- 代表的な獲得関数
  - Probability of improvement (PI)
    - 既存のデータの最大値を上回る確率で評価する
  - Expected improvement (EI)
    - 既存のデータの最大値を上回る程度の期待値で評価する
  - Upper confidence bound (UCB)
    - 信頼区間の上限(平均値 + κ ×標準偏差; 確率論的な最大値)で評価する

獲得関数

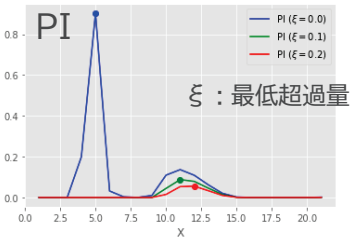
- 獲得関数は「活用」と「探索」のトレードオフを調整するためのパラメータを含む
  - 数理学的には最適なパラメータに関する議論はある
  - 初期は「探索」を重視し, データが増えてきたら「活用」を重視する
- 傾向として, PIは「活用」を重視する傾向にある
  ➔ 局所解に陥りやすい

獲得関数の違い

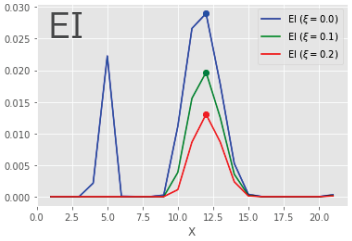
Y の予測結果

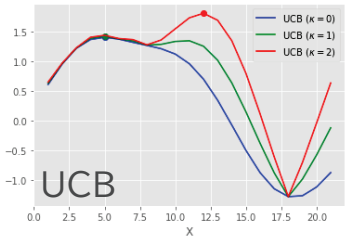


PI: 超過する確率
EI: 超過量の期待値
UCB: 平均＋κσ の値



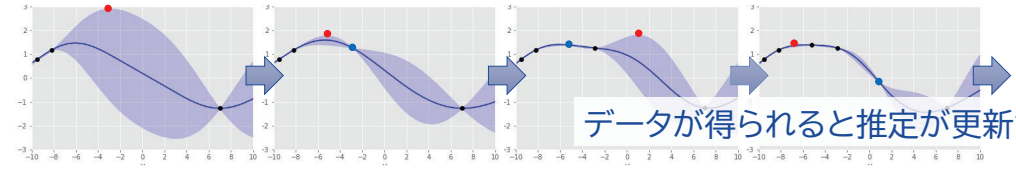
ξ：最低超過量



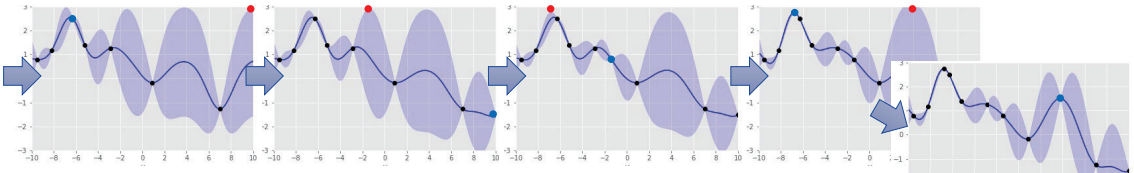


ベイズ最適化の例

平均が0の滑らかな関数を想定
黒点は観測値，青点は新しい観測値，青線・塗りは推定値（平均±2σ），赤点は次の評価候補（UCB 2σ）



データが得られると推定が更新される



推定から外れたデータが得られるとモデルが大きく更新される

最尤法

- ガウス過程回帰のカーネルパラメータは最尤法で最適化できる
  - 確率論に基づくモデルにおいて，
    データに最も適合するようにパラメータを最適化する
  - 最尤法は局所解に陥る可能性があるため，
    初期値を変えて計算を繰り返す方が望ましい
  - 汎化性能で決定しているわけではないため，過学習に注意が必要
- グリッドサーチと交差検証法でパラメータを決定することは可能だが，
  パラメータ数が多いと非常に時間がかかる

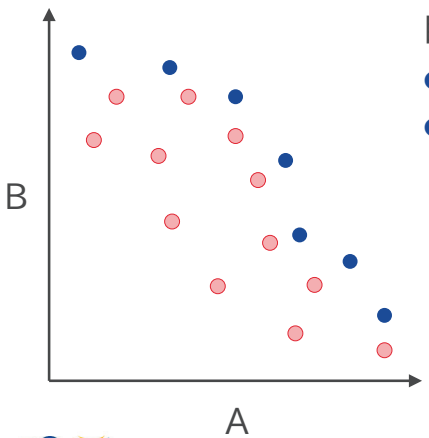
補足

- Scikit-learnでガウス過程回帰モデルの構築はできるが，
  ベイズ最適化自体(獲得関数評価・最適化・更新)はサポートされていない
  - ガウス過程回帰の場合は事後分布が正規分布となるため，
    SciPyなどの関数を使用すれば，獲得関数の計算は難しくない
- ベイズ最適化専用のパッケージがある（以下，Pythonパッケージの例）
  - BayesianOptimization
  - COMBO
  - GPyOpt
  - Hypteropt（ガウス過程回帰ではない）
  - Skopt（ガウス過程回帰以外もあり）

多目的最適化

多目的最適化とパレート最適

複数の目的変数を同時に最適化する問題を多目的最適化問題という



目的変数AとBを最大化したい場合を考える

- Aを増やすためにはBを減らさなければならない
- Bを増やすためにはAを減らさなければならない

このような状態を『パレート最適』であるという
（図中の ●）

多目的最適化

- 他に制約条件がなければ，パレート最適解同士に優劣はない
  - 多目的最適化ではパレート最適解の「集合」を求める必要がある
- いくつかの解法が提案されている
  - 制約条件を設定し，1変数最適化問題とする
    （制約条件を変更することで，異なるパレート最適解を得る）
  - 遺伝的アルゴリズム
    （複数の目的変数を最適化しつつ，解同士ができるだけ異なるようにする）
- パレート最適解を求めるだけでなく，解の「集合」に対する分析が必要である

その他

データ処理の再現性

- データ処理の再現性を確保することは重要
  - 新しくデータが追加されたときに同じように処理できなければ
    後続のモデリングの評価は困難
  - 同一の元データから同一の処理結果が得られる
- 最終形はプログラムによる自動処理でモデリング用データを得る
  - 手作業による処理やコピー＆ペーストなどが入ると
    同一処理を担保することは困難
- データ処理の途中段階では試行錯誤が必要であり，
  新しいデータに対して同一処理でよいとは限らない

サンプリング

- 所有するデータを
  「機械学習モデルを構築するためのデータ」と
  「モデルの良さを評価するためのデータ」に
  分割することが原則
- 特に分類問題において，注意すべき点（テクニック）がある
  - 層別サンプリング
  - アンダーサンプリング，オーバーサンプリング

層別サンプリング

- 汎化性能を適切に評価するためには
  テストデータは母集団と同じ分布を持つ必要がある
- 所有データは母集団と同じ分布を持つと仮定すると
  所有データの分布を保つようにテストデータをサンプリングする
  - 目的変数のクラスごとに訓練データとテストデータに分割する
    （『層別サンプリング（stratified sampling）』）
  - Scikit-learnでは，StratifiedKFold などの専用のクラスや stratify
    というオプション引数を用いることで層別サンプリングが可能

不均衡データ

- 分類問題では，訓練データのクラス分布に偏りがあると学習が難しくなる．
  特に，二値分類問題で正例と負例の割合が大きく異なる
  「インバランス（不均衡）」なデータで問題となる．
  - 通常は正例が少なく，負例が多い
  - 単純に数の多いクラスを予測するだけで高い精度が得られる
- サンプリングを工夫して，クラスの割合を補正することがある
  - 負例サンプルを減らす（『アンダーサンプリング』）
  - 正例サンプルを（人工的に）増やす（『オーバーサンプリング』）
  - Scikit-learnだけでは対応できない