

学習の種類



| 学習の種類

・学習の種類

機械学習には大きく分けて3種類の手法がある

- ・教師あり学習
- ・教師なし学習
- ・強化学習

→解きたい問題に合わせた手法を選択する必要がある

| 学習の種類

- 教師あり学習

特徴量と正解が与えられたデータ（正解ラベル）をもとに
ルールやパターンを学習していく手法のこと

→分類問題、回帰問題 を解くときに使われる手法になる

- 分類問題：データが属するカテゴリ（離散値）を予測する問題
- 回帰問題：連続値（体重、温度など）を予測する問題

| 学習の種類

- 教師あり学習

- 連続値

体重など連續した値のことで、値と値の間に無限の値を取る
→60kgと61kgの間には無限の値が存在する

- 離散値

クラス番号やカテゴリなど値が連續していないもの

| 学習の種類

- ・教師あり学習
 - ・分類問題（データが属するカテゴリを予測する問題）
 - メールを迷惑メールまたは一般のメールに分類
 - 画像を入力して犬か猫かを分類
 - 記事の文章から事前に決めたカテゴリーに分類 など

→以上のように、あらかじめ決めたカテゴリに分類していく

| 学習の種類

- 教師あり学習

- 回帰問題（連續値を予測する問題）

- 湿度、気温、曜日からビールの売上を予測する

- 気温から商品の売上や来店数を予測する

- 駅からの距離や床面積などから家賃を予測する など

→データを入力したら求めたいデータが出力される

| 学習の種類

・教師なし学習

特徴量と正解が付与されていないデータをもとに

ルールやパターンを学習していく手法のこと

→クラスタリング、次元削減（次元圧縮）などで使われる

・クラスタリング：データを類似度に基づいてクラスタに分類

・次元削減 : データを要約し、扱うデータを減らす

| 学習の種類

- ・教師なし学習
 - ・クラスタリング（データを類似度に基づいてクラスタに分類）
 - 購入データから顧客をクラスタに分類
 - 記事の文章からクラスタに分類
- クラスタリングはデータの特徴によって自動でクラスタを作成
分類問題は分類する前にカテゴリを決める必要がある

| 学習の種類

- 教師なし学習

- 次元削減（データを要約し、扱うデータを減らす）

数学・理科を理系、社会・国語を文系にまとめる など

→データの意味を保ったまま、データを要約することがポイント

→扱う特徴量を減らすことで、計算量を減らすことができる

| 学習の種類

- 半教師あり学習

教師なし学習と教師あり学習を組み合わせた学習

→ 少量の正解ラベル付きデータで学習した後に

正解ラベルが付いていない大量のデータで学習していく

→ 一般的に教師あり学習よりも精度は低くなってしまう特徴

| 学習の種類

・半教師あり学習

大量の正解ラベル付きデータを収集することが難しい場合や
正解ラベルを付けるコストを削減したいときに使用される
→正解ラベルを一つひとつ設定するのはコストがかかりすぎる

→コスト面なので余裕がある場合は教師あり学習の方が無難

| 学習の種類

・強化学習

与えられた環境下で、報酬が最大となるように

最適な行動を学習していく手法

→報酬とはエージェントの行動を評価するスコアのこと

→エージェントとは自律的に行動するシステムやプログラムのこと

お掃除ロボット、囲碁や将棋のゲーム など



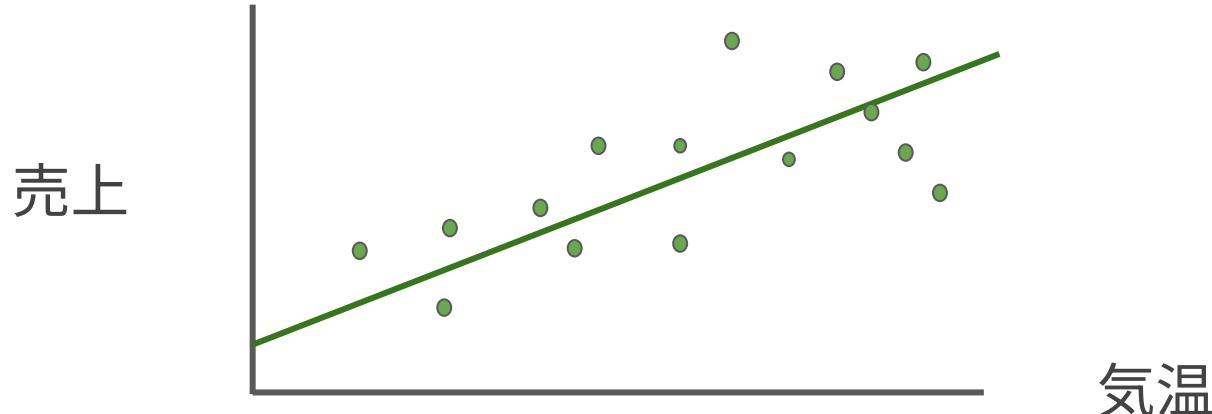
教師あり学習 線形回帰

線形回帰

・ 線形回帰

求めたい変数を**1つ以上の変数**を使って予測するための手法

→アイスクリームの売上を気温から予測したい ($y = ax + b$)



線形回帰

- 線形回帰

目的変数：予測したい変数（結果となる変数）のこと

説明変数：目的変数を説明するための変数

何かの原因になっている変数

→気温からアイスクリームの売上を予測したい場合は

アイスクリームの売上（**目的変数**）と気温（**説明変数**）

線形回帰

- ・ 単回帰分析

1つの目的変数を1つの説明変数で予測する分析手法のこと
→気温からアイスクリームの売上を予測する など

- ・ 重回帰分析

1つの目的変数を複数の説明変数で予測する分析手法のこと
→人口、気温などからアイスクリームの売上を予測する など

線形回帰

- ・ 単回帰分析

$$y = ax + b \quad (y: \text{目的変数}, \ x: \text{説明変数}, \ a: \text{傾き}, \ b: \text{切片})$$

→ $y = 0.8x + 0.5$ の場合 x が1増えると y が0.8増えることを意味

→ a は回帰係数、 b は回帰定数（定数）とも呼ばれる

線形回帰

- 重回帰分析

$$y = a_1x_1 + a_2x_2 + \dots + a_nx_n + b$$

→ a は偏回帰係数と呼ばれ、偏回帰係数の大きさによって
予測にどれだけ影響を与えていているのかを把握することができる

$$y = 0.7x_1 + 0.3x_2 + 0.25x_3 + 0.3$$

→一番前の偏回帰係数が最も影響を与えていることが分かる

線形回帰

- 重回帰分析

重回帰分析では、**多重共線性**に注意する必要がある

- 多重共線性

重回帰分析において、相関係数の高い

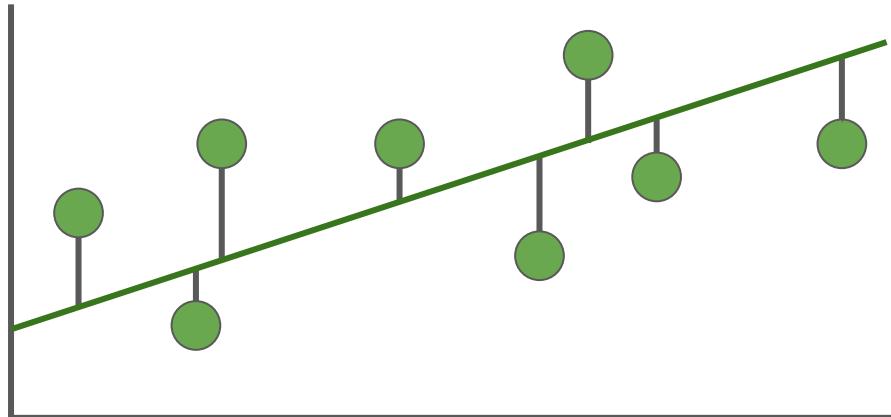
説明変数（特徴量）の組み合わせが存在すること

→計算が不安定になったり、予測精度が低下したりする

線形回帰

- 線形回帰

回帰係数 a と回帰定数 b を求める方法に**最小二乗法**がある
→ 実際の値と予想値との差の2乗和が最小になる値を探す



線形回帰

- 線形回帰

線形回帰に正則化を取り入れたラッソ回帰、リッジ回帰がある

→詳しくは今後の講義で解説

→正則化とは過学習を防ぐために使われる手法の1つ

過学習とは訓練データに対して過度に適応しすぎた結果

訓練データ以外のデータに対して予測精度が悪くなること



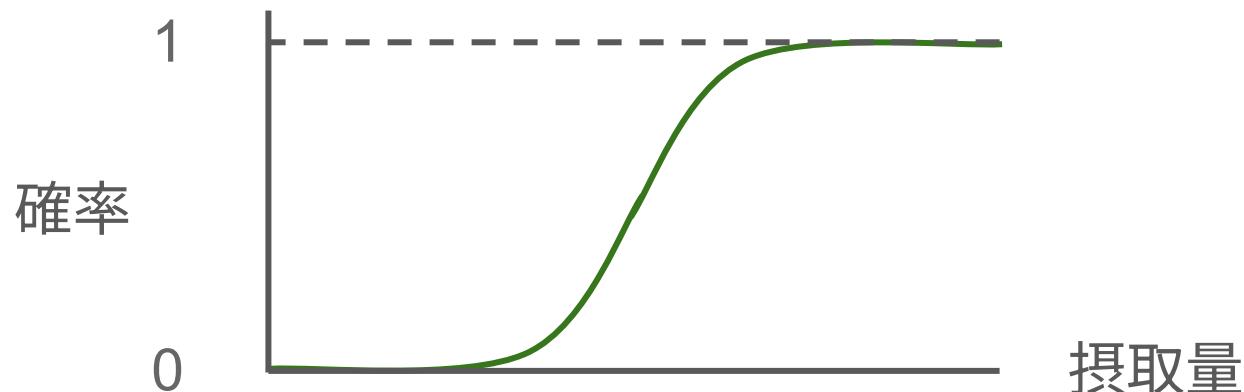
教師あり学習 ロジスティック回帰

| ロジスティック回帰

- ・ロジスティック回帰

ある事象が発生する確率を予想する手法のこと

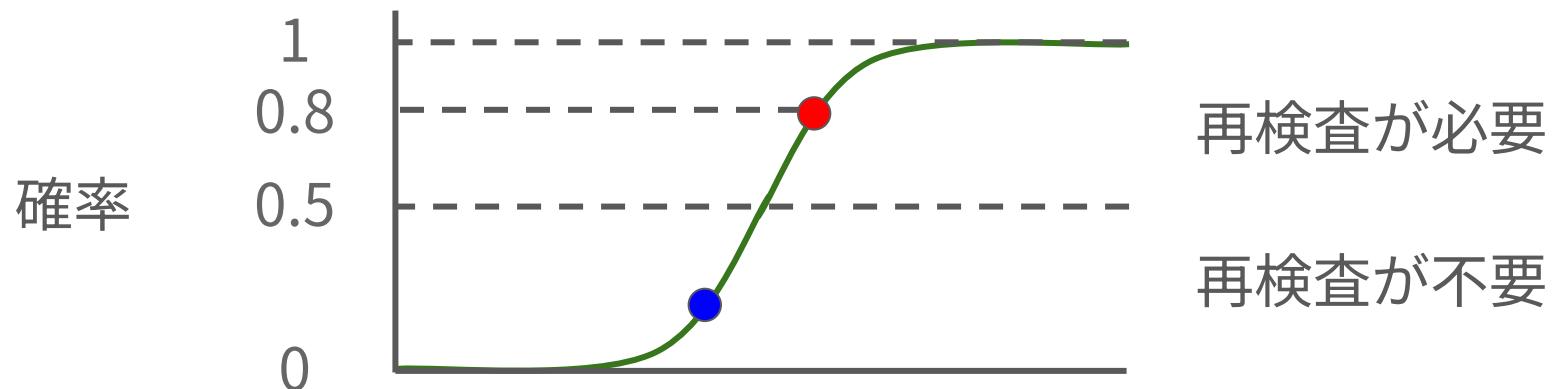
→アルコール摂取量などから病気の発症確率を求める など



| ロジスティック回帰

- ロジスティック回帰

確率を上手く活用することで分類問題に応用することができる
→確率が50%以上なら再検査が必要、50%未満は再検査が不要



| ロジスティック回帰

- ロジスティック回帰

データが**正例 (+1)**、**負例 (0)** になるかを求めていく手法

→再検査が必要な場合を**正例**、再検査が不要な場合を**負例**

→**閾値**（基準とする値）は0.5が基本だが、状況によって変わる

スパムメールを判断するときは閾値が高くなる

| ロジスティック回帰

- ロジスティック回帰

結果が2択（2値分類）のときはシグモイド関数が使用される
→再検査が必要か不要、合格するかどうか など

結果が3択以上（多クラス分類・マルチクラス分類）のときは
ソフトマックス関数が使用される
→合格判定（A判定、B判定、C判定等）、服のサイズ など

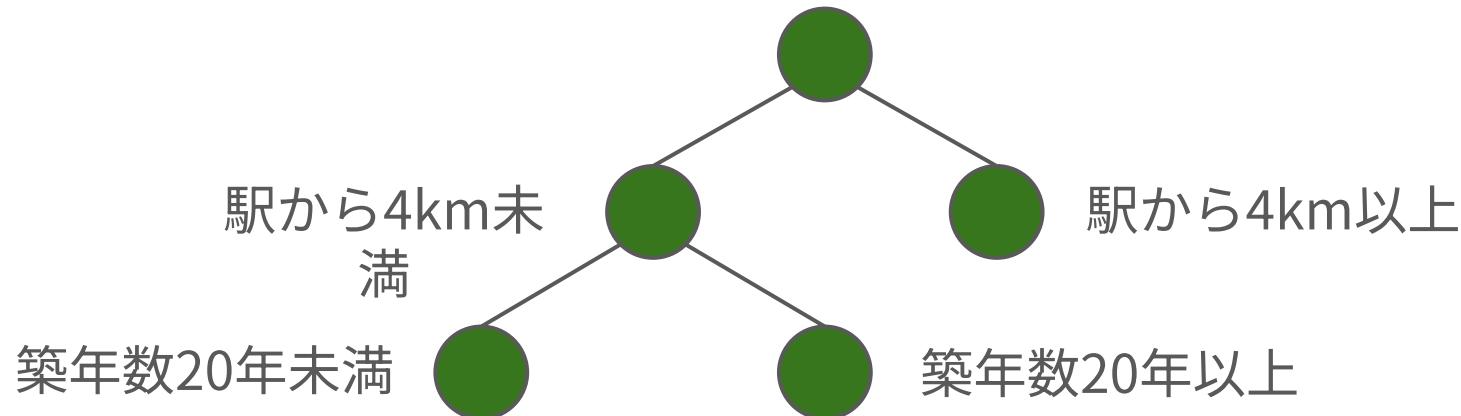


教師あり学習 決定木

| 決定木

・決定木

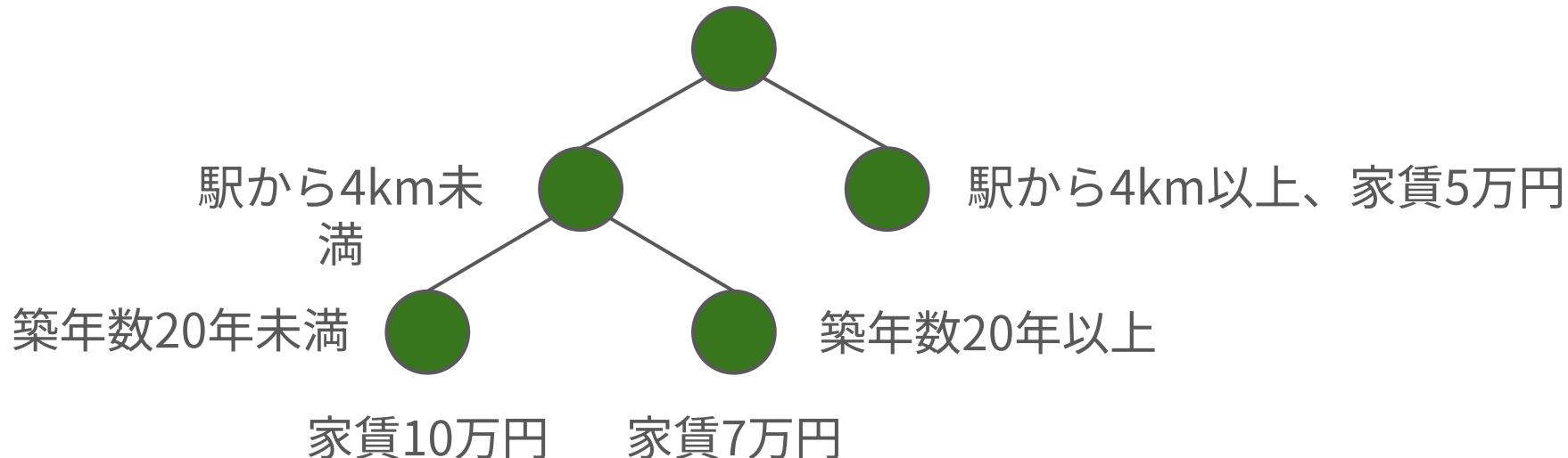
ツリー（樹形図）によってデータを分析していく手法のこと
→特徴量に基づいてグループを分割していく（分岐路を作る）



決定木

・回帰木

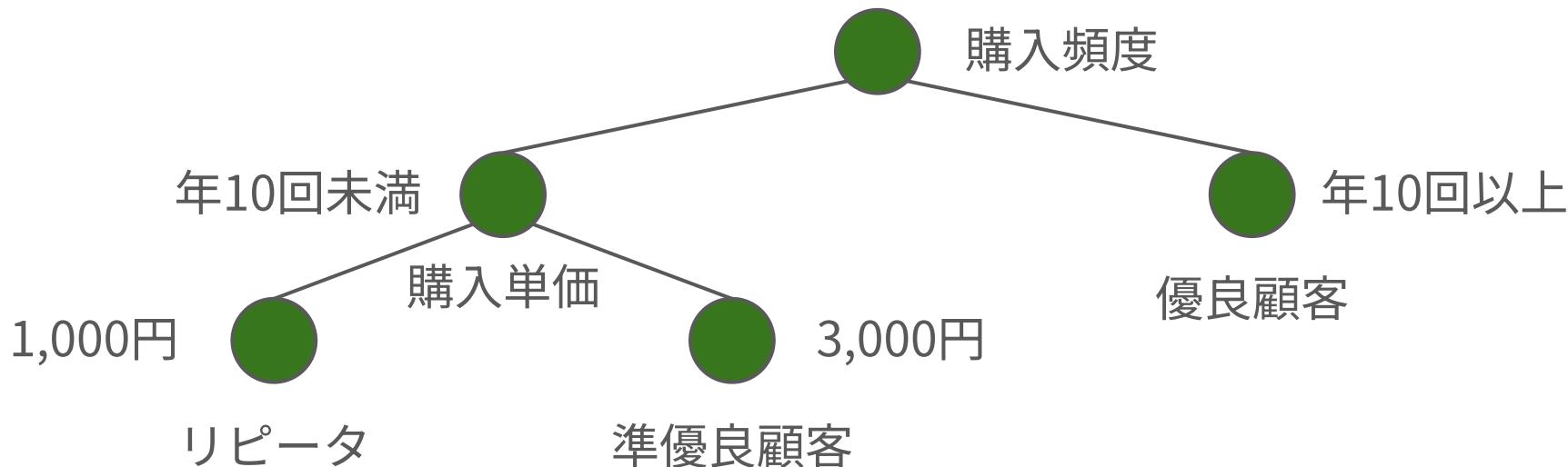
回帰で使用される決定木のこと（家賃の価格など）



| 決定木

・分類木

分類で使用される決定木のこと（顧客分類など）

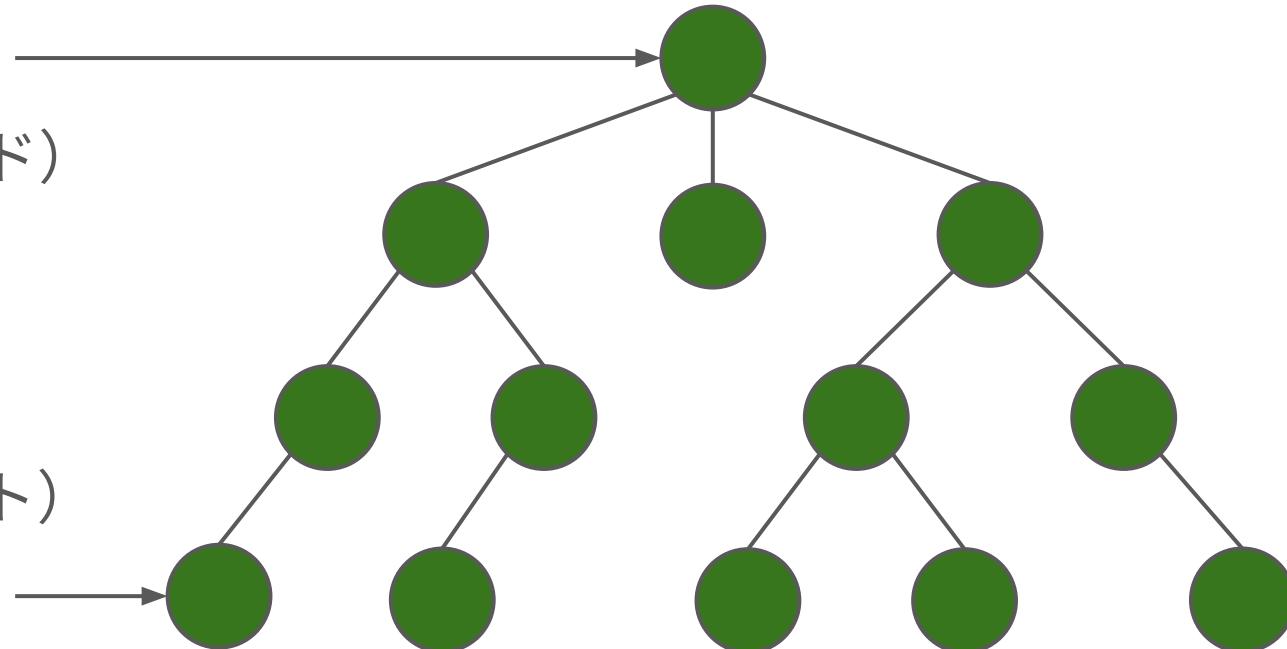


| 決定木

・決定木

根ノード
(ルートノード)

(リーフルート)
葉ノード



| 決定木

• 分岐と過学習

分岐が多くなると1つの葉に1つのデータが対応してしまう
→過学習の状態になってしまい、モデルとして良くない
訓練データに適応しすぎた結果、未知のデータに対する
推測の精度が下がってしまう現象を過学習という
→過学習を防ぐためにも、木の深さ・幅に気をつける必要がある

| 決定木

・決定木

きれいにデータを分割することが大切（分岐路の作成）

→情報利得が最大化になるように分割させる

- ・**情報利得**：分割前の不純度から分割後の不純度を引いたもの
- ・**不純度**：1つのノードに異なるクラスのサンプルがどれだけ含まれているのかという割合

| 決定木

・決定木

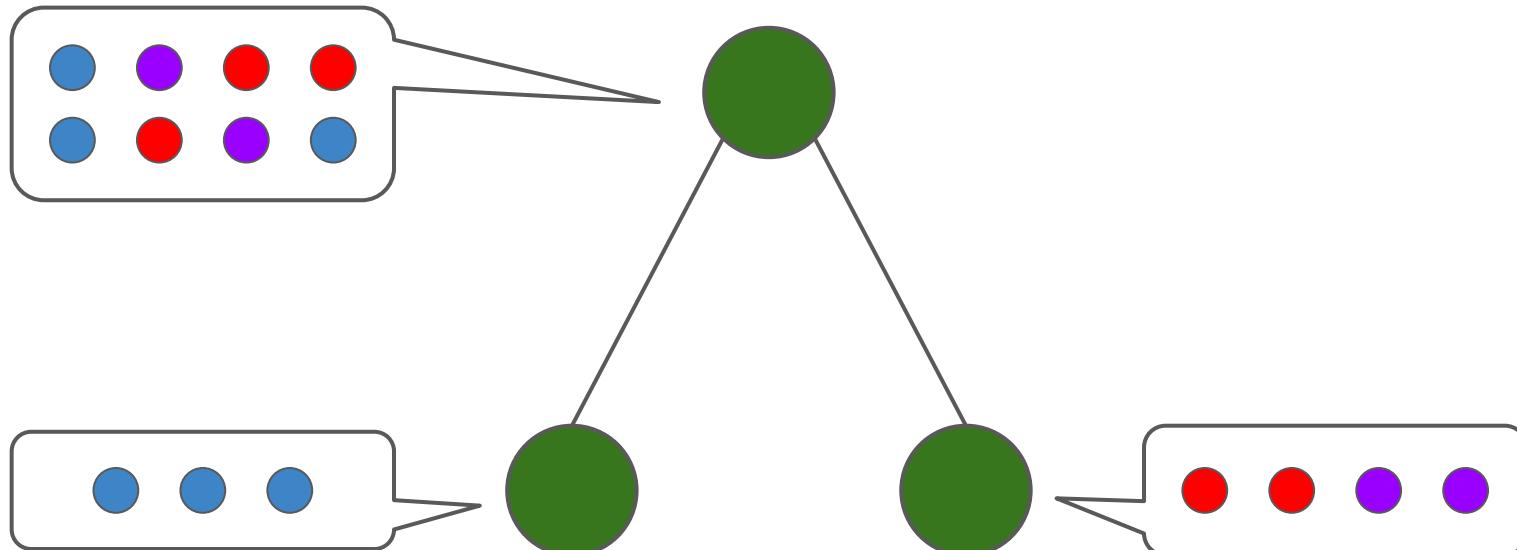
純度を計算するとき、エントロピーやジニ係数など使用する
→純度の高い（不純度が低い）モデルは予測精度が高くなりやすい

→不純度が低いモデルを実現するために
特徴量と閾値（分割する値）を求めていく（学習）

| 決定木

- 決定木

最適な特徴量と閾値を求める



| 決定木

・決定木の特徴

- 使用されている特徴量や閾値が分かるため何が重要な要素だと分かれば、
→予測に大きな影響を与える特徴量が分かることで施策に反映
- 「購入回数」が「利益」に対して重要な要素だと分かれば、
「購入回数」を増やす施策を実施することで
「利益」を伸ばすことができる可能性がある

| 決定木

・決定木の特徴

- 後から人が特徴量や閾値を容易に変更しやすいモデルである
→他のモデルの場合だと変更することが困難な場合が多い
- 外れ値が多いと適切に分割することができないことがある
データ量を増やすことで、ある程度解決することができる
→データ量を増やすことで過学習を防ぐこともできる



教師あり学習 ランダムフォレスト

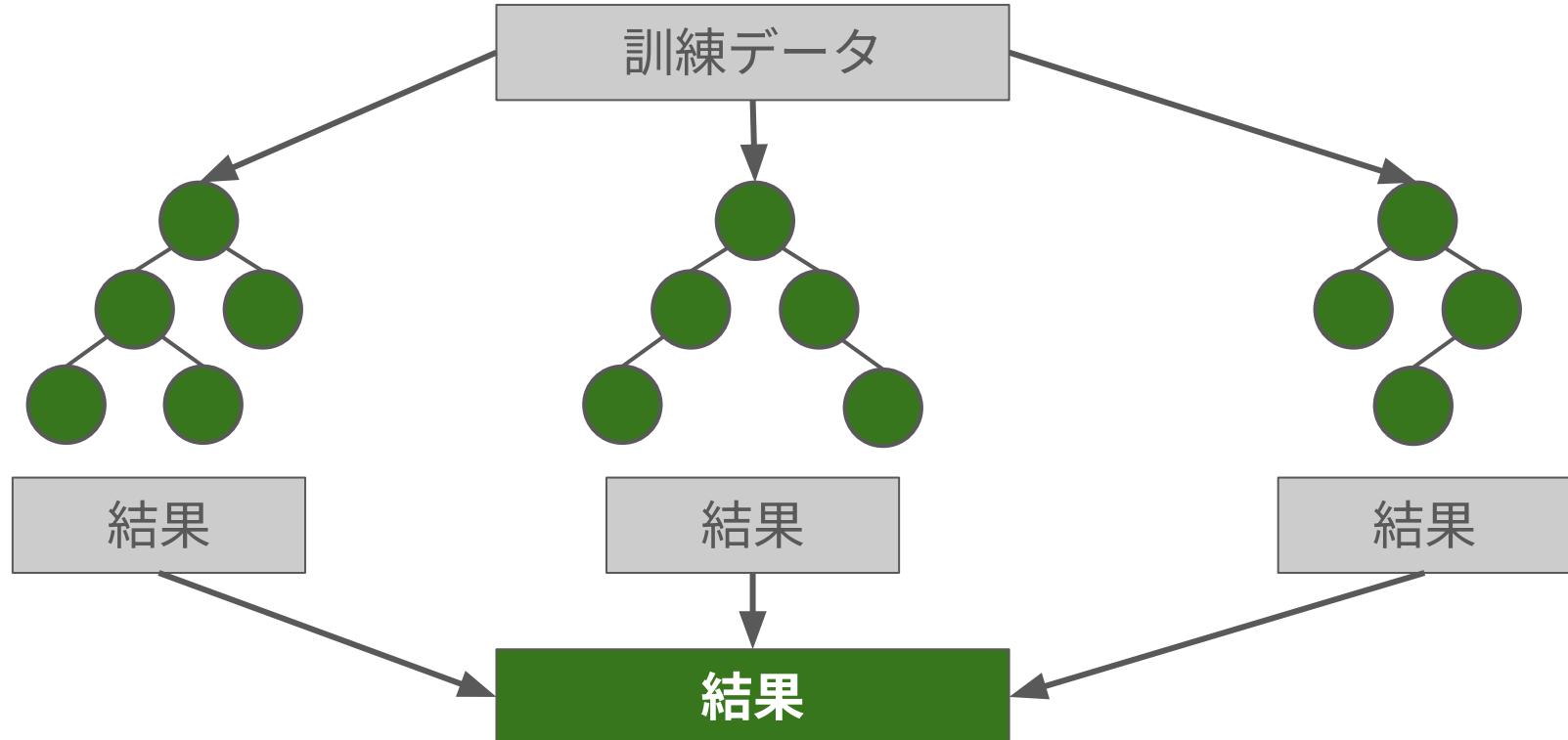
| ランダムフォレスト

・ランダムフォレスト

ランダムに抽出した訓練データで学習した決定木を複数作成し、各決定木で予測された結果の**多数決**でモデル全体の最終的な出力を決定する手法のこと（分類の場合）

→回帰の場合は多数決ではなく各決定木が予測した結果の平均値をモデル全体の最終的な出力する

| ランダムフォレスト



| ランダムフォレスト

・ランダムフォレスト

決定木同士の構造が同じになり過ぎないように

特徴量もランダムに抽出している

→決定木によって精度にばらつきが発生してしまう

→複数の決定木を組み合わせることで予測精度を高めている

1つ1つの精度が低くても、全体で精度が高くなれば良い

| ランダムフォレスト

- ランダムフォレスト

各決定木の精度の違いと特徴量を分析することで、
特徴量ごとの重要度などを算出することができる

- 決定木の数が多くなっても過学習が発生しにくい
- 1つの決定木が偏っていたとしても
全体で結果を出力するため影響が少ない

| ランダムフォレスト

・ランダムフォレスト

各決定木の学習を行うために、無作為に一部の訓練データを抽出することをランダムサンプリングという

→重複を許して無作為に一部のデータを抽出することを

ブートストラップサンプリングという

→重複したデータも使用することで決定木に**多様性**が生まれる

| ランダムフォレスト

・ ランダムフォレスト

サンプルを抽出するとき、サンプルを抽出して元に戻してから、次のサンプルを抽出することを**復元抽出**という

サンプルを抽出するとき、サンプルを抽出して元に戻さずに、次のサンプルを抽出することを**非復元抽出**という

| ランダムフォレスト

- ランダムフォレスト

「決定木」と「アンサンブル学習」を組み合わせた手法

- アンサンブル学習

精度の高くない学習器を複数組み合わせて予測を行う手法

→バギング、ブースティングという手法がある

| ランダムフォレスト

・バギング

複数のモデルを並列的に学習させていく手法

→ランダムフォレストはバギングの1つで学習時間が短い特徴

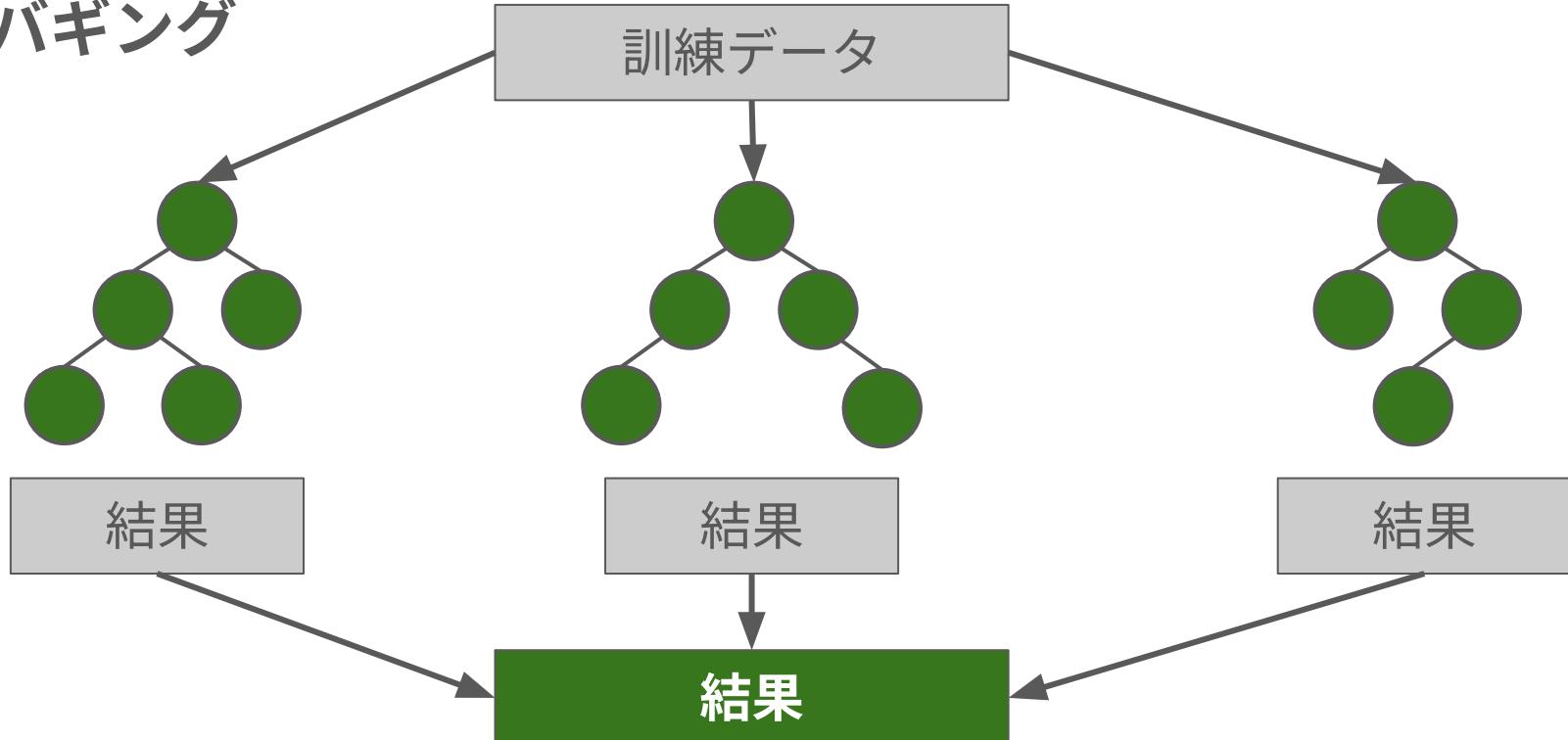
・ブースティング

モデルを直列的に学習させていく手法のこと

→前に作成したモデルを修正して新しいモデルを作成していく

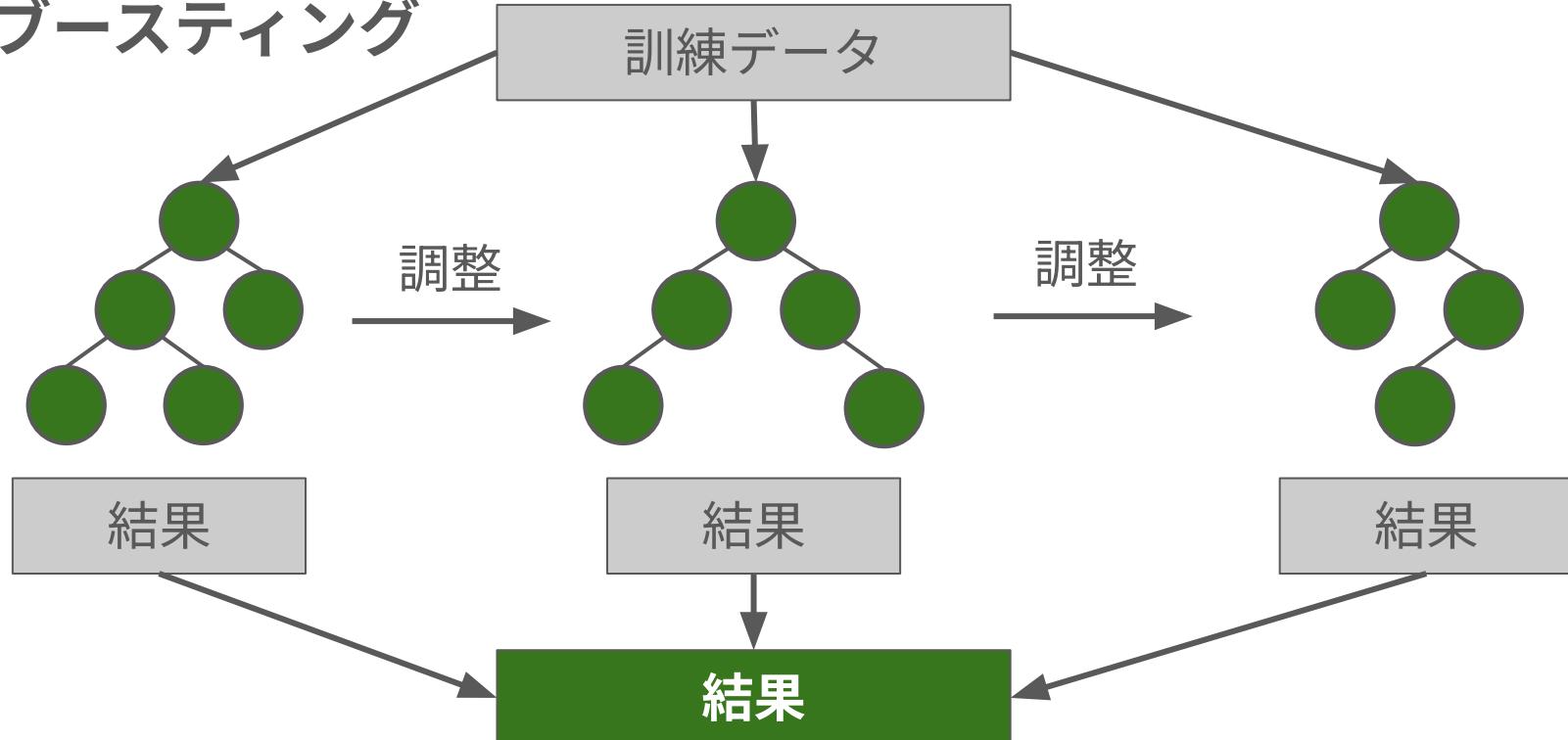
| ランダムフォレスト

・バギング



| ランダムフォレスト

・ブースティング



| ランダムフォレスト

・ブースティングの特徴

並列処理できないため学習に時間がかかるてしまう

→バギングよりも予測の精度は高いが、過学習になりやすい

→ブースティングモデルのチューニングは難易度が高い

→勾配ブースティング、AdaBoost、XGBoostなどが有名な手法

| ランダムフォレスト

- ・ブースティングの代表的な手法

勾配ブースティング：勾配降下法が使用されている

XGBoost : 決定木と勾配ブースティングを組み合わせた手法

AdaBoost : 誤分類したデータの重みを大きくして
重要度を高め、精度を高めようとする手法



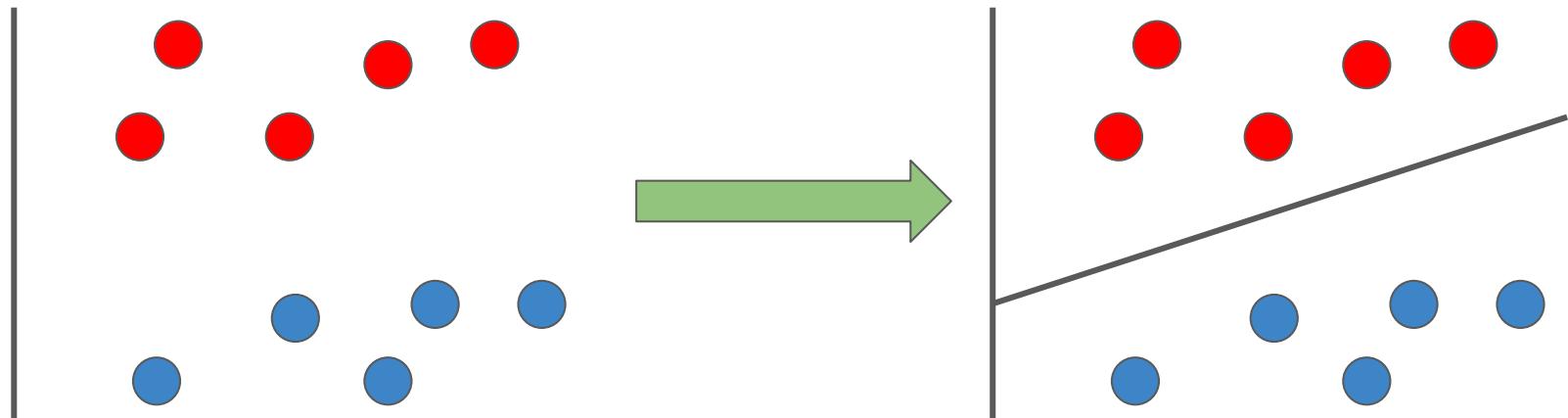
教師あり学習 サポートベクターマシン

| サポートベクターマシン

- サポートベクターマシン (SVM)

主に分類で使われるアルゴリズム

→マージン最大化という考え方を用いて、分類する線を求めていく

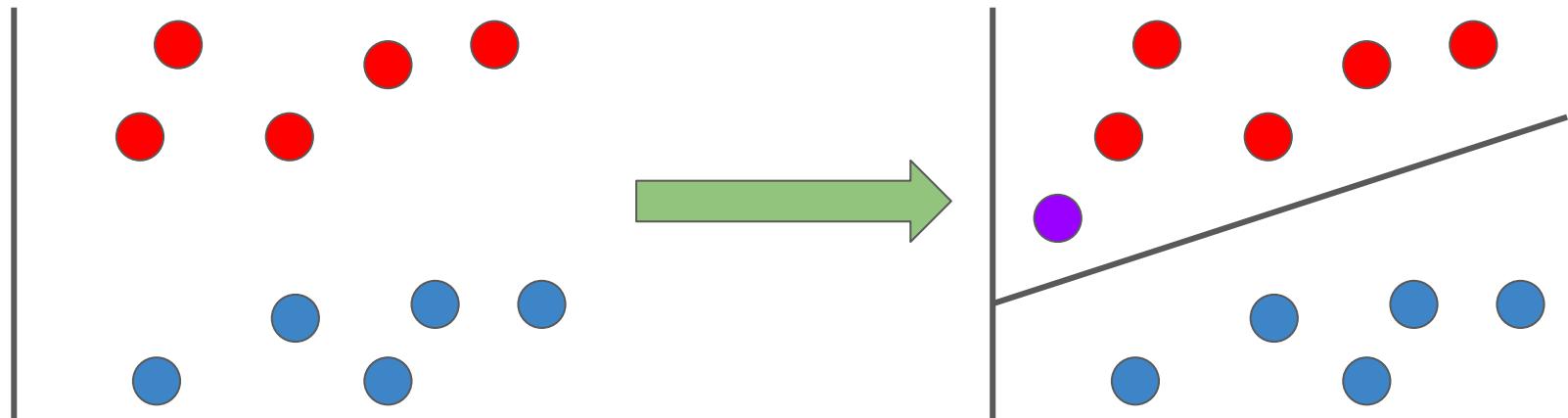


| サポートベクターマシン

- サポートベクターマシン (SVM)

主に分類で使われるアルゴリズム

→マージン最大化という考え方を用いて、分類する線を求めていく



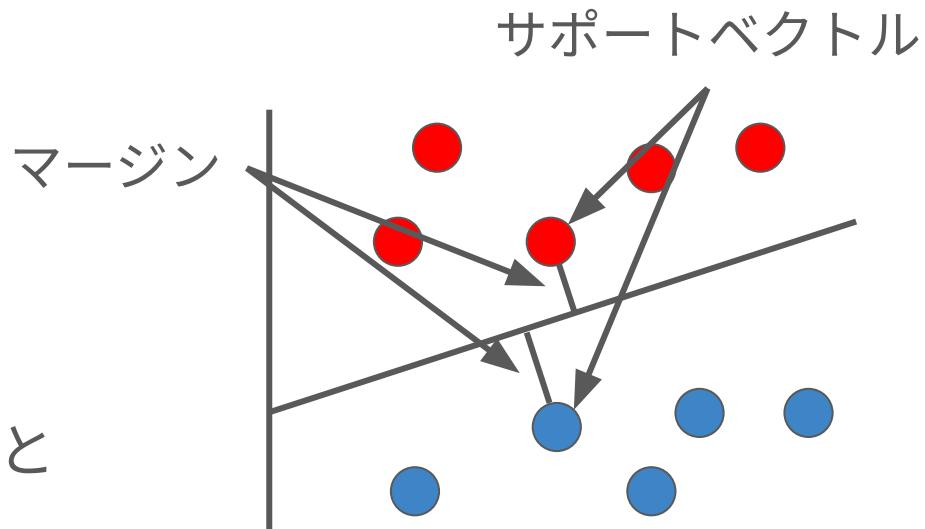
| サポートベクターマシン

- ・マージン

境界線から一番近いデータであるサポートベクトルとの距離

- ・マージン最大化

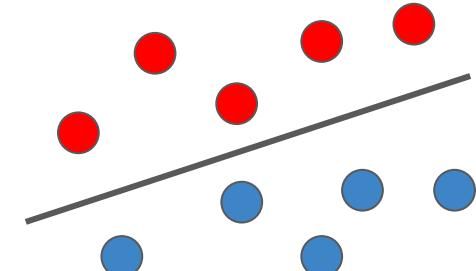
分類する線の中から、
マージンが最大になる線を
見つけようとする考え方のこと



| サポートベクターマシン

- ・ハードマージン

誤分類を禁止しながらマージンの最大化を行う

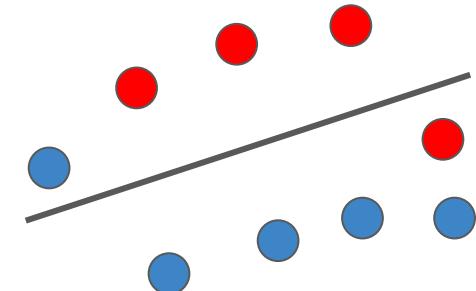


- ・ソフトマージン

誤分類を許容しながらマージンの最大化を行う

→どの程度許容するか調整する変数を

スラック変数という



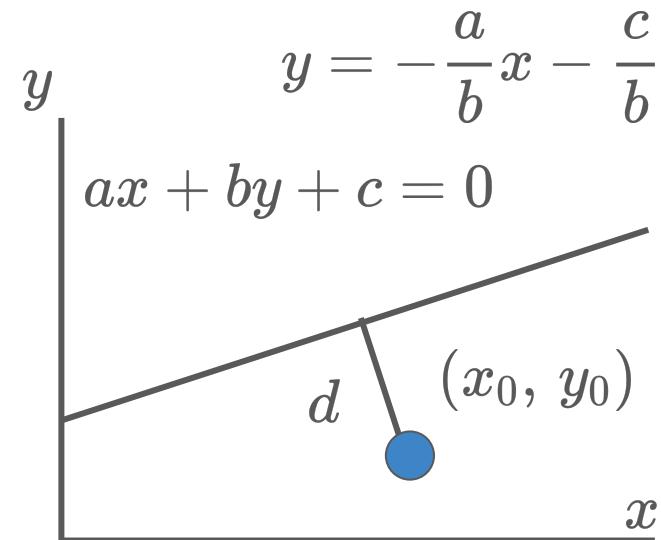
| サポートベクターマシン

- マージン

ヘッセの公式を使用することでマージンを計算することが可能
→ 点と直線の距離を求める公式のこと

→ (x_0, y_0) と $ax + by + c = 0$ の距離 d

$$d = \frac{|ax_0 + by_0 + c|}{\sqrt{a^2 + b^2}}$$



| サポートベクターマシン

- サポートベクターマシン (SVM)

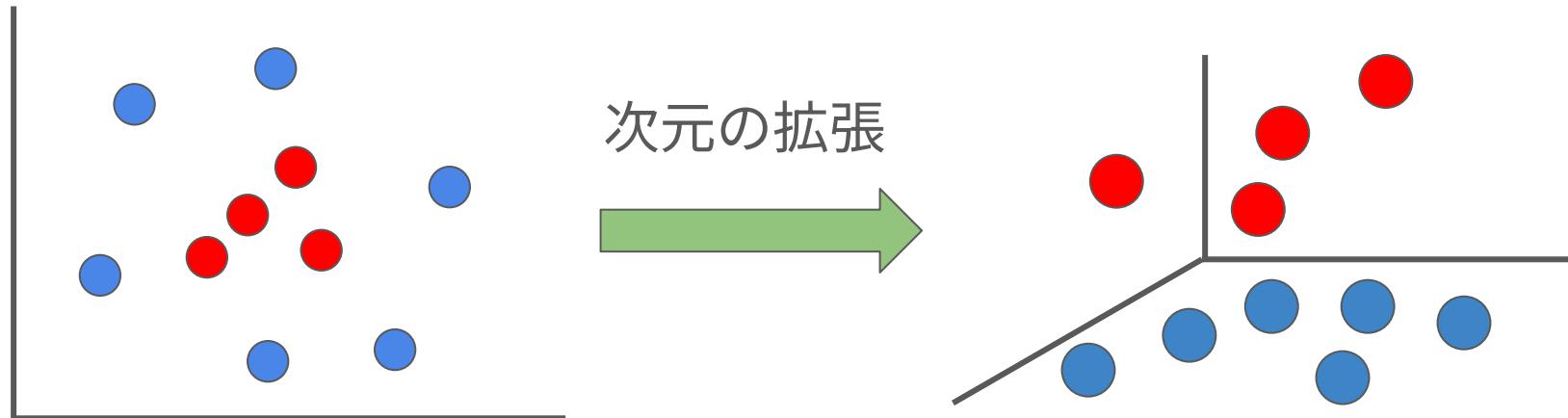
SVMは多クラス分類や回帰問題でも使用されている
→もともとは2クラス分類で使用されていた

→カーネル法を組み合わせることで、直線で分類できないものでも
分類することが可能になる
→カーネル法とは次元を拡張させる手法のこと

| サポートベクターマシン

- サポートベクターマシン (SVM)

以下のデータは直線で分類することができない



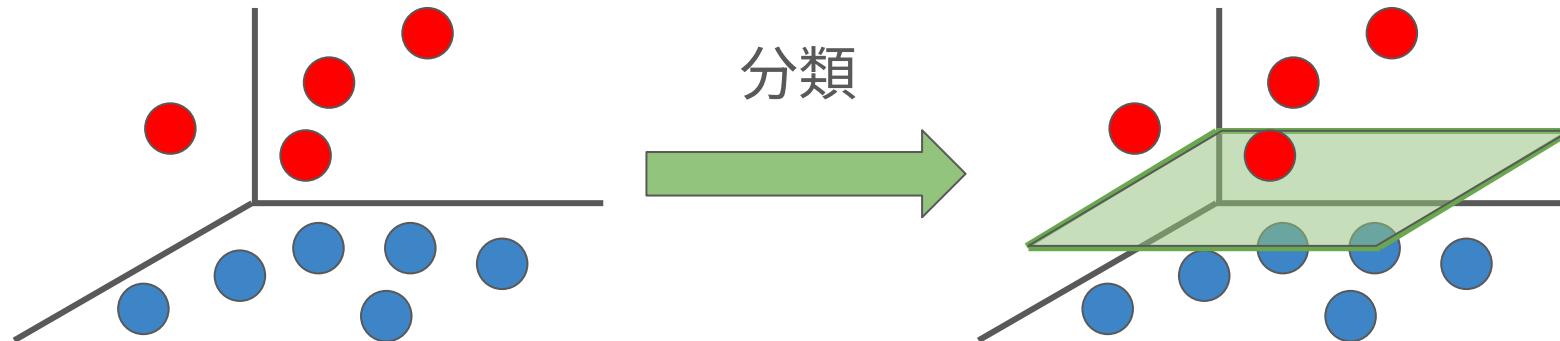
| サポートベクターマシン

- サポートベクターマシン (SVM)

次元を拡張させる関数をカーネル関数と言う

→カーネルトリックと呼ばれる方法を使うことで

計算を楽にすることができる（次元が増えると計算が大変）





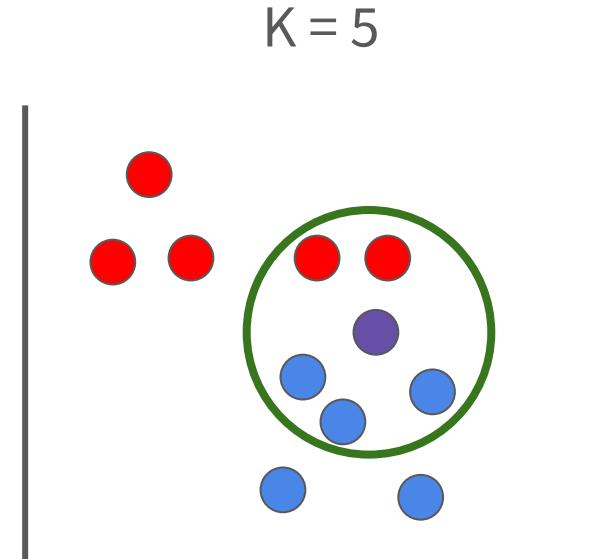
教師あり学習 K近傍法

K近傍法

- K近傍法

分類に使われる手法の1つ

- 未知のデータが与えられたときに
近くにあるデータをK個取得し、
多数決でデータが所属するクラスを分類
- 各データとの距離を計算している



K近傍法

- 問題

K近傍法において、未知のデータを与えた

$K=10$ のとき、クラスAのデータが1つ、クラスBのデータが6つ、

クラスCのデータが2つ、クラスDのデータが1つである

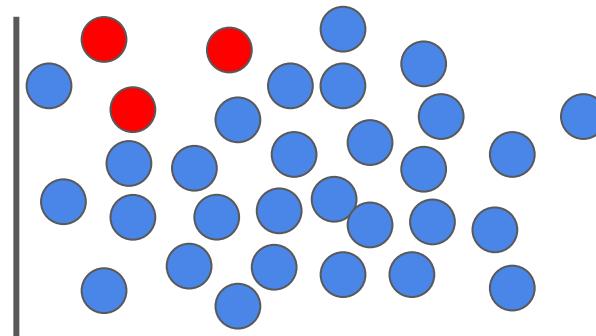
未知のデータが分類されるクラスはどれか

→クラスBのデータが最も多いため、クラスBに分類される

K近傍法

- 注意点

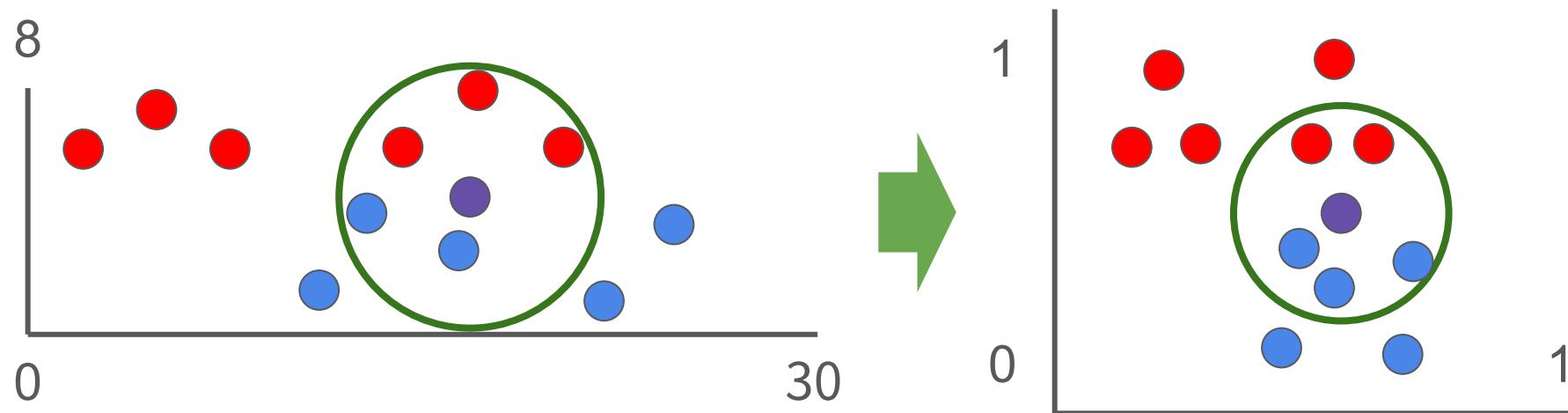
クラスごとのデータ量に偏りがある場合性能が低くなりやすい
→クラスAのデータ量とクラスBのデータ量の割合が1：9ならば、
未知のデータはクラスBに属すると予測される確率が高くなる



K近傍法

- 注意点

各特徴量のスケールが異なっていると予測精度は悪化する
→事前にデータのスケールが一定になるように調整する必要がある



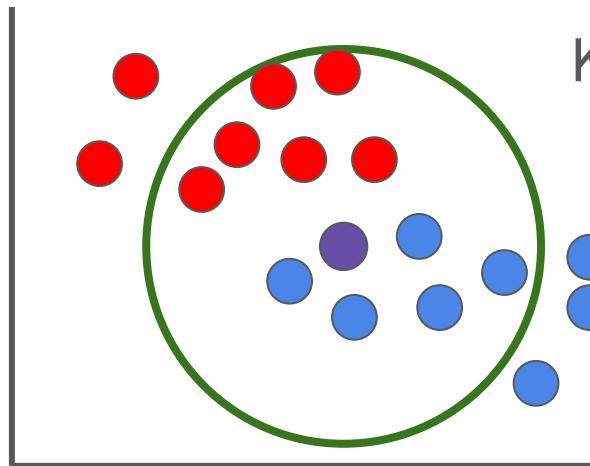
K近傍法

- 注意点

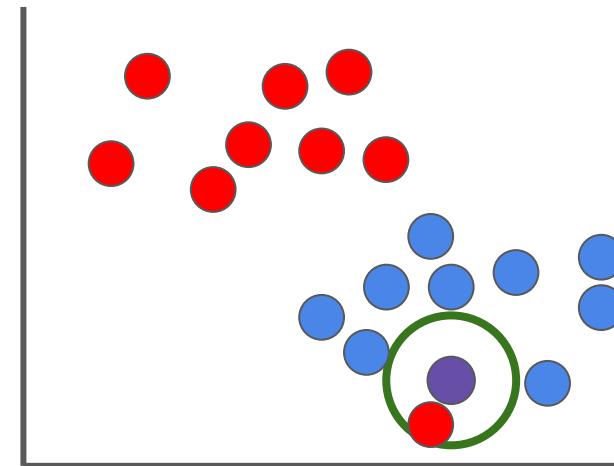
Kの値が大きすぎると予測の精度は悪化する

Kの値が1のとき外れ値の影響を受ける

K=1



K=11



K近傍法

- 注意点

全てのデータとの距離を計算する必要があるため、

データ量が多くなると計算コストが多くなる

→データ量を少なくすると計算コストも精度も低くなる

→精度と計算コストはトレードオフの関係である

K近傍法

・注意点

データの次元が大きくなると距離の差が小さくなり、

精度が低くなってしまう可能性が高い

→以上のように高次元になると生じる問題を次元の呪いという

→数学的に高次元になると距離の差が小さくなることが

分かっており、この現象を球面集中現象と呼ぶ



教師あり学習 時系列分析

時系列分析

時間的な順序で収集されたデータを分析し、特徴を抽出する手法

- ARモデル（自己回帰モデル）
- VARモデル（ベクトル自己回帰モデル）
- MAモデル（移動平均モデル）
- ARMAモデル（自己回帰移動平均モデル）
- ARIMAモデル（自己回帰和分移動平均モデル）

| 時系列分析

- ARモデル（自己回帰モデル）

過去のデータ使ってある時点の値を予測するモデル

→1日前、2日前、3日前・・・のように過去のデータを使用する

- VARモデル（ベクトル自己回帰モデル）

自己回帰モデルを多変量に拡張したもの

→説明変数を複数にしたモデルと考えることができる

| 時系列分析

- MAモデル（移動平均モデル）

ある時刻の値を過去のデータの移動平均を用いて予測するモデル

- ARMAモデル（自己回帰移動平均モデル）

ARモデルとMAモデルを組み合わせたモデルのこと

| 時系列分析

- ARIMAモデル（自己回帰和分移動平均モデル）

差分を取った時系列データにARMAモデルを適用したモデル

→時系列分析の手法は似たような用語が多いので

1つ1つ用語とキーワードを丁寧に抑えておくことが大切になる

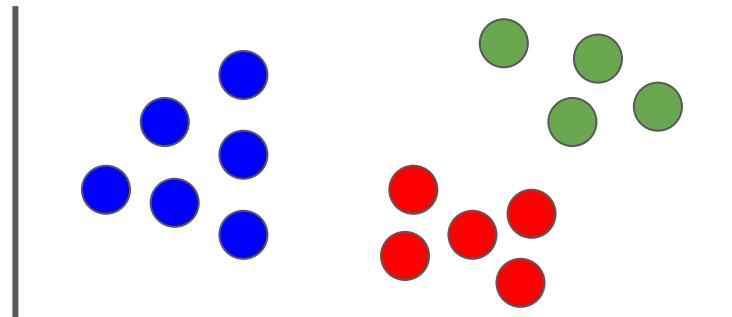


k-means法 ウォード法・群平均法

| k-means法・ウォード法・群平均法

・クラスタリング（クラスタ分析）

データの類似度に基づいてクラスタ分けする手法のこと
→大量のデータをコンピュータが自動的に分類してくれる
クラスタ分けされたものは人間が見て解釈する必要がある



| k-means法・ウォード法・群平均法

- ・ソフトクラスタリング

各データが1つ以上のクラスタに属するようなクラスタリング

- ・ハードクラスタリング

各データが1つのクラスタに属するようなクラスタリング

| k-means法・ウォード法・群平均法

- 階層クラスタリング

データ間の類似度をもとに、最も似たデータ同士から順に階層を作りながらクラスタに分けていく手法のこと

- 非階層クラスタリング

階層を作らずにデータ間の類似度から、クラスタに分けていく手法のこと

| k-means法・ウォード法・群平均法

- k-means法（階層なし、ハードクラスタリング）

データをあらかじめ設定したクラスタの数（k個）に分類する

→重心を活用してデータをクラスタに分類していく

- 重心

クラスタに属する各データの中心となる点のこと

| k-means法・ウォード法・群平均法

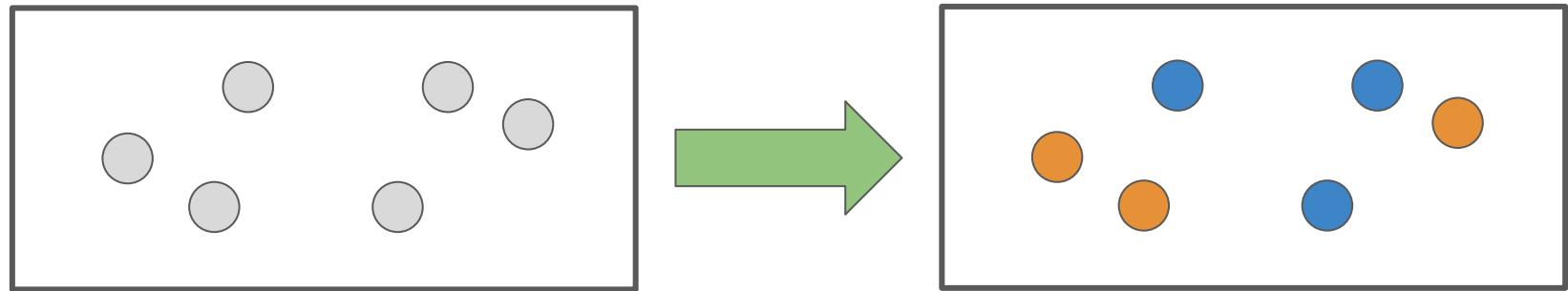
- k-means法（階層なし、ハードクラスタリング）

- 1.データをランダムにk個のクラスタに分ける
- 2.各クラスタの重心を求める
- 3.各データと重心の距離を求め、
データを最も近い重心のクラスタに割り振り直す

→2~3を重心が変化しなくなるまで何度も繰り返す

k-means法・ウォード法・群平均法

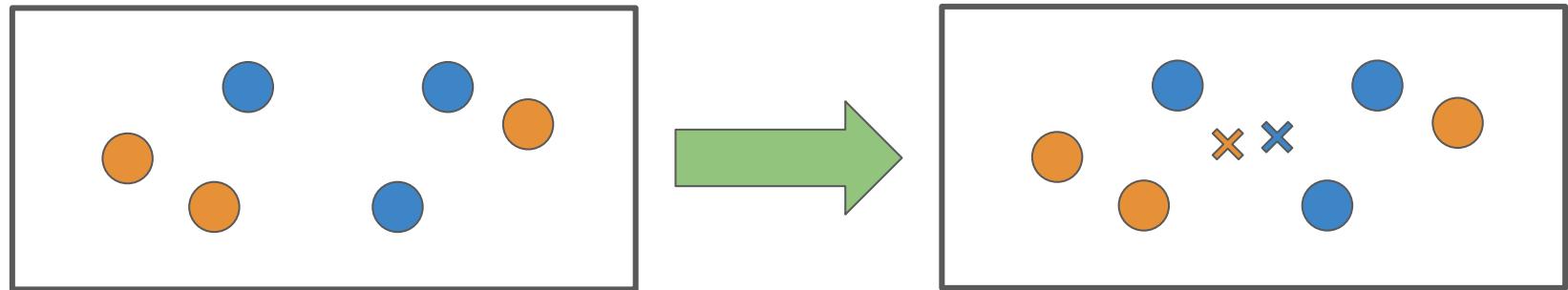
- k-means法 (階層なし、ハードクラスタリング)



データをランダムに
k個のクラスタに分ける

k-means法・ウォード法・群平均法

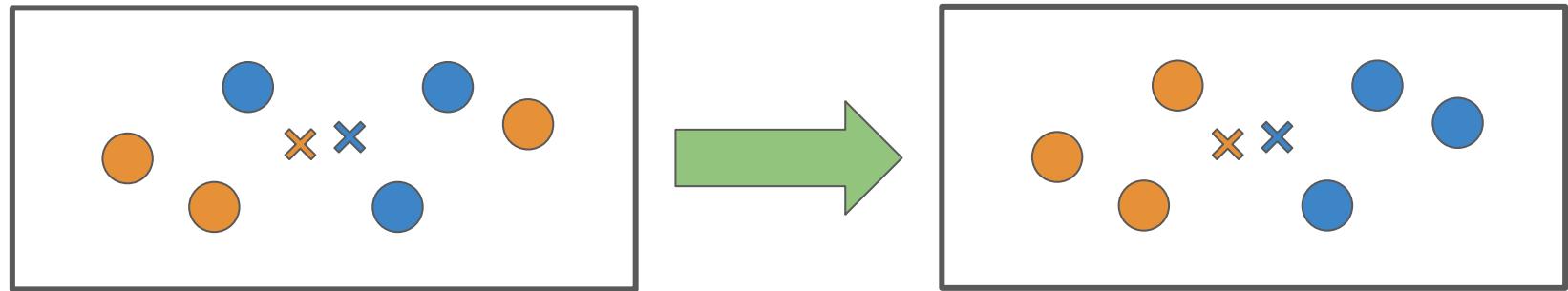
- k-means法 (階層なし、ハードクラスタリング)



各クラスタの重心を
求める

k-means法・ウォード法・群平均法

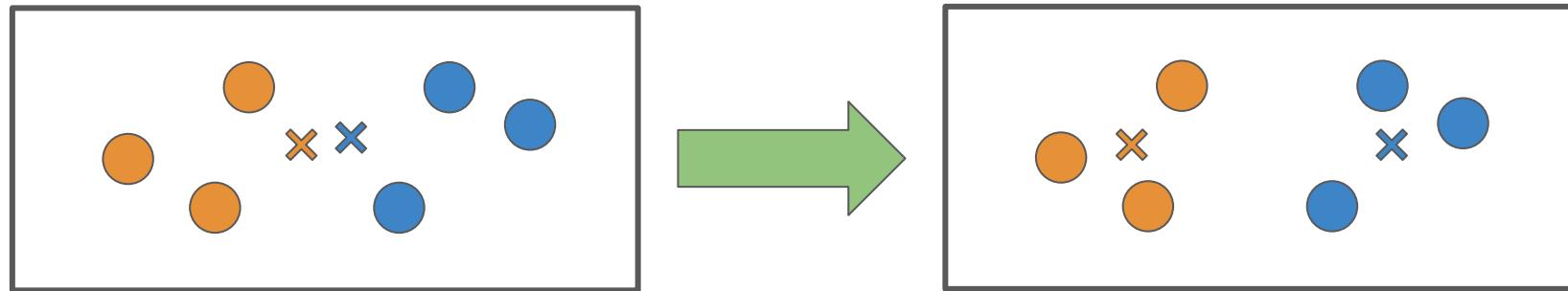
- k-means法（階層なし、ハードクラスタリング）



データを最も近い重心の
クラスタに割り振り直す

k-means法・ウォード法・群平均法

- k-means法 (階層なし、ハードクラスタリング)



各クラスタの重心を
求める

| k-means法・ウォード法・群平均法

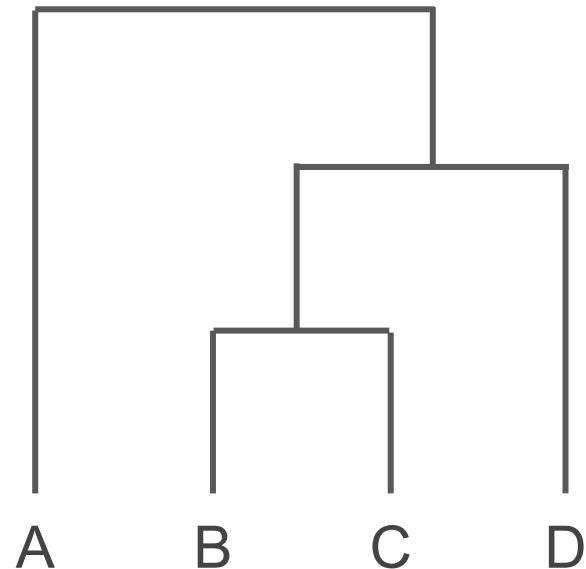
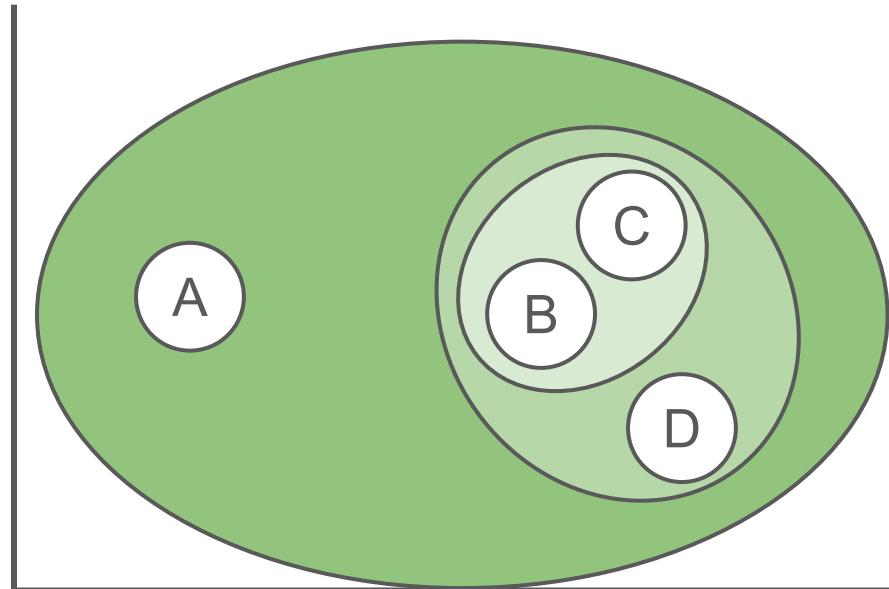
- ウォード法（階層あり、ハードクラスタリング）

分散が最小になるようにデータ同士をクラスタにまとめていく
→クラスタが1つ、またはあらかじめ設定したクラスタになるまで
データ同時をまとめていく処理を繰り返し行う

→途中経過を樹形図（デンドログラム）で表すことができる

k-means法・ウォード法・群平均法

- ・ウォード法（階層あり、ハードクラスタリング）



| k-means法・ウォード法・群平均法

- 群平均法（階層あり、ハードクラスタリング）

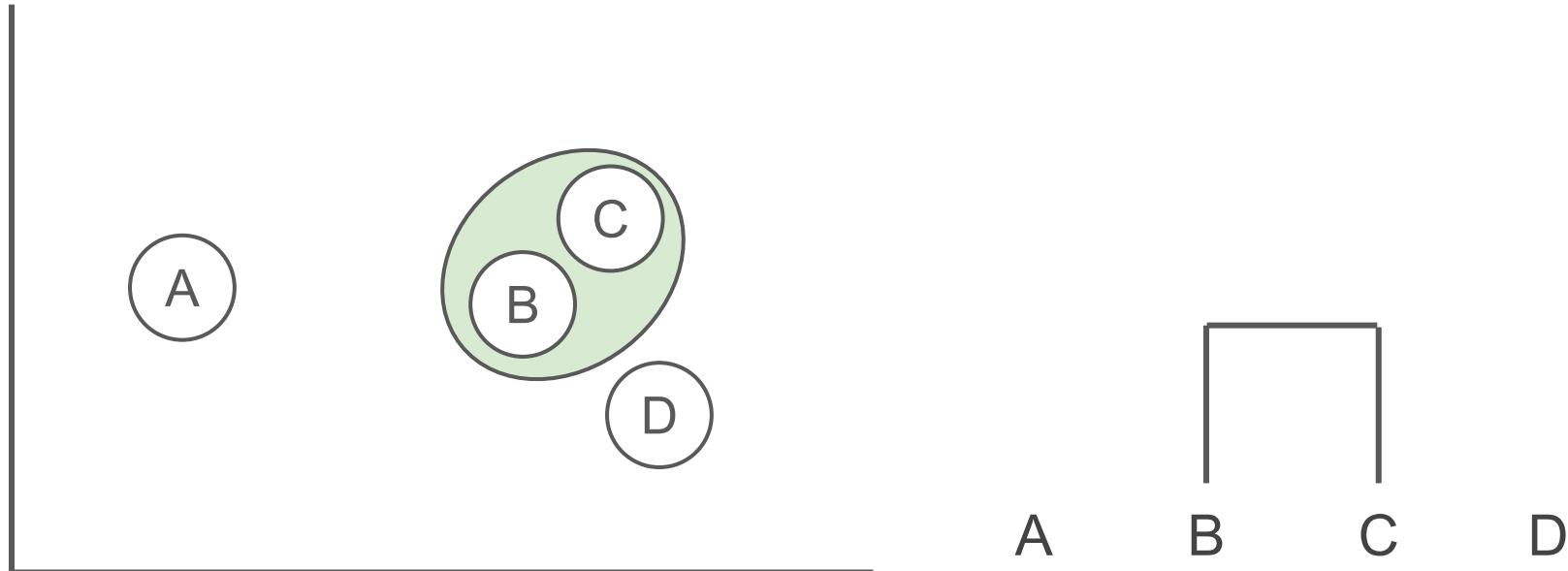
クラスタが1つ、またはあらかじめ設定したクラスタになるまで
データ同時をまとめていく処理を繰り返し行う

→データをまとめるとき距離の平均を使用する

→群平均法はクラスタ間の距離を測定する手法の
呼び名として使用されることもある

| k-means法・ウォード法・群平均法

- 群平均法（階層あり、ハードクラスタリング）



| k-means法・ウォード法・群平均法

- 群平均法（階層あり、ハードクラスタリング）

AとBの距離、AとCの距離の平均をAとBCの距離とし、

BとDの距離、CとDの距離の平均をBCとDの距離とする

→AとBC、BCとD、AとDの距離のうち近い距離の要素をまとめる

→距離の平均を使用してデータをまとめしていく



主成分分析・t-SNE法 特異値分解・多次元尺度構成法

主成分分析・t-SNE法・特異値分解・多次元尺度構成法

・主成分分析（PCA）

データの特徴量の関係性から特徴量をまとめて

より少ない特徴量に要約していく手法

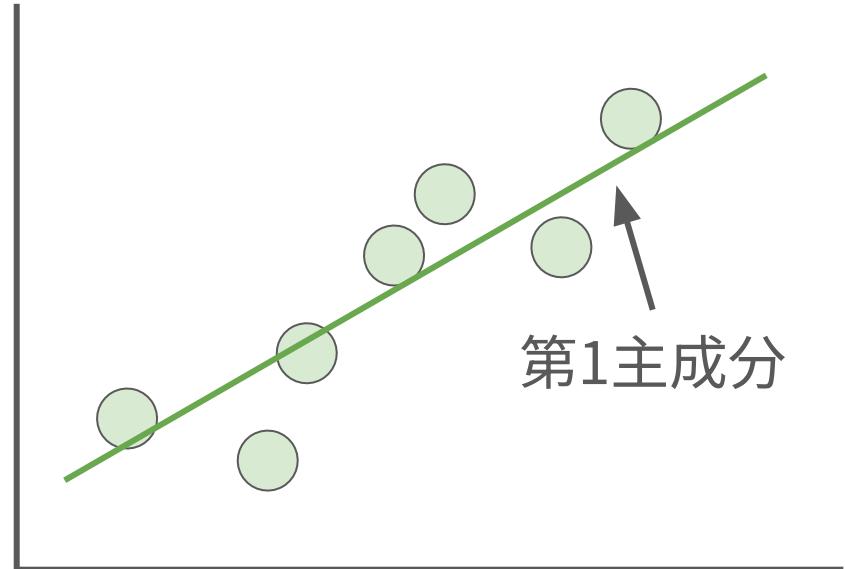
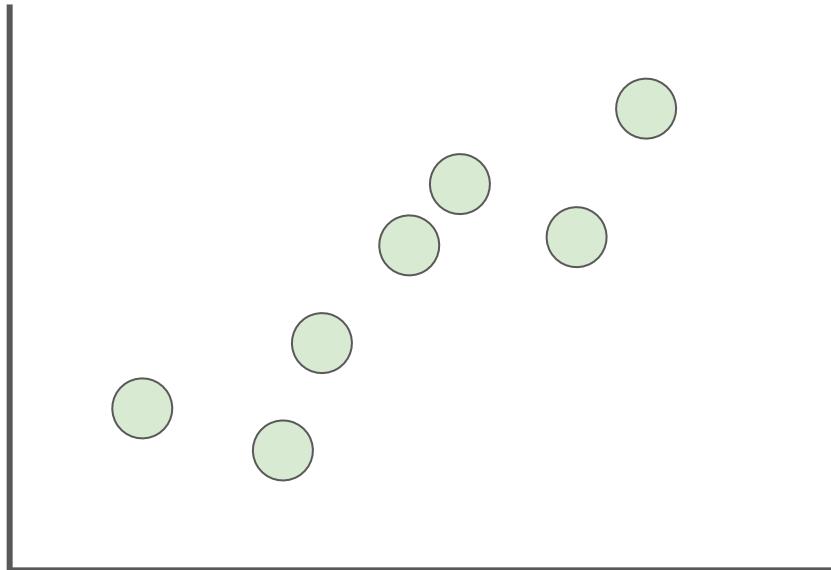
→まとめられた特徴量を**主成分**と呼ぶ

→データを要約することを**次元削減**といい、計算量を減らせる

主成分分析・t-SNE法・特異値分解・多次元尺度構成法

- 主成分分析 (PCA)

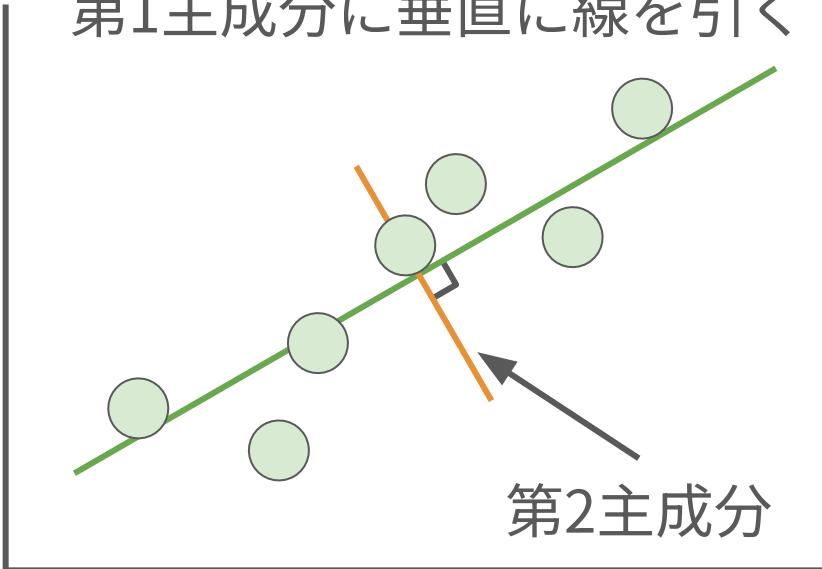
最もばらつきが
大きい方向に軸を引く



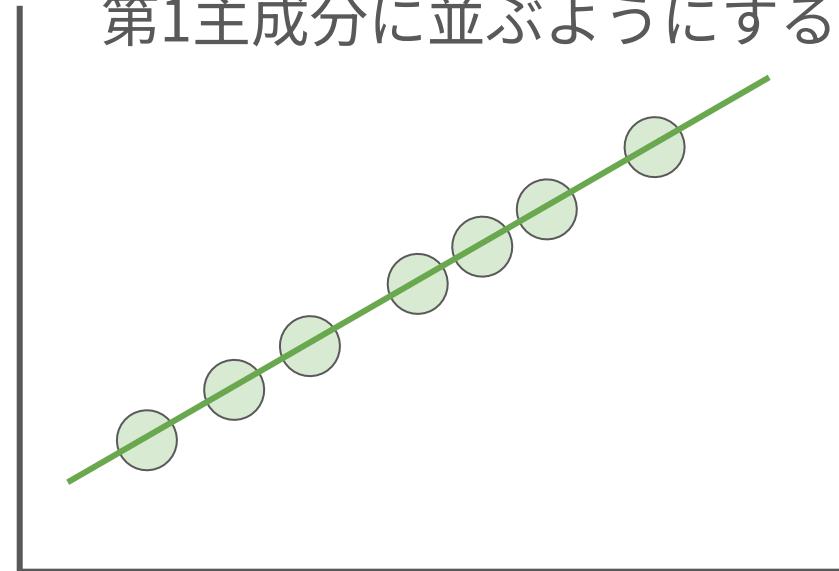
主成分分析・t-SNE法・特異値分解・多次元尺度構成法

- 主成分分析 (PCA)

第1主成分に垂直に線を引く

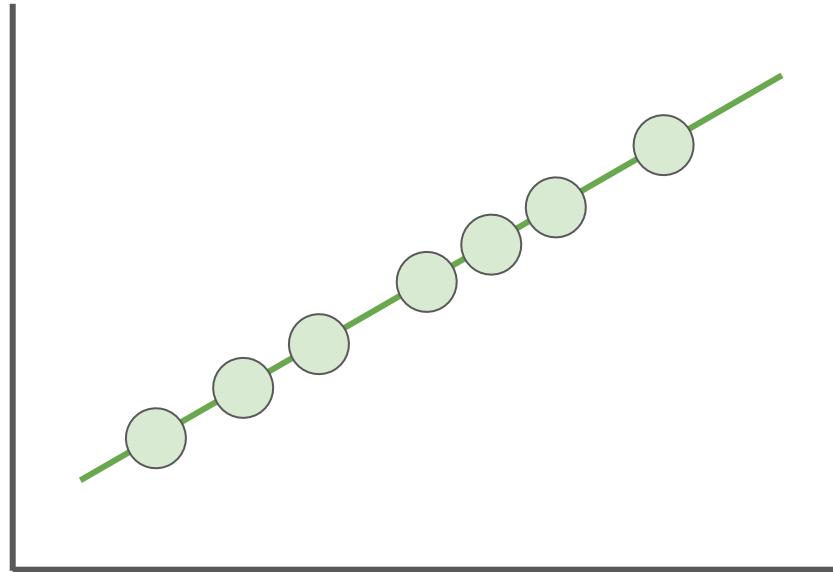


第1主成分に並ぶようにする



主成分分析・t-SNE法・特異値分解・多次元尺度構成法

- 主成分分析 (PCA)



2次元の情報を1次元に圧縮



主成分分析・t-SNE法・特異値分解・多次元尺度構成法

- 次元削減

次元を削減する方法は、主成分分析以外にも

t-SNE法、特異値分解、多次元尺度構成法などが有名

- t-SNE法

t分布（統計で使用する分布の1つ）を活用して

次元を削減する手法のこと

主成分分析・t-SNE法・特異値分解・多次元尺度構成法

・特異値分解 (SVD)

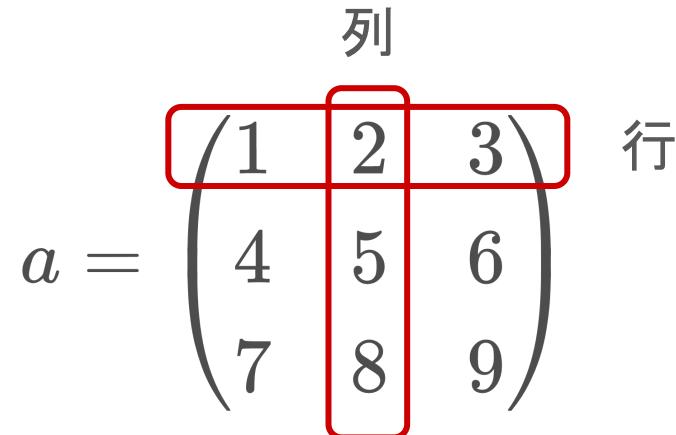
ある行列を分解して複数の行例の積で表現する手法のこと

→行列とは値や文字を縦横に並べたもの

列

$$a = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$$

行



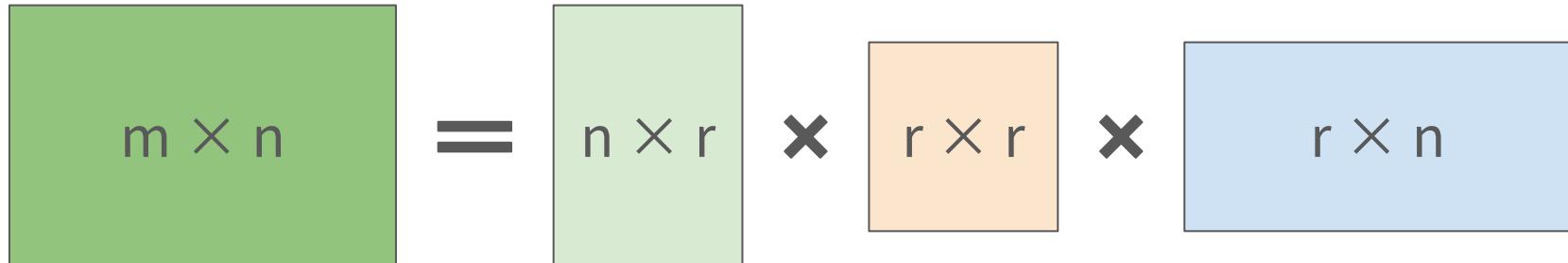
主成分分析・t-SNE法・特異値分解・多次元尺度構成法

・特異値分解 (SVD)

重要度の高い物だけを抽出し、

重要度が低いものを削除して、次元を下げる処理を行う

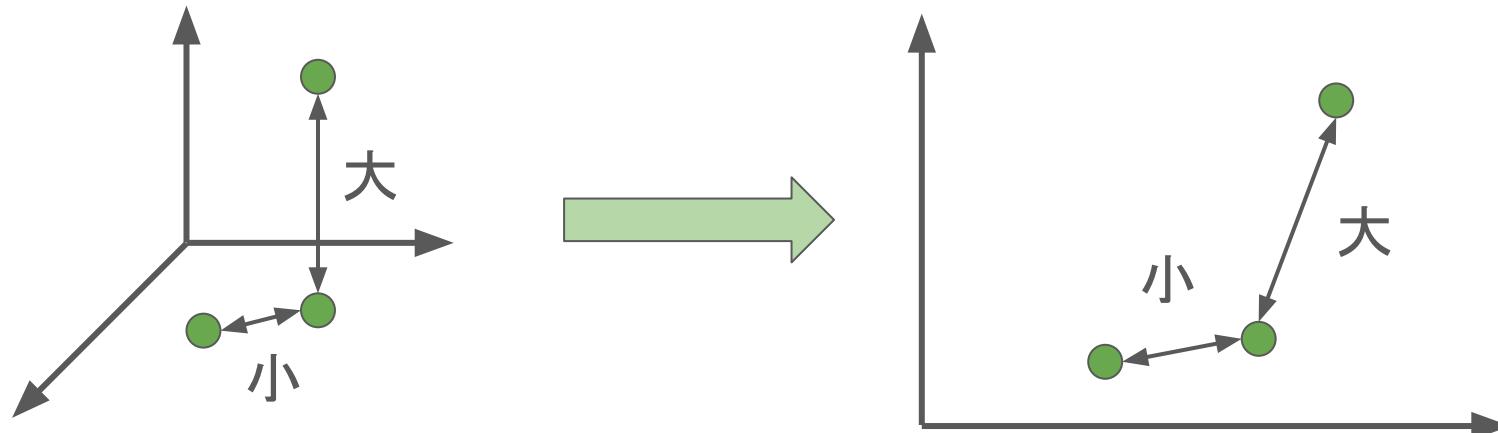
→ $m \times n$ は m 行 n 列を意味する

$$\begin{matrix} m \times n \end{matrix} = \begin{matrix} n \times r \end{matrix} \times \begin{matrix} r \times r \end{matrix} \times \begin{matrix} r \times n \end{matrix}$$


主成分分析・t-SNE法・特異値分解・多次元尺度構成法

・多次元尺度構成法（MDS）

データの関係性（類似度）やデータの距離を大小関係を維持したまま低次元空間で表現すること





教師なし学習 トピックモデル

| トピックモデル

・ トピックモデル

文章などを複数のクラスタに分類するモデルのこと

→k-means法などと異なり複数のクラスタに分類が可能

→ニュース記事を複数のカテゴリに分類するときなどに使用

「経済」 「芸能」 「スポーツ」など適切なカテゴリに分類

| トピックモデル

- トピックモデル

文章を分類するとき文章間の類似度を計算している

→似た文章をレコメンドすることも可能

→LSA、PLSA、LDAなどがトピックモデルの代表的な手法

| トピックモデル

- LSA（潜在意味解析）

文章で使われている単語の頻度から、
その文章のカテゴリを推定する手法のこと

→ 「国会」、「総理大臣」、「霞ヶ関」、「予算」など
という単語が多いならば「政治」に関する文章だと推定可能

| トピックモデル

- LSA（潜在意味解析）

→ 「車」、「自動車」のように意味的に似た単語を

別の単語として処理していくのは非効率

→ 意味的に似た単語をグルーピングしていく（特異値分解を使用）

- PLSA（確率的潜在意味解析）

LSA（潜在意味解析）に確率の概念を組み合わせた手法

| トピックモデル

- LDA（潜在的ディリクレ配分法）

PLSA（確率的潜在意味解析）を発展させたトピックモデル
→ディリクレ分布を使用している（分布の1つ）

→LSA（潜在意味解析）と同様に**特異値分解**を活用している



協調フィルタリング コンテンツベースフィルタリング アソシエーション分析

| 協調フィルタリング

- 協調フィルタリング

レコメンデーションで使われる手法の1つ

→ユーザーの興味があるものを学習し、

興味がありそうな情報を提案するシステム

→AmazonなどECサイトでオススメが表示される仕組みのこと

協調フィルタリング

・協調フィルタリング

自分と類似したユーザーが購入した商品を薦める

→動画配信ならば類似したユーザーが見た動画を薦める

提案するためにはある程度のデータ量が必要になる

→データ量が少ない場合、ユーザーに適切な情報を

提示することができない（コールドスタート問題）

| 協調フィルタリング

・協調フィルタリング

商品などの特徴を活用して類似した商品をユーザに提案することでコールドスタート問題を回避することができる
→このような手法をコンテンツベースフィルタリングという

→協調フィルタリングとコンテンツベースフィルタリングにはそれぞれにメリットとデメリットが存在する

| 協調フィルタリング

・協調フィルタリング

コンテンツベースフィルタリングはコールドスタート問題を回避することができるが、似た商品しか提案できない

- 協調フィルタリングは多様な商品を提案することができるがコールドスタート問題が発生してしまう
- 自身が知らない異なるカテゴリーの商品が提案される

| 協調フィルタリング

・アソシエーション分析

- 大量のデータをもとにデータの関連性を見つけ出す手法のこと
→顧客の購入履歴から関連性を見つけ出すことが可能である
　　オムツを購入する人はビールも一緒に購入する傾向にある　など
→実際に、オムツの近くにビールをおくと、ビールの売上が上がる

→人間が気が付かない関連性を見つけ出すことができる



半教師あり学習 FixMatch

FixMatch

- FixMatch

擬似ラベリングと一致性正則化を組み合わせた手法である

- 擬似ラベリング

正解ラベルが設定されていないデータに対して、
一定の方法により、擬似ラベルを作成すること
→モデルが予測したラベルを設定する方法 などがある

FixMatch

- FixMatch

- 一致性正則化

あるデータとそのデータに加工したデータの出力結果ができるだけ一致するように正則化する方法

FixMatch

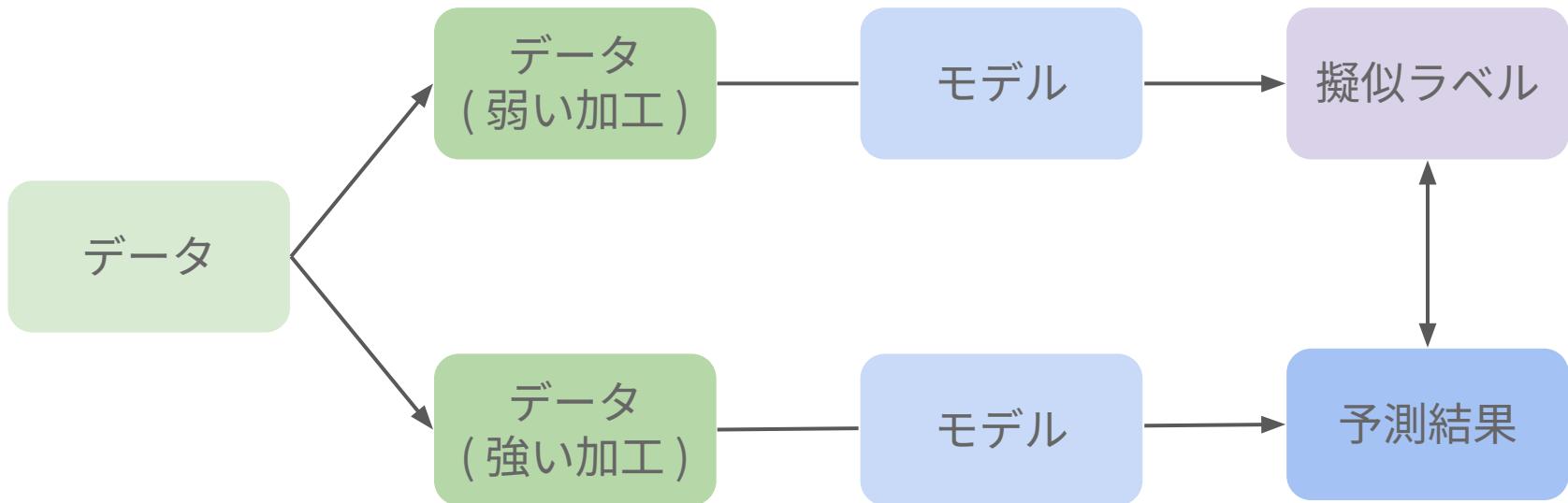
- FixMatch

正解ラベルが付与されていないデータに
弱い加工（回転など）をして擬似ラベルを作成

→同じ元のデータに対して強い加工したデータの予測結果が
擬似ラベルに一致するように学習していく

FixMatch

- FixMatch



強化学習



強化学習

・強化学習

与えられた環境下で、報酬が最大となるように

最適な行動を学習していく手法

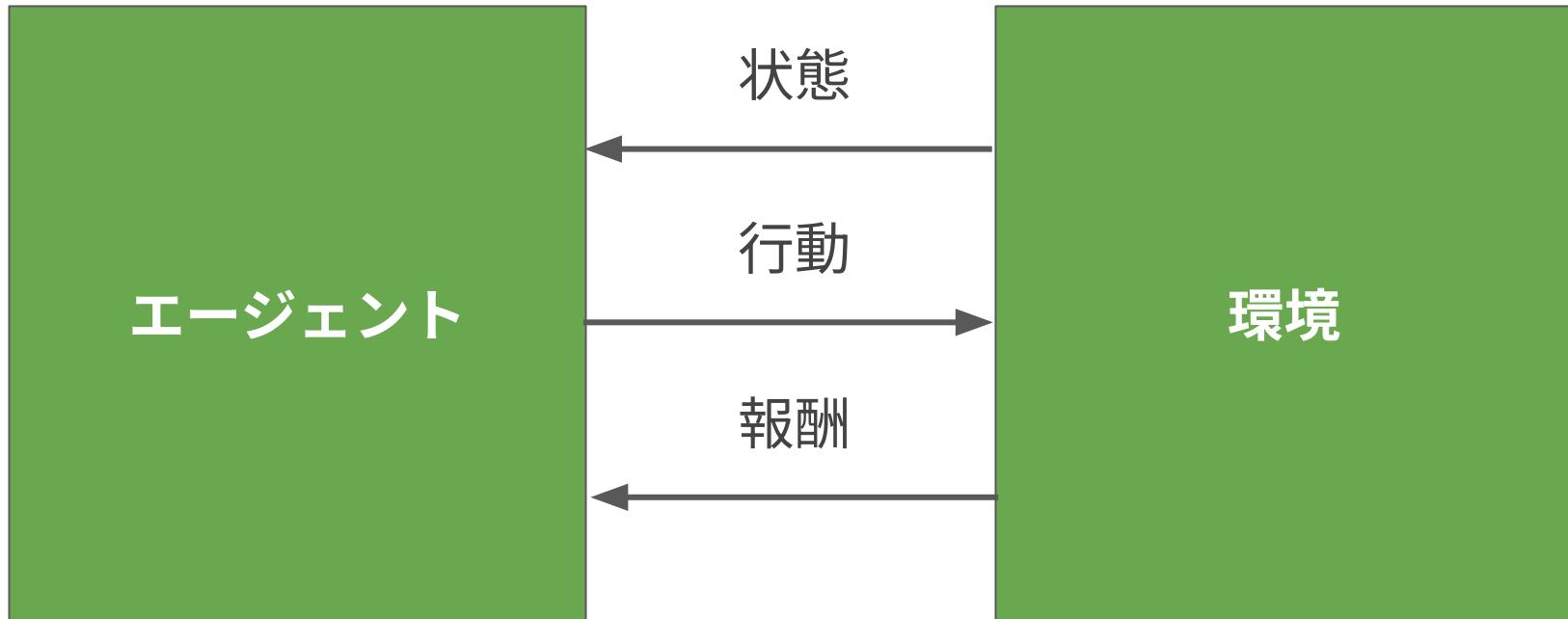
→報酬とはエージェントの行動を評価するスコアのこと

→エージェントとは自律的に行動するシステムやプログラムのこと

状態、行動、報酬 が大切なキーワードになる

強化学習

- 強化学習



強化学習

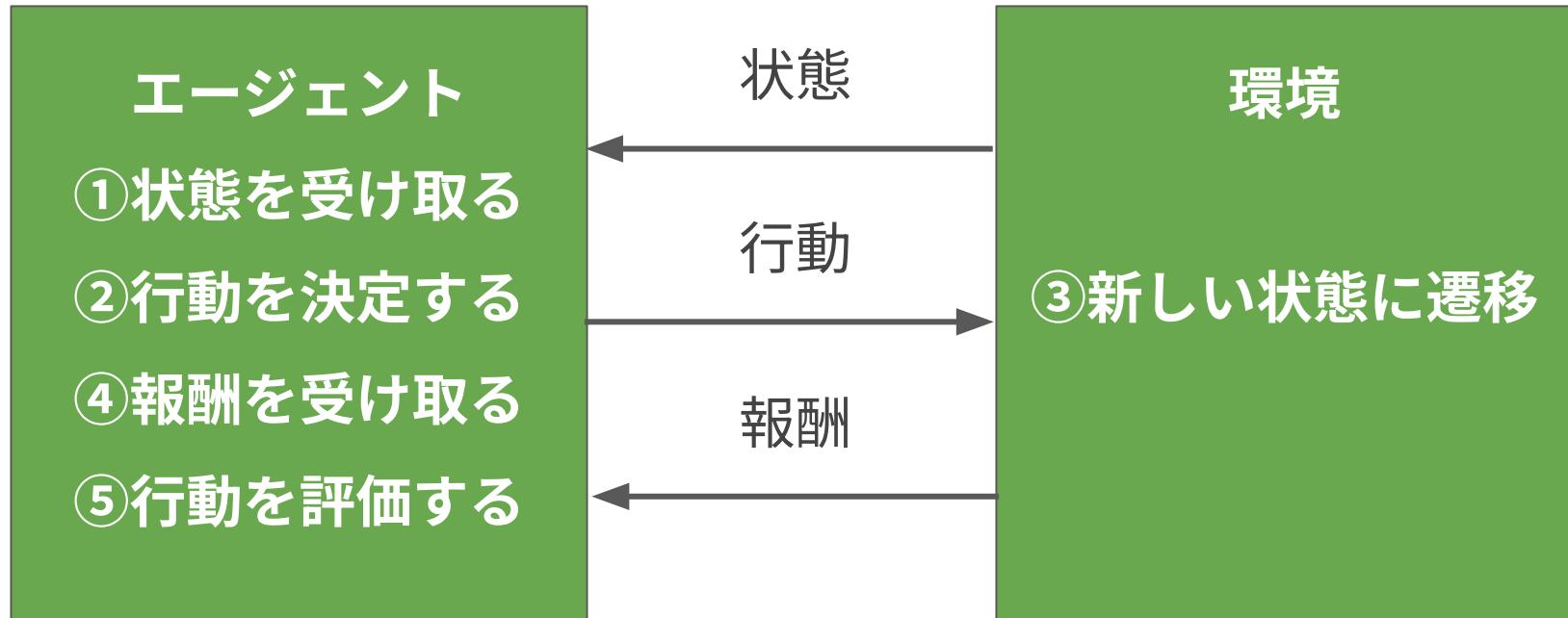
・強化学習

1. 環境から状態を受け取る
2. 状態をもとに行動を決定する
3. 行動の結果、新しい状態に遷移する
4. 遷移に応じた報酬を受け取る
5. 受け取った報酬をもとに行動を評価する

→ 1～5 を繰り返し行っていく

強化学習

・強化学習



強化学習

・強化学習

与えられた環境に対する行動の開始から
終了までの1回分の試行をエピソードという

→何度も繰り返しながら最適な行動を学習していく手法である
→報酬を上手く設定できれば、各行動に対して報酬がなくても
学習することができるという特徴がある

強化学習

・強化学習

報酬和が最大化するように行動を学習していく

→行動Aを選択すると報酬和は最大化しやすいので優先的に選択、

行動Bを選択すると報酬和は低下しやすいので選択しない など

→短期的に報酬が少なくとも、最終的に報酬和が最大になる可能性

短期的な報酬だけで行動を決定しているわけではない

強化学習

・強化学習

報酬和は各時点における報酬を足し合わせたものである

→開始時刻 t で獲得できる報酬を R_t とすると報酬和 G_t は

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} \dots$$

→ γ は割引率のことで、未来に獲得できる報酬を現在の報酬に変換するために使用されるハイパーパラメータである

強化学習

・強化学習

未来の報酬は不確実性が高いため割り引いて考える（金融）

→現在の1万円と5年後の1万円ならば現在のほうが価値が高い

→未来の報酬と現在の報酬の価値が同じようになるように変換

割引率が0.9ならば、1年後の1万円と現在の9,000円が同じ価値

→ $10,000 \times 0.9$ から9,000円を計算している

強化学習

・強化学習

割引率が0.9の場合、2年後の1万円は現在の価値にすると
→ $10,000 \text{円} \times 0.9 \times 0.9$ から8,100円になる

→未来は不確実性が高いため割り引いて考える必要がある

→以上のような考え方を強化学習でも導入している

強化学習

・強化学習

限られた資源（時間やコスト）内で最適な行動を見つけ出すことは非常に難しい

→活用と探索のバランスをどうするかも考える必要がある

活用：既存情報を使って現状において最適な行動を選択すること

探索：新しい情報を獲得するための行動を選択すること

強化学習

・強化学習

限られた資源のもとで最適な行動（活用・探索）を選択するアルゴリズムのことをバンディットアルゴリズムという

→エージェントの行動基準のことを方策と呼ぶ

活用と探索をどのような割合で選択するのかという基準

→方策には ϵ -greedy方策やUCB方策などがある

強化学習

- 方策の種類

- ϵ -greedy方策

既存情報から最適な行動を選択するが、確率 ϵ で探索を選択

- UCB方策

探索時にあまり選択されていない行動を優先的に選択

強化学習

- 強化学習

環境に対してマルコフ性と呼ばれる仮定が設定されている

- マルコフ性

状態が遷移する確率は、過去の状態に依存せず、

現在の状態で決定されるという特性のこと

→現在の状態と行動を与えれば将来の状態に遷移する確率を導出

強化学習

・マルコフ性

本来ならば、行動、状態、報酬などを計算するときは、過去の状態を考慮した方が自然であるが計算が複雑になる
→計算コストが高すぎて上手く扱うことができない

→マルコフ性を満たす遷移過程モデルのことを
マルコフ決定過程モデルという



強化学習のアプローチ方法

| 強化学習のアプローチ方法

・ 強化学習

最適な方策を見つけ出すことは難しいため、
行動の価値が最大化するように学習を行うという
アプローチ方法が考案される（価値ベースの考え方）

→価値を評価するために価値関数を使用する

価値関数には、行動価値関数や状態価値関数がある

| 強化学習のアプローチ方法

- 行動価値関数

ある状態において行動した結果、
将来的に得られる報酬の合計を返す関数

- 状態価値関数

ある状態において、将来的に得られる報酬の合計を返す関数

| 強化学習のアプローチ方法

・ 強化学習

行動価値関数から返ってきた値を**Q値**という
→**Q値**が最も高い行動をとるように学習していく
手法のことを**Q学習**という

→Q値を最適化する手法には、**Q学習**や**SARSA**がある
Q学習や**SARSA**は、**TD学習**の1種である

| 強化学習のアプローチ方法

- 強化学習
 - TD学習

エピソードの終了まで待たずに状態価値関数や
行動価値関数を更新する手法のこと

→行動する前のQ値と行動した得られたQ値の差分を
使用して学習を行っている

| 強化学習のアプローチ方法

- 強化学習
 - Q学習とSARSA

Q学習とSARSAの手法はほとんど同じだが、

価値関数を更新する式が少し異なっている

→行動後の価値を求めるときの前提が少し異なる

| 強化学習のアプローチ方法

- 強化学習

Q学習やSARSAは、価値反復法の手法の1つである

- 価値反復法

エージェントと環境が情報をやり取りながら

価値関数を更新していく手法のこと

| 強化学習のアプローチ方法

・ 強化学習

- 価値ベースの手法（Q学習など）では、
状態価値を0で初期化し、状態価値を更新していく
→状態価値が収束したら学習終了する
- 行動の選択肢が多い場面などは、価値を計算するコストが高く
最適な方策を見つけることが困難である

| 強化学習のアプローチ方法

・ 強化学習

方策をパラメータを持った関数と定義して

方策の価値が最大になるようにパラメータを

学習するというアプローチも存在する（**方策ベース**）

→方策を求める代表的なアプローチ方法として**方策勾配法**がある

→特に行動の選択肢が多い場面で使用される手法

| 強化学習のアプローチ方法

- 強化学習

- 方策勾配法

状態を入力し、行動を出力する**方策関数**を使用して

方策のパラメータを推定している

→**方策勾配法**を使用した代表的なアルゴリズムが**REINFORCE**

→**REINFORCE**はアルファ碁で活用されているアルゴリズム

| 強化学習のアプローチ方法

- 強化学習

- 方策勾配法

REINFORCEを改良したモデルとして**PPO**がある

→方策のパラメータを更新しすぎると不安定になっていた

→**PPO**では過去の方策のパラメータを参考に

パラメータの更新量を抑制することで**安定した学習が可能**

| 強化学習のアプローチ方法

- Actor-Critic

価値ベースと方策ベースの考え方を組み合わせた手法

→行動を選択するActorと評価するCriticで構成されている

CriticがActorの行動を評価し、Actorが方策を更新していく

→Actor-Criticを応用したA3Cがある

| 強化学習のアプローチ方法

- A3C (Asynchronous Advantage Actor-Critic)

複数のエージェントが非同期で並列に学習する手法

→複数のエージェントを使用するため学習が高速である

- Asynchronous (非同期)

他の処理の実行を待たずに処理できること

→他のエージェントの学習を待たずに学習を行える

| 強化学習のアプローチ方法

- A3C (Asynchronous Advantage Actor-Critic)
 - Advantage

数ステップ先を考慮してQ値などを更新していく
→今まででは1ステップ後の情報からQ値などを更新
2ステップや3ステップ後を考慮して値などを更新

| 強化学習のアプローチ方法

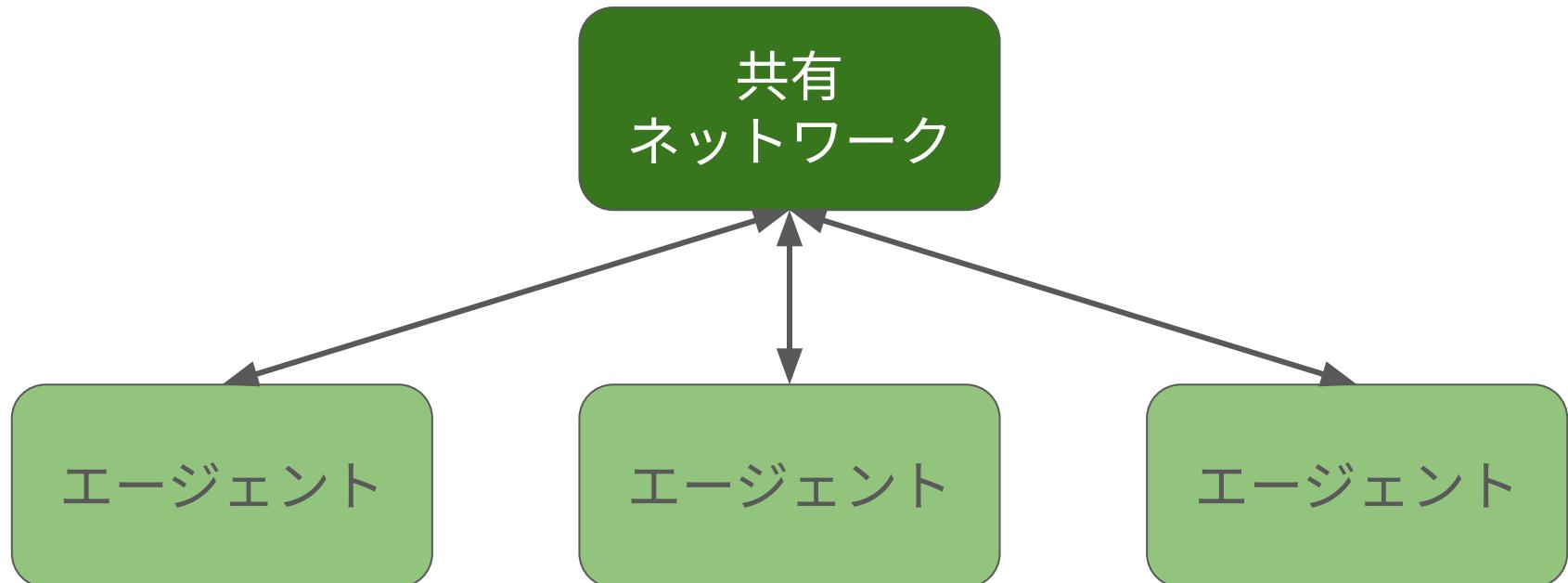
- A3C (Asynchronous Advantage Actor-Critic)

A3Cでは、エージェンは学習によりネットワークの重みを学習
→エージェンは自分の持っている情報をもとに適当な
タイミングで**共有ネットワーク** (global network) を更新する

→定期的に各エージェントは**共有ネットワーク**の重みと同期する
同期：同じ状態にすること

| 強化学習のアプローチ方法

- A3C (Asynchronous Advantage Actor-Critic)





モデルの評価の手法

| モデルの評価の手法

・モデル評価の手法

- 学習が終わったら、**モデルの性能を評価することが大切**
→未知のデータに対してどの程度予測できているのか
評価用のデータ（**テストデータ**）を使い性能を評価していく
- 学習で使用していないデータを評価用のデータとする
学習用データを使用しても未知データに対する**性能は分からぬ**

| モデルの評価の手法

- 交差検証

データを訓練データとテストデータに分けて

モデルの評価を行うこと

→訓練データを訓練データと検証データに分ける場合もある

検証データはハイパーパラメータの調整などに使用

訓練データ

検証データ

テストデータ

| モデルの評価の手法

- 交差検証の種類
 - ホールドアウト検証
 - k-分割交差検証
 - 1つ抜き（Leave One Out）法 など

| モデルの評価の手法

・ホールドアウト検証

一度だけデータを訓練データとテストデータに分けて

モデルの評価を行う手法のこと（7対3、8対2が多い）

→テストデータを活用してモデルの性能を評価

訓練データ

テストデータ

| モデルの評価の手法

・ホールドアウト検証

データが少ない場合、評価の信頼性が低下してしまう特徴

→訓練データとテストデータに偏りが生じることがある

特定の特徴のデータがテストデータにしかないなど

→一方、他の手法と比べて検証方法がシンプルであるため、

データが多くても短期間で性能評価が可能

モデルの評価の手法

• k-分割交差検証

データを複数個（k個）に分割し、そのうち1つをテストデータ
残りを訓練データとして学習と評価を行う方法



| モデルの評価の手法

- k-分割交差検証

k回、評価を行うため評価の信頼性が高い

→学習回数が増えるため、計算コストが高くなってしまう

→汎用性が高いため、多くのモデルで実施される手法

| モデルの評価の手法

- 1つ抜き (Leave One Out) 法

データ全体のうち1つをテストデータ、

残りを訓練データとしてモデルの性能を評価する方法

→訓練データとテストデータを入れ替えてテストを行う

100データなら100回評価を行う

→ジャックナイフ法とも呼ばれる

| モデルの評価の手法

- 1つ抜き（Leave One Out）法



訓練データ

•
•
•

訓練データ

| モデルの評価の手法

・データリーケージ

訓練データ・検証データにテストデータにすべき

未知のデータ（本来得られないデータ）が入ってしまうこと

→時系列データをもとに学習していくとき

訓練データや検証データにはテストデータよりも

未来のデータを入れるべきではない

| モデルの評価の手法

- ・データリーケージ

テストデータよりも未来のデータで学習しているため

テストデータで性能を評価すると

本来の性能よりも高くなってしまう特徴がある

時間軸



分類モデルの評価①

| 分類モデルの評価

・分類モデルの評価

分類モデルは混同行列を基準にして考えていく

	陽性と判断	陰性と判断
正解が陽性	真陽性 (TP : True Positive)	偽陰性 (FN : False Negative)
正解が陰性	偽陽性 (FP : False Positive)	真陰性 (TN : True Negative)

| 分類モデルの評価

・分類モデルの評価

偽陽性を第一種の過誤、偽陰性を第二種の過誤とも呼ぶ

→混同行列は2値分類などの結果をまとめた表であるが、
多クラス分類でも使用することができる

| 分類モデルの評価

・分類モデルの評価

画像を犬か、それ以外かに分けるモデルのテストを行った

1. 「犬」の画像を、正しく「犬」と判断した（**真陽性**）
2. 「犬」の画像を、間違って「犬以外」と判断した（**偽陰性**）
3. 「犬以外」の画像を、間違って「犬」と判断した（**偽陽性**）
4. 「犬以外」の画像を、正しく「犬以外」と判断した（**真陰性**）

| 分類モデルの評価

・分類モデルの評価

	犬と判断	犬以外と判断
正解が犬	真陽性 (TP : True Positive)	偽陰性 (FN : False Negative)
正解が犬以外	偽陽性 (FP : False Positive)	真陰性 (TN : True Negative)

| 分類モデルの評価

- 代表的な指標

1. 正解率

2. 適合率

3. 再現率

4. F値

| 分類モデルの評価

1. 正解率

全データのうち正しく予測できた割合のこと

→正しく予測した数を全データ数で割ることを求めることができる

$$\rightarrow \text{正解率} = (\text{TP} + \text{TN}) \div (\text{TP} + \text{FP} + \text{FN} + \text{TN})$$

$$= (\text{真陽性} + \text{真陰性}) \div \text{全データ数}$$

| 分類モデルの評価

2.適合率

陽性と判断したデータのうち実際に陽性だった割合のこと
→正しく陽性と判断した数を陽性と判断した陽性の数で
割ることで求めることができる

$$\begin{aligned}\rightarrow \text{適合率} &= \text{TP} \div (\text{TP} + \text{FP}) \\ &= \text{真陽性} \div (\text{真陽性} + \text{偽陽性})\end{aligned}$$

| 分類モデルの評価

3. 再現率

陽性データの中で、正しく陽性と判断できた割合のこと
→正しく陽性と判断した数を実際の陽性の数で割ることで
求めることができる

$$\begin{aligned}\rightarrow \text{再現率} &= \text{TP} \div (\text{TP} + \text{FN}) \\ &= \text{真陽性} \div (\text{真陽性} + \text{偽陰性})\end{aligned}$$

| 分類モデルの評価

4.F値

適合率と再現率の調和平均のこと

→ $(2 \times \text{適合率} \times \text{再現率}) \div (\text{適合率} + \text{再現率})$ で
求めることができる

→適合率と再現率をまとめて評価したいときに使用する
適合率と再現率はトレードオフの関係である

| 分類モデルの評価

・分類モデルの評価

	犬と判断	犬以外と判断
正解が犬	450 (TP:True Positive)	50 (FN:False Negative)
正解が犬以外	50 (FP:False Positive)	100 (TN:True Negative)

| 分類モデルの評価

・分類モデルの評価

$$\cdot \text{正解率} = (\text{TP} + \text{TN}) \div (\text{TP} + \text{FP} + \text{FN} + \text{TN})$$

$$= 550 \div 650 \doteq 0.85 \text{ (85\%)}$$

$$\cdot \text{適合率} = \text{TP} \div (\text{TP} + \text{FP})$$

$$= 450 \div (450 + 50) = 0.9 \text{ (90\%)}$$

| 分類モデルの評価

・分類モデルの評価

	犬と判断	犬以外と判断
正解が犬	450 (TP:True Positive)	50 (FN:False Negative)
正解が犬以外	50 (FP:False Positive)	100 (TN:True Negative)

| 分類モデルの評価

・分類モデルの評価

$$\cdot \text{再現率} = \text{TP} \div (\text{TP} + \text{FN})$$

$$= 450 \div (450 + 50) = 0.9 \text{ (90\%)}$$

$$\cdot F\text{値} = (2 \times \text{適合率} \times \text{再現率}) \div (\text{適合率} + \text{再現率})$$

$$= (2 \times 0.9 \times 0.9) \div (0.9 + 0.9)$$

$$= 1.62 \div 1.8 = 0.9 \text{ (90\%)}$$

分類モデルの評価②



| 分類モデルの評価

・分類モデルの評価

- ・ 正解率 = $(TP+TN) \div (TP+FP+FN+TN)$
- ・ 適合率 = $TP \div (TP+FP)$
- ・ 再現率 = $TP \div (TP+FN)$
- ・ F値 = $(2 \times \text{適合率} \times \text{再現率}) \div (\text{適合率} + \text{再現率})$

→評価したい項目に合わせて評価指標を選ぶことが大切

| 分類モデルの評価

- 正解率

全体の予測のうち正しい予測の割合で一般的によく使用される
→クラスに属するデータの数のバランスが取れている場合に有効

→クラスの重要性が異なる場合や
クラスに属するデータの数が極端な場合は向かない特徴

| 分類モデルの評価

- 正解率

- クラスの重要性が異なる場合

→病気の陽性か陰性か、再検査が必要かどうか など

	陽性を誤って陰性と 判断する割合	正解率	評価
モデルA	高い	高い	×
モデルB	低い	低い	○

| 分類モデルの評価

- 正解率

- クラスに属するデータの数が極端な場合

クラスAのデータが9,990件、クラスBのデータが10件の場合
→全てのデータをクラスAと判断した場合、正解率は99%

データ数が極端な場合は適合率・再現率・F値を使用する

→モデルを使わずに全てのデータをクラスAと判断しても同じ

| 分類モデルの評価

・適合率

陽性と判断したデータのうち実際に陽性だった割合のこと

→**陽性判断の正確性**を評価する指標である

→適合率を高めると陽性か陰性か曖昧なものは

陰性と判断しやすくなる傾向がある（偽陰性が発生しやすい）

→偽陰性が増えると、**再現率**は低くなりやすい特徴がある

| 分類モデルの評価

• 適合率

偽陰性を発生させたくない場合は適合率を重視しない
→病気を診断するモデルでは偽陰性を発生させたくないため
適合率を重視するよりも、再現率を重視する方が良い

→スパムメール（SPAM）の判断の場合は、誤ってSPAMと
判断したくないため**陽性判断の正確性**が求められる

| 分類モデルの評価

• 再現率

- 陽性データの中で、正しく陽性と判断できた割合のこと
→陽性判断の網羅性を評価する指標である
- 再現率を高めると陽性か陰性か曖昧なものは
 陽性と判断しやすくなる傾向がある（偽陽性が発生しやすい）
→偽陽性が増えると、適合率は低くなりやすい特徴がある

| 分類モデルの評価

• 再現率

病気を診断するとき、偽陽性が増えても大きな問題ではない
→病氣にも関わらず陰性と評価されてしまうことは問題である
適合率を重視するよりも、再現率を重視する方が良い

→SPAMの場合、偽陽性が増えると、SPAM以外のメールが
SPAMフォルダーに振り分けられてしまうため、適合率を重視

| 分類モデルの評価

- F値

適合率と再現率の調和平均のこと

→適合率と再現率の指標をまとめて評価するときに使用

→適合率と再現率のバランスを見る時に使用される

回帰モデルの評価



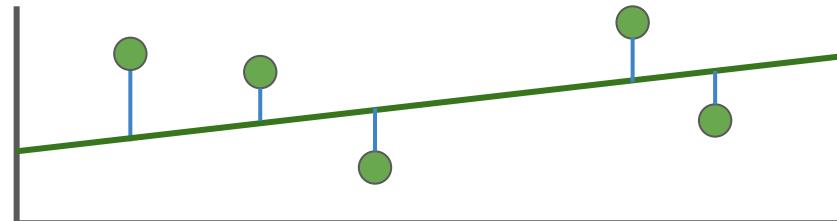
| 回帰モデルの評価

- 回帰モデルの評価

評価指標としてMSE、RMSE、MAE、決定係数などがある

- MSE（平均二乗誤差）

実際の値と予測値の誤差の2乗の平均のこと



| 回帰モデルの評価

- RMSE（二乗平均平方根誤差）

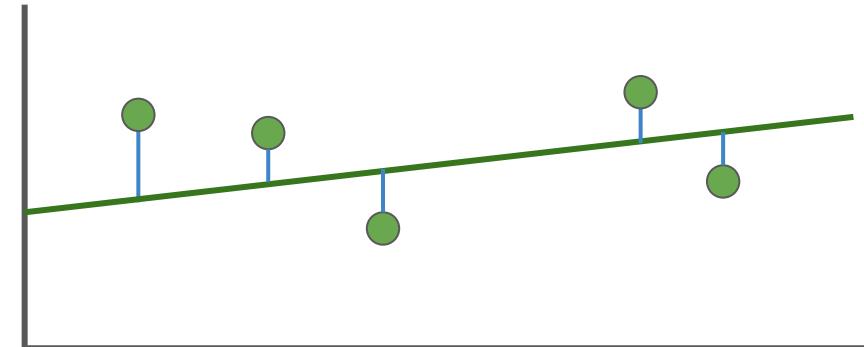
実際の値と予測値の誤差の2乗の平均の平方根

→MSE（平均二乗誤差）の平方根のこと

$$\text{RMSE} = \sqrt{\text{MSE}}$$

- MAE（平均絶対誤差）

誤差の絶対値の平均のこと



| 回帰モデルの評価

- RMSEはMAE

RMSEはMAEに比べて外れ値の影響を大きく受けてしまう
→RMSEを求めるために、一度誤差を2乗するため
RMSEは大きな誤差を許容したくない場合に使用することが多い

→MSEを求めるとき、2乗するため単位が2乗になってしまう
RMSEはMSEの平方根なので、単位を元に戻せる

| 回帰モデルの評価

・決定係数

相関係数を2乗することで求めることができる指標

→相関係数の取りうる範囲は「-1～1」であるため、

決定係数の取りうる範囲は「0～1」である

→決定係数は寄与率とも呼ばれる



過學習

過学習

- ・過学習（過剰適合、オーバーフィッティング）

訓練データに対して過度に適応しすぎた結果

訓練データ以外のデータに対して予測精度が悪くなること

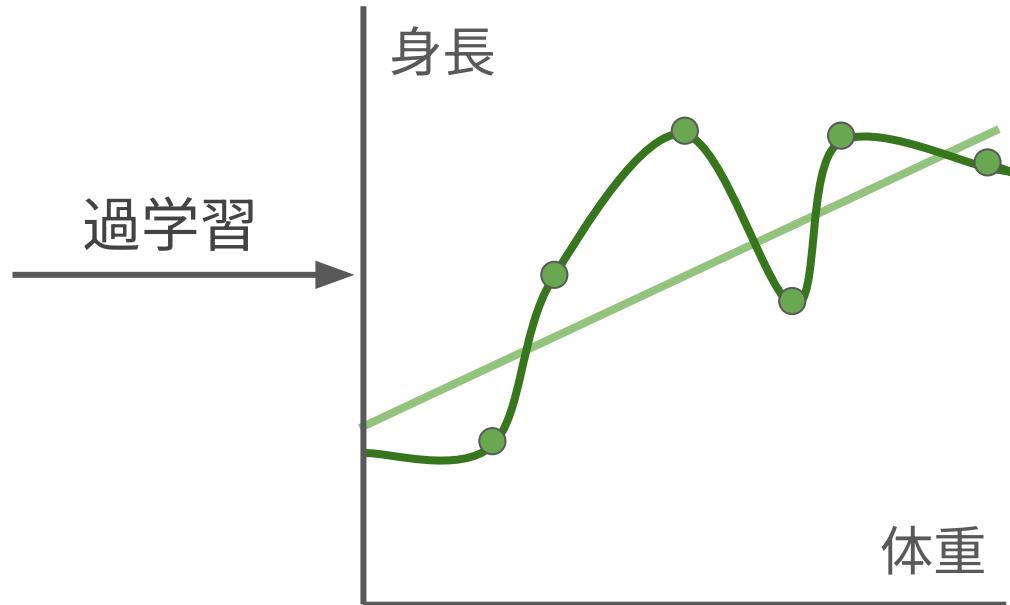
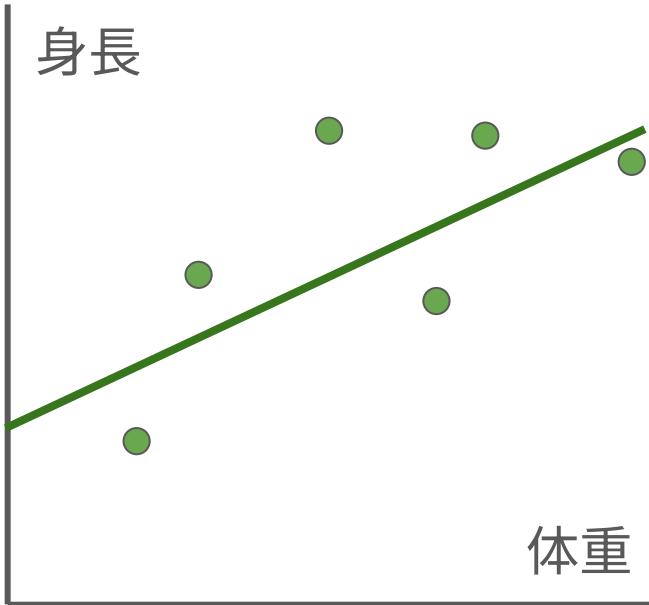
→一般的に適応しきれない方が予測精度が高くなる

モデルが複雑になるほど過学習になりやすい

→過学習を防ぐために様々なテクニックが考えられている

過学習

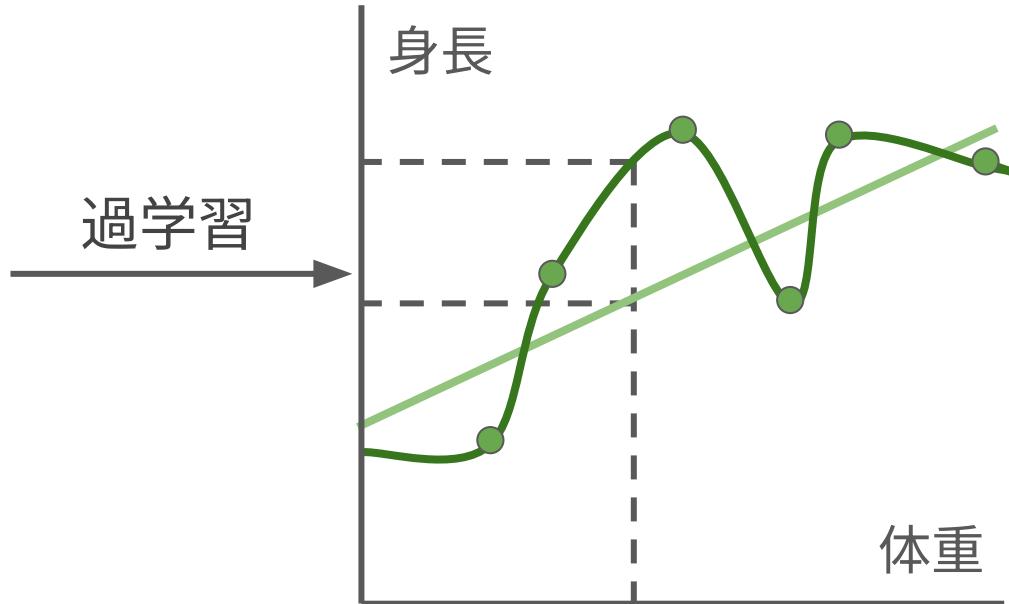
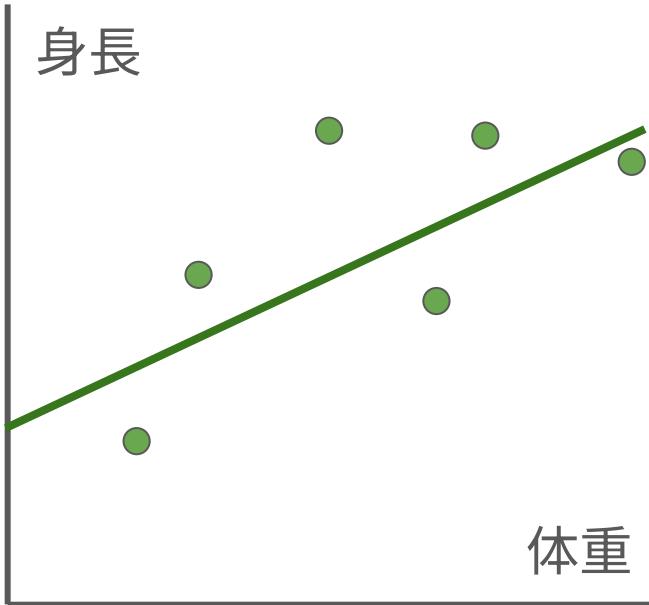
- 回帰の場合



過学習

過学習

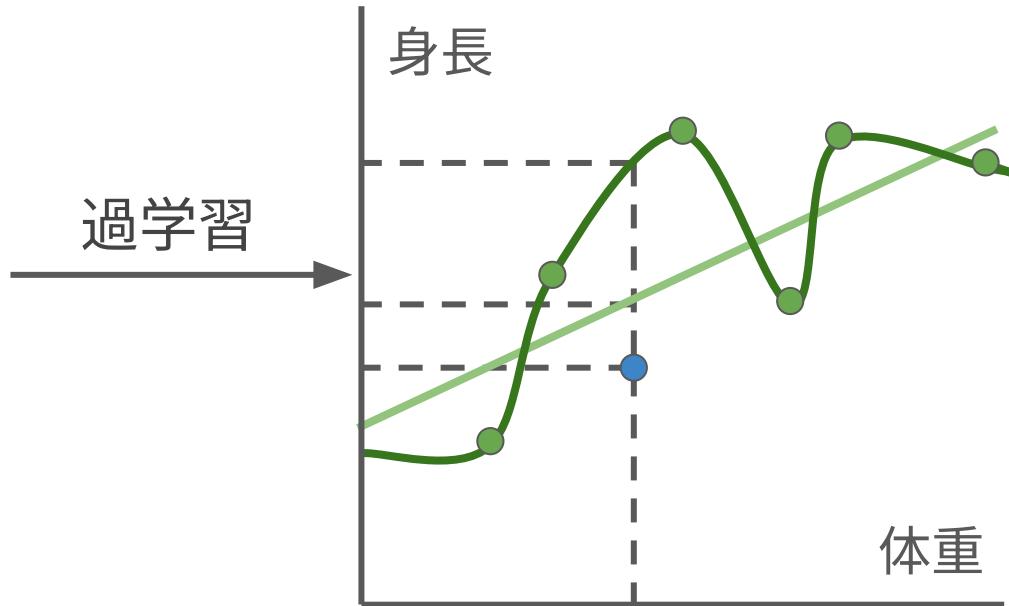
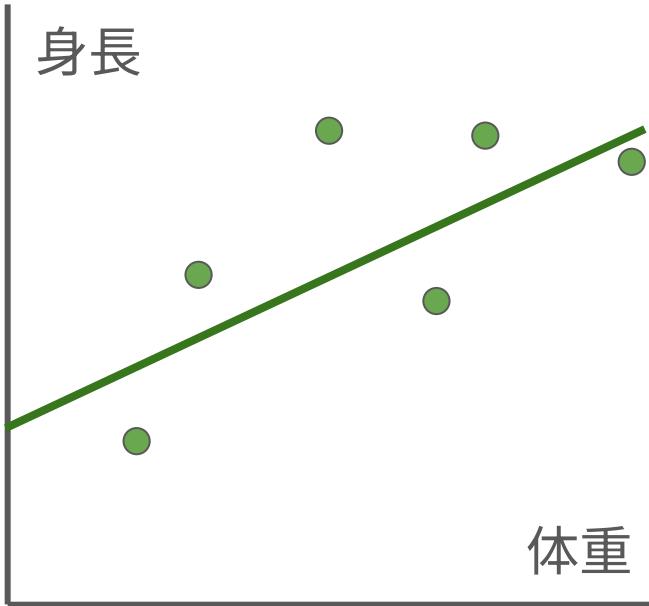
- 回帰の場合



過学習

過学習

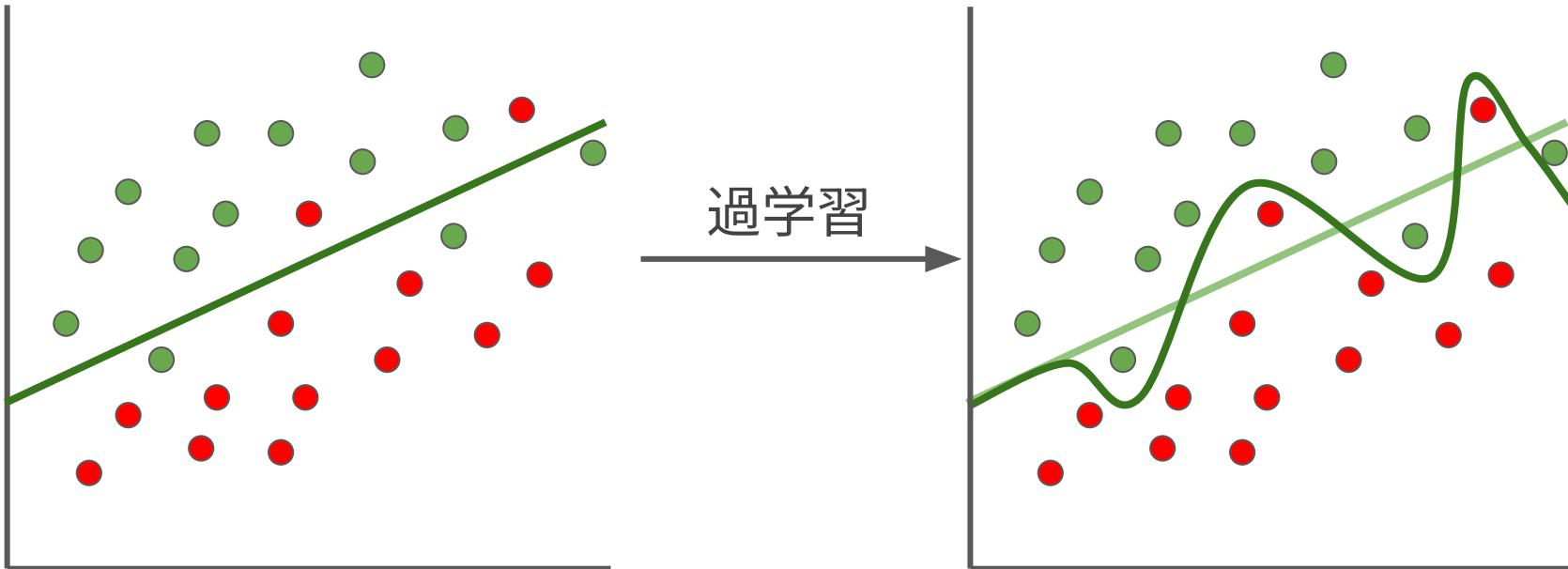
- 回帰の場合



過学習

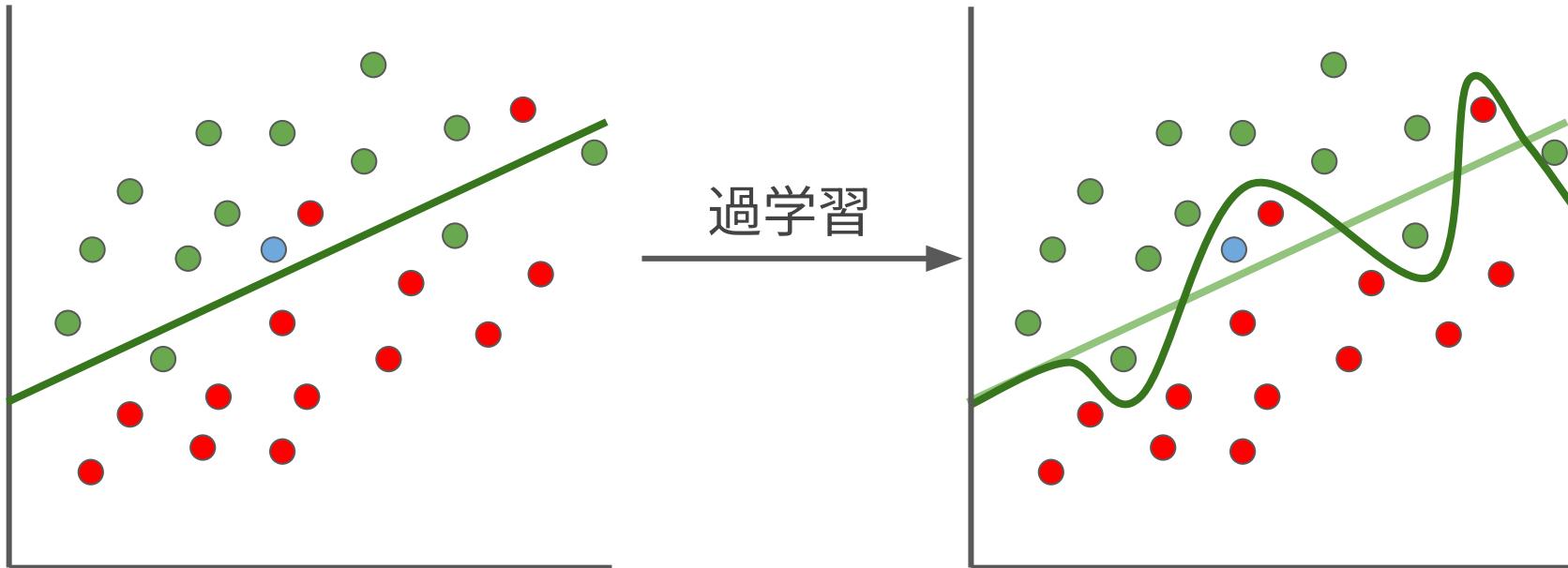
過学習

- 分類の場合



過学習

- 分類の場合



過学習

- ・過学習の代表的な原因

- ・モデルが複雑すぎる

→正則化などを行いモデルを単純化していく

- ・訓練データに対して特徴量が多すぎる

→より多種多様なデータ入手し、訓練データの量を増やす
使用する特徴量を絞り、過学習を抑制する

過学習

- ・過学習の見極め方

訓練誤差が小さくなっているにも関わらず

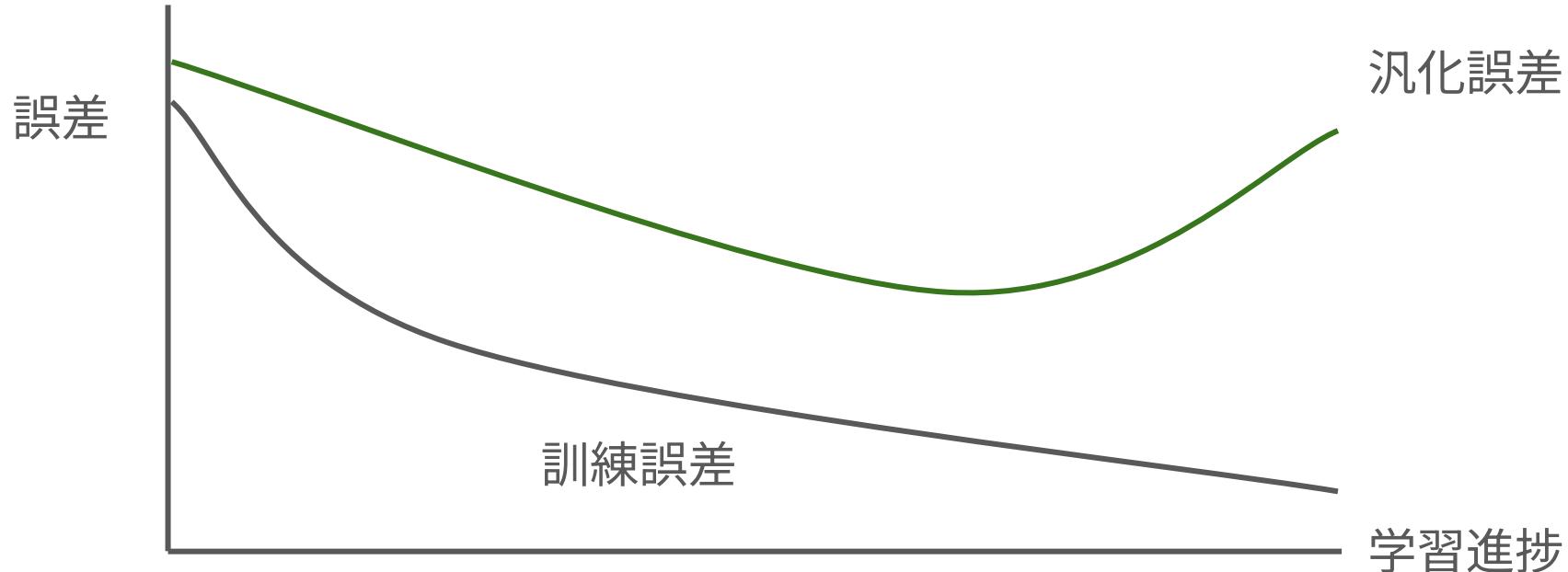
汎化誤差が小さくない場合は過学習の状態

訓練誤差：訓練データに対する予測と正解の誤差

汎化誤差：未知データに対する予測と正解の誤差

過学習

- ・過学習の見極め方



過学習

汎化誤差の要素

バイアス、バリアンス、ノイズの要素に分解できる

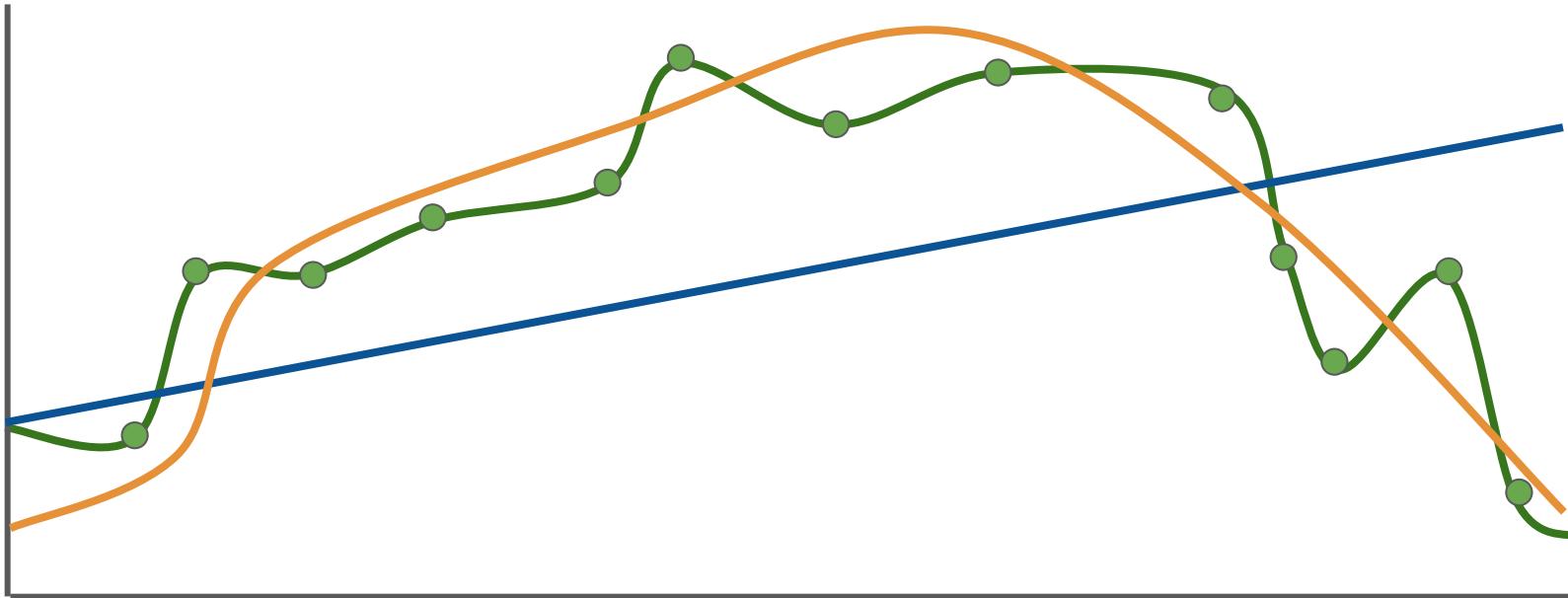
バイアス : モデルが単純すぎるために発生する誤差

バリアンス : モデルが複雑すぎるために発生する誤差

ノイズ : 減らすことができない誤差

過学習

- 学習したモデル



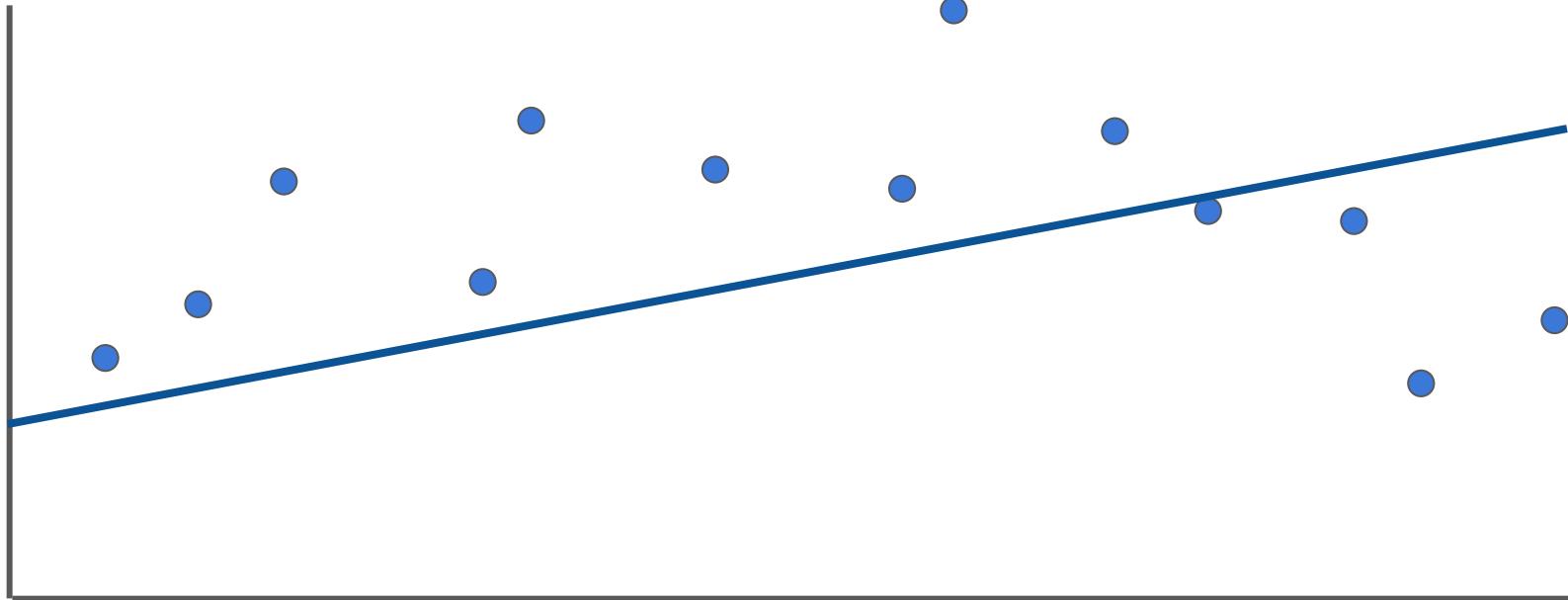
過学習

- バイアス（モデルが単純すぎるために発生する誤差）



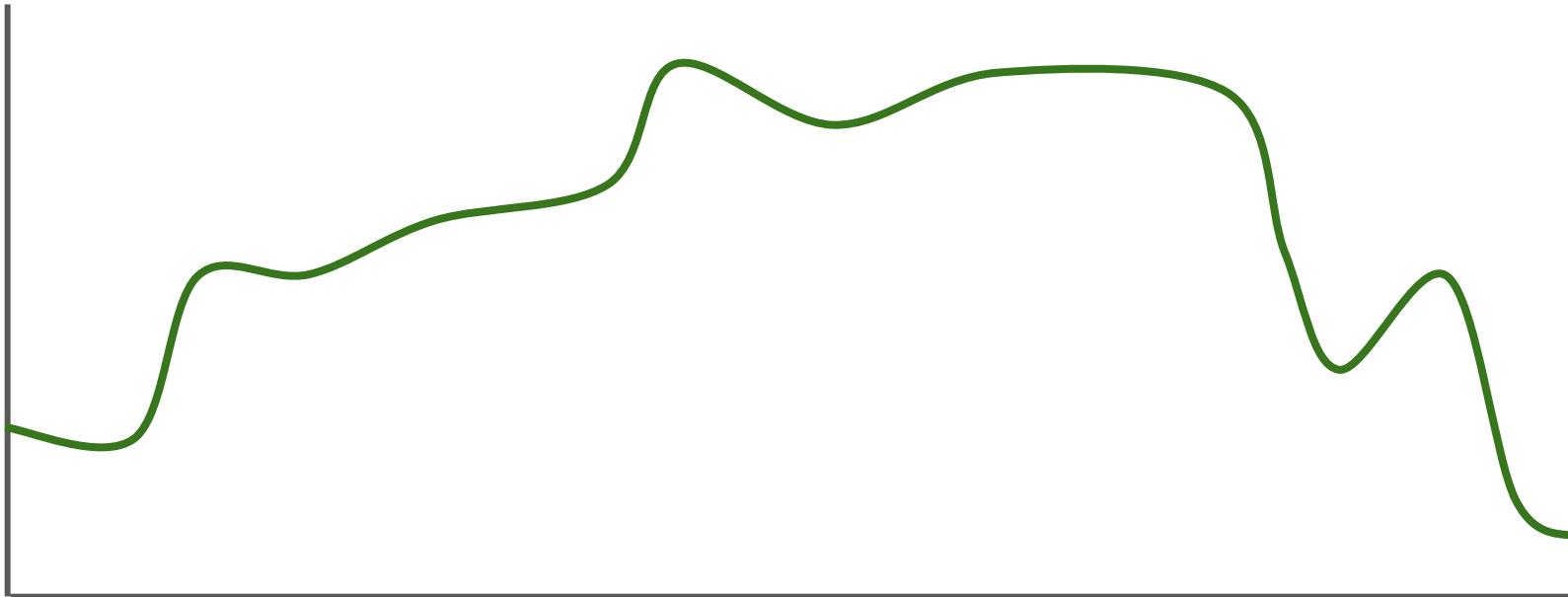
過学習

- バイアス（モデルが単純すぎるために発生する誤差）



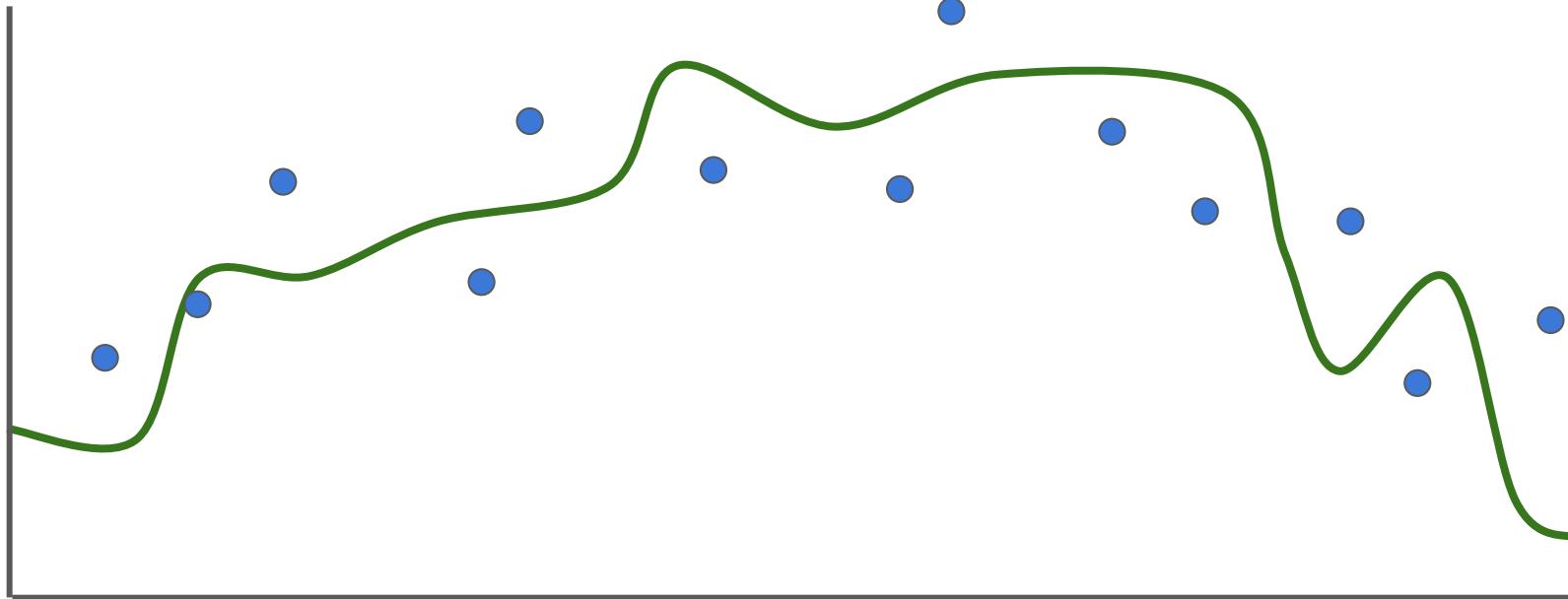
過学習

- ・バリアンス（モデルが複雑すぎるために発生する誤差）



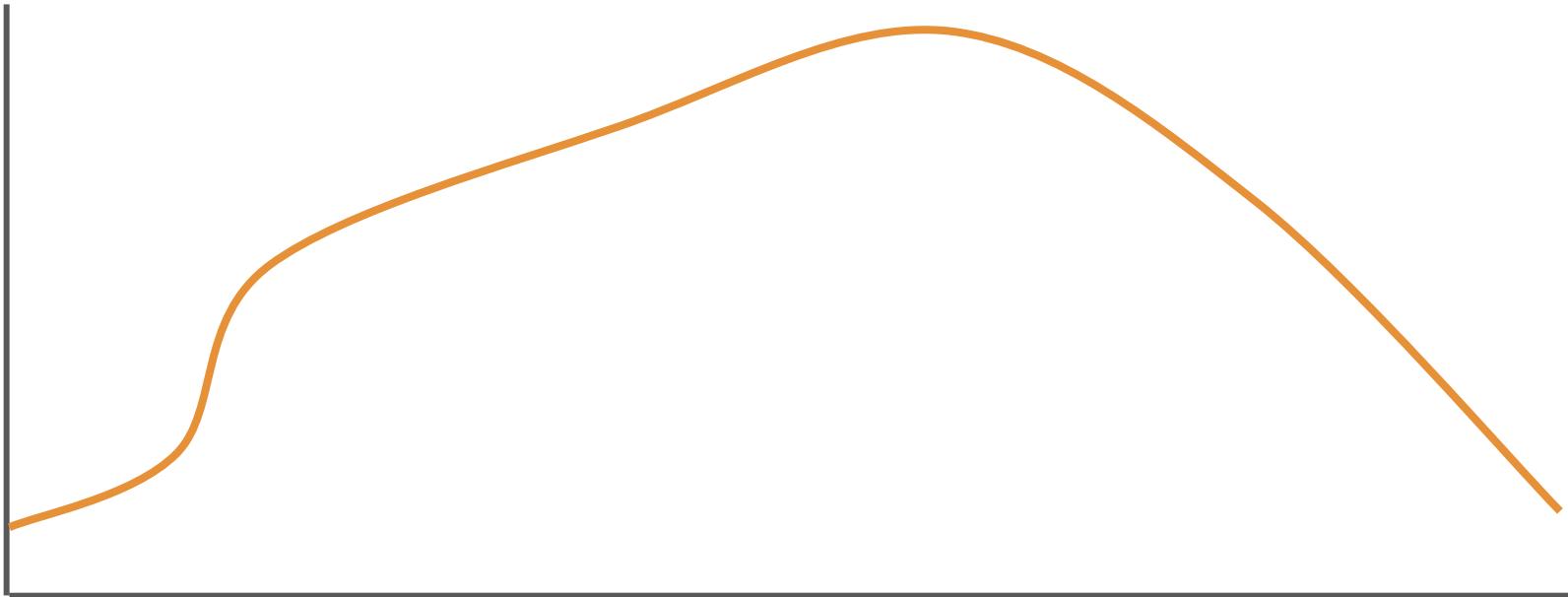
過学習

- ・バリアンス（モデルが複雑すぎるために発生する誤差）



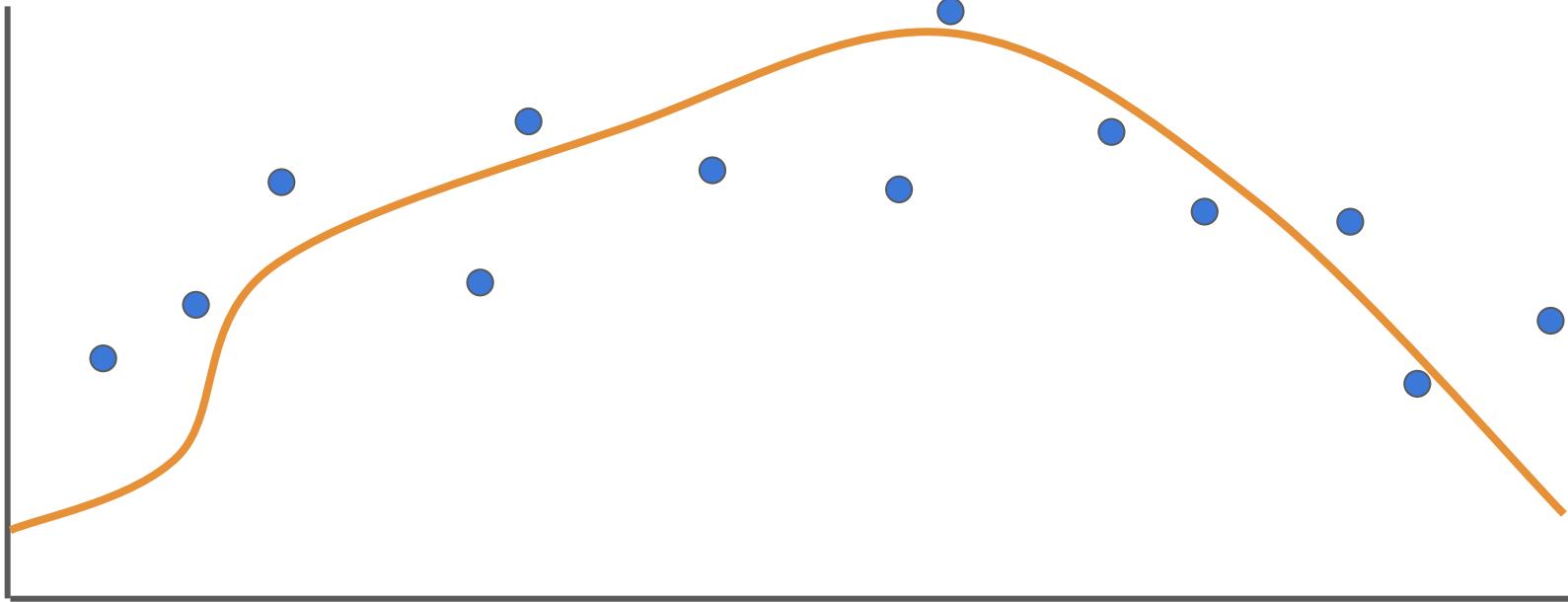
過学習

- ・ノイズ（減らすことができない誤差）



過学習

- ・ノイズ（減らすことができない誤差）



過学習

・過学習

- バイアスとバリアンスはトレードオフの関係である
→バイアスとバリアンスの和が最小になるようにモデルを調整
- 過学習はバイアスが小さくバリアンスが大きいために発生する
→未知のデータに対する精度を汎化性能という
精度は汎化誤差を使用して測定する



過学習を 防ぐテクニック

| 過学習を防ぐテクニック

- 代表的なテクニック

- アンサンブル学習
- スパース化
- データ拡張
- 正則化
- ドロップアウト
- 早期終了
- バッチ正規化

| 過学習を防ぐテクニック

- 代表的なテクニック

- アンサンブル学習
- スパース化
- データ拡張
- 正則化
- ドロップアウト
- 早期終了
- バッチ正規化



ディープラーニングの章で解説

| 過学習を防ぐテクニック

・スパース化

モデルに使用する**特徴量**（説明変数）の数を減らす手法

→特徴量が多すぎると過学習になりやすいため、

特徴量を減らしていくことで過学習を防ぐことができる

→特徴量が多いと訓練データに対する予測精度を高め過ぎてしまう

（説明ができ過ぎてしまうので説明し過ぎないようにしていく）

過学習を防ぐテクニック

- ・データ拡張（データの水増し）

既存データから様々な加工（回転、色の調整等）を行って、データのバリエーションを増やすこと

→データのバリエーションを増やすことで

特定のデータに対して適応し過ぎてしまうことを防ぐことが可能

→似たデータを増やすと過学習が起きやすいので注意が必要

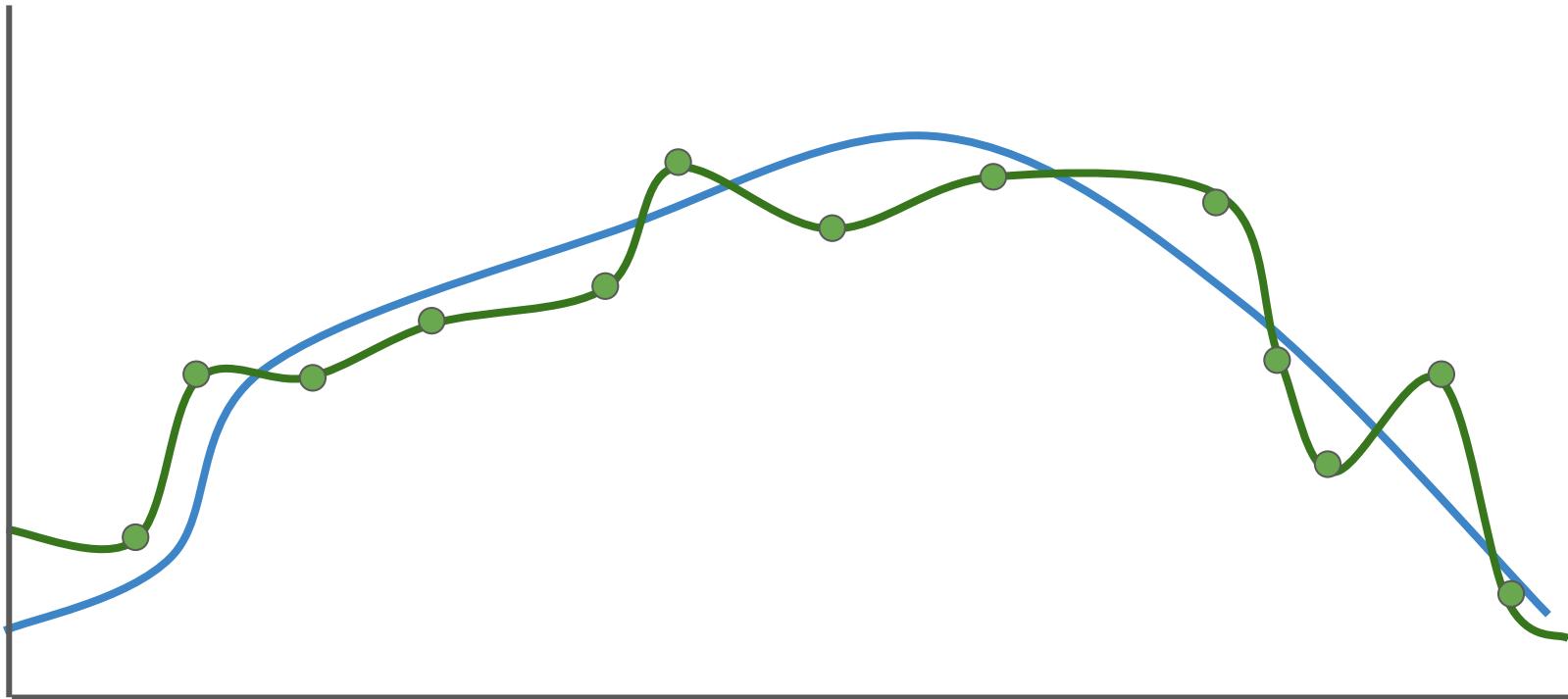
| 過学習を防ぐテクニック

- 正則化

過学習を防ぐために、パラメータに制限をかけること

→学習とは誤差関数（損失関数）で求めた誤差を最小化ために重みなどを調整していくことである

過学習を防ぐテクニック



| 過学習を防ぐテクニック

・正則化

誤差関数と正則化項（ペナルティ項）の和を
最小化することで過度な適応を避けることができる

→誤差 = 誤差関数 + 正則化項

→複雑になりすぎたモデルをシンプルにできる
制約をつけて最適化する問題を制約付き最適化問題という

| 過学習を防ぐテクニック

- 正則化

正則化にはL1正則化、L2正則化、L0正則化がある

→誤差 = 誤差関数 + 正則化項

→正則化項をどのように求めるかで名前が異なる

| 過学習を防ぐテクニック

- L0正則化

0では無い重みの数を使用して正則化項を求める

→計算コストは高いというデメリットがある

| 過学習を防ぐテクニック

- L1正則化

重みの絶対値の総和を使用して正則化項を求める

→重要度の低い重みを取り除き、

重要度の高いパラメータを残す処理を行う（特徴選択）

| 過学習を防ぐテクニック

- L2正則化

重みの2乗和を使用して正則化項を求める

→重みが大きくなりすぎないように、

重みの大きさを調整してモデルを滑らかにする処理を行う

| 過学習を防ぐテクニック

- ラッソ回帰

L1正則化を適用した線形回帰のこと

- リッジ回帰

L2正則化を適用した線形回帰のこと

- Elastic Net

ラッソ回帰とリッジ回帰を組み合わせてもの

| 過学習を防ぐテクニック

• 過学習

正則化を行いすぎると未学習になってしまう危険性がある

→未学習とは学習不足により、訓練データに対してさえ

モデルの推測精度が低いという状態のこと

→正則化を行いすぎないようにする必要がある

| 過学習を防ぐテクニック

- 過学習

モデルの表現力が高いと、過学習が起きやすい

- モデルの表現力

訓練データに対してどの程度まで説明できているのかということ

→モデルの表現力と汎化性能はトレードオフの関係である

ノルムと正則化



| ノルムと正則化

・ノルム

ベクトルなどの大きさを測るための尺度

→ノルムにはL1ノルムやL2ノルムなどがある

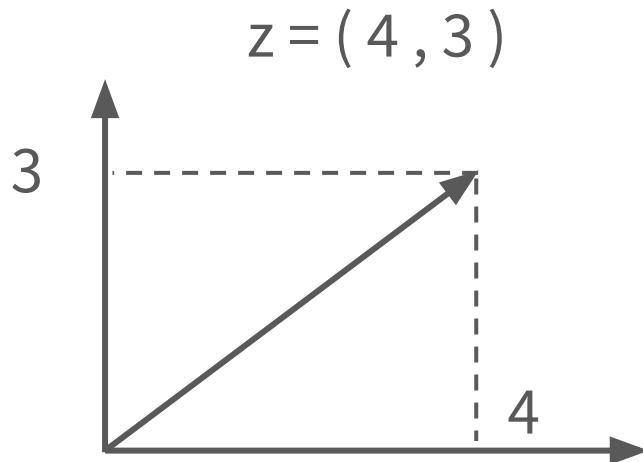
→正則化にL1ノルムの考え方を適用したものが**L1正則化**、

正則化にL2ノルムの考え方を適用したものが**L2正則化**である

| ノルムと正則化

- L1ノルム（マンハッタン距離）

ベクトルなどの要素の絶対値の総和によって求める距離のこと



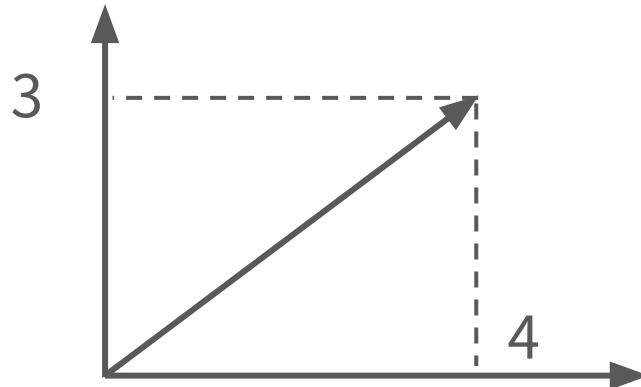
$$|4| + |3| = 7$$

| ノルムと正則化

- L2ノルム（ユークリッド距離）

ベクトルなどの要素の2乗和の平方根によって求める距離のこと

$$z = (4, 3)$$



$$\sqrt{4^2 + 3^2} = 5$$

ROC曲線とモデルの選択

ROC曲線とモデルの選択

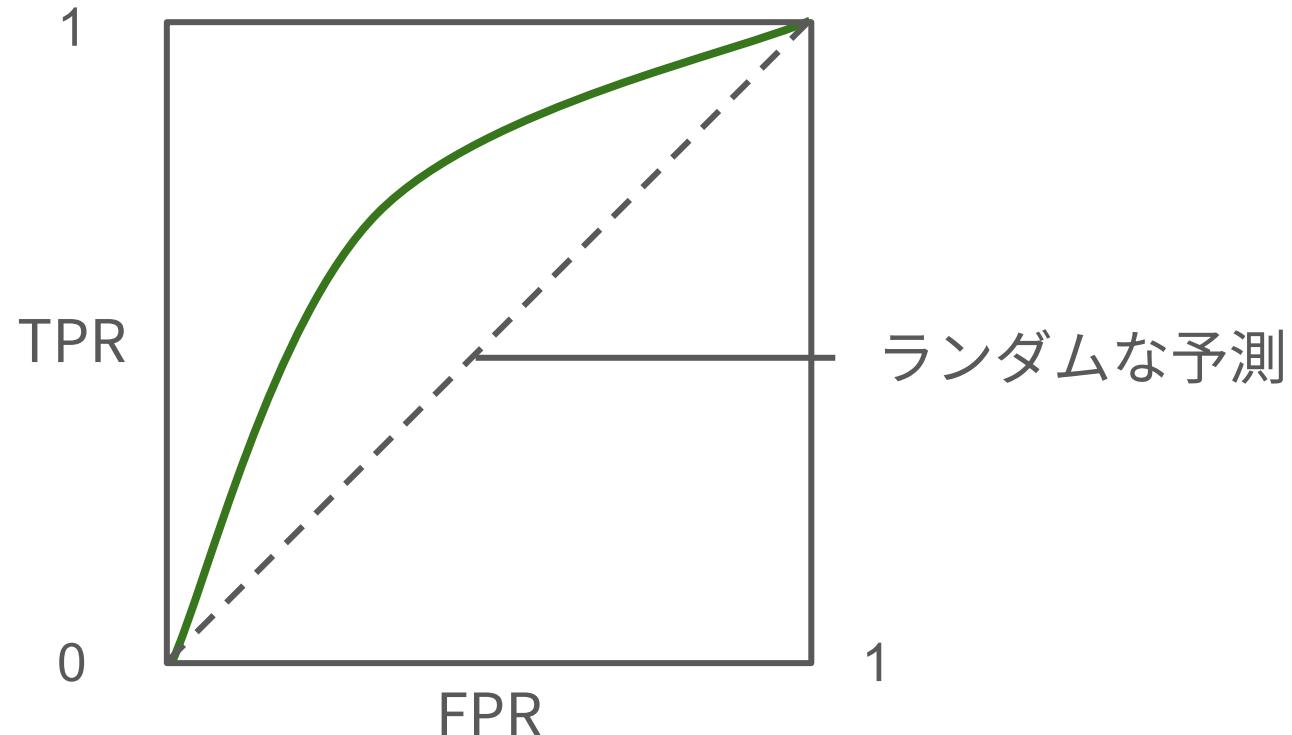
- ROC曲線

真陽性率（TPR）と偽陽性率（FPR）の関係を表した曲線

- 真陽性率 = $TP \div (TP + FN)$
- 偽陽性率 = $FP \div (FP + TN)$

→縦軸を真陽性率（TPR）、横軸を偽陽性率（FPR）をとる

ROC曲線とモデルの選択

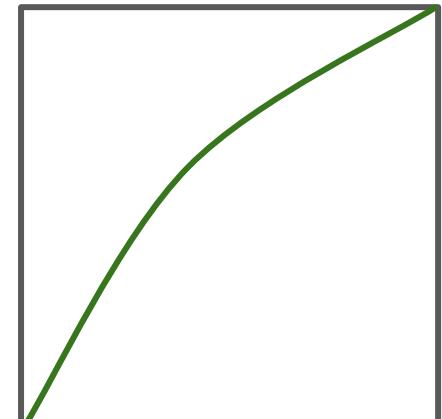
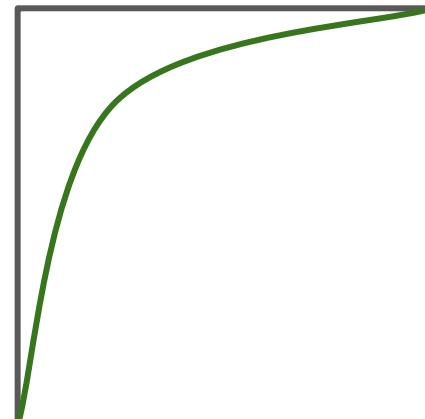
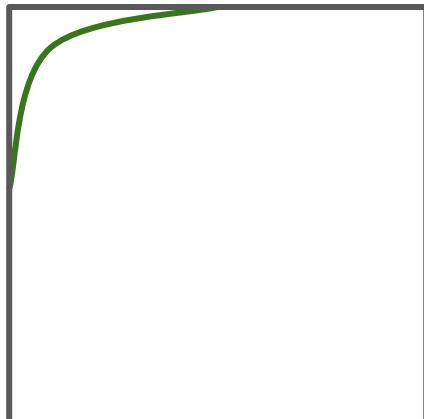


| ROC曲線とモデルの選択

- AUC

ROC曲線よりも下の面積のこと

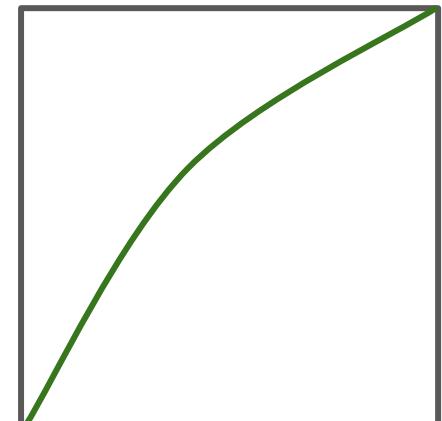
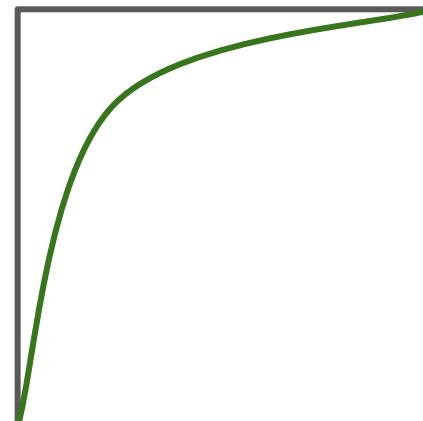
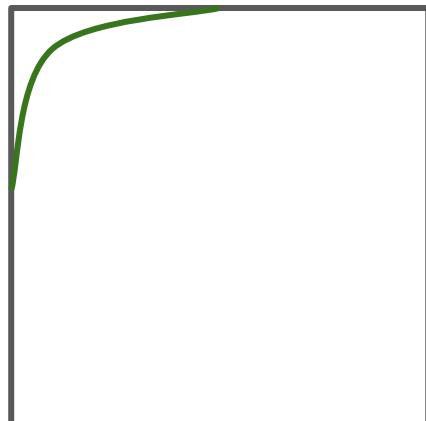
→面積が1に近いほどモデルの性能は高いとされる



| ROC曲線とモデルの選択

- AUC

AUCの値が 0.5 のとき、ランダムな予測を行うモデルと同等の性能と判断することができる



ROC曲線とモデルの選択

・モデルの選択と情報量

タスクに対して最適なモデルを選択していく

→複雑なモデルを選択すれば複雑なタスクを実行できたり、
予測精度を上げたりすることができる

→複雑なモデルは一般的に学習コストが高い特徴がある

| ROC曲線とモデルの選択

・ モデルの選択と情報量

- 簡単なモデルと複雑なモデルがあり、
予測精度が同程度ならば、簡単なモデルを選択するべき
→簡単なモデルにすることで学習コストを低下させることが可能
- 「ある事柄を説明するためには、必要以上に多くを
仮定するべきではない」という考え方をオッカムの剃刀という

ROC曲線とモデルの選択

・情報量基準

モデルの複雑さと、データとの適合度とのバランスを見るときに使用される指標・基準のこと

- モデルが複雑すぎると過学習が起きやすく、学習コストも高い
- モデルが簡単すぎると予測精度が低くなってしまう
- モデルの複雑さをちょうど良いバランスにする必要がある

ROC曲線とモデルの選択

- 情報量基準

赤池情報量規準 (AIC) やベイズ情報量規準 (BIC) がある

→使用する規準によってモデルの評価が変わることがある

→BICでは計算にデータ数が大きく関与する

データ数が大きい場合にはシンプルなモデルが評価されやすい

→データ数の大小によって影響を受けにくいAICが選択される場合