DeepLearninig 勉強会

5 章前半

B4 T.Ochiai

May 13, 2019

Nagoya Institute of Technology
Takeuchi & Karasuyama Lab

目次

Next Section

Deep Learning について

- 機械学習の一種
  - 機械学習の基礎を理解する必要がある

今回は機械学習の基礎についてお話しします

Next Section

学習アルゴリズムについて

- 機械学習はデータを元にして学習を行う
- ここでの学習とは何か？
  - tasks T
  - experiences E
  - performance measures P

タスク T

- 機械学習は人間界において解決するのが難しいタスクを解決する
  - 学習過程自体がタスクではない
  - 例: ロボットを歩かせるための学習では「歩かせる」ことがタスクである
- 標本をどのような過程で処理するかという面で説明される
  - 標本とは特徴の集合である
  - 標本はよくベクトル $\boldsymbol{x} \in \mathbb{R}^n$ で表す
- 機械学習ではたくさんのタスクを解くことができる

Classification

- k 個のカテゴリの中のどれに所属するのかを決定
  - 関数 $f : \mathbb{R}^n \to \{1, ..., k\}$ を学習
  - $y = f(\boldsymbol{x})$ においてベクトル $\boldsymbol{x}$ は入力 (標本), y はカテゴリに対応する数値

例

- 画像の分類
  - $\boldsymbol{x}$: 画像
  - y : 画像のカテゴリを識別するラベル
- 撮った写真を自動でタグ付けしたりする技術など

Classification with missing inputs

- 入力に欠損データがあるときの Classification
- 1 つの関数を学習するより, 複数の関数の集合を学習する方がよい
- 各特徴の欠損の有無から, 最大で 2^n 個の関数を考えれば解決できる
- 現実的には単一の同時分布を考えれば良い
  - 欠損データで周辺化することで対応できる

Regression

- 入力に対してある数値を予測する
  - 関数 $f : \mathbb{R}^n \to \mathbb{R}$ を学習
- 出力がラベルではないこと以外は Classification と同じ

例

- 被保険者が支払う額の予測
- 証券の将来の価格の予測

Transcription

- 非構造化されたデータをテキストデータに変換する

例

- 画像中の文字の認識
  - $\boldsymbol{x}$：あるテキストを含んだ画像
  - y：画像中に含まれているテキスト
- スピーチの認識
  - $\boldsymbol{x}$：ある文字列が話されたときの音波の波形
  - y：実際に話された文字列

Machine translation

- ある言語を他の言語に翻訳する
- 一般的に, 自然言語に対して適用される

Structured output

- 出力がベクトルであるようなタスク全般
  - または, 複数の値を含むようなデータ構造を出力するタスク
  - 先程の転写や機械翻訳も含まれる

例

- 自然言語の品詞へのパース
- 画像内の各ピクセルをあるカテゴリに分類するタスク

Anomaly detection

- ある物事が正常か異常かを検知するタスク

例

- クレジットカード詐欺の検知
  - 普段とは違うタイプの買い物がなされたときに対して異常検知を行う
  - 異常が検知された場合はクレジットカードの利用を止めて詐欺を防ぐことができる

Synthesis and sampling

- 訓練データを元に新たな標本を生成する
- 入力が与えられたとき, サンプリングや合成処理によって出力を生成する
- ある入力に対しての正しい出力は存在しない
  - 複数のパターンの出力

例

- ゲームにおけるテクスチャや地形の自動生成
- 文章からその文章を読み上げるオーディオ波形の生成

Imputation of missing values

- 入力データ中の欠損値の補完

Denoising

- 入力に含まれるノイズを除去する
- ノイズの含まれる入力 $\tilde{\boldsymbol{x}} \in \mathbb{R}$
- ノイズを除去した入力 $\boldsymbol{x} \in \mathbb{R}$
- 一般的には予測分布 $p(\boldsymbol{x}|\tilde{\boldsymbol{x}})$ を求める

Density estimation

- 関数 $p_{\mathrm{model}} : \mathbb{R}^n \to \mathbb{R}$ を学習する
  - p_{model} は確率密度関数であると解釈できる
- 分布を捉えることができる
- 欠損データにも用いることができる
  - i 番目の特徴が欠損している $\boldsymbol{x}_{-i}$
  - $p(x_i|\boldsymbol{x}_{-i})$ を考えることで欠損データにも対応可能

The Performance Measure P

- 機械学習アルゴリズムの性能を定量的に計測する必要がある
  - accuracy: モデルが正しい出力を生成する事例の割合
  - error rate: 誤った出力を生成した事例の割合
- 実際は未知のデータに対する機械学習アルゴリズムの性能を計測したい
  - training data で学習を行い **test data** で未知データへの性能を評価
- 問題設定にあった性能指標を見つけるのは実際難しい
  - 何を計測するべきかを決めるのが難しい
  - タスクによって異なる指標を選択する必要がある

例

- 頻繁に中程度の誤りを起こすもの/稀に大規模な誤りを起こすもの
  - どちらに重いペナルティを課すべきか

The Experience, E

- 機械学習アルゴリズムは 2 つの種類に分けることができる
  - 教師なし学習
  - 教師あり学習
- 機械学習アルゴリズムはデータセットをもとに学習を行う
- データセットは標本の集合で, 標本をデータ点とも呼ぶ

Unsupervised learning algorithms

- データセットの構造の特性を学習する
  - 深層学習の観点では, データセットを生成した確率分布 $p(\boldsymbol{x})$ を学習

例

- 密度推定
- ノイズ除去
- クラスタリング

Supervised learning algorithms

- 教師なし学習とは違い訓練データに **label** が含まれる
  - label は **target** とも呼ばれる
- 入力: $\boldsymbol{x}$, label: y を用いて $\boldsymbol{x}$ から y を予測できるように学習を行う
  - $p(y|\boldsymbol{x})$ を推定するなど

Iris データセットの例

- $\boldsymbol{x}$ はあやめのいくつかの特徴
- y はあやめの種類を表すコード

補足　教師なし学習/教師あり学習は, 正式に定義されている用語ではない:

$$p(\boldsymbol{x}) = \prod_{i=1}^{n} p(x_i|x_1, \ldots, x_{i-1})$$

- $p(\boldsymbol{x})$ の教師なし学習をn 個の教師あり学習に分割して解くことができる

教師あり学習，教師なし学習，その他の学習

- 教師あり学習と教師なし学習は大まかに適応すべきケースが決まっている
- 教師あり学習
  - 回帰
  - クラス分類
  - structured output problems
- 教師なし学習
  - Density estimation
- そのほかにもいくつかの学習が存在する
  - semi supervised learning
    - いくつかの標本にのみ label が存在する
  - multi-instance learning
    - label を含む，含まないという情報のみは与えられるが label の値はわからない
  - reinforcement learning
    - 環境との相互作用で報酬を最大化するように行動を学習

計画行列

- ほとんどの機械学習アルゴリズムでは標本の集合を用いて学習を行う
- 標本の集合は通常，計画行列として表わされる
  - 計画行列は各行に各標本, 各列に各特徴をもつ
- 教師あり学習では計画行列は $\boldsymbol{X}$，ラベルは $\boldsymbol{y}$ として表されることが多い
  - y_i は i 番目の標本のラベル

$$\underbrace{\boldsymbol{X} = \left[\begin{array}{c} \vdots \\ \boldsymbol{x}_i^\top \\ \vdots \end{array}\right]}_{\text{特徴数}} \left.\vphantom{\begin{array}{c} \vdots \\ \boldsymbol{x}_i^\top \\ \vdots \end{array}}\right\} \text{標本数}, \quad \boldsymbol{y} = \left[\begin{array}{c} \vdots \\ y_i \\ \vdots \end{array}\right]$$

Next Section

線形回帰

- 入力 $\boldsymbol{x} \in \mathbb{R}^n$ から出力 $y \in \mathbb{R}$ を予測する
- 予測値を $\hat{y}$, $\boldsymbol{w}$ を重みパラメータとして以下のような回帰問題を考える
  - $\boldsymbol{w}$ は $\boldsymbol{x}$ の各要素に対する重み
  - w_i の絶対値が大きいほど x_i が回帰の結果に強く寄与する

$$\hat{y} = \boldsymbol{w}^\top \boldsymbol{x} \tag{1}$$

- $\boldsymbol{w}$ を訓練データによって学習することを目指す

Mean Squared Error

- 線形回帰の性能指標として **Mean Squared Error** (MSE) を用いる
- テストデータの計画行列を $\boldsymbol{X}^{(\mathrm{test})}$, targets を $\boldsymbol{y}^{(\mathrm{test})}$, 予測値を $\hat{\boldsymbol{y}}^{(\mathrm{test})}$ とするとテストデータにおける $\mathrm{MSE}_{\mathrm{test}}$ は以下のように表される:

$$\mathrm{MSE}_{\mathrm{test}} = \frac{1}{m} \| \hat{\boldsymbol{y}}^{(\mathrm{test})} - \boldsymbol{y}^{(\mathrm{test})} \|_2^2 \tag{2}$$

  - m: 標本数
- targets と予測値の距離が遠いほど $\mathrm{MSE}_{\mathrm{test}}$ の値は大きくなる
- $\mathrm{MSE}_{\mathrm{test}}$ の値を可能な限り小さくしたい
  - targets と予測値を近くしたい

MSE によるパラメータの最適化

- 訓練データ $(\boldsymbol{X}^{(\text{train})}, \boldsymbol{y}^{(\text{train})})$ を用いて線形回帰モデルを学習する
- $\text{MSE}_{\text{train}}$ を最小にするような $\boldsymbol{w}$ を求める
  - $\text{MSE}_{\text{train}}$ を $\boldsymbol{w}$ で微分してその値 (勾配) が $\boldsymbol{0}$ となる $\boldsymbol{w}$ を求める

$$\frac{\partial}{\partial \boldsymbol{w}} \text{MSE}_{\text{train}} = 0 \tag{3}$$

$$\frac{\partial}{\partial \boldsymbol{w}} \frac{1}{m} \| \hat{\boldsymbol{y}}^{(\text{train})} - \boldsymbol{y}^{(\text{train})} \|_2^2 = 0 \tag{4}$$

$$\frac{1}{m} \frac{\partial}{\partial \boldsymbol{w}} \| \boldsymbol{X}^{(\text{train})} \boldsymbol{w} - \boldsymbol{y}^{(\text{train})} \|_2^2 = 0 \tag{5}$$

$$\vdots$$

$$\boldsymbol{w} = (\boldsymbol{X}^{(\text{train})\top} \boldsymbol{X}^{(\text{train})})^{-1} \boldsymbol{X}^{(\text{train})\top} \boldsymbol{y}^{(\text{train})} \tag{6}$$

- 式 (6) の方程式のことを一般に正規方程式と呼ぶ

MSE によるパラメータの最適化の例

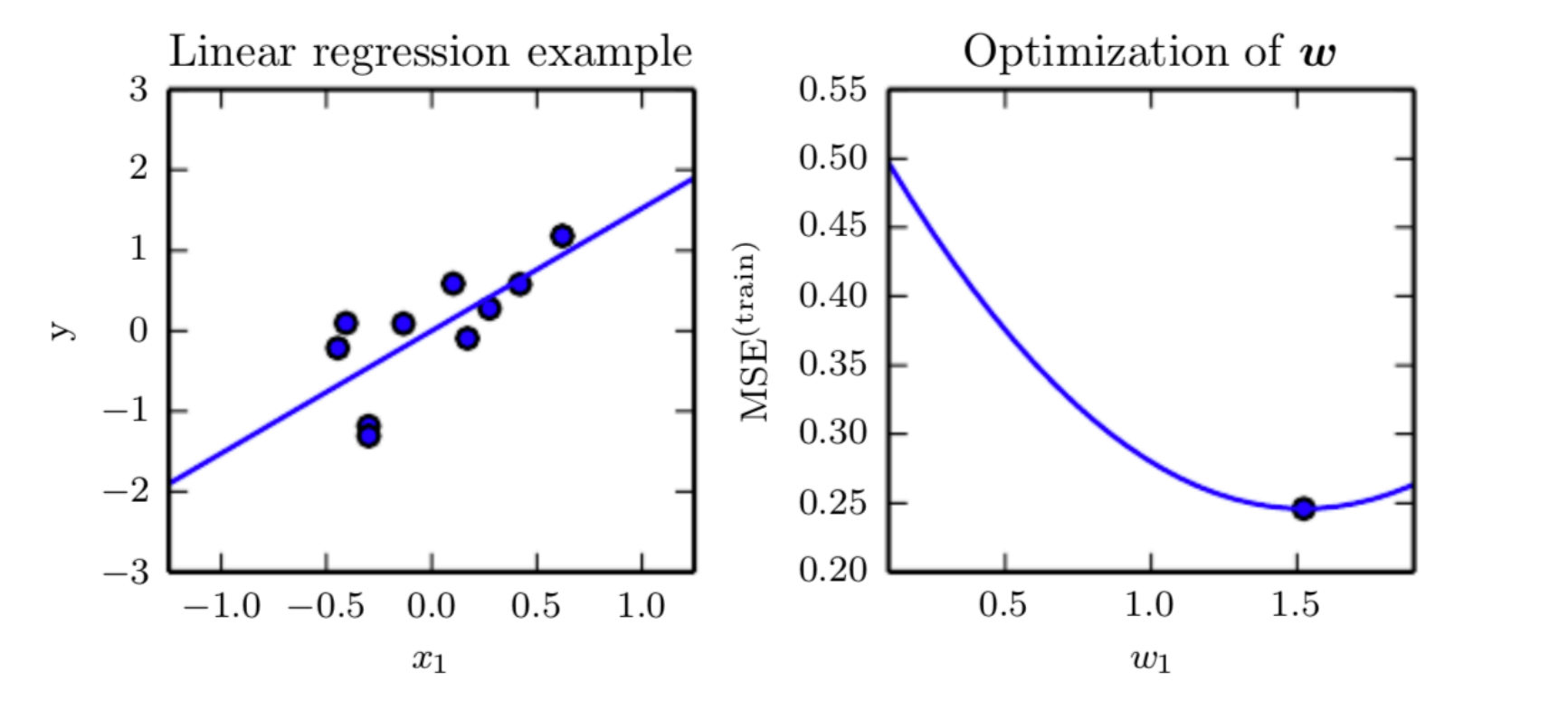


Figure 1: w の最適化の結果

- 最適化によって訓練データ点をうまく直線回帰できている

バイアス項を含む線形回帰

- 一般に線形回帰を考えるときは上述の線形モデル対してさらにバイアス項 b を考慮したモデルを考えることが多い

$$\hat{y} = \boldsymbol{w}^\top \boldsymbol{x} + b \tag{7}$$

- 入力が全くない場合にモデルの出力が b に偏るという観点に由来する
- 統計的バイアスの考えとは異なることに注意
- $\boldsymbol{x}$ に 1 を追加すれば, b を特別に扱う必要はない:

$$\hat{y} = \begin{bmatrix} \boldsymbol{w} \\ b \end{bmatrix}^\top \begin{bmatrix} \boldsymbol{x} \\ 1 \end{bmatrix} = \tilde{\boldsymbol{w}}^\top \tilde{\boldsymbol{x}}$$

Next Section

汎化性能

- 機械学習では未知の入力に対しての汎化性能が重要
- モデルを学習させるためには訓練誤差を小さくする
  - これだけならば単なる最適化問題
- 機械学習が最適化問題と異なるのは学習したモデルでテスト誤差も小さくしたい点
  - テスト誤差： $\frac{1}{m}\|\boldsymbol{X}^{(\text{test})}\boldsymbol{w} - \boldsymbol{y}^{(\text{test})}\|_2^2$

i.i.d. 仮定

- 訓練データとテストデータはデータ生成過程と呼ばれるデータ集合の確率分布から生成される
- 通常は i.i.d. 仮定という仮定を置く
  - independent and identically distributed
  - 各データ集合の標本が互いに独立
  - 訓練集合とテスト集合が同一の分布にしたがう
- 仮定をおくことで訓練誤差とテスト誤差の関係を数学的に調べることが可能
  - 無作為に選択されたモデルの期待訓練誤差と期待テスト誤差が等しい

過学習と未学習

- 機械学習がどの程度うまく動作するかには以下の要素がある
  - 訓練誤差を小さくする
  - 訓練誤差とテスト誤差の差を小さくする
- これら 2 つの要素は未学習, 過学習に対応する
  - 訓練集合で十分小さな誤差が得られない $\rightarrow$ 未学習
  - 訓練誤差とテスト誤差の差が大きすぎる $\rightarrow$ 過学習

モデルの容量

- モデルの容量: モデルが多様な関数に適合する能力
- 過学習や未学習を起こしやすいかどうかはモデルの容量で制御できる
  - 容量が小さい場合は訓練集合を適合させにくい
  - 容量が大きい場合は訓練集合の特徴を記憶し過ぎてしまう
- タスクの複雑さと訓練データの量に対して, 適切な容量があるときに最もよい性能を発揮する
- モデルの容量を変更する方法
  - 仮説空間を選ぶ
    - 仮説空間: 学習アルゴリズムが解として選択できる関数の集合
  - (例) 入力される特徴量の数を変更し，対応するパラメータを追加する

$$\underbrace{\hat{y} = b + w_1 x}_{\text{一次多項式}} \quad \begin{array}{c} \text{特徴に } x^2 \text{ を追加} \\ \longrightarrow \\ \text{パラメータ } w_2 \text{ を追加} \end{array} \quad \underbrace{\hat{y} = b + w_1 x + w_2 x^2}_{\text{二次多項式}}$$

過学習と未学習の例

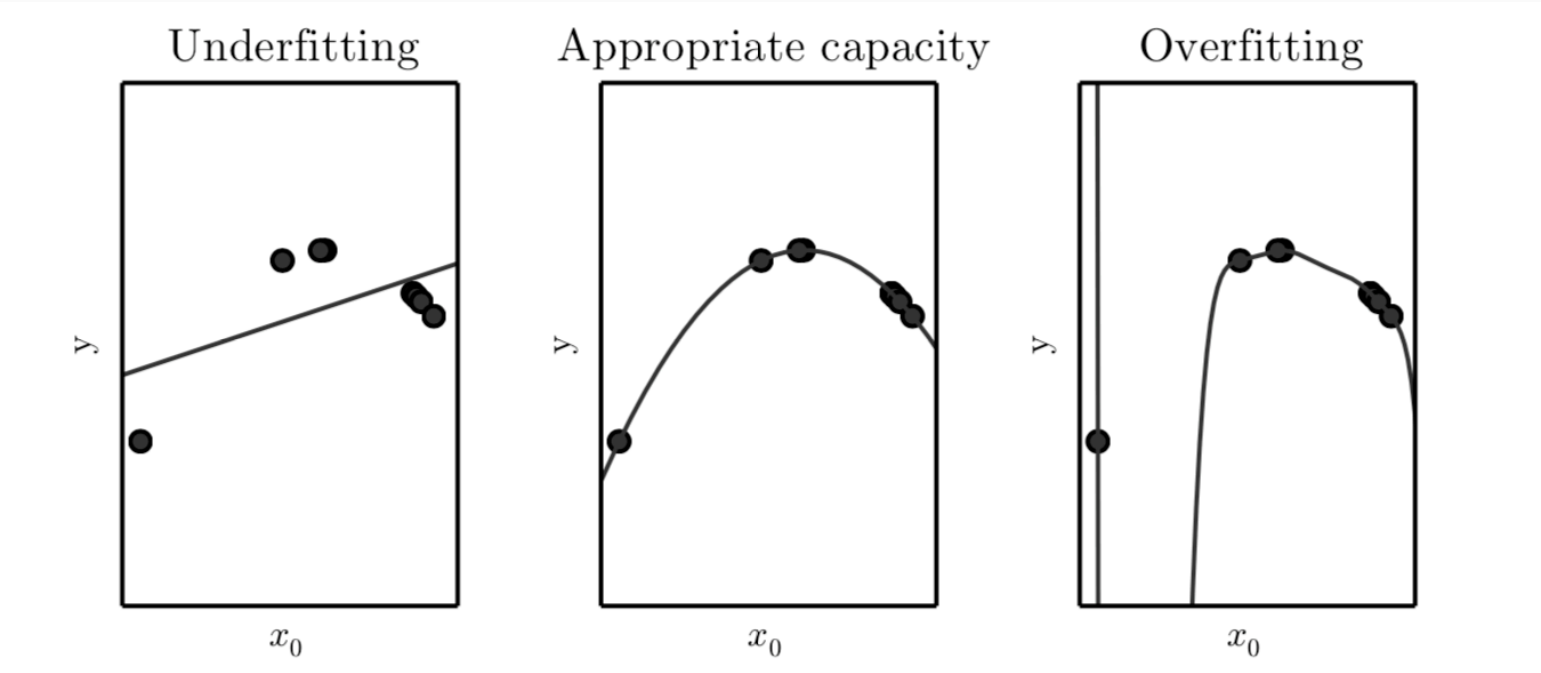
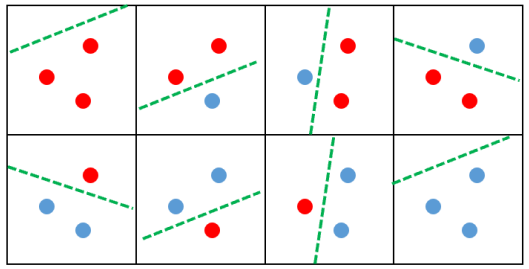


Figure 2: 真の関数が 2 次であるような問題に対して (左) 線形，(真ん中) 2 次，(右) 9 次 の予測モデルで回帰した結果

- 線形モデルの場合は容量が小さく未学習
- 9 次の場合は容量が大きすぎるため過学習

オッカムの剃刀, VC 次元

- オッカムの剃刀: 競合する複数の仮説が既知の観察を同様にうまく行える場合, 「最も単純な」仮説を選ぶべき
  - 同様の性能が得られるならば最も容量の小さいモデルを選択すべき
- モデルの容量を定量化する方法として VC 次元 [1] がある
  - 二項分類の容量を測るもの
- 訓練誤差とテスト誤差の差はモデルの容量が増えるほど大きくなる
  - 容量が定量化できると定量的な予測を行うことが可能となる
- 任意の高い容量というモデルを選ぶならばノンパラメトリックを選べばよい



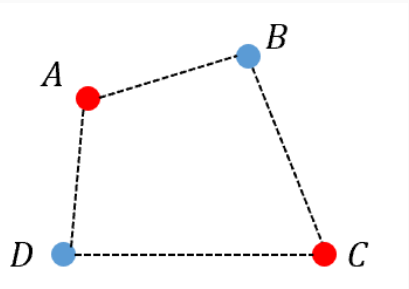


Figure 3: 左: 3 点の任意のラベル付けに対して線形分離可能. 右: 4 点だと無理な例がある. 2 次元平面の VC 次元は 3.

[1] 二項分類によって任意にラベル付けできる m 個の異なる点 x の訓練集合が存在するときに, m が取り得る最大値で定義される

容量と誤差との関係

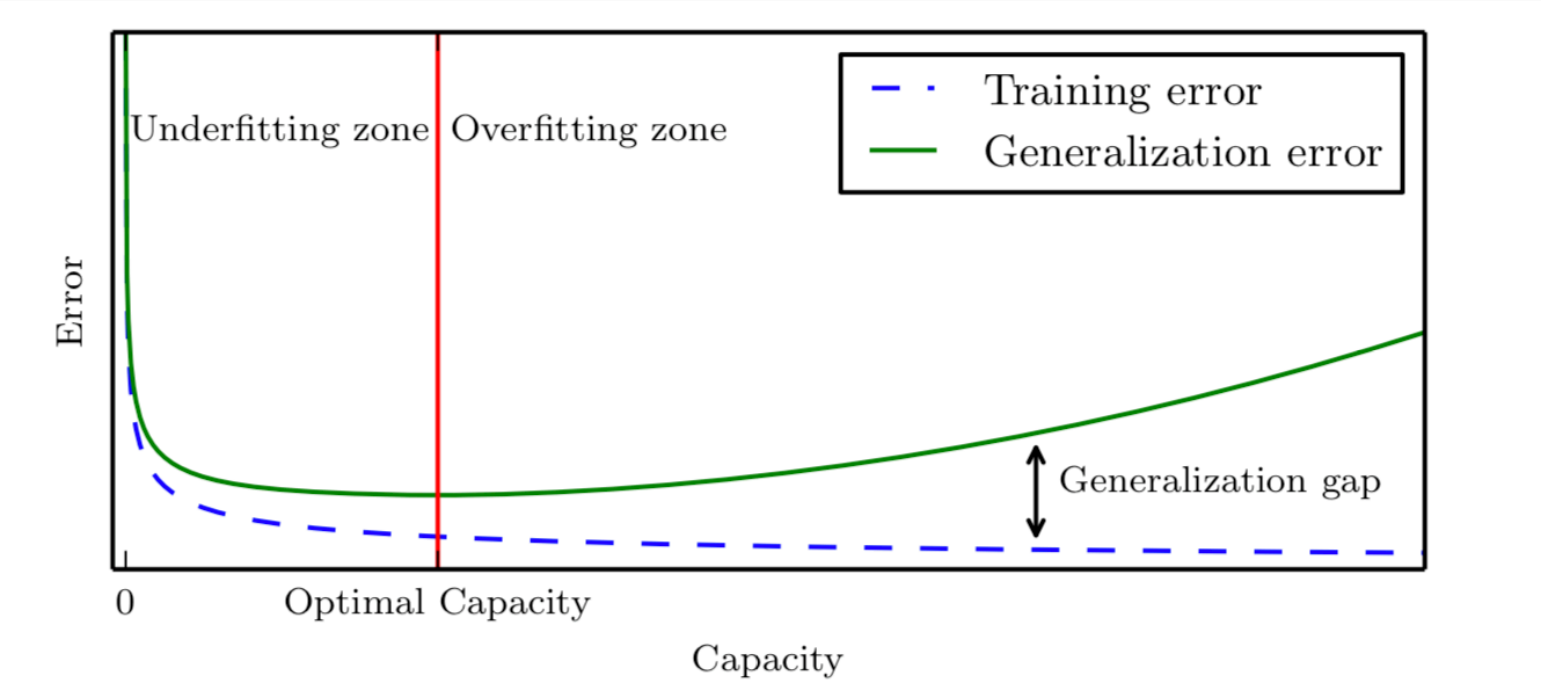


Figure 4: 容量と誤差の関係．青点線は訓練誤差，緑実線はテスト誤差，赤線の位置は最適な容量

- 最適な容量になるまでは未学習の状態
- 最適な容量を超えると各誤差の差が大きくなり過学習の状態

ノーフリーランチ定理

- データを生成する分布全てを平均すると，どの分類アルゴリズムも，未知のデータに対する誤差率は同じになる
- 他の機械学習アルゴリズムよりも普遍的に良いと呼ばれるアルゴリズムはないと言われている
  - ノーフリーランチ定理という
  - 全て問題に対して高性能なアルゴリズムは存在しない
- データを生成するすべての分布を平均化した場合にのみ成立する
  - 実世界の応用で出現する確率分布の種類に仮定を設けることで，その分布に対して良い性能を発揮するアルゴリズムを設計できる

正則化

- 仮説空間内のある解を他の解より優先させることが可能
- 例えば以下のような重み減衰によって重みパラメータの訓練基準を変更することができる

$$J(\boldsymbol{w}) = \mathrm{MSE}_{\mathrm{train}} + \lambda \boldsymbol{w}^{\top} \boldsymbol{w} \tag{8}$$

- λ が大きいほど重みパラメータは小さくなる
- 重み減衰によって未学習や過学習の傾向を制御することができる
- 一般には正則化項というペナルティをコスト関数に加えてモデルを正則化できる
- 正則化は訓練誤差ではなく汎化誤差を減少させる目的でアルゴリズムに変更を加える

Next Section

ハイパーパラメータ

- ほとんどの機械学習アルゴリズムにはその挙動を制御するハイパーパラメータという設定値がある
- ハイパーパラメータは学習によって最適化はされない
  - 訓練集合によって学習されたハイパーパラメータは常に最大のモデル容量を選択するため過学習を起こす

検証集合

- ハイパーパラメータを決定するためには検証集合を用いる
- 検証集合は訓練データの一部から構築される
  1. 訓練データを 2 つの別々のセットに分割する
  2. 分割されたセットの 1 つでパラメータの学習を行う
  3. もう片方のセットを検証集合として用いて汎化誤差の推定に用いる
  4. 汎化誤差に応じてパラメータを更新する
- テスト集合は検証集合に用いてはいけない
  - 学習にテスト集合を用いてはいけないのと同様
- 検証集合を用いると汎化誤差は過小評価される
  - 訓練誤差よりはその度合いは小さい

交差検証

- データ集合を訓練集合とテスト集合に分割するとテスト集合が小さくなったときに問題が生じる可能性がある
  - 平均テスト誤差に対する不確実性が大きくなり，アルゴリズムの性能の比較が難しくなる
- 無作為に訓練集合とテスト集合を分割して性能を評価することを繰り返すことですべてのデータにおいて平均テスト誤差を評価できる
  - このような方法を交差検証と呼ぶ
- 一般的なものとして k-分割交差検証法がある

k-分割交差検証法の例

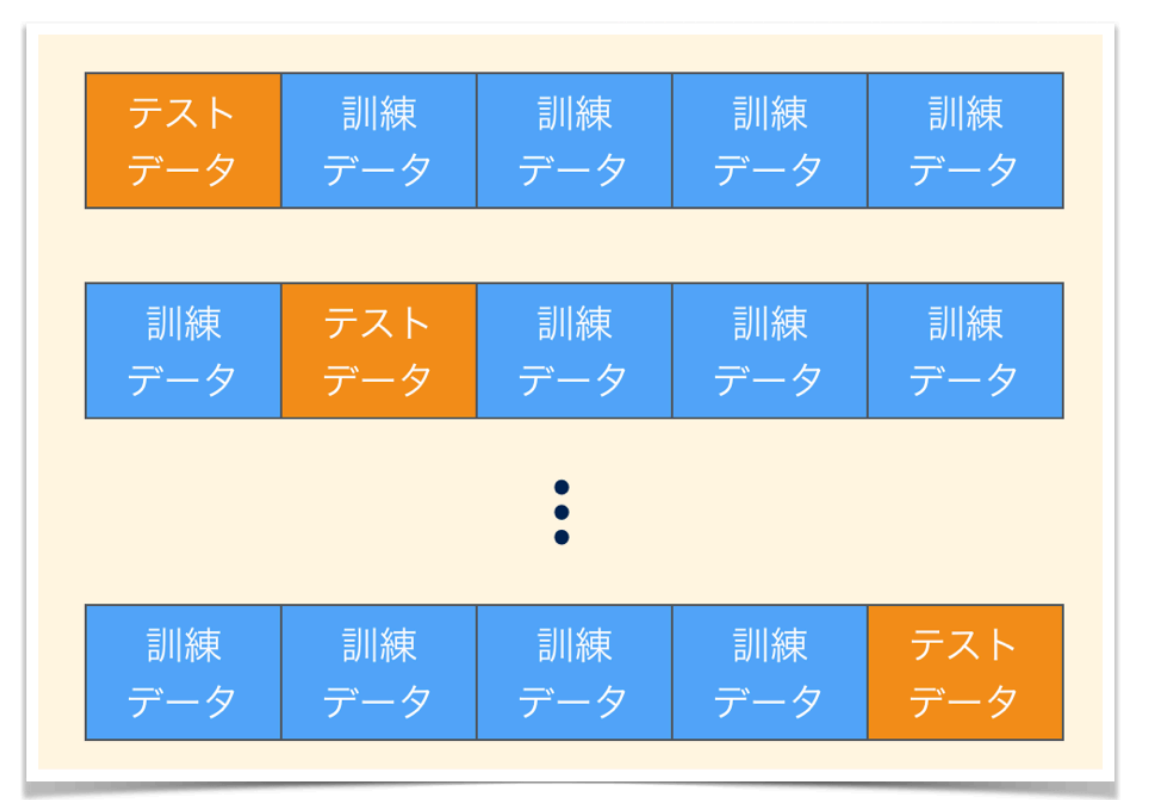


Figure 5: k-分割交差検証法の例

- テストデータと訓練データを分割して各ステップでテスト誤差を求める
- すべてのステップが終わった後に平均テスト誤差を求める

Next Section

点推定量

- 関心のある量について最良の予測を 1 つ推定する試み
- パラメータ $\boldsymbol{\theta}$ の点推定を $\hat{\boldsymbol{\theta}}$ と表す
- $\{\boldsymbol{x}^{(1)}, ..., \boldsymbol{x}^{(m)}\}$ を m 個における点推定量はデータの任意の関数となる
  - $\{\boldsymbol{x}^{(1)}, ..., \boldsymbol{x}^{(m)}\}$ は独立同分布 (i.i.d.) に従う

$$\hat{\boldsymbol{\theta}}_m = g(\boldsymbol{x}^{(1)}, ..., \boldsymbol{x}^{(m)}) \tag{9}$$

- g が $\boldsymbol{\theta}$ の真の値に近い値を返す必要はない
- g の値域が $\boldsymbol{\theta}$ に許容される値の集合である必要はない
  - ほとんどの関数が推定量として利用可能
  - ただし良い推定量は $\boldsymbol{\theta}$ の真の値に近い出力を出す関数
- データは確率的な過程から抽出されるため $\hat{\boldsymbol{\theta}}$ は確率変数

関数推定

- 入力ベクトル x に対する出力 y を予測
- y と x の関係を記述する関数 $f(x)$ が存在すると仮定
- よくある設定として以下の問題を考える
  - ϵ は x からは予測できないノイズ項

$$y = f(x) + \epsilon \tag{10}$$

- モデルか関数推定 $\hat{f}$ で f を近似したい
  - 関数推定は θ を推定するのと同じ
- 関数推定量 $\hat{f}$ は関数空間における点推定量にすぎない

バイアス

- 推定量のバイアスは以下のように表される
  - $\mathbb{E}(\hat{\boldsymbol{\theta}}_m)$ は全データに関する期待値
  - $\boldsymbol{\theta}$ は潜在的なパラメータの真の値

$$\mathrm{bias}(\hat{\boldsymbol{\theta}}_m) = \mathbb{E}(\hat{\boldsymbol{\theta}}_m) - \boldsymbol{\theta} \tag{11}$$

- $\mathrm{bias}(\hat{\boldsymbol{\theta}}_m) = \mathbf{0}$ である場合は不偏であるという
  - $\mathbb{E}(\hat{\boldsymbol{\theta}}_m) = \boldsymbol{\theta}$ であるのと等価
- $\lim_{m\to\infty} \mathrm{bias}(\hat{\boldsymbol{\theta}}_m) = 0$ の場合は漸近不偏であるという
  - $\lim_{m\to\infty} \mathbb{E}(\hat{\boldsymbol{\theta}}_m) = \boldsymbol{\theta}$ であるのと等価
- 一般的にバイアスは小さい方が良いとされる
  - 推定量の期待値と真の値が近いということになるため

具体例はテキストを参照

分散，標準誤差

- 推定量がデータのサンプル関数としてそれだけ変化すると予想されるか
- 推定量のバリアンスとは分散 $\mathrm{Var}(\hat{\theta})$ のこと
- 分散の平方根は標準誤差と呼ばれ, $\mathrm{SE}(\hat{\theta})$ と表される
- 推定量の分散や標準誤差はデータ集合を繰り返しサンプリングする際に推定量がどの程度変化するのかを示す尺度
- バイアスと同様バリアンスも小さい方が好まれる

サンプル平均の標準誤差

- サンプル平均の標準誤差は以下のように表される

$$\mathrm{SE}(\hat{\mu}_m) = \sqrt{\mathrm{Var}\left[\frac{1}{m}\sum_{i=1}^{m} x^{(i)}\right]} = \frac{\sigma}{\sqrt{m}} \tag{12}$$

  - σ^2 はサンプル $x^{(i)}$ の分散
- 真の標準誤差を過小評価してしまっている
  - m が非常に大きい場合はこの過小評価は無視できる

MSE とバイアス，バリアンス

- 大きなバイアスか大きなバリアンスをもつモデルのどちらかしか得られない場合はどのようにモデルを選択すべきか
  - 交差検証
  - 平均二乗誤差 (MSE) を比較
- MSE は以下のように表すことができる

$$\mathrm{MSE} = \mathbb{E}[(\hat{\theta}_m - \theta)^2] = \mathrm{bias}(\hat{\theta}_m)^2 + \mathrm{Var}(\hat{\theta}_m) \tag{13}$$

- MSE の評価にはバイアスとバリアンスの両方が組み込まれてる
- 望ましい推定量は以下のような推定量である
  - MSE を小さくする
  - バイアスとバリアンスをある程度抑えている

バイアス, バリアンスの関係

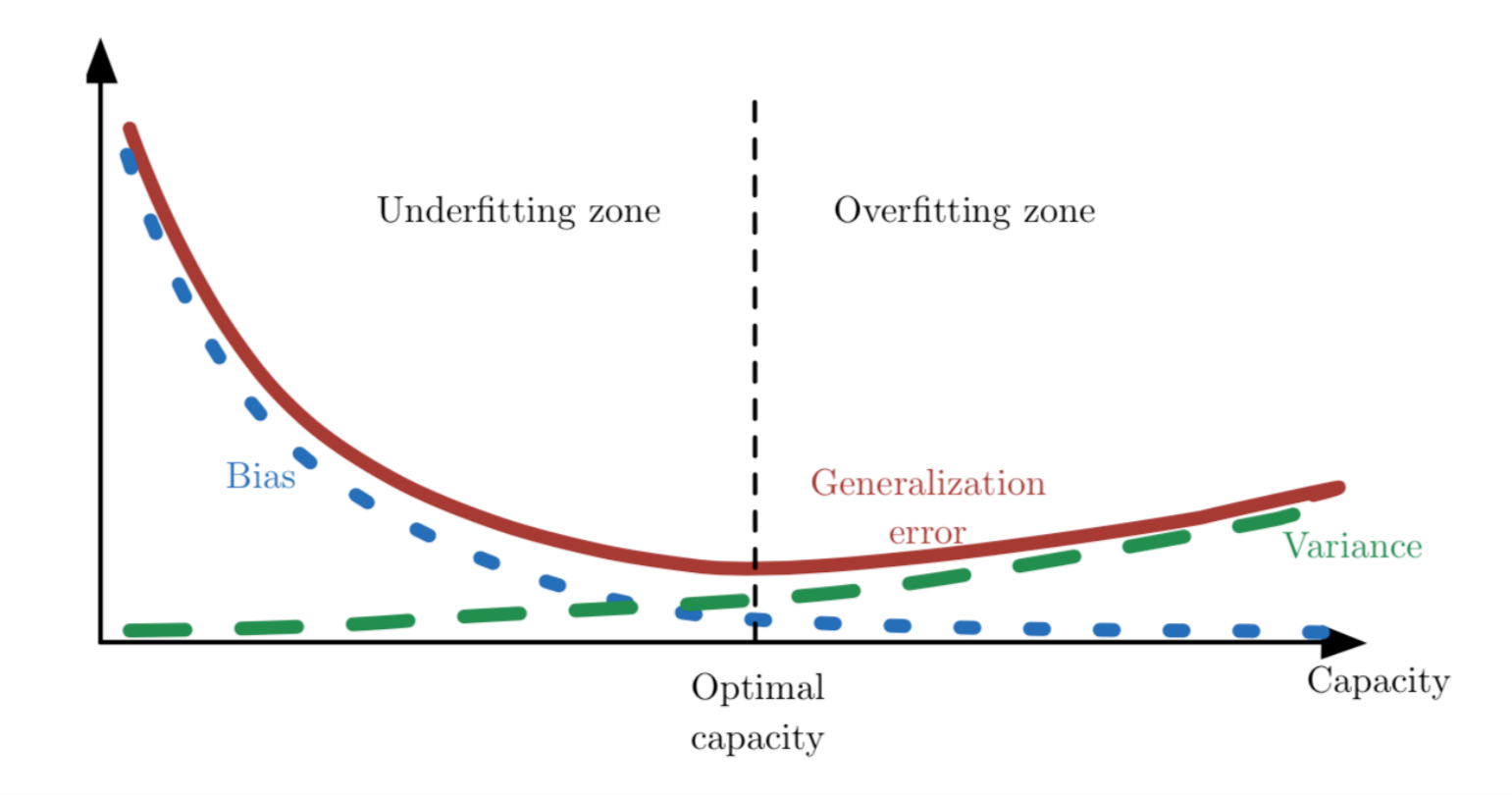


Figure 6: バイアス, バリアンスの関係. 青点線がバイアス, 緑点線がバリアンス, 赤線がテスト誤差, 黒点線が最適な容量の位置

- バイアスとバリアンスの関係は過学習, 未学習と密接に関係している
- 容量を増加させるごとにバイアスが減少し, バリアンスが増加する

一致性

- これまで訓練データ量が固定の場合を見てきた
- 訓練データの量が増える場合の推定量の挙動
- 具体的には以下のようになってほしい

$$\lim_{m \to \infty} \hat{\theta}_m = \theta \tag{14}$$

- このような条件のことを一致性と呼ぶ
- 一致性によってデータ数が増えるとバイアスが減少することが保証される
  - その逆は真ではない
  - 漸近的に不偏であることは一致性を意味しない