クレジット:
UTokyo Online Educationデータマイニング入門 2018 森　純一郎

ライセンス:
利用者は、本講義資料を、教育的な目的に限ってページ単位で利用することができます。特に記載のない限り、本講義資料はページ単位でクリエイティブ・コモンズ 表示-非営利-改変禁止 ライセンスの下に提供されています。
http://creativecommons.org/licenses/by-nc-nd/4.0/

本講義資料内には、東京大学が第三者より許諾を得て利用している画像等や、各種ライセンスによって提供されている画像等が含まれています。個々の画像等を本講義資料から切り離して利用することはできません。個々の画像等の利用については、それぞれの権利者の定めるところに従ってください。

東京大学
THE UNIVERSITY OF TOKYO

CC BY NC ND

データマイニング入門 第9回

2018年度

UTokyo Online Education データマイニング入門 2018 森　純一郎 CC BY-NC-ND

学習目標

機械学習の概念について理解する

教師あり学習・回帰の概念について理解する

線形回帰

- コスト関数の最小化について理解する
- パラメータの推定
  - 最急降下法について理解する
- パラメータの解析解
  - 正規方程式について理解する

Pythonで線形回帰の実装を理解する

UTokyo Online Education データマイニング入門 2018 森　純一郎 CC BY-NC-ND

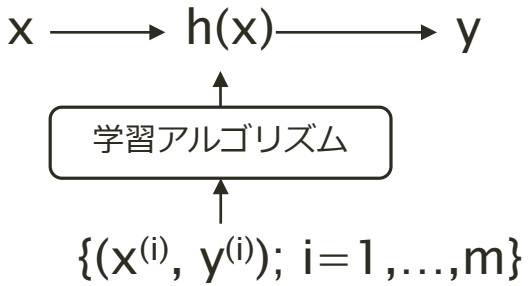
教師あり学習

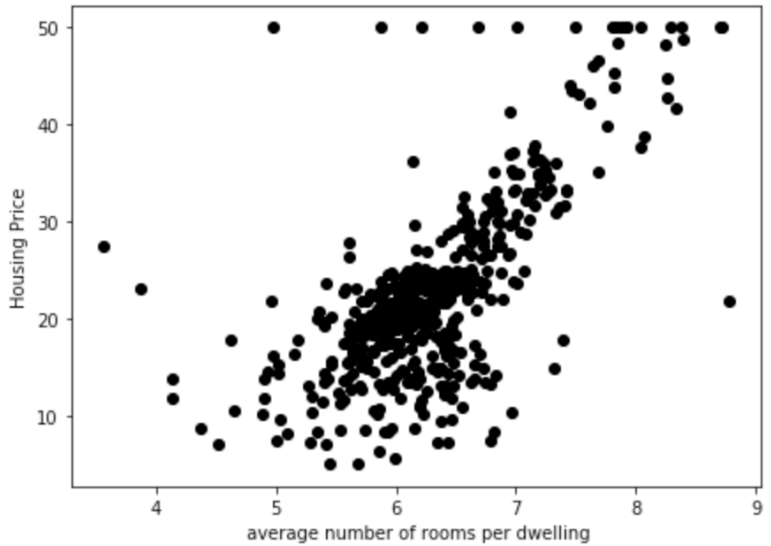
入力から出力を予測したい

- 特徴量ベクトル（入力）： $x^{(i)}$
- 正解（出力）： $y^{(i)}$
- 訓練データ：$(x^{(i)}, y^{(i)})$
- （訓練）データセット： $\{(x^{(i)}, y^{(i)}); i=1,\ldots,m\}$
  - iは学習データセットのインデックス

教師あり学習（supervised learning）

- データセットを元に入力から出力を予測する関数h(x)を学習する
  - hは仮説関数と呼ばれる





UTokyo Online Education データマイニング入門 2018 森　純一郎 CC BY-NC-ND

教師あり学習

特徴量ベクトル（入力）： $x^{(i)}$

正解（出力）： $y^{(i)}$

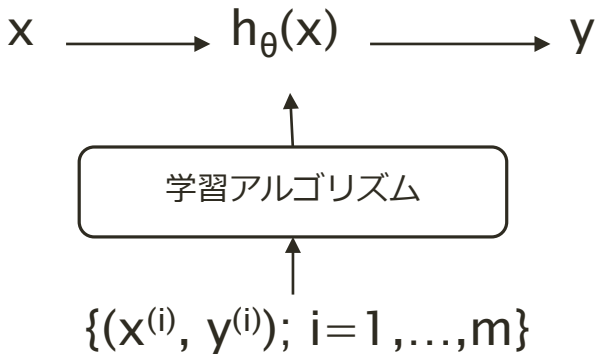
- 連続値：回帰
- 離散値：分類

データセット： $\{(x^{(i)}, y^{(i)}); i=1,\ldots,m\}$

仮説関数h(x)のパラメータ： θ

学習アルゴリズム

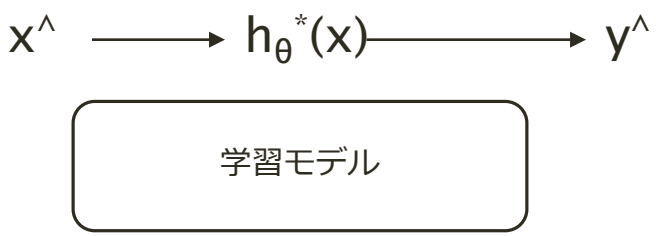
- データセットを元に仮説関数h(x)のパラメータを学習
  - コスト関数に基づくパラメータの最適化



未知の入力：x^　　予測： y^

学習モデル

- 未知の入力に対して尤もらしい出力を予測
- h(x)（最適パラメータθ^*）を元に入力x^からy^を予測



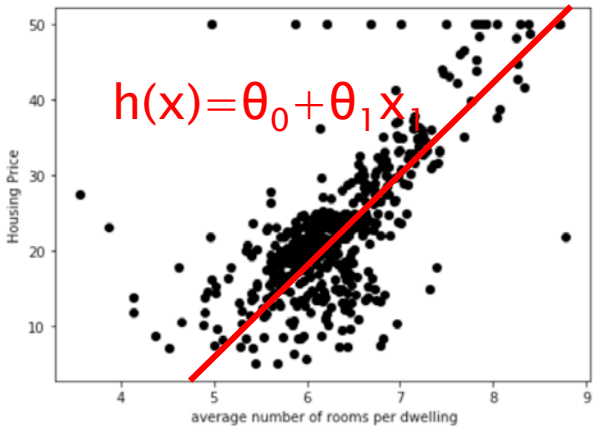
 CC BY-NC-ND

線形回帰

1変数の入力に対して出力を予測する線形回帰を考える

仮説関数h(x)の入力に対するパラメータ（重み）をθ_iとしてh(x)は

- $h(x)=\theta_0+\theta_1 x_1=\Sigma\theta_k x_k=\theta^T x$
  - $\theta=(\theta_0, \theta_1)^T$
  - $x=(x_0, x_1)^T$ (バイアス項$x_0=1$)
- $y=h(x)=b+ax$と同じ
  - θ_0は切片b, θ_1は傾きa



データセットを元に入力xから出力yを予測する関数h(x)のパラメータθを学習

学習した$h_\theta^*(x)$を元に新たな入力x^に対する出力y^を予測したい

 CC BY-NC-ND

コスト関数

どのように仮説関数のパラメータを学習するか

- データセット$\{(x^{(i)}, y^{(i)}); i=1,\ldots,m\}$について
- 仮説関数の出力$h(x^{(i)})$と実際の出力$y^{(i)}$がなるべく近くなるようなパラメータθを見つけたい

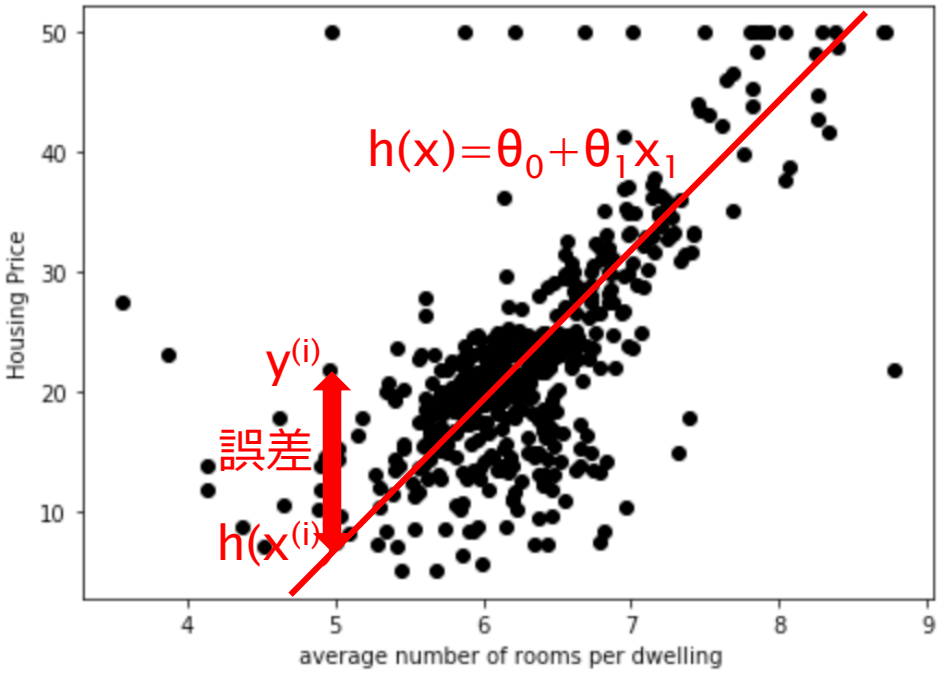
最小二乗法

- コスト関数（残差平方和）の最小化

$$J(\theta) = \frac{1}{2m}\Sigma_{i=1}^{m}(h(x^{(i)}) - y^{(i)})^2$$

データセット全体について$h(x^{(i)})$と$y^{(i)}$の誤差を少なくするようなパラメータθを見つけたい

入力が1変数の場合は誤差を少なくするような直線のパラメータ、傾きθ_1と切片θ_0、を見つけることになる



 CC BY-NC-ND

線形回帰

コスト関数

入力の変数が多変数（n次元の特徴量ベクトル）の時、仮説関数h(x)は

$$h(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \ldots + \theta_n x_n$$

この時、コスト関数は入力が1変数の時と同様に

$$J(\theta) = \frac{1}{2m}\Sigma_{i=1}^{m}(h(x^{(i)}) - y^{(i)})^2$$

m個のn+1次元入力データを以下の様に表すと（x_0はバイアス項で$x_0=1$とする）

$$X = \begin{pmatrix} x_0^{(1)} & x_1^{(1)} & \ldots & x_n^{(1)} \\ \ldots & \ldots & \ldots & \ldots \\ x_0^{(m)} & x_1^{(m)} & \ldots & x_n^{(m)} \end{pmatrix}$$

コスト関数は

$$J(\theta) = \frac{1}{2m}(X\theta - y)^T(X\theta - y)$$

$$\theta = (\theta_0, \theta_1, \ldots, \theta_n)^T$$

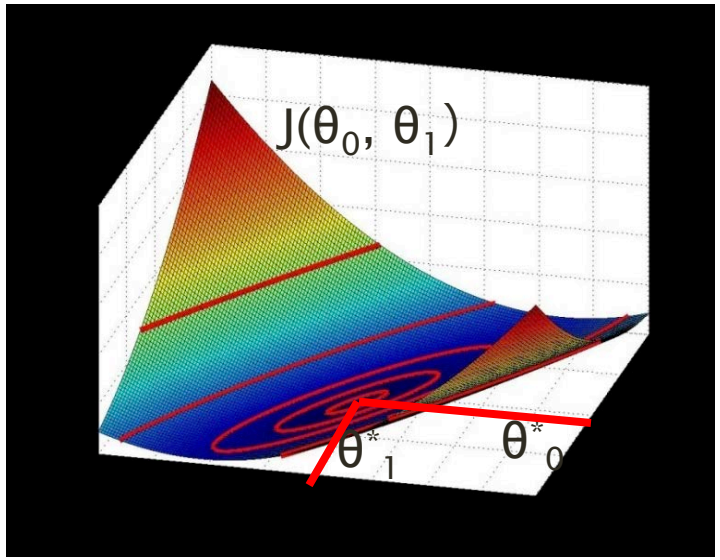
$$y = (y^{(1)}, y^{(2)}, \ldots, y^{(m)})^T$$

コスト関数の最小化

コスト関数を最小化するようなパラメータθをどのように求めるか

入力が1変数の場合で考えてみる

コスト関数$J(\theta)=J(\theta_0, \theta_1)$はパラメータ$\theta_0, \theta_1$について凸関数であり、$J(\theta_0, \theta_1)$が最小となるようなパラメータ$\theta^*_0, \theta^*_1$が見つけられればよい



 CC BY-NC-ND

勾配ベクトル

コスト関数$J(\theta)=J(\theta_0, \theta_1)$）をパラメータ$\theta_0, \theta_1$でそれぞれ偏微分した時のベクトル

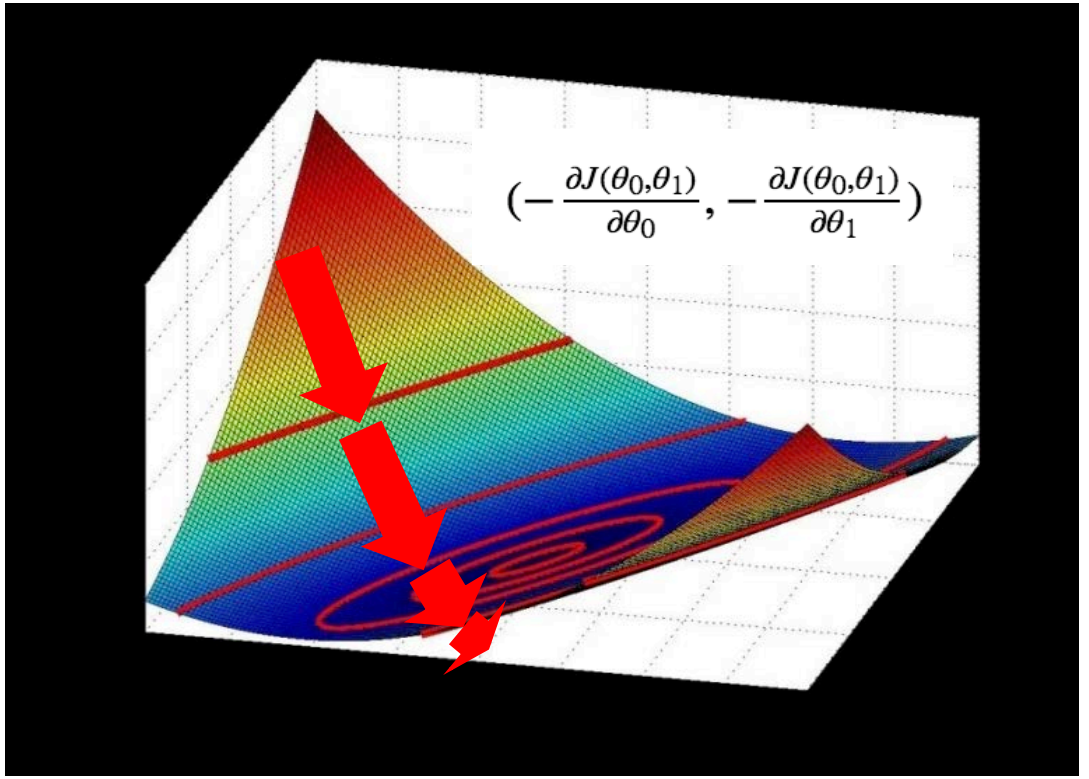
$$\nabla J(\theta_0, \theta_1) = \left(\frac{\partial J(\theta_0,\theta_1)}{\partial \theta_0}, \frac{\partial J(\theta_0,\theta_1)}{\partial \theta_1} \right)$$

勾配ベクトルの方向：

- その点から移動した時に関数の値が最も大きく（逆方向は小さく）なる方向
  - 図の曲面の等高線の法線ベクトルとなっている

勾配ベクトルの大きさ：

- その点から勾配ベクトルの方向に移動した時に関数の値は $\|\nabla J(\theta_0, \theta_1)\|$ だけ増える（逆方向は減る）



 CC BY-NC-ND

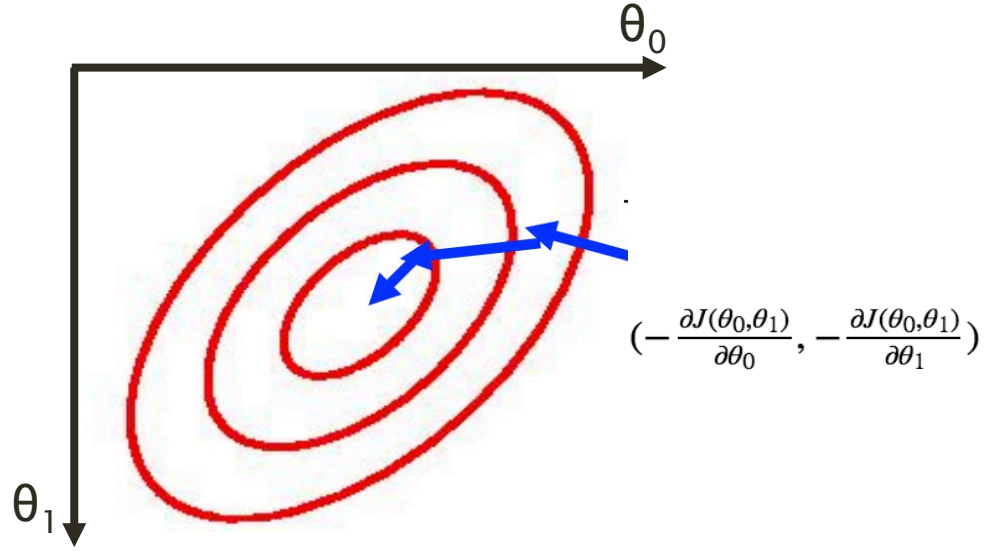
最急降下法

勾配ベクトルの定義より、$-\nabla J(\theta_0, \theta_1)$ の方向に進むようにパラメータを更新すれば $J(\theta_0, \theta_1)$ を減少させることができる

この時、どれぐらい勾配ベクトルの方向へ進むようかを定めるハイパーパラメータを α（学習率）として，パラメータθは任意の初期値から開始して$J(\theta_0, \theta_1)$ を減少させる方向に以下の様に更新できる

$$\theta_0 := \theta_0 - \alpha \frac{\partial J(\theta_0,\theta_1)}{\partial \theta_0} = \theta_0 - \alpha \Sigma_{i=1}^{m} (h(x^{(i)}) - y^{(i)})/m$$

$$\theta_1 := \theta_1 - \alpha \frac{\partial J(\theta_0,\theta_1)}{\partial \theta_1} = \theta_1 - \alpha \Sigma_{i=1}^{m} ((h(x^{(i)}) - y^{(i)})x^{(i)})/m$$



 CC BY-NC-ND

補：勾配ベクトルの導出

$$\frac{\partial J(\theta_0,\theta_1)}{\partial \theta_0} = \frac{1}{2m}\Sigma_{i=1}^{m}\frac{\partial(\theta_0+\theta_1 x^{(i)}-y^{(i)})^2}{\partial \theta_0} = \frac{1}{m}\Sigma_{i=1}^{m}(\theta_0+\theta_1 x^{(i)}-y^{(i)})$$

$$\frac{\partial J(\theta_0,\theta_1)}{\partial \theta_1} = \frac{1}{2m}\Sigma_{i=1}^{m}\frac{\partial(\theta_0+\theta_1 x^{(i)}-y^{(i)})^2}{\partial \theta_1} = \frac{1}{m}\Sigma_{i=1}^{m}(\theta_0+\theta_1 x^{(i)}-y^{(i)})\frac{\partial(\theta_0+\theta_1 x^{(i)}-y^{(i)})}{\partial \theta_1} = \frac{1}{m}\Sigma_{i=1}^{m}(\theta_0+\theta_1 x^{(i)}-y^{(i)})x^{(i)}$$

 CC BY-NC-ND

最急降下法

入力の変数が多変数（n次元の特徴量ベクトル）の時も同様に以下の様に各パラメータθ_jを更新してコスト関数J(θ)が最小となるパラメータを求めればよい

$$\theta_j := \theta_j - \alpha \frac{\partial J(\theta)}{\partial \theta_j} = \theta_j - \alpha \Sigma_{i=1}^{m} ((h(x^{(i)}) - y^{(i)}) x_j^{(i)}) / m$$

訓練データ$x^{(i)}$の誤差の大きさに応じてj次元のパラメータを調整

データセットの入力Xと正解yを元にコスト関数J(θ)を最小にするようなパラメータθを求める更新は以下のように表すことができる

$$\theta := \theta - \frac{\alpha}{m} X^T (X\theta - y)$$

$$X = \begin{pmatrix} x_0^{(1)} & x_1^{(1)} & \dots & x_n^{(1)} \\ \dots & \dots & \dots & \dots \\ x_0^{(m)} & x_1^{(m)} & \dots & x_n^{(m)} \end{pmatrix} \qquad y = (y^{(1)}, y^{(2)}, \dots, y^{(m)})^T$$

$$\theta = (\theta_0, \theta_1, \dots, \theta_n)^T$$

 CC BY-NC-ND

最急降下法

バッチ勾配降下法

- データセット全体の誤差に基づく勾配によりパラメータを更新
  - パラメータ更新の計算が遅い
  - 大規模なデータセットではオンメモリで処理できない
  - 非凸なコスト関数では局所最適解に陥る可能性がある

$$\theta_j := \theta_j - \alpha \frac{\partial J(\theta)}{\partial \theta_j} = \theta_j - \alpha \Sigma_{i=1}^{m} ((h(x^{(i)}) - y^{(i)}) x_j^{(i)}) / m$$

確率的勾配降下法

- ランダムに選択した各訓練データの誤差に基づく勾配によりパラメータを更新
  - パラメータ更新の計算が早い
  - 局所最適解から抜け出せる可能性がある
  - 最適解への収束が不安定
  - 学習率を徐々に小さくすることにより収束を安定させることができる

$$\theta_j := \theta_j - \alpha (h(x^{(i)}) - y^{(i)}) x_j^{(i)}$$

 CC BY-NC-ND

最急降下法

訓練データセット：$\{(x^{(i)}, y^{(i)}); i=1,\ldots,m\}$

1. 入力の各特徴量を標準化（必要であれば出力も標準化）
   - 特徴量のスケールを揃えることで最急降下法の収束を早めることができる
2. 入力の各データ$x^{(i)}$にバイアス項（$x_0^{(i)}=1$）を追加
3. パラメータの初期値を与える
4. 訓練データセットを元にパラメータを更新
   - パラメータが収束するあるいは一定の繰り返し回数が終了するまで
5. 学習されたパラメータを元にテストデータに対して予測を行いモデルを評価する
   - *学習モデルの選択と評価については次回説明

 CC BY-NC-ND

線形回帰

学習モデルの評価

訓練データセットで学習されたモデルのよさは、訓練データとは別に用意したテストデータに対して学習モデルを用いて予測を行い評価する

決定係数

- 学習モデルの予測値y^と実際のデータyとの相関係数の2乗
  - 観測データの分散のうちどれぐらいを学習モデルが説明できているかを表す
  - 1に近いほどデータへの学習モデルのあてはまりがよい

$$R^2 = \frac{(\Sigma_{i=1}^{m}(\hat{y}^{(i)}-\bar{\hat{y}})(y^{(i)}-\bar{y}))^2}{(\Sigma_{i=1}^{m}(\hat{y}^{(i)}-\bar{\hat{y}})^2)(\Sigma_{i=1}^{m}(y^{(i)}-\bar{y})^2)} = \frac{\Sigma_{i=1}^{m}(\hat{y}^{(i)}-\bar{\hat{y}})^2}{\Sigma_{i=1}^{m}(y^{(i)}-\bar{y})^2}$$

ただし$\bar{\hat{y}} = \bar{y}$

テストデータに対する残差平方和や十分小さい・決定係数が十分大きいならよい

一方、訓練データに学習モデルが適合しすぎてしまい、テストデータのような未知のデータでは予測がうまくいかなくなることがある（過学習の問題）

正則化などにより過学習を防ぎ、なるべく未知のデータへもあてはまるようにしたい（汎化）

 CC BY-NC-ND

正規方程式

コスト関数を最小化するようなパラメータは解析的に求めることが可能

コスト関数の勾配が0になるようなパラメータを求める

$$J(\theta) = \frac{1}{2m}(X\theta - y)^T(X\theta - y) \qquad \nabla J(\theta) = 0$$

以下の正規方程式が得られる

- 導出にはトレースを用いた行列の微分を使う

$$\theta = (X^T X)^{-1} X^T y$$

X^TXの逆行列が計算できればコスト関数を最小化する最適パラメータを解析的に求められる

X^TXが正則（フルランク）であること（階数rank(X^TX)が最大であること）

データ数 << 特徴量数の時は解が一意に定まらない（計算も大変$O(n^3)$）

- 正則でない場合はX^TXの対角成分に正の定数を加えて正則にする（過学習に対する正則化に関連）

 CC BY-NC-ND

多重共線性

入力データの特徴量間に強い相関があるような場合は注意が必要

正規方程式によりパラメータを求める場合のX^TXの正則性に関連

- X^TXが正則であるには、Xの各列（特徴量）が1次独立である必要がある

具体的な問題としては

- 相関がある特徴量それぞれにパラメータの重み与えようとしてパラメータの分散が大きくなる
- 特にパラメータの重みにより学習モデルを解釈する場合は問題になる

お互いに相関が強い特徴量があれば特徴量選択などによりいずれかの特徴量を採用するなどする

 CC BY-NC-ND

復習：正規分布

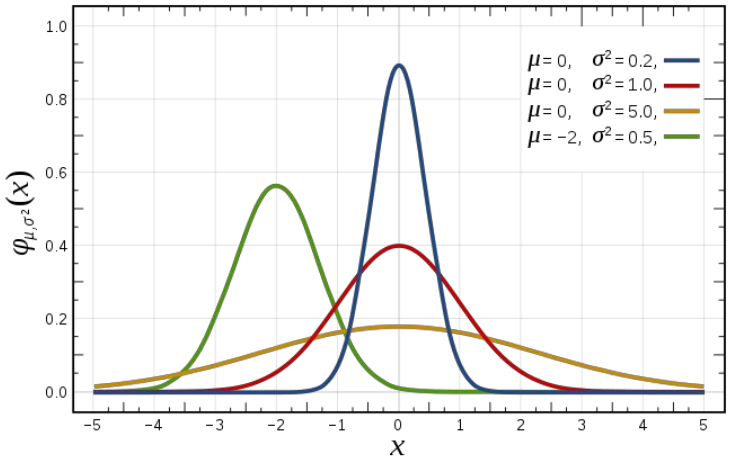
確率変数が正規分布に従うとき、その確率密度関数は

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \quad (x \in \mathbb{R})$$

正規分布に従う確率変数の平均（期待値）E(X)、分散V(X)は

- $E(X)=\mu$, $V(X)=\sigma^2$

正規分布は平均、分散の2つの母数で決まる



確率変数の平均（期待値）と分散

- 離散型
  - 平均
    - 確率変数のとりうる値とその確率の積

$$E(X) = \Sigma_i x_i P(X = x_i)$$

  - 分散

$$V(X) = \Sigma_i (x_i - E(X)) P(X = x_i)$$

- 連続型
  - 平均

$$E(X) = \int_{-\infty}^{\infty} x f(x) dx$$

  - 分散

$$V(X) = \int_{-\infty}^{\infty} (x - E(X))^2 f(x) dx$$

復習：最尤法によるパラメータ推定

最尤法によるパラメータ推定

- 最尤原理
  - 現実の標本は確率最大のものが実現した
- 尤度関数
  - パラメータθの元で、観測したデータがどの程度起こりうるかを表す関数
    - 確率分布（確率密度関数）の積をパラメータθの関数とみなしたもの

$$L(\theta) = \Pi_{i=1}^{n} f(x_i|\theta)$$

  - 尤度関数を（パラメータがとる値の集合の空間で）最大にするものを観測したデータに対するパラメータの推定量とする
- 対数尤度
  - 尤度関数の対数をとって和の形にしたもの
  - 観測したデータについて対数尤度を最大にするパラメータの推定：最尤推定

$$logL(\theta) = \Sigma_{i=1}^{n} f(x_i|\theta) \qquad \frac{\partial logL(\theta)}{\partial \theta} = 0$$

 CC BY-NC-ND

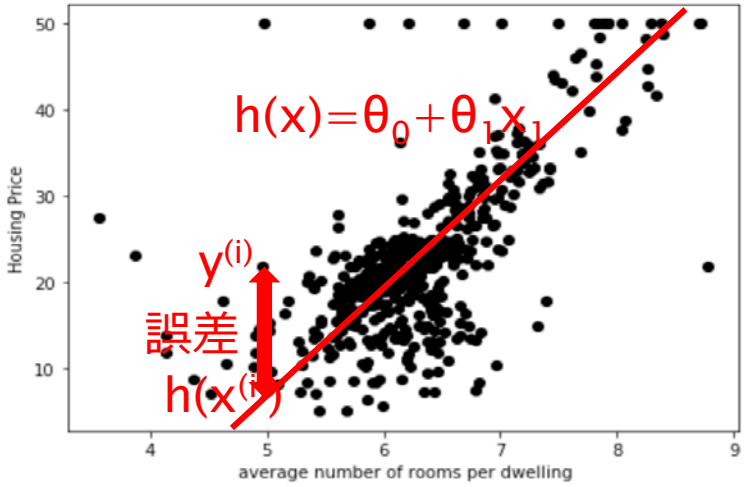
線形回帰
最尤推定による解釈

訓練データセットの出力$y^{(i)}$は平均$h(x^{(i)})$、分散σ^2の正規分布に従う確率変数の実現値であるとする

- $y^{(i)}$は$h(x^{(i)})$に平均0、分散σ^2の正規分布に従うようなノイズ$\epsilon^{(i)}$が合わさり観測されるとも考えられる
  - $y^{(i)} = h(x^{(i)}) + \epsilon^{(i)}$
- $\epsilon^{(i)}$は独立同分布と仮定する

$\epsilon^{(i)}$の確率密度関数（データ$x^{(i)}$が与えられた時の$y^{(i)}$の分布（パラメータはθ））は

$$f(\epsilon^{(i)}) = f(y^{(i)}|x^{(i)};\theta) = \frac{1}{\sqrt{2\pi}\sigma} exp(-\frac{(y^{(i)}-h(x^{(i)}))^2}{2\sigma^2})$$



最尤推定によりパラメータをする

対数尤度は

$$logL(\theta) = log\Pi_{i=1}^{m} f(y^{(i)}|x^{(i)};\theta) = log\Pi_{i=1}^{m} \frac{1}{\sqrt{2\pi}\sigma} exp(-\frac{(y^{(i)}-h(x^{(i)}))^2}{2\sigma^2})$$

 CC BY-NC-ND

最尤推定による解釈

対数尤度（つづき）

$$logL(\theta) = log\Pi_{i=1}^{m} \frac{1}{\sqrt{2\pi}\sigma} exp(-\frac{(y^{(i)}-h(x^{(i)}))^2}{2\sigma^2})$$

$$= \Sigma_{i=1}^{m} log \frac{1}{\sqrt{2\pi}\sigma} exp(-\frac{(y^{(i)}-h(x^{(i)}))^2}{2\sigma^2})$$

$$= mlog \frac{1}{\sqrt{2\pi}\sigma} - \frac{1}{2\sigma^2} \Sigma_{i=1}^{m} (y^{(i)} - h(x^{(i)}))^2$$

第1項は定数、対数尤度を最大にするには第2項を最小化すればよい

コスト関数に用いた残差平方和を最小化することに等しい $\Sigma_{i=1}^{m} (h(x^{(i)}) - y^{(i)})^2$

最小二乗法による線形回帰はパラメータの最尤推定をしているとも考えられる

 CC BY-NC-ND