機械学習入門

第5,6,7回　教師あり学習/線形回帰
重回帰モデル

前回までの復習から

単回帰モデル

$$Y = wX + b$$

このwとbを求めたい！

重回帰モデル

説明変数2つ以上

標準化/正規化

多重共線性に注意

重回帰モデル

- $Y = w_0 + w_1 X_1 + \cdots + w_n X_n$
- 単回帰モデルでは、切片と傾きの二つを求めたが、今回はたくさん。
- やることは同じ！
- 実測値とモデルの誤差が小さくなるようにwを決める→最小二乗法

再掲(コスト関数)

$$J(w) = \frac{1}{2N} \left\| y - w^{\mathrm{T}} \mathrm{X} \right\|^2$$

$$(\hat{w}) = argmin\, J(w)$$

この時のJを損失関数(cost function)という

（この損失関数を最小化する手法を最小二乗法(least square estimation)という）

→最小値を求めるには？

単回帰の場合(再掲)

$$J(a,b) = \sum_{k=1}^{\text{データ数}} (y_k - (aX + b))\text{^}2$$

$$(\hat{a}, \hat{b}) = argmin\, J(a,b)$$

$$\frac{\partial J(a,b\ )}{\partial a} = 0$$

$$\frac{\partial J(a,b\ )}{\partial b} = 0$$

という連立方程式を解けばよかった．が．今回は？？

重回帰モデル

$$\frac{\partial J(w_0, w_1, \cdots, w_n)}{\partial w_0} = 0$$

$$\frac{\partial J(w_0, w_1, \cdots, w_n)}{\partial w_1} = 0$$

$$\vdots$$

$$\frac{\partial J(w_0, w_1, \cdots, w_n)}{\partial w_n} = 0$$

微分値が 0 が最小値！

→でも変数がn個もある！

上の式を解くのは大変。。。（やってみるとわかる）

パラメータの求め方

- 勾配降下法：最小二乗法で定義したコスト関数の最小値を計算機で求める方法
- 正規方程式を解く：解析的に求める方法

線形回帰モデル(勾配降下法)

各変数を傾きに応じて変化させる.

$$w_0 \coloneqq w_0 - \alpha \frac{\partial J(w_0, w_1, \cdots, w_n)}{\partial w_0}$$

$$w_1 \coloneqq w_1 - \alpha \frac{\partial J(w_0, w_1, \cdots, w_n)}{\partial w_1}$$

$$\vdots$$

$$w_n \coloneqq w_n - \alpha \frac{\partial J(w_0, w_1, \cdots, w_n)}{\partial w_n}$$

勾配降下法によって，Jが最小になるところを求める.

線形回帰モデル

- 勾配降下法の実装

1,初期値wを決める

2,学習率αを決める

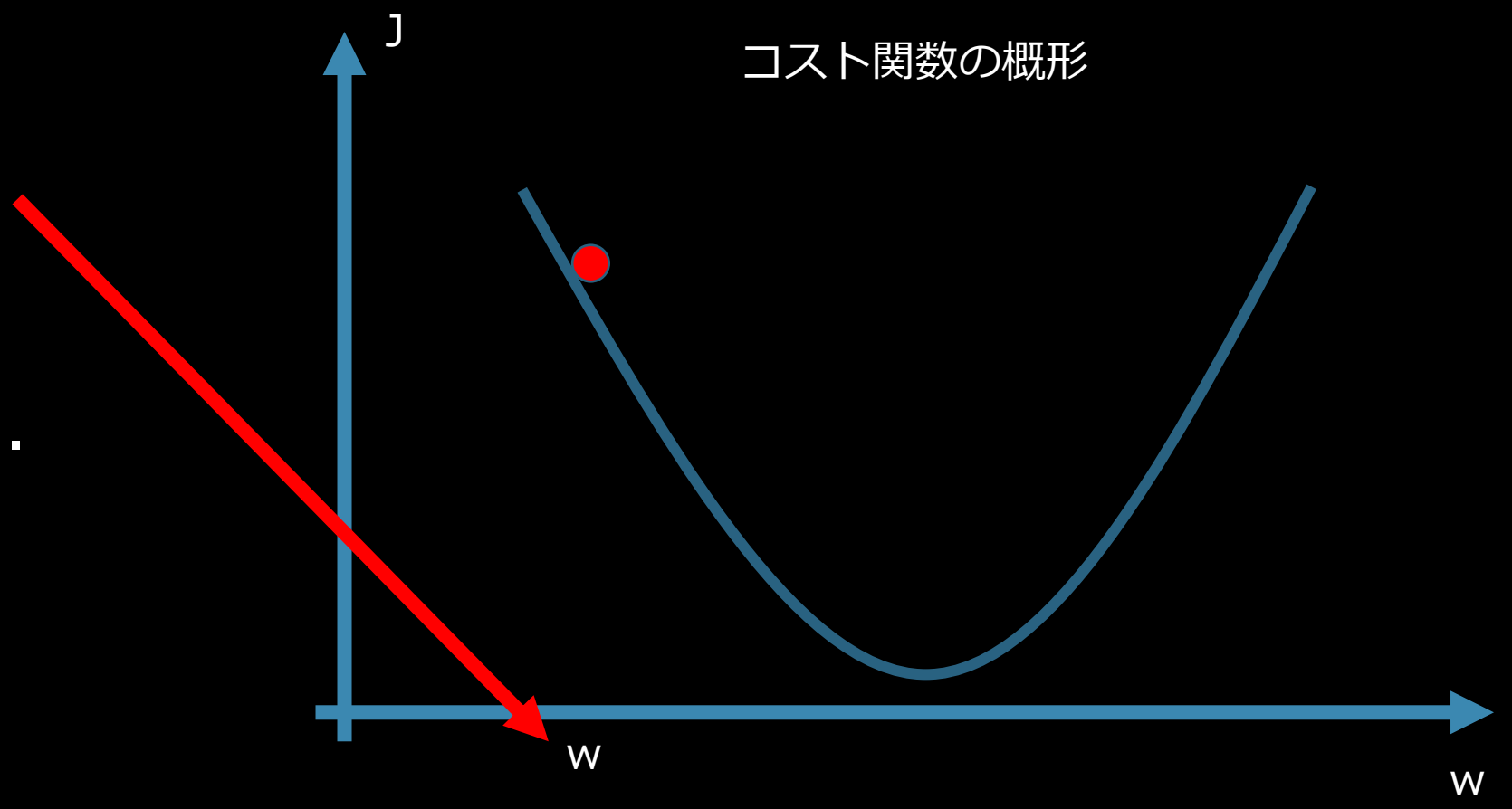
3,先の式に従ってwを更新する

勾配降下法

1. 初期値を決める.

→Jが求まる.

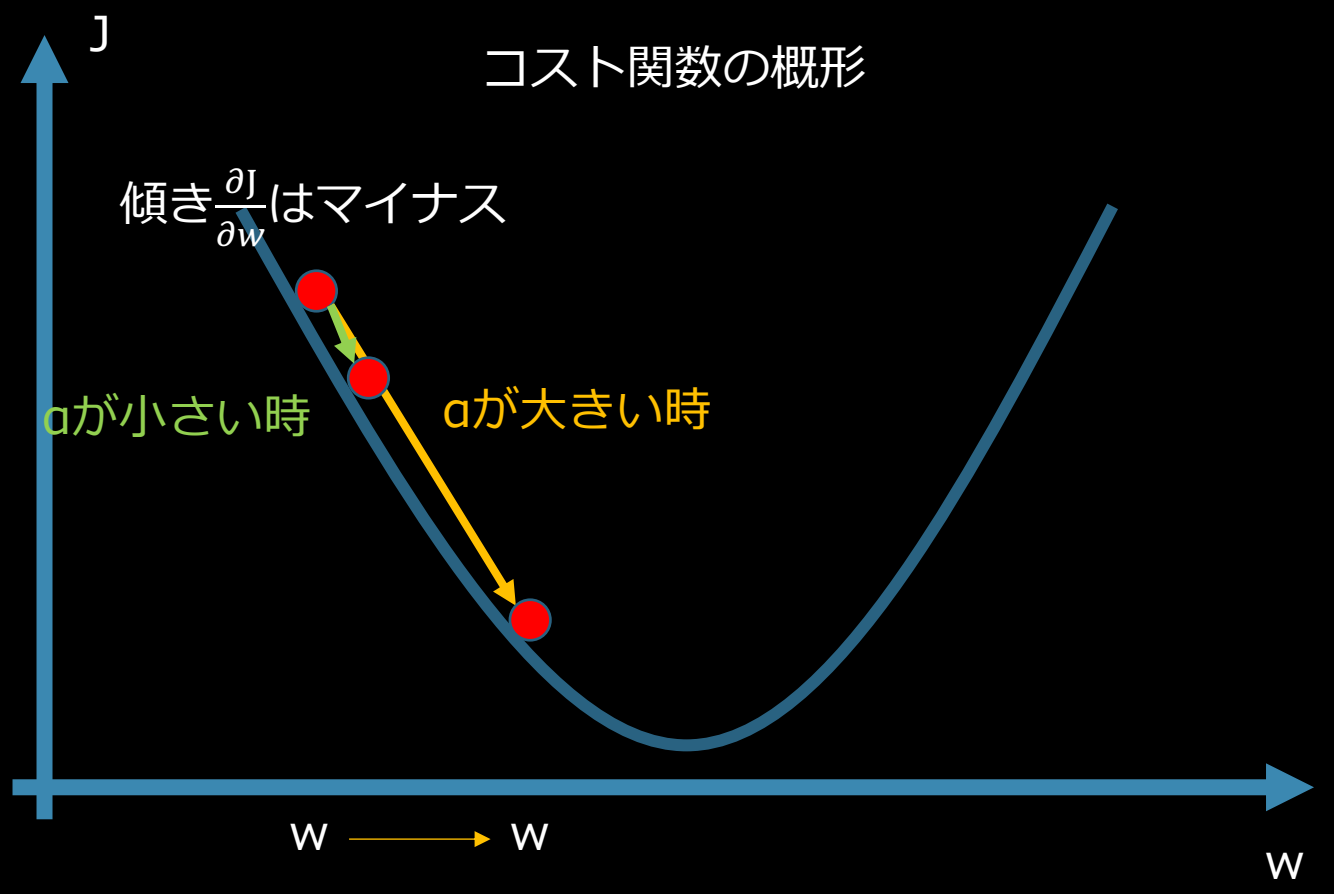
2. 学習率αも決める.



勾配降下法

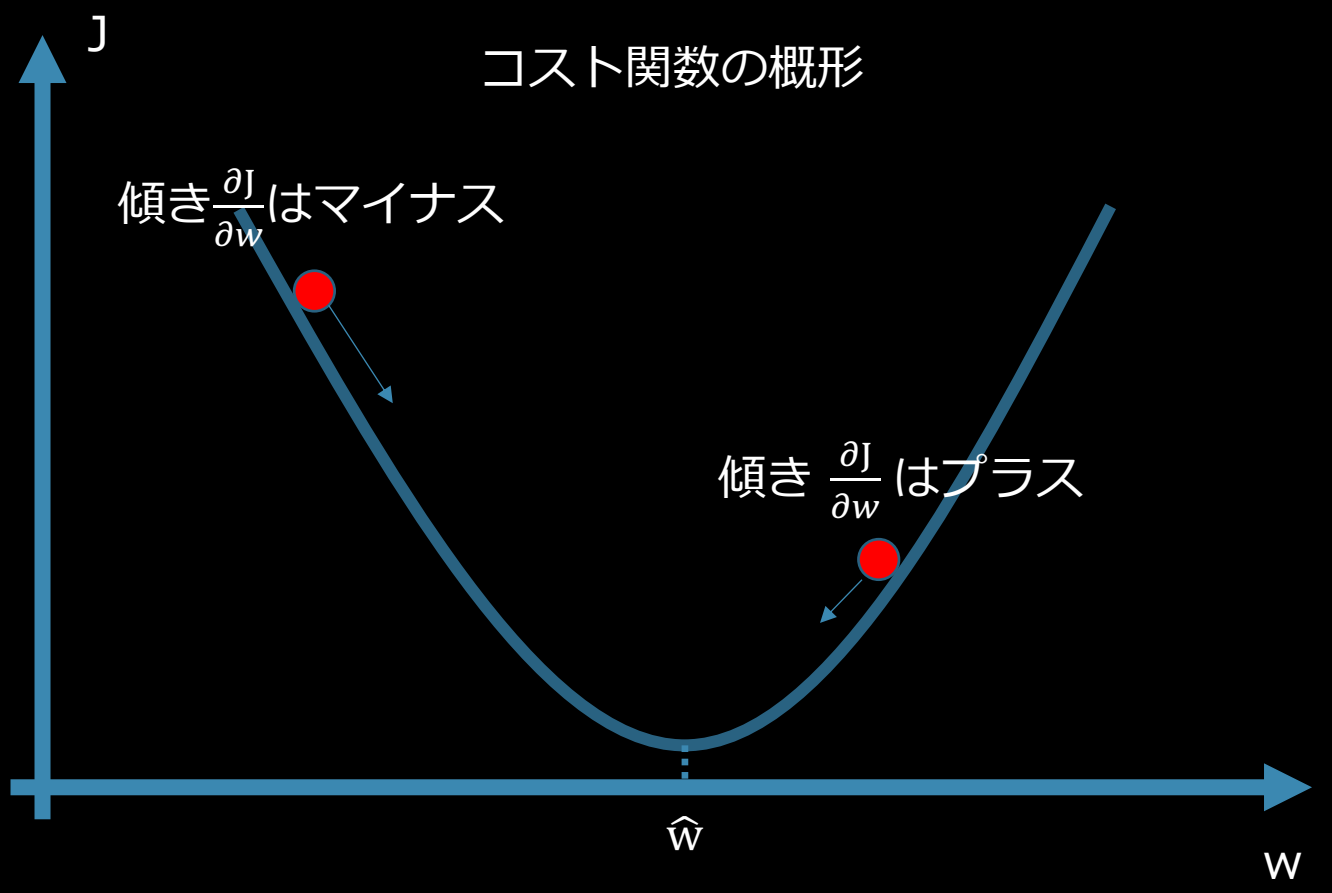
3. wを微分値に従って更新する.

$$w := w - \alpha \underbrace{\frac{\partial J(w)}{\partial w}}_{\text{マイナス}}$$



勾配降下法

- このような坂にボールを転がすのに似ている.
- wを傾きに従って更新する.
- 初期値、上手くパラメータを設定して行う.



線形回帰モデル

- 勾配降下法の実装

1,初期値wを決める

2,学習率αを決める

3,先の式に従ってwを更新する

4,決められた繰り返し回数に達するまで 3 を繰り返す

覚えてて欲しいこと（これだけは！）

学習率α、初期値wによって最適解がもとまらないことがある！

notebookへ

バッチ学習とミニバッチ学習

全データを一度に読み込ませることをバッチ学習といい，データセットを分割して，一部分ずつ読み込ませることをミニバッチ学習という．

尚，バッチ学習の場合は，１回の読み込みに対して，１エポックで有るが，ミニバッチ学習は複数回読み込ませて，全データ集合を使い切ったタイミングで１エポックである．

線形回帰モデル

行列の掛け算の式を使うと、線形回帰の式が簡単に書ける

・h(x)=$w_0 + w_1X_1 + \cdots + w_nX_n$

(Xは説明変数，X_0はバイアス項ですべて要素が1のベクトル)

• Y=$(w_0 \quad w_1 \quad \cdots \quad w_n)\begin{pmatrix} 1 \\ X_1 \\ \vdots \\ X_n \end{pmatrix} = w^TX$で、最終的に$w^T$を求めれば良い。

実装例を見てみる

線形回帰モデル

sklearnを用いて重回帰モデルを実装せよ

california_housingデータセットを使用せよ

コードは重回帰.ipynbを参考に。

わからない部分があったらきくこと！

パラメータの意味

目的変数y：あるケーキ屋の月の売上（万円）

$\boldsymbol{x_1}$:　月のインスタグラムの投稿数（回）

$\boldsymbol{x_2}$:　月の来客数（人）

$$y\text{（万円）} = w_0 + w_1\boldsymbol{x_1} + w_2\boldsymbol{x_2}$$

この$\boldsymbol{w_1}$とか$\boldsymbol{w_2}$は他の説明変数を一定にして，その説明変数を１単位だけ増加させたら目的変数がどう変わるか？を表す.

標準化/正規化

・ざっくりいうと、
データをある基準に合わせて比べられるようにしよう！
ということ

標準化/正規化

目的変数y：あるケーキ屋の月の売上（万円）

$\boldsymbol{x_1}$:　月のインスタグラムの投稿数（回）

$\boldsymbol{x_2}$:　月の来客数（人）

$$\boldsymbol{y}\text{（万円）} = w_0 + w_1\boldsymbol{x_1} + w_2\boldsymbol{x_2}$$

来客数を1000人，インスタの投稿数を20回，売上を30万円とするとパラメータのオーダーはいくらくらいになりそうか？

標準化/正規化

目的変数y：あるケーキ屋の月の売上（万円）

$\boldsymbol{x_1}$:　月のインスタグラムの投稿数（回）

$\boldsymbol{x_2}$:　月の来客数（人）

$$y\text{（万円）} = w_0 + w_1\boldsymbol{x_1} + w_2\boldsymbol{x_2}$$

来客数を1000人，インスタの投稿数を20回，売上を30万円とするとパラメータのオーダーはいくらくらいになりそうか？

→およそ，w_1は0.01くらい，w_2は１くらいのオーダー

→つまり，インスタの投稿数が同じ月で，月の来客数が1001人の月は売り上げが0.01万円上がるという予測が立つ.

→それでは，回帰に一番寄与の大きい説明変数はどれだろうか？

標準化/正規化

目的変数y：あるケーキ屋の月の売上（万円）

$\boldsymbol{x_1}$:　月のインスタグラムの投稿数（回）

$\boldsymbol{x_2}$:　月の来客数（人）

$$y\text{（万円）} = w_0 + w_1 x_1 + w_2 x_2$$

回帰係数の影響度を調べるために，標準化/正規化という（単位の影響を無くす）操作を行う．

標準化/正規化.ipynbへ

標準化回帰モデルを実装してみよう

線形回帰モデル

多重共線性(multicollinearity)

説明変数間に強い相関がある場合に起こる

通称マルチコ。

係数が安定しなかったり、未知データに対する予測精度が落ちたり、統計的には回帰係数の値が解釈しづらかったりする。

機械学習の文脈では、正則化を行う、相関が高いデータの片方を削除するなどで対策する。

線形回帰モデル

多重共線性(multicollinearity)

機械学習におけるマルチコ.

一般には過学習を引き起こす.

今回の例では，そこまで過学習していないっぽいが、例えば説明変数の次元が多くなったとき、そのうちいくつもの説明変数の相関が高ければ、過学習を引き起こす可能性がある.

線形回帰モデル

多重共線性(multicollinearity)

統計的な解釈におけるマルチコ.

偏回帰係数は他の変数を一定に保ち、その変数だけ１単位増加させたときに目的変数がどのように変化するかを表している.
しかし互いに相関の高い説明変数が存在する場合、その変数だけを１単位増加させるという考えが成り立たなくなる.

それに加えて，推定した回帰係数の分散が大きくなる．これによりも推定量の信頼性を損なわせる.

線形回帰モデル

そこで．．．**正則化**という考え方を導入する．

実は，線形回帰の係数の推定式は次のように行列で書ける．

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

これを計算したいが，多重共線性が起きている場合には
$(X^T X)^{-1}$が存在しない．よって係数を推定できない．．．

どうする．．．

Tips

実は、行列は連立方程式を解くために作り出されたもの.

$$\begin{cases} x + y = 2 \\ x - y = 4 \end{cases} \rightarrow \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 2 \\ 4 \end{pmatrix}$$

左から行列を掛け算して、(x,y)を求められないだろうか.

つまり、 $\mathrm{B}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} = B \begin{pmatrix} 2 \\ 4 \end{pmatrix}$

となるようなBがないだろうか.

Tips

B$=\begin{pmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & -\frac{1}{2} \end{pmatrix}$とすれば、 B$\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix}$と

なって、B @$\begin{pmatrix} 2 \\ 4 \end{pmatrix} = \begin{pmatrix} 3 \\ -1 \end{pmatrix}$と解が求まる.

このときのBをA=$\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$の逆行列といい，$A^{-1}$と書く.

逆行列の計算方法は割愛．ここから本題．多重共線性が起きているときに逆行列を持たないとはどういうときなんだろうか？

Tips

$$\begin{cases} x + y = 2 \\ 2x + 2y = 4 \end{cases}$$

中学でやった通り、上記の連立方程式は無数の解を持つ.

すなわち、解が不定である.

行列で書くと、$\begin{pmatrix} 1 & 1 \\ 2 & 2 \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 2 \\ 4 \end{pmatrix}$ であるが、相関が強い場合データがこのようになっているのが多重共線性である.（2つの式は本質的に同値という意味)

なので、多重共線性がある->解が決まらない->逆行列を持たない、のである.

線形回帰モデル

$$\begin{cases} x + y = 2 \\ 2x + 2y = 4 \end{cases}$$

このままでは解を持たないので、正則化が何をするのかというと、

$$\begin{pmatrix} 1 & 1 \\ 2 & 2 \end{pmatrix} \rightarrow \begin{pmatrix} 1+\lambda & 1 \\ 2 & 2+\lambda \end{pmatrix}$$

と、少しだけ値を加えてやる．そうすることで解を持つようになり、逆行列の計算が行える．

この操作を正則化という．

線形回帰モデル

$$\hat{\theta} = (X^T X)^{-1} X^T y \rightarrow (X^T X + \lambda I)^{-1} X^T y$$

これは**Ridge正則化**と呼ばれる.

よくみる形は

$$J(\theta) = \left\| y - \theta^{\mathrm{T}} \mathrm{X} \right\|^2 + \lambda \sum \|\theta\|^2$$

$$\left(\hat{\theta}\right) = argmin\, J(\theta)$$

これを微分して、解析的に求めると上記の式が得られる.

線形回帰モデル

$$J(\theta) = \left\| y - \theta^{\mathrm{T}} \mathrm{X} \right\|^2 + \lambda \sum \|\theta\|$$

$$(\hat{\theta}) = argmin\, J(\theta)$$

ちなみに．このような正則化を**Lasso正則化**という．

これの最適解は座標降下法とかを使って求める．(微分のような解析操作ができない)

線形回帰モデル

Ridge回帰とLasso回帰を実装してみよう.

hint:

sklearn.linear_model.Lasso

sklearn.linear_model.Ridge

線形回帰モデル

正則化は過学習を防ぐのにも使える.

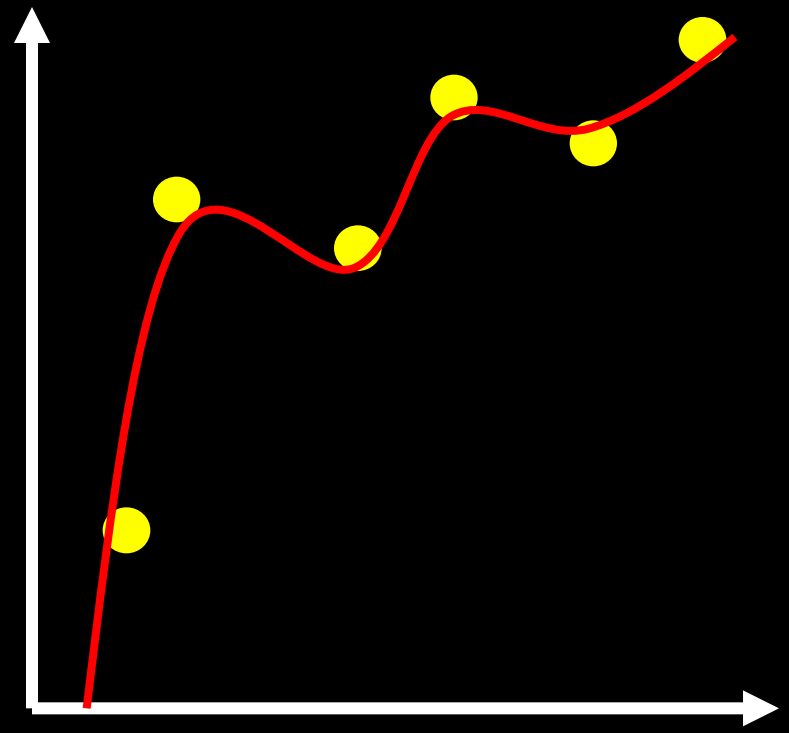
$$J(\theta) = \left\| y - \theta^{\mathrm{T}} \mathrm{X} \right\|^2 + \lambda \sum \|\theta\|^2$$

$$J(\theta) = \left\| y - \theta^{\mathrm{T}} \mathrm{X} \right\|^2 + \lambda \sum \|\theta\|$$

この最後の項によってコスト関数が0に近くなりすぎないようになっている.

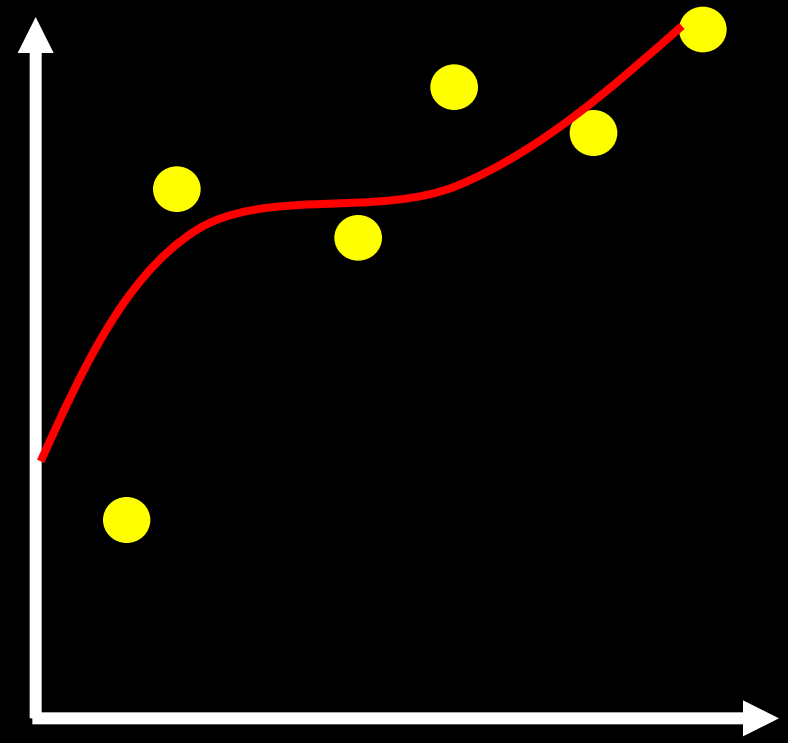
＃正則化

過学習！



正則化

ちょっとマシ。



以下は以前に作った参考資料です

行列(matrix)

- 行列とは、「数字を並べたもの」
- 例えば、$(1), (2 \quad 3), \begin{pmatrix} 4 \\ 5 \\ 6 \end{pmatrix}, \begin{pmatrix} 7 & 8 \\ 9 & 10 \\ 11 & 12 \end{pmatrix}$のようなもの
- 横に並んだものを行、縦に並んだものを列という。
- 行数と列数によって、(1)は1×1行列,(2　3)は1×2行列, $\begin{pmatrix} 4 \\ 5 \\ 6 \end{pmatrix}$は3×1行列,$\begin{pmatrix} 7 & 8 \\ 9 & 10 \\ 11 & 12 \end{pmatrix}$は3×2行列、などと言う(行数×列数)。

行列の成分

- $\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1j} \\ a_{22} & a_{22} & \ldots & a_{2j} \\ \vdots & \vdots & \ddots & \vdots \\ a_{i1} & a_{i2} & \ldots & a_{ij} \end{pmatrix}$
- 書き並べた数字(要素)は成分と呼ばれ、第i行目、第j列目の成分を、行列の(i,j)成分という。
- 例えば、$\begin{pmatrix} 7 & 8 \\ 9 & 10 \\ 11 & 12 \end{pmatrix}$の(3,1)成分は、11。

行列の計算

- まずは、足し算(引き算)から。
- 足し算(引き算)は成分ごとに行う。(すなわち、行列の行数と列数が一致している必要がある。)
- ex) $(2 \quad 3)+(0 \quad -1)=(2+0 \quad 3+(-1))=(2 \quad 2)$

$$\begin{pmatrix}4\\5\\6\end{pmatrix}+\begin{pmatrix}-4\\3\\2\end{pmatrix}=\begin{pmatrix}4+(-4)\\5+3\\6+2\end{pmatrix}=\begin{pmatrix}0\\8\\8\end{pmatrix}$$

$$\begin{pmatrix}7 & 8\\9 & 10\\11 & 12\end{pmatrix}+\begin{pmatrix}1 & -3\\-1 & 2\\-9 & 8\end{pmatrix}=\begin{pmatrix}7+1 & 8+(-3)\\9+(-1) & 10+2\\11+(-9) & 12+8\end{pmatrix}=\begin{pmatrix}8 & 5\\8 & 12\\2 & 20\end{pmatrix}$$

行列の計算

- 次は掛け算。
- スカラー(実数)倍。全部にスカラー値をかけてあげればよい。
- $3\times\begin{pmatrix}7 & 8\\ 9 & 10\\ 11 & 12\end{pmatrix}=\begin{pmatrix}7\times3 & 8\times3\\ 9\times3 & 10\times3\\ 11\times3 & 12\times3\end{pmatrix}=\begin{pmatrix}21 & 24\\ 27 & 30\\ 33 & 36\end{pmatrix}$

行列の計算

- 行列同士の掛け算
- ちょっと複雑。
- $A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1j} \\ a_{22} & a_{22} & \cdots & a_{2j} \\ \vdots & \vdots & \ddots & \vdots \\ a_{i1} & a_{i2} & \cdots & a_{ij} \end{pmatrix}, \mathrm{B} = \begin{pmatrix} b_{11} & b_{12} & \cdots & b_{1j} \\ b_{22} & b_{22} & \cdots & b_{2j} \\ \vdots & \vdots & \ddots & \vdots \\ b_{i1} & b_{i2} & \cdots & b_{ij} \end{pmatrix}$

  とする。

行列の計算

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1j} \\ a_{21} & a_{22} & \cdots & a_{2j} \\ \vdots & \vdots & \ddots & \vdots \\ a_{i1} & a_{i2} & \cdots & a_{ij} \end{pmatrix}, \mathrm{B} = \begin{pmatrix} b_{11} & b_{12} & \cdots & b_{1k} \\ b_{21} & b_{22} & \cdots & b_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ b_{j1} & b_{j2} & \cdots & b_{jk} \end{pmatrix}$$

この時、積AB=

$$\begin{pmatrix} \sum_{m=1}^{i} a_{1m}b_{m1} & \sum_{m=1}^{i} a_{1m}b_{m2} & \cdots & \sum_{m=1}^{i} a_{1m}b_{mk} \\ \sum_{m=1}^{i} a_{2m}b_{m1} & \sum_{m=1}^{i} a_{2m}b_{m2} & \cdots & \sum_{m=1}^{i} a_{2m}b_{mk} \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{m=1}^{i} a_{im}b_{m1} & \sum_{m=1}^{i} a_{im}b_{m2} & \cdots & \sum_{m=1}^{i} a_{im}b_{mk} \end{pmatrix}$$ となる。

行列の計算

- 例えば、 $(2 \quad 3)\times \begin{pmatrix}1\\0\end{pmatrix}=(2\times1+3\times0)=(2)$,

$$(1 \quad 0 \quad 3)\times\begin{pmatrix}4\\5\\6\end{pmatrix}=(1*4+0*5+3*6)=(4+0+18)=(22)$$

$$\begin{pmatrix}2 & -3 & 1\\-4 & -1 & 5\end{pmatrix}\begin{pmatrix}7 & 8\\9 & 10\\11 & 12\end{pmatrix}=$$

$$\begin{pmatrix}2*7+(-3)*9+1*11 & 2*8+(-3)*10+1*12\\(-4)*7+(-1)*9+5*11 & (-4)*8+(-1)*10+5*12\end{pmatrix}$$

$$=\begin{pmatrix}14-27+11 & 16+30+12\\-28-9+55 & -32-10+60\end{pmatrix}=\begin{pmatrix}-2 & 58\\18 & 18\end{pmatrix}$$

行列の計算

- 注意点
- i×m行列とm×l行列の積はi×l行列になる。
- 赤字のところが一致していないと行列の掛け算はできない。(定義より)
- 回帰モデルを作るのにreshapeメソッドを使っていたが、行列計算を行うのに上記のような理由があったから。
- 行列のかけ算は順番が大事。(A*B)≠(B*A)

行列の計算

- 転置行列(transpose matrix)と逆行列(inverse matrix)
- Aの転置行列をA^T,逆行列をA^{-1}と書く。
- $A=\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1j} \\ a_{21} & a_{22} & \cdots & a_{2j} \\ \vdots & \vdots & \ddots & \vdots \\ a_{i1} & a_{i2} & \cdots & a_{ij} \end{pmatrix}$,としたとき

$A^T=\begin{pmatrix} a_{11} & a_{21} & \cdots & a_{i1} \\ a_{12} & a_{22} & \cdots & a_{i2} \\ \vdots & \vdots & \ddots & \vdots \\ a_{1j} & a_{1j-1} & \cdots & a_{ij} \end{pmatrix}$である。

線形回帰モデル

- 今説明した掛け算の式を用いると線形モデルの式が簡単に書ける。
- h(x)=$\theta_0 X_0 + \theta_1 X_1 + \cdots + \theta_n X_n$(Xは授業内で用いた、平均の部屋数や、犯罪率などが入っている説明変数、n=1の時が単回帰モデルの時の式)(X_0はバイアス項ですべて要素が1の行列)
- Y=$(\theta_0 \quad \theta_1 \quad \cdots \quad \theta_n)\begin{pmatrix} X_0 \\ X_1 \\ \vdots \\ X_n \end{pmatrix} = \theta^T X$で、最終的に$\theta^T$を求めれば良い。

線形回帰モデル

- 誤差を考える。

$$J(\theta_0, \theta_1, \cdots, \theta_n) = \frac{1}{2m}(\theta^T X - Y)^2$$

これをθ_i(i=0,1,・・・n)について偏微分して、連立方程式を解くか、最急勾配法を用いてJ(θ)のminを求め、その時のθをパラメータとする。

線形回帰モデル(最急降下法)

$$\theta_0 \coloneqq \theta_0 - \alpha \frac{\partial J(\theta_0, \theta_1, \cdots, \theta_n)}{\partial \theta_0}$$

$$\theta_1 \coloneqq \theta_1 - \alpha \frac{\partial J(\theta_0, \theta_1, \cdots, \theta_n)}{\partial \theta_1}$$

$$\vdots$$

$$\theta_n \coloneqq \theta_n - \alpha \frac{\partial J(\theta_0, \theta_1, \cdots, \theta_n)}{\partial \theta_n}$$

Θを次々に更新していくと、先のスライドにあるような坂を
ボールが運動するようなイメージでパラメータが更新される。

このとき、αを正しく設定する必要がある。

線形回帰モデル(正規方程式)

- データが独立(データ間の相関が強くない)などの条件であれば正規方程式によってパラメータθを求めることが出来る。

$$\text{正規方程式}：\theta = (X^T X)^{-1} X^T y$$

導出は割愛。

正規方程式と勾配降下法

	勾配降下法	正規方程式
メリット	特徴量が増えても遅くなりにくい	繰り返し計算が不要 αを決めなくて良い
デメリット	初期値やαを上手く決める必要がある。 繰り返しによって計算が遅くなる可能性	特徴量が増えると計算が遅くなる。(次元の3乗) 多重共線性によって逆行列を持たない場合があり、その場合は回帰係数が安定しない