Regression (回歸分析)

原理

1. Cost function

任何能夠衡量模型預測出來的值h(θ)與真實值y之間的差異的函數都可以叫做代價函數C(θ)。其中，代價函數是參數θ的函數，如果有多個樣本，則可以將所有代價函數的取值求均值，記做J(θ)。代價函數可以用來評價模型的好壞，代價函數越小說明模型和參數越符合訓練樣本(x, y)。當我們確定了模型h，後面做的所有事情就是訓練模型的參數θ。因此訓練參數的過程就是不斷改變θ，從而得到更小的J(θ)的過程。理想情況下，當我們取到代價函數J的最小值時，就得到了最優的參數θ。

在線性回歸中，最常用的是均方誤差(Mean squared error)，即

$$E(\theta_0, \theta_1) = \frac{1}{2}\sum_{i=1}^{m}\left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$$

2. 最小平方法 vs 梯度下降法

兩種方法都是在給定已知資料（independent & dependent variables）的前提下對 dependent variables 算出出一個一般性的估值函式。然後對給定新資料的 dependent variables 進行估算。都是在已知資料的框架內，使得估算值與實際值的總平方差儘量更小(未必一定要使用平方)。

最小平方法是直接對△求導找出全域性最小，是非迭代法。而梯度下降法是一種迭代法，先給定一個β，然後向△下降最快的方向調整β，在若干次迭代之後找到區域性最小。

最小平方法:

找到一組θ_0、θ_1，使的E(θ_0, θ_1)最小。

$$\frac{\partial E(\theta_0, \theta_1)}{\partial \theta_0} = \sum\left(\theta_0 + \theta_1 x^{(i)} - y^{(i)}\right) \equiv 0$$

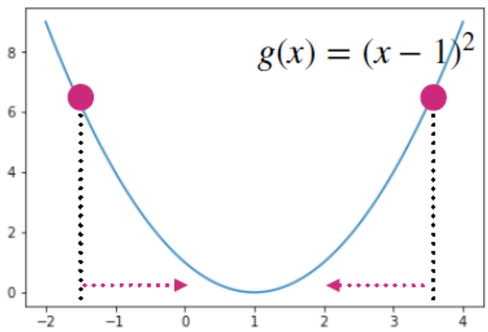
$$\frac{\partial E(\theta_0, \theta_1)}{\partial \theta_1} = \sum\left(\theta_0 + \theta_1 x^{(i)} - y^{(i)}\right) \cdot x^{(i)} \equiv 0$$

$$\theta_1 = \frac{n\sum x^{(i)}y^{(i)} - \sum x^{(i)}\sum y^{(i)}}{n\sum (x^{(i)})^2 - \left(\sum x^{(i)}\right)^2} = \frac{\sum\left(x^{(i)} - \bar{x}\right)\left(y^{(i)} - \bar{y}\right)}{\sum\left(x^{(i)} - \bar{x}\right)^2}$$

$$\theta_0 = \bar{y} - \theta_1\bar{x}$$

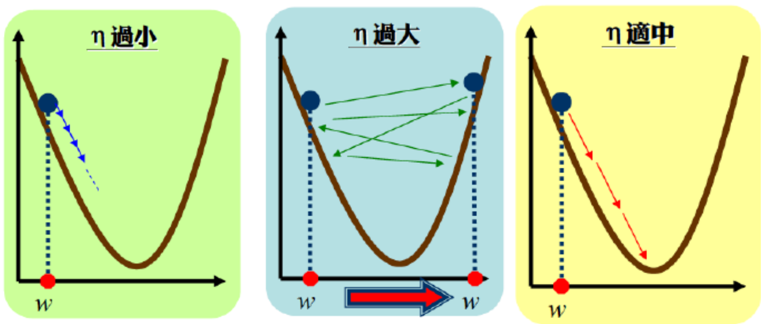
梯度下降法:

梯度下降法是一種不斷去更新參數(這邊參數用 x 表示)找「解」的方法，所以一定要先隨機產生一組初始參數的「解」，然後根據這組隨機產生的「解」開始算此「解」的梯度(倒函數、一次微分、切線斜率)方向大小，然後將這個「解」去減去梯度，往與導函數相反的方向移動，就會往最小值的方向移動。而找「解」的時候公式是往梯度的方向更新，一次要更新多少，就是由學習率(learning rate)來控制，學習率的大小會影響最佳化的效果。



$$x := x - \eta \frac{d}{dx} g(x)$$

η : learning rate



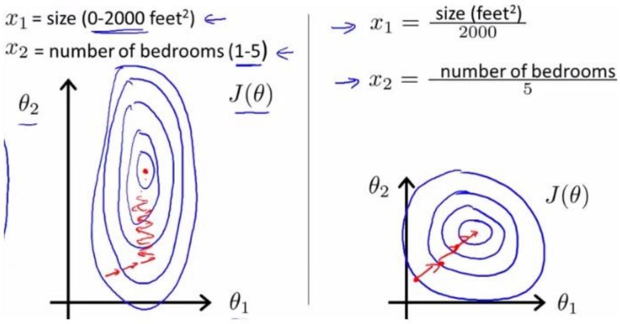
$$\theta_0 := \theta_0 - \eta \frac{\partial E}{\partial \theta_0}$$

$$\theta_1 := \theta_1 - \eta \frac{\partial E}{\partial \theta_1}$$

- 特徵正規化 (normalization)

  將特徵資料按比例縮放，讓資料落在某一特定的區間。可以提升優化速度並提高精確度。

  如下方圖示，藍色圈圈代表的是特徵的等高線，左圖的特徵 X1,X2 區間相差非常大，所以對應的等高線非常尖，會導致在使用梯度下降法尋求最佳解時，需要迭代多次才可以收斂。



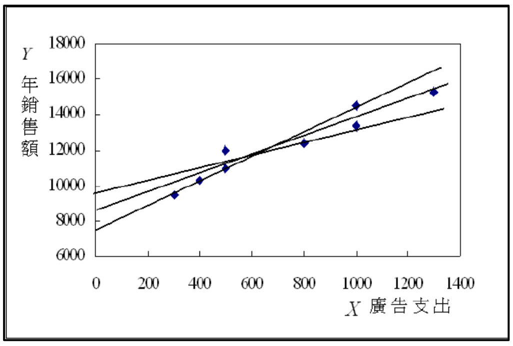
除了上述說明了可以優化梯度下降法外，還可以提高精準度。有的分類器需要計算樣本間的距離，例如之前所提到的 KNN，一個特徵值的範圍非常大，那麼距離計算通常就會取決於這個特徵，若情況是範圍小的特徵比較重要的話，就會與我們所要的結果是相反的。

常有兩種標準化的方法: 1. min max normalization，會將特徵數據按比例縮放到 0 到 1 的區間，(或是-1 到 1)。2. standard deviation normalization，會將所有特徵數據縮放成平均為 0、平方差為 1。

3. 簡單線性迴歸 (Simple Linear Regression)

簡單線性迴歸包含一個自變數(x)和一個因變數(y)，兩個變數的關係用一條直線來模擬。

$E(y) = \beta_0+\beta_1 x$



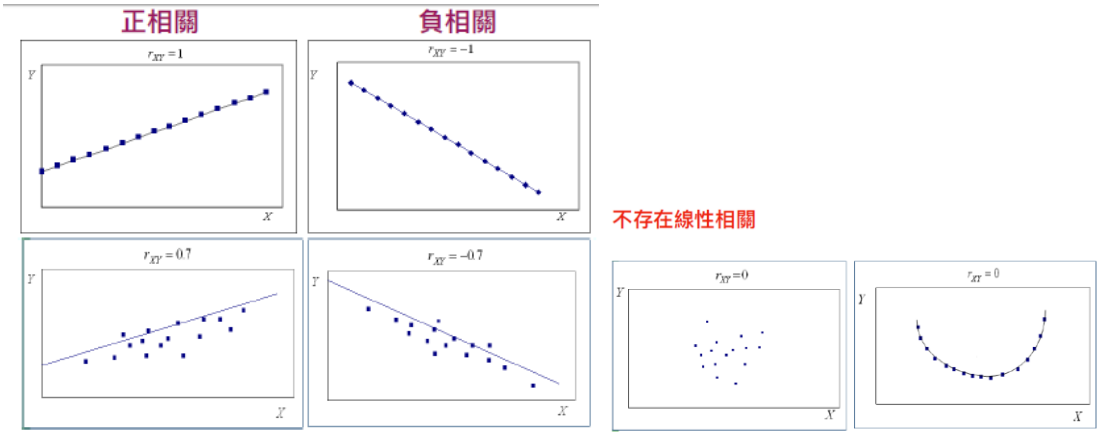
這個方程對應的影象是一條直線，稱作迴歸線。其中，β_0是迴歸線的截距，β_1是迴歸線的斜率。E(y)是在一個給定 x 值下 y 的期望值(均值)。

$$\theta_1 = \frac{n\sum x^{(i)}y^{(i)} - \sum x^{(i)}\sum y^{(i)}}{n\sum (x^{(i)})^2 - \left(\sum x^{(i)}\right)^2} = \frac{\sum \left(x^{(i)} - \bar{x}\right)\left(y^{(i)} - \bar{y}\right)}{\sum \left(x^{(i)} - \bar{x}\right)^2}$$

$$\theta_0 = \bar{y} - \theta_1\bar{x}$$

● 相關係數 (Correlation)

$$r_{xy} = \frac{S_{xy}}{S_x S_y} = \frac{\sum (x-\bar{x})(y-\bar{y})}{\sqrt{\sum (x-\bar{x})^2}\sqrt{\sum (x-\bar{x})^2}} \in [-1,1]$$



● 判定係數 (R 平方)

$$\underbrace{\sum_{i=1}^{n}(Y_i-\bar{Y})^2}_{\textbf{SST}} = \underbrace{\sum_{i=1}^{n}(\hat{Y}-\bar{Y})^2}_{\textbf{SSR}} + \underbrace{\sum_{i=1}^{n}(Y_i-\hat{Y}_i)^2}_{\textbf{SSE}}$$

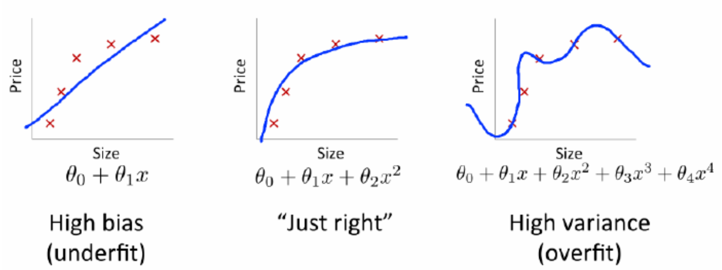
$$R^2 = \frac{\mathrm{SSR}}{\mathrm{SST}} = \frac{\sum(\hat{Y}_i-\bar{Y})^2}{\sum(Y_i-\bar{Y})^2}$$

SST 為資料總變異，SSR 為回歸模型解釋的變異，SSE 為模型無法解釋的變異數。R^2 (判定係數) 為回歸模型能解釋的整體變異比例。

4. 多重線性迴歸 (Multiple linear regression，MLR)

研究一個因變數與多個自變數之間的數量依存關係。y_hat=b0＋b１x1+b2x2+ ... +bpxp 一個樣本被用來計算β0，β１，β2... βp 的點估計 b0, b1, b2,..., bp。

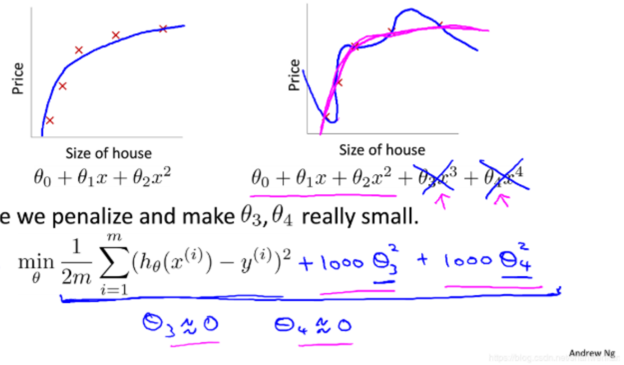
● Overfitting:



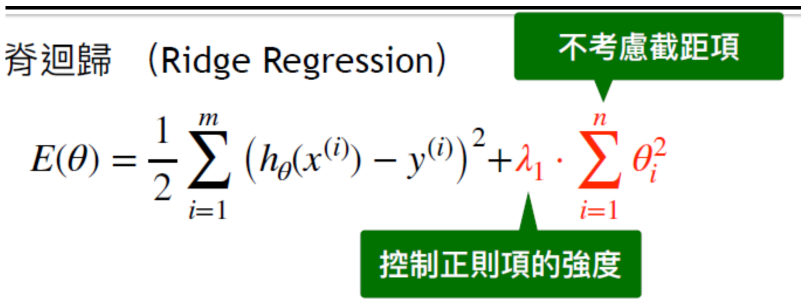
如果過多因子可能會有 overfitting 的問題。

- 正規化 (Regularization)

我們在利用資料來進行曲線擬合的時候會出現三種情況，欠擬合(underfitting)，合適（just right)，過擬合（overfitting)。欠擬合的情況一般是由於變數太少，而過擬合的原因一般是變數太多。下面我們主要考慮過擬合的問題。過擬合的解決方法一種是減少特徵的數量，一種就是正則化。正規化採用的方法就是修改代價函式，將其加上我們認為不那麼重要的項，例如下面這個例子我們加上θ3和θ4，我們可以知道這樣在優化的時候θ3和θ4會很小，這樣這兩項對函式的影響就很小，相當於這兩項消失了，也就相當於減少了特徵的數量，解決了過擬合的問題。



5. Ridge Regression



複雜度越低的模型在訓練集上的表現越差，但泛化的能力會更好。如果我們更在意模型在泛化方面的能力，應該選擇 Ridge 而非線性迴歸。

增加α後，模型的分數大幅降低，然而 test 分數 > train 分數，若模型出現 overfitting，可以透過提高α值來降低 overfittng 的程度。

非常小的α值，會使結果很接近線性迴歸；增加α會降低係數，使其趨近於 0，降低訓練集的分數，但有助於泛化。

	Linear Regression	Ridge $\alpha = 1$	Ridge $\alpha = 10$	Ridge $\alpha = 0.1$
R^2 值 Train	0.530	0.433	0.151	0.522
R^2 值 Test	0.459	0.433	0.162	0.473

overfitting

6. LASSO

最小絕對壓縮挑選機制 （Least Absolute Shrinkage and Selection Operator, LASSO）

$$E(\theta) = \frac{1}{2}\sum_{i=1}^{m}\left(h_\theta(x^{(i)}) - y^{(i)}\right)^2 + \lambda_2 \cdot \sum_{i=1}^{n}|\theta_i|$$

使某些係數變為0

Lasso 的得分都很低，模型只使用 3 個特徵。降低 後，模型的分數大幅增加，模型較複雜 （7 個特徵）。相對 Ridge, Lasso 表現稍好，且只用 7 個特徵，使模型較好理解。非常小的 α 值，使用了全部特徵，會使結果很接近線性迴歸。

	Linear Regression	Lasso $\alpha = 1$	Lasso $\alpha = 0.1$	Lasso $\alpha = 0.001$
R^2 值 Train	0.530	0.362	0.519	0.530
R^2 值 Test	0.459	0.366	0.480	0.460

overfitting

7. Ridge v.s. Lasso

- 實作時，Ridge 通常是首選
- 但如果特徵太多，且只有一小部分是真正重要的，那應該選擇 Lasso
- 如果須解釋模型，Lasso 也更更好理理解，因為使用較少特徵

8. 回歸優缺點

Pros:

- 簡單，直覺，易於運算
- 迴歸係數能得到有用的訊息

Cons:

- 易受異異常值影響
- 相關預測因子的權重會被扭曲
- 曲線趨勢

參考資料

1. 機器學習 代價函數（cost function）
https://www.itread01.com/articles/1491123443.html
2. 機器學習 迴歸分析 (regression analysis)
https://www.itread01.com/content/1546306589.html
3. 機器/深度學習-基礎數學(二):梯度下降法(gradient descent)
https://medium.com/@chih.sheng.huang821/%E6%A9%9F%E5%99%A8%E5%AD%B8%E7%BF%92-%E5%9F%BA%E7%A4%8E%E6%95%B8%E5%AD%B8-%E4%BA%8C-%E6%A2%AF%E5%BA%A6%E4%B8%8B%E9%99%8D%E6%B3%95-gradient-descent-406e1fd001f
4. 機器學習：特徵標準化！
https://ithelp.ithome.com.tw/articles/10197357?sc=iThelpR
5. Class Handout, Lee, Chia Jung professor, MDM64001, School of Big Data Management, Soochow University