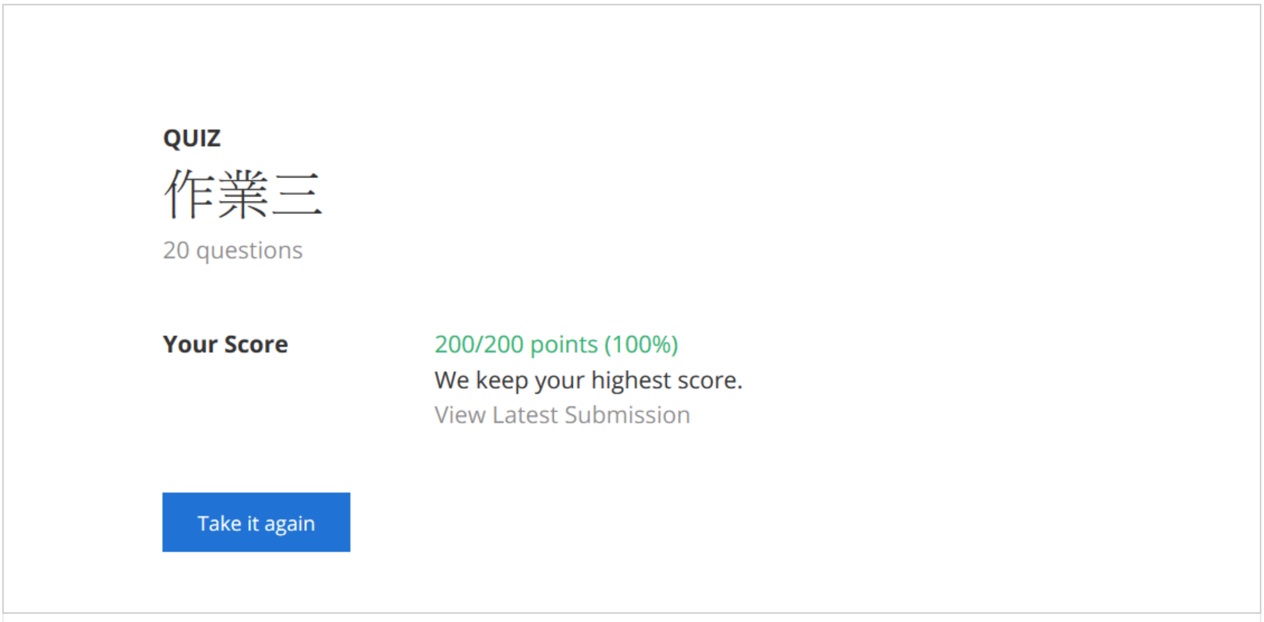
ML Foundation: HW3

b04902053 鄭淵仁

December 31, 2017

1



2

Claim: $H^2 = H$.
Proof:

$$\begin{aligned}
H^2 &= (X(X^TX)^{-1}X^T)^2 \\
&= (X(X^TX)^{-1}X^T)(X(X^TX)^{-1}X^T) \\
&= X(X^TX)^{-1}[(X^TX)(X^TX)^{-1}]X^T \\
&= X(X^TX)^{-1}X^T \\
&= H
\end{aligned}$$

With the claim above, we can prove that:

$$\begin{aligned}
(I-H)^2 &= I^2 - 2IH + H^2 \\
&= I - 2H + H \\
&= I - H
\end{aligned}$$

3

TODO: 看不懂題目在寫什麼？QQ

4

$$\hat{E}_2(\Delta u, \Delta v) = E(u,v) + \nabla E(u,v) \cdot (\Delta u, \Delta v) + \frac{1}{2}(\Delta u, \Delta v)^T \nabla^2 E(u,v)(\Delta u, \Delta v)$$

Set the partial differences of $\hat{E}_2(\Delta u, \Delta v)$ be 0, we have :

$$\begin{cases} 0 = \dfrac{\partial \hat{E}_2(\Delta u, \Delta v)}{\partial \Delta u} = \dfrac{\partial E}{\partial u} + \dfrac{1}{2}\left(2\dfrac{\partial^2 E}{\partial u^2}\Delta u + 2\dfrac{\partial^2 E}{\partial u \partial v}\Delta v\right) \\ \qquad\qquad\qquad\qquad = \dfrac{\partial E}{\partial u} + \dfrac{\partial^2 E}{\partial u^2}\Delta u + \dfrac{\partial^2 E}{\partial u \partial v}\Delta v \\ 0 = \dfrac{\partial \hat{E}_2(\Delta u, \Delta v)}{\partial \Delta v} = \dfrac{\partial E}{\partial v} + \dfrac{\partial^2 E}{\partial v^2}\Delta v + \dfrac{\partial^2 E}{\partial v \partial u}\Delta u \end{cases}$$

Simplify the equations :

$$\begin{cases} 0 = \dfrac{\partial E}{\partial u} + \dfrac{\partial^2 E}{\partial u^2}\Delta u + \dfrac{\partial^2 E}{\partial u \partial v}\Delta v \\ 0 = \dfrac{\partial E}{\partial v} + \dfrac{\partial^2 E}{\partial v^2}\Delta v + \dfrac{\partial^2 E}{\partial v \partial u}\Delta u \end{cases}$$

Now combine the two equations to one equation by vector (u, v) :

$$\begin{aligned} 0 &= \nabla E(u,v) + \nabla^2 E(u,v) \cdot (\Delta u, \Delta v) \\ -\nabla^2 E(u,v) \cdot (\Delta u, \Delta v) &= \nabla E(u,v) \\ (\Delta u, \Delta v) &= -\left(\nabla^2 E(u,v)\right)^{-1} \nabla E(u,v) \end{aligned}$$

Q.E.D.

5

$$\max_h \prod_{n=1}^{N} h_y(x_n) = \max_w \prod_{n=1}^{N} \frac{\exp\left(w_{y_n}^T x_n\right)}{\sum_{k=1}^{K} \exp(w_k^T x_n)}$$

Take natural log on it :

$$\begin{aligned} &\max_w \ln \prod_{n=1}^{N} \frac{\exp\left(w_{y_n}^T x_n\right)}{\sum_{k=1}^{K} \exp(w_k^T x_n)} \\ &= \max_w \sum_{n=1}^{N} \ln \frac{\exp\left(w_{y_n}^T x_n\right)}{\sum_{k=1}^{K} \exp(w_k^T x_n)} \\ &= \max_w \sum_{n=1}^{N} \left(\ln\left(\exp\left(w_{y_n}^T x_n\right)\right) - \ln \sum_{k=1}^{K} \exp\left(w_k^T x_n\right) \right) \\ &= \min_w \sum_{n=1}^{N} \left(\ln \sum_{k=1}^{K} \exp\left(w_k^T x_n\right) - w_{y_n}^T x_n \right) \end{aligned}$$

Therefore the E_{in} is :

$$E_{in} = \frac{1}{N} \sum_{n=1}^{N} \left(\ln \sum_{k=1}^{K} \exp\left(w_k^T x_n\right) - w_{y_n}^T x_n \right)$$

6

First compute :

$$\frac{\partial\left(\sum_{n=1}^{N}\left(\ln\sum_{k=1}^{K}\exp\left(w_k^T x_n\right)\right)\right)}{\partial w_i}$$
$$=\sum_{n=1}^{N}\left(\frac{\exp\left(w_i^T x_n\right)}{\sum_{k=1}^{K}\exp(w_k^T x_n)}x_n\right)$$
$$=\sum_{n=1}^{N}\left(h_i(x_n)x_n\right)$$

Therefore the answer is :

$$\frac{\partial E_{in}}{\partial w_i}=\frac{\partial\left(\frac{1}{N}\sum_{n=1}^{N}\left(\ln\sum_{k=1}^{K}\exp\left(w_k^T x_n\right)-w_{y_n}^T x_n\right)\right)}{\partial w_i}$$
$$=\frac{1}{N}\sum_{n=1}^{N}\left((h_i(x_n)x_n)-[[y_n=i]]x_n\right)$$
$$=\frac{1}{N}\sum_{n=1}^{N}\left(((h_i(x_n))-[[y_n=i]])\,x_n\right)$$

7

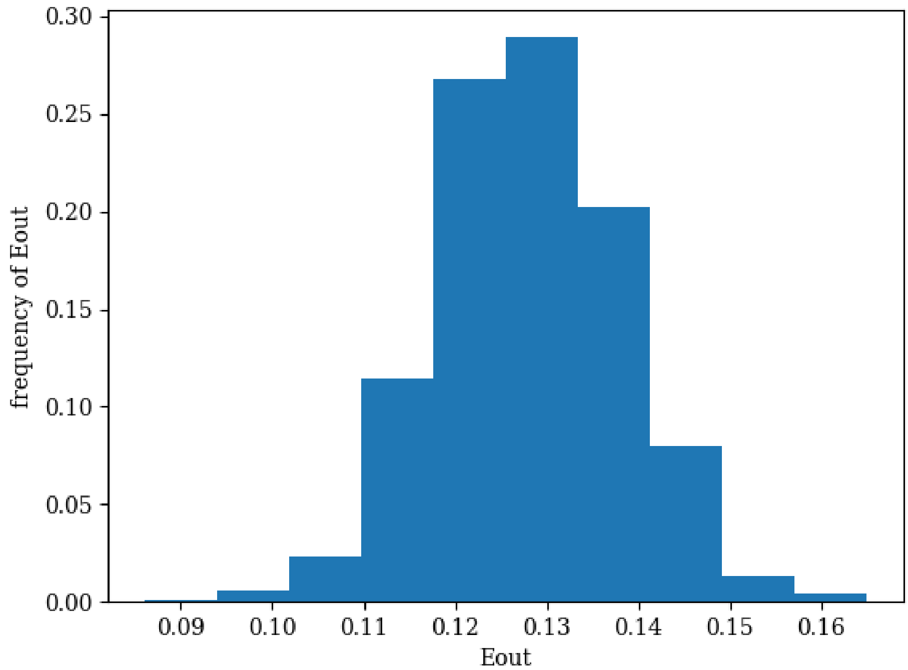
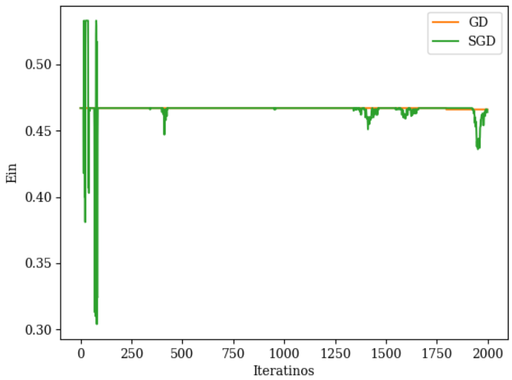


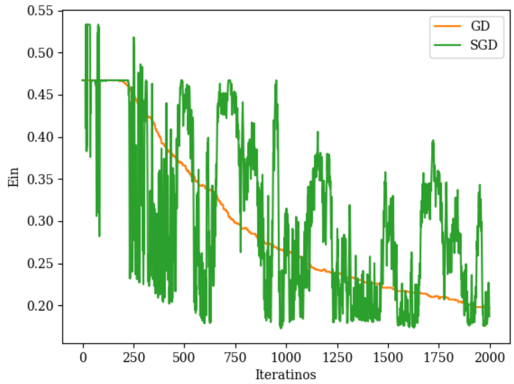
Figure 1: Histogram of E_{out}

Figure 1 shows the histogram of E_{out}.

8



(a) $lr = 0.001$



(b) $lr = 0.01$

Figure 2: Comparison between GD and SGD in E_{in}.

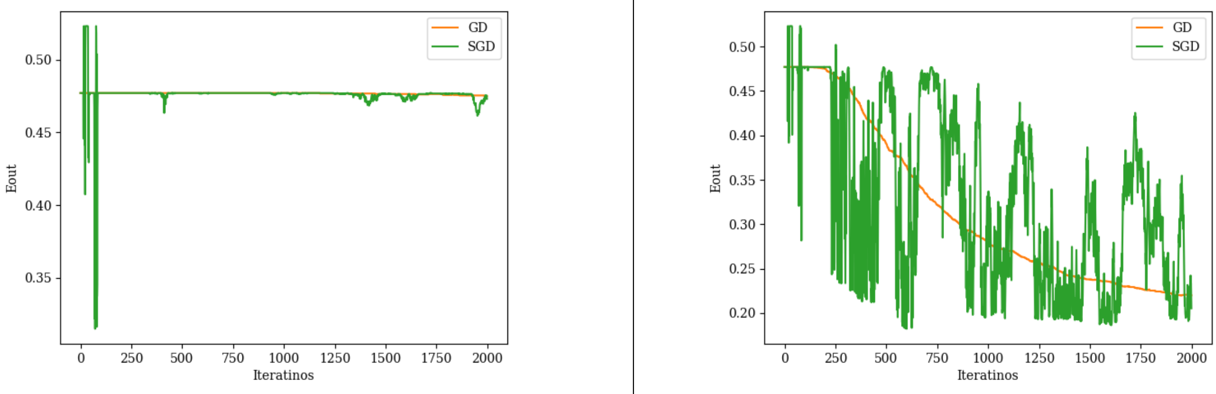
從上面兩張圖中，我發現有以下三點現象：

1. GD 和 SGD 的差異：
   我發現 GD 的 E_{in} 很快就會穩定不變，或是穩定下降，而不會上下亂跳；相較之下，SGD 的 E_{in} 則是很容易上下浮動。
   我想這是因為 SGD 一次只會取一筆資料來計算 gradient ，如果這一筆資料有 noise 的話，算出來的 gradient 很容易會被這個 noise 影響；相較之下，GD 一次會用所有資料來計算 gradient ，所以算出來的結果一定會讓所有 training data 的 E_{in} 變小或幾乎不變。

2. $lr = 0.001$ 和 $lr = 0.01$ 的差異：
   我發現無論是 GD 或 SGD ，$lr = 0.001$ 的時候，E_{in} 除了一開始上下亂跳以外，接下來就幾乎固定在 0.46 左右了；而 $lr = 0.01$ 的時候，E_{in} 會一直下降到 0.21 。
   我想這是因為 0.001 的 learning rate 太小了，下降的速度太慢，甚至很容易卡在局部極值出不去；而 0.01 的 learning rate 則是比較洽當的值，所以 E_{in} 才能一直下降。

3. 我發現到最後 GD 和 SGD 收斂到很接近的值。
   我想這是因為以期望值而言，SGD 算出來的 gradient 和 GD 的 gradient 會是一樣的，但是 SGD 會有 noise ，所以一開始 GD 和 SGD 的結果會差很多。但是夠多的 iteration 之後，SGD 的 E_{in} 會因為跑過所有的點很多次，而且有相同的 learning rate，所以有比較高的機率可以到相同的極值。

9



(a) $lr = 0.001$

(b) $lr = 0.01$

Figure 3: Comparison between GD and SGD in E_{out}.

從上面兩張圖中，我發現有以下 5 點現象：

1. E_{out} 的結果和 E_{in} 的結果很像。
   我想這是因為 E_{out} 和 E_{in} 的 noise 沒有太多，而且取的資料量又都相對夠多，所以 E_{out} 的結果才會和 E_{in} 很像。

2. GD 和 SGD 的差異：
   我發現 GD 的 E_{in} 很快就會穩定不變，或是穩定下降，而不會上下亂跳；相較之下，SGD 的 E_{in} 則是很容易上下浮動。
   我想這個原因跟 E_{in} 的這個現象的原因是有關係的：因為 SGD 一次只會取一筆資料來計算 gradient ，如果這一筆資料有 noise 的話，算出來的 gradient 很容易會被這個 noise 影響，所以 E_{out} 的值也會被影響；相較之下，GD 一次會用所有資料來計算 gradient ，所以算出來的結果比較穩定，不會只受單一資料的 noise 影響，所以 E_{out} 比較不會上下大幅變動。

3. $lr = 0.001$ 和 $lr = 0.01$ 的差異：
   我發現無論是 GD 或 SGD ，$lr = 0.001$ 的時候，E_{in} 除了一開始上下亂跳以外，接下來就幾乎固定在 0.47 左右了；而 $lr = 0.01$ 的時候，E_{in} 會一直下降到 0.22 。
   我想這個原因跟 E_{in} 的這個現象的原因是一樣的：因為 0.001 的 learning rate 太小了，下降的速度太慢，甚至很容易卡在局部極值出不去；而 0.01 的 learning rate 則是比較洽當的值，所以 E_{out} 才能一直下降。

4. 我發現到最後 GD 和 SGD 收斂到很接近的值。
   我想這個原因跟 E_{in} 的這個現象的原因是一樣的：因為以期望值而言，SGD 算出來的 gradient 和 GD 的 gradient 會是一樣的，但是 SGD 會有 noise ，所以一開始 GD 和 SGD 的結果會差很多。但是夠多的 iteration 之後，SGD 的 E_{in} 會因為跑過所有的點很多次，而且有相同的 learning rate，所以有比較高的機率可以到相同的極值。

5. 另外，從上一現象可以發現：在這次的 data 裡面，其實可以只使用速度較快、運算資源不用太多的 SGD 來做 training ，也可以得到和 GD 相近的效果。
   而如果給定相同的時間用 GD 和 SGD 來做的話，SGD 可能可以比 GD 更早下降到極值，或是 SGD 可能會有比 GD 更好的效果。