MAT 5314

Assignment 1

Soufiane Fadel

January 31, 2019

Exercise 1

> *Suppose $m < d < \infty$ then what can you conclude about the rank of the $d \times d$ matrix A and why does this imply that there are infinitely many w that minimize empirical risk?*

solution:

By **”Rank–nullity theorem”** and the result **Null(X^tX)=Null(X)**[Gram-matrix-note-course] we have: Rank$(X^tX) = d-$Nullity$(X^tX) = d-$Nullity$(X) \underset{\text{RNT on X}}{=} d-(d-\text{Rank}(X)) \underset{\text{Rank}(X)\leq m}{\leq} d-(d-m) < d$ then $A = X^tX \notin \mathbb{GL}_d(\mathbb{R}) \Rightarrow$ we have ∞ many sol (because $A\mathcal{Y} = b \perp \text{Null}(A)$ see ”Revisiting three cases”notes).

Exercise 2

a) part 1

> *With all the above assumptions, show that Recursive Least Squares will take time $\mathcal{O}(d^3)$ to find the optimal $w_* = w_{n+1}$ knowing the previous $w_* = w_n$ and matrices from that previous computation*

solution:

Let’s defined some notations that we will use on our pseudo-code :

$$\underset{(n+1\times d)}{x_{(n+1)}} = \begin{bmatrix} x_1^t \\ \vdots \\ x_{n+1}^t \end{bmatrix} = \begin{bmatrix} x_{(n)} \\ x_{n+1}^t \end{bmatrix} \qquad \text{and} \qquad y_{(n+1)} = \begin{bmatrix} y_1 \\ \vdots \\ y_{n+1} \end{bmatrix} = \begin{bmatrix} y_{(n)} \\ y_{n+1} \end{bmatrix}$$

we sow in class that at step $(n+1)$:

$$w_{n+1} = (x_{(n+1)}^t x_{(n+1)})^{-1} x_{(n+1)}^t y_{(n+1)}$$

On the other hand we have a rule for updating our predictor that we sow in class, it express the solution w_{n+1} in terms of things we computed before on step n, so we can show that :

$$w_{n+1} = w_n + P_{n+1} x_{n+1} \left(y_{n+1} - x_{n+1}^t w_n \right)$$

where

$$P_{n+1}^{-1} = P_n^{-1} - x_{n+1} x_{n+1}^t$$

to analyse this algorithm let sketch a pseudo-code for it :
In order to well analysing the complexity of this algorithm here is a table that summarising all the basic operations (additions and multiplications) :

Algorithm 1 Recursive Least Squares

```
Require: data matrix $\underset{(n\times d)}{X}$, labels $\underset{(n\times 1)}{Y}$
1: Initialize:
       $P_0 \leftarrow I_d$
2: for ($i \leftarrow 0$ to $n-1$) do                                                    ▷ Complexity : $\sum_{i=0}^{n-1}$
3:     $P_{i+1}^{-1} \leftarrow P_i^{-1} - x_{i+1}x_{i+1}^t$                              ▷ Complexity : $\mathcal{O}(d^2)$
4:     inverte $P_{i+1}^{-1}$ with Gaussien elimination algorithm to get $P_{i+1}$        ▷ Complexity : $\mathcal{O}(d^3)$
5:     compute $P_{i+1}x_{n+1}$                                                          ▷ Complexity : $\mathcal{O}(d^2)$
6:     compute $y_{i+1} - x_{i+1}^t w_i$                                                  ▷ Complexity : $\mathcal{O}(d)$
7:     $w_{i+1} \leftarrow w_i + P_{i+1}x_{i+1}\left(y_{i+1} - x_{i+1}^t w_i\right)$       ▷ Complexity : $\mathcal{O}(d)$
8: return $w_n$
```

	number of additions	**number of multiplications**	**Total**
iteration 3	d^2	d^2	$2d^2 = \mathcal{O}(d^2)$
iteration 4	***		$\mathcal{O}(d^3)$ (using gauss-elimination)
iteration 5	d^2	d^2	$2d^2 = \mathcal{O}(d^2)$
iteration 6	1 + d	d	$1 + 2d = \mathcal{O}(d)$
iteration 7	d	d	$2d = \mathcal{O}(d)$
			$\mathcal{O}(d^3)$

Note: Because of the iterations from 0 to $n-1$ (**Interation 2**) then the time-complexity of the Recursive Least Squares algorithm is : $\mathcal{O}(nd^3)$

Conclusion:

♣ The time-Complexity to find the optimal $w_* = w_{n+1}$ knowing the previous $w_* = w_n$ and matrices from that previous computation is : $\mathcal{O}(d^3)$

♣ The time-Complexity of the Recursive Least Squares algorithm is : $\mathcal{O}(nd^3)$

a) part 2

show that performing Ordinary Least Squares on the full data set $(x_i, y_i), i = 1, \cdots n+1$ *would take time* $\mathcal{O}((n+1)(d+d^2)+d^3) = \mathcal{O}(nd^2+d^3)$

solution:
An ERM algorithm is one that performs empirical risk minimization, i.e.

$$w_* = \text{argmin} \sum_{i=1}^{n} (y_i - wx_i)^2$$

we show in class that by differentiation we find that the gradient of the smooth function $J(w) = (Y - Xw)(Y - Xw)^t$ is zero iff the matrix equation :

$$(XX^t)w_* = X^tY$$

here is the pseudo-code for the Ordinary Least Squares :
As in the previous questions here is a table that summarising all the basic operations (additions and multiplications):

```
Algorithm 2 Ordinary Least Squares
Require: data matrix $\underset{(n+1\times d)}{X}$, labels $\underset{(n+1\times 1)}{Y}$
1: compute $X^tX$                                  ▷ Complexity : $\mathcal{O}(d^2(n+1))$
2: compute $X^tY$                                  ▷ Complexity : $\mathcal{O}(d(n+1))$
3: inverse $X^tX$ to get $(X^tX)^{-1}$             ▷ Complexity : $\mathcal{O}(d^3)$
4: compute $w_* \leftarrow (X^tX)^{-1}X^tY$        ▷ Complexity : $\mathcal{O}(d^2)$
5: return $w_*$
```

	number of additions	**number of multiplications**	**Total**
iteration 1	$d^2(n+1)$	$d^2(n+1)$	$2d^2(n+1) = \mathcal{O}(d^2(n+1))$
iteration 2	$d(n+1)$	$d(n+1)$	$2d(n+1) = \mathcal{O}(d(n+1))$
iteration 3	***		$\mathcal{O}(d^3)$ (using gauss-elimination)
iteration 4	d^2	d^2	$2d^2 = \mathcal{O}(d^2)$
			$\mathcal{O}((n+1)(d^2+d)+d^2+d^3)$ $= \mathcal{O}(nd^2+nd+d^2+d+d^2+d^3)$ $= \mathcal{O}(nd^2+d^3)$

Conclusion:

♣ The time-Complexity of the Ordinary Least Squares algorithm is : $\mathcal{O}(nd^2+d^3)$

b)

What situations does this suggest should be handled by Recursive Least Squares instead of Ordinary Least Squares? Justify your answer

solution:
In order to answer to this question we should compare the complexity of **Ordinary least square** and **Recursive least square** under a order condition between the size of the data "**n**" and the size of the features "**d**"

♣ **if:** $d > n$
in this case we have : $nd^2 + d^3 < 2d^3 < nd^3$ and then:

$$\mathcal{O}(nd^2+d^3) \subset \mathcal{O}(nd^3)$$

♣ **if:** $d < n$
in this case we have : $nd^2 + d^3 > 2d^3$ and then:

$$\mathcal{O}(2d^3) \subset \mathcal{O}(nd^2+d^3)$$

Or since n is small than d then :

$$\mathcal{O}(d^3) = \mathcal{O}(2d^3) = \mathcal{O}(nd^3) \subset \mathcal{O}(nd^2+d^3)$$

Conclusion:

♣ We can see from the general perspective by analysing the time complexity the it's preferable to use Recursive Least Squares instead of Ordinary Least Squares when we have more features than observations.

♣ On the other hand we can think on Memory issues, in fact when we are dealing with very large datasets, matrices can become huge and memory issues may arise when using the ordinary least square . for example in case when we are using the **data slices** the Recursive Least Squares may perform better the ordinary least square.

Annexe 1 :(Gaussian Elimination)

(ASIDE not for credit: think about why this runtime can be achieved using Gaussian Elimination)

To find the inverse of an $n \times n$ matrix A we augment $(A|I)$ and use Gauss-elimination. To show that here is the pseudo-code to Construct an algorithm using Gaussian elimination to find A^{-1}. But before that let's defined what's a augmented matrix:

$$B = \underset{(n\times 2n)}{(A|I_n)} = \left[\begin{array}{cccc|cccc} a_{1,1} & a_{1,2} & \cdots & a_{1,n} & 1 & 0 & \cdots & 0 \\ a_{2,1} & \cdots & \cdots & a_{2,n} & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & 0 & \ddots & 0 \\ a_{n,1} & \cdots & \cdots & a_{n,n} & 0 & 0 & \cdots & 1 \end{array}\right]$$

Here is the pseudo-code for the Inverting-Gauss-elimination :

Algorithm 3 Inverting-Gauss-elimination

```
Input: augmented matrix $B = (A|I_n)$
Output: The inverse of $A$ : $A^{-1}$
for $i \leftarrow 0$ to $n-1$ do        ▷ Search for maximum in this column with complexity $\sum_{i=0}^{n-1} \mathcal{O}(1)$
    max $\leftarrow |A_{i,i}|$
    argmaxRw $\leftarrow i$
    for ($k \leftarrow i+1$ to $n-1$) do        ▷ Complexity : $\sum_{k=i+1}^{n-1} \mathcal{O}(1)$
        if $|A_{k,i}| > max$ then
            max $\leftarrow A_{k,i}$
            argmaxRw $\leftarrow k$
    for ($k \leftarrow i$ to $2n$) do        ▷ Swap maximum row with current row with complexity : $\sum_{k=i}^{2n} \mathcal{O}(1)$
        tmp $\leftarrow A_{argmaxRw,k}$
        $A_{argmaxRw,k} \leftarrow A_{i,k}$
        $A_{i,k} \leftarrow tmp$
    for ($k \leftarrow i+1$ to $n$) do        ▷ Make all rows below this one 0 in current column with : $\sum_{k=i+1}^{n} \mathcal{O}(1)$
        c $\leftarrow -\frac{A_{k,i}}{A_{i,i}}$
        for ($k \leftarrow i$ to $2n$) do        ▷ Complexity : $\sum_{j=i}^{2n} \mathcal{O}(1)$
            if $i == j$ then
                $A_{k,i} \leftarrow 0$
            else
                $A_{k,i} \leftarrow A_{k,i} + c.A_{k,i}$
return: $[B_{i,j}]_{\substack{1\leq i\leq n \\ 1\leq j\leq 2n}}$
```

By summing all this Big-O's for all the loops we have :

$$
\begin{aligned}
run - time &= \mathcal{O}\left(\sum_{i=0}^{n-1}\left(\sum_{k=i+1}^{n-1} 1\right)\right) + \mathcal{O}\left(\sum_{i=0}^{n-1}\left(\sum_{k=i}^{2n} 1\right)\right) + \mathcal{O}\left(\sum_{i=0}^{n-1}\left(\sum_{k=i+1}^{n-1}\sum_{j=i}^{2n} 1\right)\right) \\
&= \mathcal{O}\left(\sum_{i=0}^{n-1}\left(\sum_{k=i+1}^{n-1}\sum_{j=i}^{2n} 1\right)\right) \\
&= \mathcal{O}\left(\sum_{i=0}^{n-1}\left(\sum_{k=i+1}^{n-1}(2n-i+1)\right)\right) \\
&= \mathcal{O}\left(\sum_{i=0}^{n-1}(n-i-1)(2n-i+1))\right)
\end{aligned}
$$

On the other hand we know that :

$$
\begin{aligned}
\sum_{i=1}^{n} i &= \frac{n(n+1)}{2} = \mathcal{O}(n^2) \\
\sum_{i=1}^{n} i^2 &= \frac{n(n+1)(2n+1)}{6} = \mathcal{O}(n^3)
\end{aligned}
$$

therefore:

$$
\sum_{i=0}^{n-1}(n-i-1)(2n-i+1) = \underbrace{\sum_{i=0}^{n-1}(n-1)(2n+1)}_{\mathcal{O}(n^3)} + \underbrace{\sum_{i=0}^{n-1} i^2}_{\mathcal{O}(n^3)} + \underbrace{\sum_{i=0}^{n-1} -i(2n+1)}_{\mathcal{O}(n^3)} + \underbrace{\sum_{i=0}^{n-1} -i(n-1)}_{\mathcal{O}(n^3)}
$$

Finally we have :

$$
\begin{aligned}
Run - time &= \mathcal{O}\left(\sum_{i=0}^{n-1}(n-i-1)(2n-i+1))\right) \\
&= \mathcal{O}(n^3)
\end{aligned}
$$

Exercise 3

a)

> *Consider a slightly more general assumptions where the Gaussians have standard deviation that may depend on the class (value of Y). Derive the form of $\mathbb{P}(Y|X)$ under this new more general version of GNB and say whether it still corresponds to logistic regresssion.*

solution:

In this question we will proceed as in section 3.1" Form of $\mathbb{P}(Y|X)$ for Gaussian Naive Bayes Classifier " (Chapter 3 of Mitchell). We will derive the form of $\mathbb{P}(Y|X)$ entailed by the following modeling assumptions:

- ♣ Y is boolean, governed by a Bernoulli distribution, with parameter $\pi = \mathbb{P}(Y = 1)$
- ♣ $X =< X_1, \cdots, X_n >$, where each X_i is a continuous random variable
- ♣ For each X_i , $\mathbb{P}(X_i|Y = k)$ is a Gaussian distribution of the form $\mathcal{N}(\mu_{i,k}, \sigma_{i,k})$ where, $i = 1 \cdots n$ and $k = 0, 1$.

♣ For all i and $j \neq i$, X_i and X_j are conditionally independent given Y

Fist let's express $\pi = \mathbb{P}(Y=1|X)$ according to $\pi = \mathbb{P}(X_i|Y=1)$ and π :

$$\begin{aligned}
\mathbb{P}(Y=1|X) &= \frac{\mathbb{P}(Y=1)\mathbb{P}(X|Y=1)}{\mathbb{P}(X)} \qquad (BayseRule) \\
&= \frac{\mathbb{P}(Y=1)\mathbb{P}(X|Y=1)}{\mathbb{P}(Y=1)\mathbb{P}(X|Y=1)+\mathbb{P}(Y=0)\mathbb{P}(X|Y=1)} \\
&= \frac{1}{1+\frac{\mathbb{P}(Y=0)\mathbb{P}(X|Y=0)}{\mathbb{P}(Y=1)\mathbb{P}(X|Y=1)}} \\
&= \frac{1}{1+\exp\ln\left(\frac{\mathbb{P}(Y=0)\mathbb{P}(X|Y=0)}{\mathbb{P}(Y=1)\mathbb{P}(X|Y=1)}\right)} \\
&= \frac{1}{1+\exp\left(\ln\frac{1-\pi}{\pi}+\ln\frac{\mathbb{P}(X|Y=0)}{\mathbb{P}(X|Y=1)}\right)} \\
&= \frac{1}{1+\exp\left(\ln\frac{1-\pi}{\pi}+\sum_{i=1}^n\ln\frac{\mathbb{P}(X_i|Y=0)}{\mathbb{P}(X_i|Y=1)}\right)}
\end{aligned}$$

On the other hand we have :

$$\begin{aligned}
\sum_{i=1}^n \ln\frac{\mathbb{P}(X_i|Y=0)}{\mathbb{P}(X_i|Y=1)} &= \sum_{i=1}^n \ln\frac{\frac{1}{\sqrt{2\pi\sigma_{i,0}^2}}\exp\left(\frac{-(X_i-\mu_{i,0})^2}{2\sigma_{i,0}^2}\right)}{\frac{1}{\sqrt{2\pi\sigma_{i,1}^2}}\exp\left(\frac{-(X_i-\mu_{i,1})^2}{2\sigma_{i,1}^2}\right)} \\
&= \sum_{i=1}^n \ln\frac{\sigma_{i,1}}{\sigma_{i,0}} + \sum_{i=1}^n\left(\frac{(X_i-\mu_{i,1})^2}{2\sigma_{i,1}^2}-\frac{(X_i-\mu_{i,0})^2}{2\sigma_{i,0}^2}\right) \\
&= \sum_{i=1}^n \ln\frac{\sigma_{i,1}}{\sigma_{i,0}} + \sum_{i=1}^n \frac{(\sigma_{i,0}^2-\sigma_{i,0}^2)X_i^2+2(\mu_{i,0}\sigma_{i,1}^2-\mu_{i,1}\sigma_{i,0}^2)X_i+\mu_{i,1}^2\sigma_{i,0}^2-\mu_{i,0}^2\sigma_{i,1}^2}{2\sigma_{i,0}^2\sigma_{i,1}^2} \\
&= \sum_{i=1}^n\left(\ln\frac{\sigma_{i,1}}{\sigma_{i,0}}+\frac{\mu_{i,1}^2\sigma_{i,0}^2-\mu_{i,0}^2\sigma_{i,1}^2}{2\sigma_{i,0}^2\sigma_{i,1}^2}\right)+\sum_{i=1}^n\frac{\mu_{i,0}\sigma_{i,1}^2-\mu_{i,1}\sigma_{i,0}^2}{\sigma_{i,0}^2\sigma_{i,1}^2}X_i+\sum_{i=1}^n\frac{\sigma_{i,0}^2-\sigma_{i,1}^2}{2\sigma_{i,0}^2\sigma_{i,1}^2}X_i^2
\end{aligned}$$

Finnaly the form of $\pi = \mathbb{P}(Y=1|X)$ is given by :

$$\begin{aligned}
\mathbb{P}(Y=1|X) &= \frac{1}{1+\exp\left(\ln\frac{1-\pi}{\pi}+\sum_{i=1}^n\ln\frac{\mathbb{P}(X_i|Y=0)}{\mathbb{P}(X_i|Y=1)}\right)} \\
&= \frac{1}{1+\exp\left(w_0+\sum_{i=1}^n w_iX_i+\sum_{i=1}^n v_iX_i^2\right)}
\end{aligned}$$

where :

$$\begin{aligned}
w_0 &= \ln\frac{1-\pi}{\pi}+\sum_{i=1}^n\left(\ln\frac{\sigma_{i,1}}{\sigma_{i,0}}+\frac{\mu_{i,1}^2\sigma_{i,0}^2-\mu_{i,0}^2\sigma_{i,1}^2}{2\sigma_{i,0}^2\sigma_{i,1}^2}\right) \\
w_i &= \frac{\mu_{i,0}\sigma_{i,1}^2-\mu_{i,1}\sigma_{i,0}^2}{\sigma_{i,0}^2\sigma_{i,1}^2} \\
v_i &= \frac{\sigma_{i,0}^2-\sigma_{i,1}^2}{2\sigma_{i,0}^2\sigma_{i,1}^2}
\end{aligned}$$

Conclusion:

- ♣ when standard deviation may depend on the class and then $\sigma_{i,1} \neq \sigma_{i,0}$ so we have a quadratic terme. Therefore the form $\mathbb{P}(Y = 1|X)$ it's differente from the form of logistic regression.

b)

Finally consider a less naive algorithm where conditional independence of the random variables in $X =< X_1, \cdots, X_n >$ is not assumed - specifically suppose $\mathbb{P}(X|Y = k)$ is multivariate normal with mean depending on k, but covariance matrix now NOT depending on k. Derive $\mathbb{P}(X|Y)$ and say if it has the same form as in logistic regression.

solution:
In this question we will proceed as in the previews question. We will derive the form of $\mathbb{P}(Y|X)$ entailed by the following modeling assumptions:

- ♣ Y is boolean, governed by a Bernoulli distribution, with parameter $\pi = \mathbb{P}(Y = 1)$
- ♣ $X =< X_1, \cdots, X_n >$, where each X_i is a continuous random variable
- ♣ $X =< X_1, \cdots, X_n >$ are not conditionally independent given Y, and $\mathbb{P}(X|Y = k)$ is a multivariate normal distribution $\mathcal{N}_n(\mu_k, \Sigma)$ where, μ_k is the mean vector that depending on k and Σ is the covariance matrix not depending on k.

Before starting expressing $\mathbb{P}(Y|X)$ let's remember the density of the multivariate normal distribution $\mathcal{N}_n(\mu, \Sigma)$:

$$f(x) = f(x_1, \cdots x_n) = \frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma|^{1/2}} \exp^{\frac{-(x-\mu)^t \Sigma^{-1} (x-\mu)}{2}} \qquad \forall x \in \mathbb{R}^n$$

Now we have all the tools to start the calculation:

$$\begin{aligned}
\mathbb{P}(Y = 1|X) &= \frac{\mathbb{P}(Y = 1)\mathbb{P}(X|Y = 1)}{\mathbb{P}(X)} \qquad (BayseRule) \\
&= \frac{\mathbb{P}(Y = 1)\mathbb{P}(X|Y = 1)}{\mathbb{P}(Y = 1)\mathbb{P}(X|Y = 1) + \mathbb{P}(Y = 0)\mathbb{P}(X|Y = 1)} \\
&= \frac{1}{1 + \frac{\mathbb{P}(Y=0)\mathbb{P}(X|Y=0)}{\mathbb{P}(Y=1)\mathbb{P}(X|Y=1)}} \\
&= \frac{1}{1 + \exp \ln \left(\frac{\mathbb{P}(Y=0)\mathbb{P}(X|Y=0)}{\mathbb{P}(Y=1)\mathbb{P}(X|Y=1)} \right)} \\
&= \frac{1}{1 + \exp \left(\ln \frac{1-\pi}{\pi} + \ln \frac{\mathbb{P}(X|Y=0)}{\mathbb{P}(X|Y=1)} \right)}
\end{aligned}$$

On the other hand we have :

$$\begin{aligned}
\ln \frac{\mathbb{P}(X|Y = 0)}{\mathbb{P}(X|Y = 1)} &= \ln \frac{\frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma|^{1/2}}}{\frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma|^{1/2}}} + \ln \exp \left(\frac{1}{2} \left[(x - \mu_1)^t \Sigma^{-1} (x - \mu_1) - (x - \mu_0)^t \Sigma^{-1} (x - \mu_0) \right] \right) \\
&= \ln \exp \left((\mu_0^t - \mu_1^t) \Sigma^{-1} x + \frac{1}{2} \mu_1^t \Sigma^{-1} \mu_1 - \frac{1}{2} \mu_0^t \Sigma^{-1} \mu_0 \right)
\end{aligned}$$

Finally we got :

$$\begin{aligned}
\mathbb{P}(Y = 1|X) &= \frac{1}{1 + \exp \left(\ln \frac{1-\pi}{\pi} + \frac{1}{2} \mu_1^t \Sigma^{-1} \mu_1 - \frac{1}{2} \mu_0^t \Sigma^{-1} \mu_0 + (\mu_0^t - \mu_1^t) \Sigma^{-1} x \right)} \\
&= \frac{1}{1 + \exp (w_0 + w^t x)}
\end{aligned}$$

where :

$$\begin{aligned} w_0 &= \ln \frac{1-\pi}{\pi} + \frac{1}{2}\mu_1^t \Sigma^{-1} \mu_1 - \frac{1}{2}\mu_0^t \Sigma^{-1} \mu_0 \\ w &= \left((\mu_0^t - \mu_1^t)\Sigma^{-1}\right)^t = \Sigma^{-1}(\mu_0 - \mu_1) \qquad \text{because the covariance matrix is symmetric} \end{aligned}$$

Conclusion:

♣ When conditional independence of the random variables in $X =< X_1, \cdots, X_n >$ is multivariate normal with mean depending on Y, but covariance matrix now NOT depending on Y the $\mathbb{P}(X|Y)$ have the form of logistic regression