406_a2

January 28, 2018

1 Exercise 1

1.0.1 a)

$$A = \begin{bmatrix} 1 & x_1^1 & x_2^1 \\ 1 & x_1^2 & x_2^2 \\ & \vdots & \\ 1 & x_1^n & x_2^n \end{bmatrix}, x = \begin{bmatrix} c_0 \\ c_1 \\ c_2 \end{bmatrix}, b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

1.0.2 b)

```
In [4]: data = CSV.read("C:\\Users\\Kyle\\Desktop\\lsq_classification.csv"; datarow=1)
        mtx = convert(Array, data)
        n = size(mtx, 1)
        c = mtx[:,1:2]
        A = hcat(ones(n,1), mtx[:,1:2])
        betas = mtx[:,3:3]
```

```
Out[4]: 115×1 Array{Float64,2}:
          1.0
          1.0
          1.0
          1.0
          1.0
          1.0
          1.0
          1.0
          1.0
          1.0
          1.0
          1.0
          1.0
          ⋮
         -1.0
         -1.0
         -1.0
         -1.0
         -1.0
```

```
 -1.0
 -1.0
 -1.0
 -1.0
 -1.0
 -1.0
 -1.0
```

i)

```
In [5]: # Solution via normal equation A^TAx = A^Tb
        xls = \(*(transpose(A),A), *(transpose(A),betas))

Out[5]: 3×1 Array{Float64,2}:
         -0.368611
          0.576988
          0.63353
```

ii)

```
In [6]: # Solution via QR
        QR = qrfact(A)
        Q = QR[:Q]
        R = QR[:R]
        m = size(R,1)

        y = *(transpose(Q), betas)
        y = y[1:m]
        xqr = \(R, y)

Out[6]: 3-element Array{Float64,1}:
         -0.368611
          0.576988
          0.63353
```

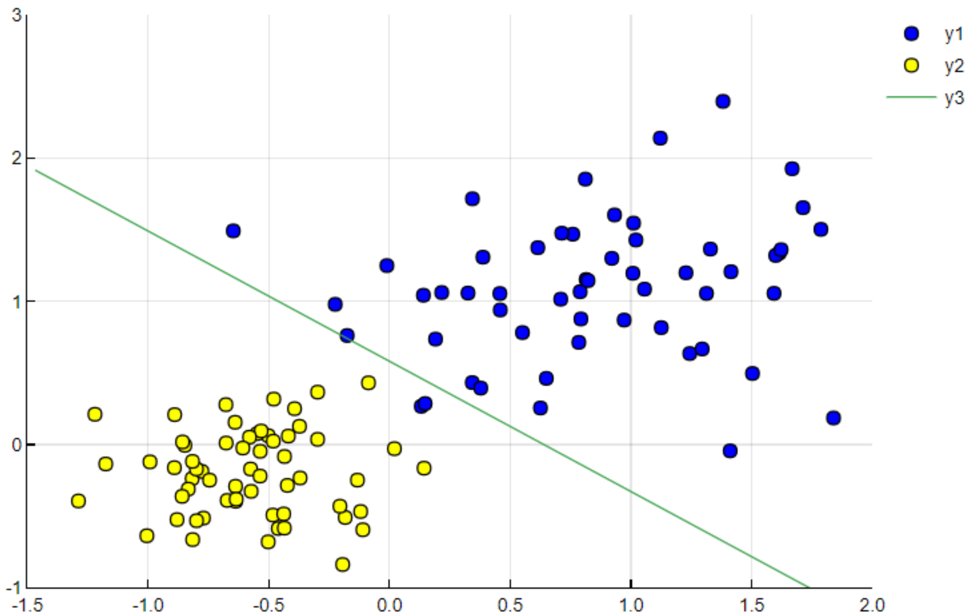
1.0.3 c)

```
In [7]: # define coefficients and function
        c0, c1, c2 = reshape(xls[1:1],1)[1], reshape(xls[2:2],1)[1], reshape(xls[3:3],1)[1]
        f(x1) = (-c0 -c1*x1) / c2

        # # # (x1,x2) where label = 1
        l11 = mtx[:,1:1][mtx[:,3:3] .== 1]
        l12 = mtx[:,2:2][mtx[:,3:3] .== 1]

        # # # (x1,x2) where label = -1
        l21 = mtx[:,1:1][mtx[:,3:3] .== -1]
        l22 = mtx[:,2:2][mtx[:,3:3] .== -1]
```

```
# # # plot data
scatter(l11, l12, xlim=(-1.5,2), ylim=(-1, 3), colour="blue")
scatter!(l21, l22, colour="yellow")
plot!(f)
```



2 Exercise 2

`1-norm`

$f(\alpha) = ||\alpha e - b|| = \sum_{i=1}^{n} |\alpha - b_i|$
This shows that by minimizing f(a) we are really just minimizing the sum of the distances between a and the various values of b. To do this we should let a = median(b)

`2-norm`

$\min\ f(\alpha) = ||\alpha e - b|| \Rightarrow \min\ f(\alpha)^2 = \min\ \sum_{i=1}^{n} (\alpha - b_i)^2$
$\Rightarrow 2\alpha n - 2\sum_{i=1}^{n} b_i$
$\Rightarrow \alpha n = \sum_{i=1}^{n} b_i$
$\Rightarrow \alpha = \frac{1}{n}\sum_{i=1}^{n} b_i$
$\Rightarrow \alpha = mean(b)$

`inf-norm`

$\min_\alpha f(\alpha) = min_\alpha\{max_i\{|\alpha - b_i|\}\} = min_\alpha\{max_i\{\alpha - b_i, b_i - \alpha\}\}$
To maximize $\alpha - b_i$ we would pick the smallest b_i. To maximize $b_i - \alpha$ we would pick the largest b_i.Therefore, to minimize both components we should let $\alpha = \frac{max b_i - min b_i}{2}$

3 Exercise 3

$f(x) = \frac{1}{2}(Ax-b)^T W(Ax-B) = \frac{1}{2}(Ax-b)^T QDQ(Ax-b)$
$= \frac{1}{2}(Ax-b)^T Q\sqrt{D}(\sqrt{D})^T Q(Ax-b)$
$Let\ Q\sqrt{D} = P$
$\Rightarrow \frac{1}{2}(Ax-b)^T PP^T(Ax-b)$
$= \frac{1}{2}(P^T Ax - P^T b)^T (P^T Ax - P^T b)$
$= \frac{1}{2}(Bx-d)^T(Bx-d)$

4 Exercise 4

a) Since A is underdetermined and consistent, should pick x such that

$$minimize\ ||x||^2$$
$$subject\ to : Ax = b$$

$let\ L(x,\mu) = ||x||^2 - \mu^T(b - Ax)$
$\frac{\partial}{\partial x} = 2x - u^T A$

$\Rightarrow x = \frac{1}{2}A^T\mu$
$\frac{\partial}{\partial\mu} = b - Ax$
$\Rightarrow b = Ax$
$\Rightarrow b = A(\frac{1}{2}A^T\mu)$
$\Rightarrow \mu = 2(AA^T)^{-1}b \Rightarrow x = A^T(AA^T)^{-1}b$
$Notice\ that\ A^T\ is\ overdetermined\ so\ let\ A^T = QR$
$\Rightarrow x = QR(R^TQ^TQR)^{-1}b = QR^{-T}b$
$We\ know\ the\ norm\ is\ invariant\ to\ orthogonal\ transformations\ therefore:$
$||x_{ls}|| = ||R^{-T}b||$

b)

```
In [11]: m = 5
         n = 10
         A = randn(5,10)
         b = randn(5)
         At = A'
         QR =qrfact(At)
         Q = QR[:Q]
         R = QR[:R]
         xls = *(*(Q,inv(R')),b)
```

```
Out[11]: 10-element Array{Float64,1}:
           1.01408
           0.3956
           0.187194
           0.598118
          -0.370701
```

```
-0.00259992
-0.391118
 0.245214
 0.801711
-0.0776825
```

5 Exercise 5

Proof: RLS has a unqiue solution, then null(A)^null(L) = {0}.

$f(x) = x^T A^T Ax - 2x^T A^T b + b^T b + \lambda x^T L^T Lx$
$\nabla f(x) = 2A^T Ax - 2A^T b + 2\lambda L^T Lx$
$\Rightarrow A^T Ax + \lambda L^T Lx = A^T b$
$\Rightarrow (A^T A + \lambda L^T L)x = A^T b$
We know that $(A^T A + \lambda L^T L)^{-1}$ must exist since RLS has a unique solution. So 0 is not an eigenvalue
$\Rightarrow y^T(A^T A + \lambda L^T L)y \Rightarrow y^T A^T Ay + \lambda y^T L^T Ly \Rightarrow ||Ay||^2 + \lambda||Ly||^2 > 0$
Since there is not y s.t the above equation is 0, it must be the case that the $null(A) \wedge null(L) = \{0\}$

Proof: $null(A) \wedge null(L) = \{0\}$ then RLS has a unqiue solution. Proof by contradiciton, suppose $null(A) \wedge null(L) = \{0\}$ and RLS does not have a unique solution

$Let\ x, y \in \mathbb{R}^n, x \neq y\ and\ \phi(x) = \phi(y) = c_i = the\ min\ to\ the\ RLS$
$Consider\ 3\ cases:$
$1)\ x, y \in null(A) \wedge null(L)$
$\Rightarrow since\ null(A) \wedge null(L) = \{0\}, it\ must\ be\ that\ x = y = 0$
$But\ we\ assumed\ x \neq y\ so\ this\ cannot\ be\ the\ case$
$2)\ x, y \in null(A) \wedge x, y \notin null(L)$
$f(x) = ||b||^2 + \lambda||Lx||^2$
$\nabla f(x) = 2\lambda L^T Lx \Rightarrow L^T Lx = 0 \rightarrow x \in null(L^T L) \rightarrow x \in null(L)$
$But,\ it\ was\ assumed\ that\ x \notin null(L)\ so\ this\ case\ cannot\ be\ possible,\ a\ similar\ argument\ can\ be\ made\ for\ y$
$3)\ \ x, y \in null(L) \wedge x, y \notin null(A)$
$f(x) = ||Ax - b||^2$
$\nabla f(x) = 2A^T Ax - 2A^T b \rightarrow A^T Ax = A^T b$

6 Exercise 6

```
In [14]: x1 = [0,0.5,1,1,0]
         x2 = [0,0,0,1,1]
         function circle_fit(A)
             A = A'
             new_A = 2*A[:,1:1]
             new_b = A[:,1:1].^2
             for i in 2:size(A,2)
                 new_A = hcat(new_A, 2*A[:,i:i])
                 new_b = new_b + A[:,i:i].^2
             end
             new_A = hcat(new_A, ones(size(A,1)))
```

```
    xls = *(inv(*(new_A',new_A)),*(new_A',new_b))
    return xls[1:size(xls,1)-1], xls[size(xls,1)]
end

x, r = circle_fit(vcat(x1',x2'))
```

Out[14]:
```
([0.5, 0.541667], -0.08333333333333304)
```