Written Exercises

1.  
Maximize: aT B a

Subject to: aT W a = 1

Apply Lagrange Multiplier:  
l(λ) = aT B a – λ (aT W a − 1)

∂l/∂a = (B + BT) a – λ (W + WT) a = 0

= (B + BT) a = λ (W + WT) a

= (W + WT)-1 (B + BT) a = λ a  
because matrices B and W are symmetric, we reduce this to  
W-1 B a = λ a

This becomes a standard eigenvalue problem T(a) = λ a where we apply transformation of (W-1 B) over a.

2.

(a)

δk(x) = xT Σ −1µk – ½ µTk Σ −1µk + log πk

When δ2(x) > δ1(x), LDA rule classifies to class 2 if  
xT Σˆ−1 (ˆµ2 − µˆ1) > ½ (ˆµ2 + ˆµ1)T Σˆ−1 (ˆµ2 − µˆ1) + log(N1/N) − log(N2/N)  
= ½ ˆµ2T Σˆ−1 ˆµ2 − ½ ˆµ1T Σˆ−1 µˆ1 + log(N1/N) − log(N2/N)

(b)

(c)  
ΣˆBβ = (ˆµ2 − µˆ1)(ˆµ2 − µˆ1)T β

Since (ˆµ2 − µˆ1)T β is a scalar and Σˆβ is a linear combination of B in the direction of (ˆµ2 − µˆ1), βˆ ∝ Σˆ−1(ˆµ2 − µˆ1).

(d)

We can replace N with N1N2/N (y2-y1):

β = N1N2/N(N-2) [(y2-y1) - (ˆµ2 − µˆ1)T β] Σˆ−1(ˆµ2 − µˆ1) ∝ Σˆ−1(ˆµ2 − µˆ1)

(e)

βˆo = 1/N 1T (Y − Xβ)

= -1/N(N1 µˆ1T+ N2 µˆ2T) β

f(x) = βo + βT x

🡪 f(x) = 1/N(Nx T - N1 µˆ1T- N2 µˆ2T) β

= 1/N(Nx T - N1 µˆ1T- N2 µˆ2T) λ Σˆ−1(µˆ2 - µˆ1)

When f(x) > 0,

NxT λΣˆ −1 (ˆµ2 − µˆ1) > (N1µˆ T 1 + N2µˆ T 2 ) λΣˆ −1 (ˆµ2 − µˆ1)

xT Σˆ −1 (ˆµ2 − µˆ1) > 1/N (N1µˆ T1 + N2µˆ T2) Σˆ −1 (ˆµ2 − µˆ1)

3.

(a)

Reduce M to row reduced form:  
R2 = R2 – 3R1  
[[1, 0, 3], [0, 7, -7], [2, -2, 8], [0, -1, 1], [5, 8, 7]]

R3 = R3 – 2R1

[[1, 0, 3], [0, 7, -7], [0, -2, 2], [0, -1, 1], [5, 8, 7]]

R5 = R5 – 5R1  
R2 = R2 / 7

R3 = R3 + 2R2

R4 = R4 + R2

R5 = R5 – 8R2

[[1, 0, 3], [0, 1, -1], [0, 0, 0], [0, 0, 0], [0, 0, 0]]

MMT  = [[1, 0, 3], [0, 1, -1], [0, 0, 0], [0, 0, 0], [0, 0, 0]] [[1, 0, 0, 0, 0], [0, 1, 0, 0, 0], [3, -1, 0, 0, 0]] = [[10, -3, 0, 0, 0], [-3, 2, 0, 0, 0], [0, 0, 0, 0, 0], [0, 0, 0, 0, 0], [0, 0, 0, 0, 0]]

m = np.array([[1,0,3],[3,7,2],[2,-2,8],[0,-1,1],[5,8,7]]

m.dot(m.transpose())

MMT =

MTM = [[1, 0, 0, 0, 0], [0, 1, 0, 0, 0], [3, -1, 0, 0, 0]] [[1, 0, 3], [0, 1, -1], [0, 0, 0], [0, 0, 0], [0, 0, 0]] = [[1, 0, 3], [0, 1, -1], [3, -1, 10]]

m.transpose().dot(m)

MTM =

(b, c)

Eigenvalues for MMT:

vals, vecs = np.linalg.eig(m.dot(m.transpose()))

array([ 2.14670489e+02, -8.88178420e-16, 6.93295108e+01,

-3.34838281e-15, 7.47833227e-16])

array([[-0.16492942, -0.95539856, 0.24497323, -0.54001979, -0.78501713],

[-0.47164732, -0.03481209, -0.45330644, -0.62022234, 0.30294097],

[-0.33647055, 0.27076072, 0.82943965, -0.12704172, 0.2856551 ],

[-0.00330585, 0.04409532, 0.16974659, 0.16015949, 0.43709105],

[-0.79820031, 0.10366268, -0.13310656, 0.53095405, -0.13902319]])

Eigenvalues and eigenvectors for MTM

vals, vecs = np.linalg.eig(m.transpose().dot(m))

array([ 2.14670489e+02, 9.32587341e-15, 6.93295108e+01])

array([[ 0.42615127, 0.90453403, -0.01460404],

[ 0.61500884, -0.30151134, -0.72859799],

[ 0.66344497, -0.30151134, 0.68478587]])

(d) SVD

M = U T

U is the eigenvectors of MMT, and V is eigenvectors of MTM.

=

U =

VT =

M = array([[-1.05943914, -2.97100607, -0.20731136],

[-2.8894697 , -1.50157122, -7.16683789],

[-2.20140628, -8.05956859, 1.45534974],

[-0.04126397, -1.05877815, 0.93498625],

[-4.96708402, -6.38480526, -8.51644679]])

(e) =

M = U T

M =

= array([[-1.02967352, -1.4859942 , -1.60302635],

[-2.94454898, -4.24948552, -4.58416142],

[-2.10062471, -3.03155911, -3.27031502],

[-0.02063881, -0.02978531, -0.03213111],

[-4.98325721, -7.19168861, -7.75808302]])