Machine Learning CS 6140
Miles Benjamin
Assignment 3

1. PCA Objective Value

We can write our objective function for PCA as:

$tr(YY^T UU^T)\ s.t.\ U^T U = I_d$

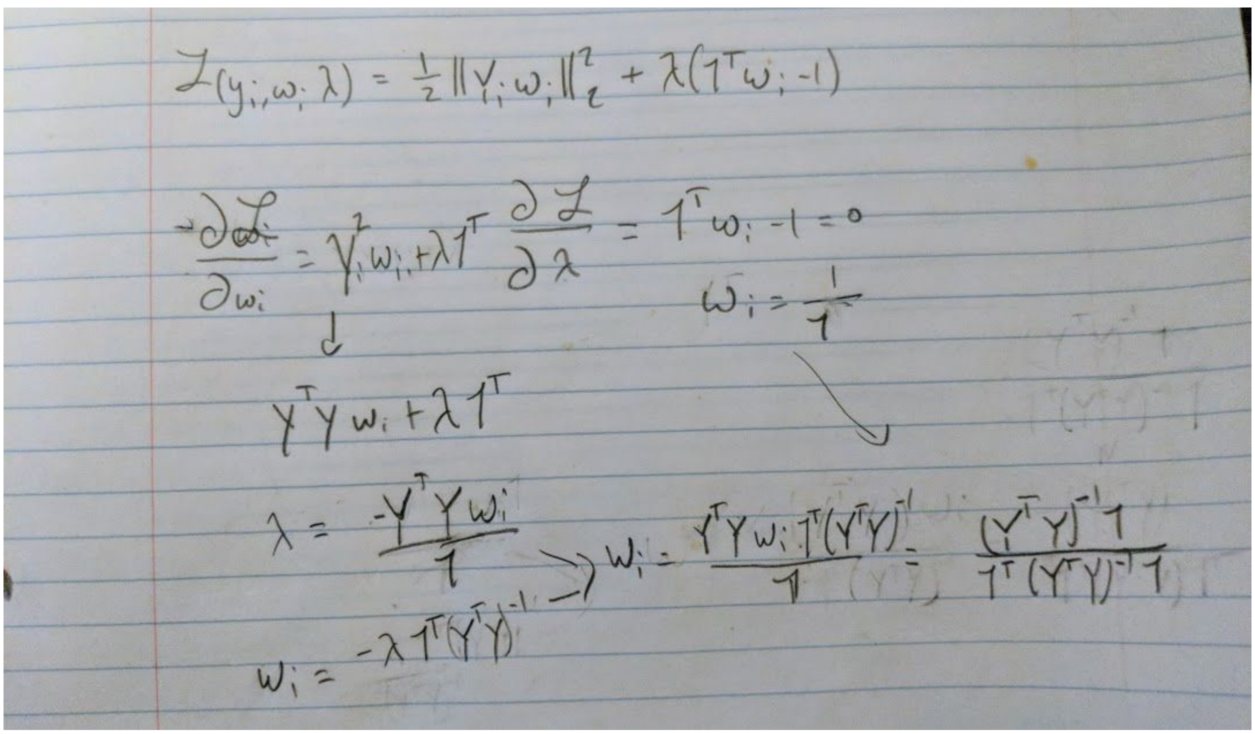
From here we can Van Newman's Trace Inequality Lemma, setting $A = YY^T$ and $B = UU^T$

$tr(AB^T) = \Sigma \sigma_i(A)\sigma_i(B)$

Since $U^T U = I_d$ we can say that the whole expression is equal to $\sum_{i=d+1}^{N} \sigma_i(Y)$

2. Locally Linear Embedding

A. This proof can be done by forming the lagrangian dual optimization, taking the derivatives with respect to the variables and solving:



$$\mathcal{L}(y_i, w_i, \lambda) = \frac{1}{2}\|Y_i w_i\|_2^2 + \lambda(1^T w_i - 1)$$

$$\frac{\partial \mathcal{L}}{\partial w_i} = Y_i^2 w_i + \lambda 1^T \qquad \frac{\partial \mathcal{L}}{\partial \lambda} = 1^T w_i - 1 = 0$$

$$w_i = \frac{1}{1}$$

$$Y^T Y w_i + \lambda 1^T$$

$$\lambda = \frac{-Y^T Y w_i}{1} \Rightarrow w_i = \frac{Y^T Y w_i 1^T (Y^T Y)^{-1}}{1} = \frac{(Y^T Y)^{-1} 1}{1^T (Y^T Y)^{-1} 1}$$

$$w_i = -\lambda 1^T (Y^T Y)^{-1}$$

B. For step 1 of LLE (find each point yi and take KNN) this is O(N^2) work since each point needs to scan all other points to find the nearest neighbors.

For step 2 of LLE (solve for Wij) this involves multiplying NxD matrices for each W (which is N work) This makes the whole thing 2 * N*D * N or O(D*N^2).

3. Laplacian Matrix

A. $x^T Lx = x^T(D - W)x$ Since D can be re-written as:

$$D = \sum_{i=1}^{n} E_i(W1)1^T$$

Where Ei is a nxn matrix with 1 at position (i,i). So the whole expression can be written as:

$x^T L x = x^T(\sum_{i=1}^{n} E_i(W1)1^T - W)x$

B. To show that L is positive semi definite we simply need to show that $x^T L x \geq 0$

If L is positive semi definite we can write it as $L = A^T A$

$x^T L x = x^T A^T A x = (Ax)^T(Ax)$

If we define some y = Ax then we can show:

$x^T L x = x^T A^T A x = (Ax)^T(Ax) = y^T y = ||y||^2 \geq 0$

C. L is a positive semi-definite matrix and is square, so it is invertible as long as it's determinant isn't 0. Since the diagonal of the Laplacian matrix is very large and positive (compared to the other values) this will dominate the computation of the determinant. Not to mention that since the values not on the diagonal are all negative and some of them are subtracted, the overall product is definitely more than 0.

D. L has as many 0 singular values as the graph has components. If the graph is all connected (one component) it will have exactly 1 0-value singular value.

4. **Neural Network**

A. If we define:

$A = c(w_3x_1 + w_5x_2 + w_1)$ *and* $B = c(w_4x_1 + w_6x_2 + w_2)$ *and* $D = (w_8A + w_9B + w_7)$

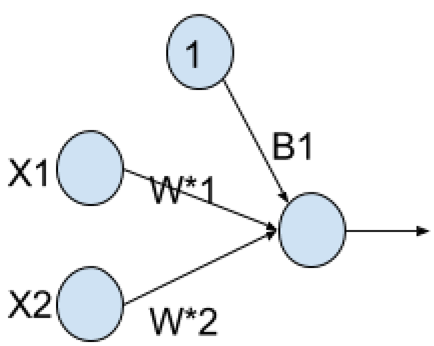
Then the output of the neural net

$P(y = 1|x, w) = 1/1 + e^{-D}$

The decision boundary is as always

$0.5 = 1/1 + e^{-D}$

B.



Our weights for W*1, W*2 and B1 can be expressed as follows:

$w_1^* = w_8cw_3 + w_9cw_4$

$w_2^* = w_8cw_5 + w_9cw_6$

$b_1 = w_8cw_1 + w_9cw_2 + w_7$

C. All neural nets with a linear hidden layer can be expressed as a neural net without that hidden layer. This only holds true if the activation function of the hidden layer is linear.