SDS 365 Midterm 2 Solutions

Curtis McDonald

May 19, 2021

1 Problem 1

For k fold cross validation split the data into k parts. Pick one part to validate, rest to train and train a model. Do this so each part get a chance being validation set, average to get approximate loss. Do this for a variety of λ values and pick the one with lowest average validation loss. Then train model with this λ on full data set to get final model.

2 Problem 2

a^2 is a good embedding because it is trying to make a value close to the original data point to minimize loss, but there is a bottle neck since it must express this point in fewer dimensions. This results in a lower dimensional point that captures the main formation (i.e. minimizes loss).

For SGD, define,

$$l_i = \|W^2 g(W^1 x_i) - x_i\|^2$$
$$\frac{\partial}{\partial W^2} l_i = 2(W^2 g(W^1 x_i) - x_i) g(W^1 x_i)^T$$

$$\frac{\partial}{\partial a^2} l_i = 2(W^2)^T (W^2 a^2 - x_i)$$

let $z^2 = W^1 x_i$

$$\frac{\partial}{\partial z^2} l_i = \frac{\partial l_i}{\partial a^2} \frac{\partial a^2}{\partial z^2} = 2(W^2)^T (W^2 a^2 - x_i) \circ (z^2)_+$$

(only values $z^2 > 0$ will contribute).

$$\begin{aligned} \frac{\partial l_i}{\partial W^1} &= \frac{\partial l_i}{\partial z^2} \frac{\partial z^2}{\partial W^1} \\ &= 2(W^2)^T (W^2 a^2 - x_i) \circ (z^2)_+ x_i^T \end{aligned}$$

then for SGD update pick a random index $j \subset [n]$ to train on,

$$W^2_{t+1} = W^2_t - \eta_t \frac{\partial l_j}{\partial W^2}$$
$$W^1_{t+1} = W^1_t - \eta_t \frac{\partial l_j}{\partial W^1}$$

3 Problem 3

Boosting is a sequential method. At iteration k we have our model so far $F_k(x_i)$. A iteration $k+1$ train new model to minimize

$$f_{k+1} = \text{argmin} \sum_i |f_{k+1}(x_i) - (y_i - F_k(x_i))|$$

(training on residuals). Gradient boosting also fine.

4 Problem 4

Initialize some assignments for the data to clusters. For those assignments, cluster center is median not average of points since we are looking at l_1 loss not sum of square loss. With these new centroids, assign each point to closest cluster in l_1 distance. Repeat alternating finding median of assigned point and assigning points to closest cluster center until termination condition. Either number of iterations termination or no points change clusters.

5 Problem 5

Note here crucially the **columns** represent out data x_i, therefore the vectors we are projecting onto are the columns of the U_k matrix, that is to find the coordinates in this basis we have $A = U_k^T$ which returns the inner product of the columns of X (our data points) with the columns of U_k (our orthonormal basis). We have

$$U_k^T U S V^T = S_k V_k^T$$

α_i are columns of this matrix,

$$\alpha_i = (S_k V_k^T)_{:,i}$$

6 Problem 6

$$X = USV^T, M = VS^2V^T = (SV^T)^T(SV^T)$$

See that M is a matrix transpose times itself, therefore we should approximate these matrices in low rank. That is, if A is the matrix with the embeddings α_i as it's columns we have

$$A = S_k V_k^T, \alpha_i = (S_k V_k^T)_{:,i}$$

7 Problem 7

Take gradient and set equal to zero,

$$\begin{aligned} X^T(X\beta - y) + \lambda\beta &= 0 \\ \beta &= (X^TX + \lambda I)^{-1}X^Ty \\ &= (VSU^TUSV^T + \lambda VV^T)^{-1}USV^Ty \end{aligned}$$

note $VV^T = 1$ since it is square orthogonal matrix, $U^TU = I$.

$$\begin{aligned}
&(VSSV^T + \lambda VIV^T)^{-1}VSU^Ty \\
&= V(S^2 + \lambda I)^{-1}V^TVSUy \\
&= V(S^2 + \lambda I)^{-1}SU^Ty \\
&= V_k(S_k^2 + \lambda I)^{-1}S_kU_k^Ty
\end{aligned}$$

then note S^2 is a diagonal matrix, λI is a diagonal matrix and this the inverse is a diagonal matrix with the entries reciprocal.

$$V_k(S_k^2 + \lambda I)^{-1}S_kU_k^Ty = \sum_{j=1}^{k} \frac{\sigma_j}{\sigma_j^2 + \lambda} v_j u_j^T y$$

8 Problem 8

Note here our data x_i is now the **rows** of our matrix X, not the columns as in problem 5.

If you would like, think of X^T as having the data as it's columns, then X^T has SVD VSU^T and the columns of V_k are what we would project onto, $\tilde{X}^T = V_k^T X^T$. Transpose back and it is clear our embedding is $A = XV = US$.

Therefore, our embedding is

$$A = XV_k = U_kS_k$$

taking the gradient we have

$$\begin{aligned}
A^TA\beta &= A^Ty \\
S_kU_k^TU_kS_k\beta &= S_kU_k^Ty \\
S_k^2\beta &= S_kU_k^Ty \\
\beta &= S_k^{-1}S_kU_k^Ty \\
&= S_k^{-1}U_k^Ty
\end{aligned}$$

9 Problem 9

By the optimality condition,

$$\begin{aligned}
(USV^T)^T(USV^T\beta - y) &= 0 \\
VS^TSV^T\beta &= VSU^Ty
\end{aligned}$$

note that rank X is only n dimensions, call this space W. Express $y = y' + y^\perp$ where $y' \in W, y^\perp \notin W$. Then we have

$$VS^TSV^T\beta = VSU^Ty'$$

if we set $\beta = VS^{-1}U^Ty' + q$ for any $q \perp V$ we have

$$\begin{aligned}
VS^TSV^T(VS^{-1}U^Ty' + q) &= VS^TSV^T(VS^{-1}U^Ty') \\
&= VSU^Ty'
\end{aligned}$$

and we have our result.

10 Problem 10

From above, the minimum choice of β is to set $q = 0$ (don't include anything orthogonal to V, just adds norm and is lost in optimality condition).