Problem 2 Cross-validation

Suppose sample size n is large, we could use k-fold cross-validation ($k=10$)

① set up a grid of λ e.g. $\lambda = 0 \sim 1$

② Randomly divide the data set into 10 folds

③ For each λ

Iterate through data set for 10 times

1). for the b^{th} iteration, $b = 1, \cdots, 10$

use b^{th} fold as validation set

use the rest 9 folds as training set

compute mean squared error (MSE) of validation set

$$MSE_b(\lambda) = \frac{1}{N} \sum_{n=1}^{N} (f(x) - y)^2$$

where N is the number of data in the b^{th} fold

2) compute mean MSE

$$\overline{MSE}(\lambda) = \frac{1}{10} \sum_{b=1}^{10} MSE_b(\lambda)$$

④ Choose λ with the smallest $\overline{MSE}$

Problem 2 Non-linear embeddings and gradients

① Why a^2 is a good low-dimensional embedding of data point x?

From the perspective of PCA, the low-dimensional embedding of data point x is $\alpha = U_k^T x$, which is similar to $a^2 = g(W^1 x)$ in Neural Net

The reconstructed version of x is $\tilde{x} = U_k \alpha = U_k(U_k^T x)$

Which is similar to $\tilde{x} = W^2(a^2) = W^2(g(W^1 x))$ in Neural Net

Also, benefits of ReLU are (1) sparsity when $W^1 x \le 0$

(2) a reduced likelihood of gradient to vanish

(3) Computationally efficient because of the non-saturation of gradient, which accelerates convergence of stochastic gradient descent.

② Stochastic Gradient Descent

For W^1:

$$W^1_k = W^1_{k-1} - \eta_{k-1} \nabla f_{J_k}(W^1_{k-1})$$

$$\nabla f_{J_k}(W^1_{k-1}) = \left[W^2_{k-1}[g(W^1_{k-1} x_{J_k})] - x_{J_k}\right] \cdot \left[[W^2_{k-1}]^T \circ g'(W^1_{k-1} x_{J_k})\right] x_{J_k}^T$$

For W^2:

$$W^2_k = W^2_{k-1} - \eta_{k-1} \nabla f_{J_k}(W^2_{k-1})$$

$$\nabla f_{J_k}(W^2_{k-1}) = \left[W^2_{k-1}[g(W^1_{k-1} x_{J_k})] - x_{J_k}\right] \cdot \left[g(W^1_{k-1} x_{J_k})\right]^T$$

Where η_{k-1} is learning rate, $J_k \in [n]$ is a uniform random variable.

Problem 3 Boosting

Use generalize gradient boosting

① initialize the first model $S_0(x)$ to be the median of response

$$S_0(x) = f_0(x) = \tilde{y}_i \qquad \text{Since Loss is least absolute deviation}$$

② Iterate for M times

For the $m+1$ th iteration, $m = 1, 2, \cdots, M$

(1) compute negative gradient $r_m \in R^n$

$$r_m = -\frac{\partial L(f(x), y)}{\partial f(x)} = -\frac{\partial \sum_{i=1}^{n} |f(x_i) - y_i|}{\partial f(x)} = \sum_{i=1}^{n} \text{sign}(f(x_i) - y_i)$$

(2) create a working data set W_m

$$W_m = (x, r_m)$$

(3) use a weak learner f_m (tree) to fit the working data set W_m by minimizing loss function of this working data set

$$f_m = \arg\min_f L(r_m, f) = \arg\min_f \sum_{i=1}^{n} |f_i - r_{mi}|$$

f_m is just the median of $\{r_{mi}; i \in n\}$

(4) Pick optimal step size λ_m by minimizing loss function of original training set

$$\lambda_m = \arg\min_\lambda L(y, S_{m-1}(x) + \lambda f_m(x))$$

$$= \arg\min_\lambda \sum_{i=1}^{n} \left| [S_{m-1}(x_i) + \lambda f_m(x_i)] - y_i \right|$$

(5) Update the model $S_m(x)$ using a small fraction λ_m of f_m

$$S_{m+1}(x) = S_m(x) + \lambda f_m(x)$$

③ Return the final model $S_M(x)$

$$S_M(x) = f_0(x) + \lambda f_1(x) + \dots + \lambda f_M(x)$$

Problem 4 : Clustering with least absolute deviation

Algorithm :

(1) Start with K initial random guess for μ_j where $j \in [K]$

(2) Compute π : assign each observation to the closest μ_j

(3) Compute μ_j : μ_j is the median of each cluster

(4) Repeat steps 1~3 until convergence

Problem 5 : PCA

Since $\alpha_i = A x_i \quad A \in R^{k\times p}$

Then $A = U_k^T$

where $U_k \in R^{p\times k}$, $[U_k]_{(:,i)} = u_i$

The column of matrix U_k are the top k left singular vectors of matrix X

matrix $\tilde{\alpha} = AX = U_k^T X = U_k^T (USV^T) = U_k^T USV^T = \underset{k\times p}{[I_k | 0]} SV^T = \underset{k\times p}{[S_k | 0]} V^T = S_k V_k^T$

$\Rightarrow$ column vector $\boxed{\tilde{\alpha}_i = [S_k V_k^T]_{(:,i)} \in R^k}$

$\tilde{\alpha}_i$ is the ith column of matrix $[S_k V_k^T]$ $\quad [\tilde{\alpha}_i]_{(j)} = \sigma_i [V_k]_{(ij)}$

where $S_k \in R^{k\times k} \quad [S_k]_{(ii)} = S_{(ii)} = \sigma_i$

The diagonal entry of matrix S_k is the top k singular values of data X.

$V_k \in R^{n\times k} \quad [V_k]_{(:,i)} = v_i$

The column of matrix V_k are the top k right singular vectors of data X.

Problem 6 : Lost your data

① Suppose the SVD of X is

$$X = USV^T$$

where $U \in R^{p\times p}$, $S \in R^{p\times p}$, $V \in R^{n\times p}$

U and V are orthonormal matrix $U^TU = I_p$, $V^TV = I_p$

S is diagonal matrix

$$M = X^TX = (USV^T)^T(USV^T) = VS^TU^TUSV^T = VS^TSV^T = VSSV^T = VS^2V^T$$

$$= \begin{pmatrix} | & | & & | \\ v_1 & v_2 & \cdots & v_p \\ | & | & & | \end{pmatrix}_{n\times p} \begin{pmatrix} \sigma_1^2 & & & \\ & \sigma_2^2 & & \\ & & \ddots & \\ & & & \sigma_p^2 \end{pmatrix}_{p\times p} \begin{pmatrix} — & v_1^T & — \\ — & v_2^T & — \\ & \vdots & \\ — & v_p^T & — \end{pmatrix}_{p\times n}$$

where $\begin{cases} U' = V & \in R^{n\times p} \\ S' = S^2 & \in R^{p\times p} \\ V' = V & \in R^{n\times p} \end{cases}$ $\boxed{M = VS^2V^T}$.

② $$\underset{\alpha_i \in R^k}{\arg\min} \sum_{i,j} (\alpha_i^T\alpha_j - M_{(ij)})^2 = \underset{\alpha \in R^{k\times n}}{\arg\min} \|\alpha^T\alpha - M\|_F^2$$

The optimal solution is SVD of M

$$\hat{\alpha}^T\hat{\alpha} = V_kS_k^2V_k^T = (S_kV_k^T)^T(S_kV_k^T)$$

$$\Rightarrow \hat{\alpha} = S_kV_k^T = \begin{pmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_k \end{pmatrix}_{k\times k} \begin{pmatrix} — & v_1^T & — \\ — & v_2^T & — \\ & \vdots & \\ — & v_k^T & — \end{pmatrix}_{k\times n} = \sum_{i=1}^{k} S_{(ii)} v_i^T$$

$$\Rightarrow \hat{\alpha}_i = [S_kV_k^T]_{(:,i)} \in R^k$$

Embeddings $\alpha_i \in R^k$ are the i^{th} column of matrix $[S_kV_k^T]$

Problem 7: Regularization and SVD

Take gradient with respect to β

$$\frac{\partial L}{\partial \beta} = \frac{\partial \frac{1}{2}\|X\beta-y\|_2^2 + \frac{\lambda}{2}\|\beta\|_2^2}{\partial \beta}$$

$$= \frac{1}{2}\frac{\partial (X\beta-y)^T(X\beta-y) + \lambda\beta^T\beta}{\partial\beta}$$

$$= \frac{1}{2}\left(X^T(X\beta-y) + \lambda\beta\right)$$

set gradient to be 0

$$X^T(X\beta-y) + \lambda\beta = 0$$

$$\hat{\beta} = (X^TX + \lambda I_p)^{-1}X^Ty$$

plug in $X = USV^T = \sum_{i=1}^{\text{rank}(x)} S_{(ii)}u_iv_i^T = \sum_{i=1}^{k} S_{(ii)}u_iv_i^T = U_kS_kV_k^T$

$$\hat{\beta} = \left[(U_kS_kV_k^T)^T(U_kS_kV_k^T) + \lambda I_k\right]^{-1}(U_kS_kV_k^T)^Ty$$

$$= \left[V_kS_k^TU_k^TU_kS_kV_k^T + \lambda I_k\right]^{-1}V_kS_k^TU_k^Ty$$

$$= \left[V_kS_k^2V_k^T + \lambda I_k\right]^{-1}V_kS_kU_k^Ty \qquad (S_k^T = S_k,\ U_k^TU_k = I_k)$$

$$= (V_kS_k^2V_k^T)^{-1}V_kS_kU_k^Ty + (\lambda I_k)^{-1}V_kS_kU_k^Ty$$

$$= V_k(S_k^2)^{-1}V_k^{-1}V_kS_kU_k^Ty + V_k(\lambda I_k)^{-1}S_kU_k^Ty \qquad (V_k^T = V_k^{-1})$$

$$= V_k(S_k^2)^{-1}S_kU_k^Ty + V_k(\lambda I_k)^{-1}S_kU_k^Ty \qquad (V_k^{-1}V_k = I_k)$$

$$\boxed{= V_k(S_k^2+\lambda I_k)^{-1}S_kU_k^Ty}$$ where $(S_k^2+\lambda I_k)^{-1}$ is a diagonal matrix

$$\hat{\beta} = \begin{pmatrix} | & | & & | \\ v_1 & v_2 & \cdots & v_k \\ | & | & & | \end{pmatrix}\begin{pmatrix} \frac{1}{\sigma_1^2+\lambda} & & & \\ & \frac{1}{\sigma_2^2+\lambda} & & \\ & & \ddots & \\ & & & \frac{1}{\sigma_k^2+\lambda}\end{pmatrix}\begin{pmatrix}\sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_k\end{pmatrix}\begin{pmatrix} - u_1^T - \\ - u_2^T - \\ \vdots \\ - u_k^T - \end{pmatrix}\begin{pmatrix} | \\ y \\ | \end{pmatrix}$$

$$= \begin{pmatrix} | & | & & | \\ v_1 & v_2 & \cdots & v_k \\ | & | & & | \end{pmatrix}\begin{pmatrix} \frac{\sigma_1}{\sigma_1^2+\lambda} & & & \\ & \frac{\sigma_2}{\sigma_2^2+\lambda} & & \\ & & \ddots & \\ & & & \frac{\sigma_k}{\sigma_k^2+\lambda}\end{pmatrix}\begin{pmatrix} - u_1^T - \\ - u_2^T - \\ \vdots \\ - u_k^T - \end{pmatrix}\begin{pmatrix} | \\ y \\ | \end{pmatrix}$$

$$\boxed{= \sum_{j=1}^{k}\frac{\sigma_j}{\sigma_j^2+\lambda}v_ju_j^Ty}$$

Problem 8 Low-rank regression

$$\beta_{lr} = \arg\min_{\beta_{lr}} \|A\beta_{lr} - y\|_2^2$$

$$\hat{\beta}_{lr} = (A^TA)^{-1} A^Ty$$

From PCA, we have

matrix $\alpha = U_k^T X = U_k^T (USV^T) = S_k V_k^T \in \mathbb{R}^{k\times n}$ with column α_i

Since $A \in \mathbb{R}^{n\times k}$ with row α_i

Then $A = \alpha^T = [S_k V_k^T]^T = V_k S_k$, $A^T = \alpha$

$$\Rightarrow \hat{\beta}_{lr} = [(S_k V_k^T)\ V_k S_k]^{-1} (S_k V_k^T) y$$

$$= (S_k S_k)^{-1} (S_k V_k^T) y$$

$$= S_k^{-1} V_k^T y$$

$$= \begin{pmatrix} \frac{1}{\sigma_1} & & & \\ & \frac{1}{\sigma_2} & & \\ & & \ddots & \\ & & & \frac{1}{\sigma_k} \end{pmatrix} \begin{pmatrix} - & v_1^T & - \\ - & v_2^T & - \\ & \vdots & \\ - & v_k^T & - \end{pmatrix} \begin{pmatrix} | \\ y \\ | \end{pmatrix}$$

$$= \sum_{j=1}^{k} \frac{1}{\sigma_j} v_j^T y$$

$$\boxed{\hat{\beta}_{lr} = S_k^{-1} V_k^T y}$$

Problem 9 Over-complete least squares

Since $\text{rank}(X) = n < p$, the SVD of X is now:

$$X = USV^T = \sum_{i=1}^{n} \sigma_i u_i v_i^T$$

where $U \in R^{n\times n}$ is square orthogonal matrix $U^TU = I_n$

$S \in R^{n\times n}$ is square diagonal matrix with positive entries (delete zero singular values)

$V \in R^{p\times n}$ is rectangular orthogonal matrix $V^TV = I_n$

Then $X^TX = (USV^T)^T(USV^T) = VS^TU^TUSV^T = VS^2V^T$

gradient optimality condition is:

$$X^T(X\hat{\beta} - y) = 0$$

$$X^TX\hat{\beta} = X^TXy$$

$$VS^2V^T\hat{\beta} = VSU^Ty$$

$$S^{-2}V^TVS^2V^T\hat{\beta} = S^{-2}V^TVSU^Ty$$

$$V^T\hat{\beta} = S^{-1}U^Ty$$

Because V^T is not invertible, solution is not unique

we have a guess, $\boxed{\hat{\beta} = VS^{-1}U^Ty = \sum_{i=1}^{p} v_i\sigma_i^{-1}u_i^Ty}$

$\Rightarrow V^T\hat{\beta} = V^T(VS^{-1}U^Ty) = S^{-1}U^Ty$

So $\hat{\beta} = VS^{-1}U^Ty$ is a potential solution

Problem 10 Extending Previous problem

$$\hat{\beta} = (I_p - U_n U_n^T)\beta \qquad \text{where } \beta \in V,\ U_n \in R^{p\times n}$$

Proof:

The energy (variance) of β (l_2-norm of β) can be decomposed to a part lives in K and a part lives in $K^\perp$

Where K is the column space of U

$K^\perp$ is the orthogonal complement space of K

suppose $\beta = v + w$

where $v \in K$, $w \in K^\perp$

Since $v = UU^T\beta$ where UU^T is an orthogonal projection onto space K

Then $w = \beta - UU^T\beta = (I - UU^T)\beta$ where $I - UU^T$ is an orthogonal projection onto $K^\perp$

$$\begin{aligned}
\|\beta\|_2^2 &= \|v+w\|_2^2 \\
&= \|v\|_2^2 + \|w\|_2^2 + 2\langle v, w\rangle \\
&= \|v\|_2^2 + \|w\|_2^2 \\
&= \|UU^T\beta\|_2^2 + \|(I-UU^T)\beta\|_2^2
\end{aligned}$$

$$\underset{\beta\in V}{\arg\min}\|\beta\|_2 = \underset{v\in K}{\arg\min}\|v\|_2^2 = \underset{w\in K^\perp}{\arg\max}\|w\|_2^2 = \underset{\beta\in V}{\arg\max}\|(I-UU^T)\beta\|_2^2$$

$$\Rightarrow \hat{\beta} = (I_P - U_n U_n^T)\beta$$

where $\beta \in V = \{\beta \mid X^TX\beta = X^Ty\}$

$U_n \in R^{p\times n}$ is rectangular orthogonal matrix $U^TU = I_n$

$$U_n = \begin{bmatrix} | & | & & | \\ u_1 & u_2 & \cdots & u_n \\ | & | & & | \end{bmatrix}_{p\times n}$$