Final Exam, CPSC 8420, Fall 2024

Alshurbaji, Mohammad

Dr. Kai Lu.

Problem 1: Lasso Problem

$$\min_{\beta} \frac{1}{2} \|y - X\beta\|_2^2 + \lambda \|\beta\|_1$$

1) Prove that if $\lambda \geq \|X^T y\|_\infty$, then $\beta^* = 0$

Sol.:

Infity Norm: $\|X^T y\|_\infty = \max_i |(X^T y)_i|$

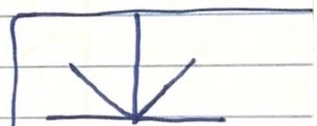
Note: In general when λ is large, some coe. in β becomes zero, Removing less Important features. (From its name: Shrinkage and Selection).

So, To find optimal sol., take gradient $= 0$:

$$-X^T(y - X\beta^*) + \lambda z = 0 \; ; \; z_i = \begin{cases} \text{Sign}(\beta^*_i), & \beta^*_i \neq 0 \\ \in [-1,1], & \beta^*_i = 0 \end{cases}$$

$$\therefore X^T(y - X\beta^*) = \lambda z$$

let's start from $\beta^* = 0$, and see what we get:



Absolute Function

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

$$\boxed{\therefore \lambda z = X^T y}$$

Now: when $\beta^*_i = 0$, $z_i \in [-1,1]$ for

$$|X^T y| \leq \lambda$$

hence:

$$\lambda \geq \|X^T y\|_\infty = \max |(X^T y)_i|$$

means:

$$X^T y = \lambda z \quad \text{with } z \in [-1,1]$$

2) This has been proved in Python, using two methods:

i) Descent Coordination.

ii) Sklearn Lasso Library. #

Problem 2:

1) SVD Decomposition : $X = U \Sigma V^T$

Define $\|X\|_2 = \Sigma(1,1)$, $\|X\|_F = \sqrt{\sum_i \sum_j |x_{ij}|^2}$

Prove: $\|X\|_F \geq \|X\|_2$ and Indicate when the equality holds.

Sol.:

1) $\|X\|_2 = \Sigma(1,1) = \sigma_1$ (spectral Norm means largest singular value)

2) $\|X\|_F = \sqrt{\sum_i \sum_j |x_{ij}|^2}$

Considering SVD: $X = U \Sigma V^T$

$X^T X = (U \Sigma V^T)^T (U \Sigma V^T)$

$= V \Sigma^2 V^T$

So, The diagonal elements of Σ^2 is σ^2.

$UU^T = I$, $VV^T = I$ (since they are orthogonal).

$\therefore \text{Trace}(X^T X) = \text{Trace}(\Sigma^2) = \sigma_1^2 + \sigma_2^2 + \sigma_3^2 + \dots + \sigma_r^2$

r: rank of X.

$\therefore \|X\|_F = \sqrt{\text{Trace}(X^T X)} = \sqrt{\sigma_1^2 + \sigma_2^2 + \dots + \sigma_r^2}$

3) Now : $\|X\|_F \overset{??}{\geq} \|X\|_2$

$\sqrt{\sigma_1^2 + \sigma_2^2 + \dots + \sigma_r^2} \geq \sigma_1$ ✓

Taking into considerations, that $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r$

Note: The equality $\|X\|_F = \|X\|_2$ holds ~~if~~ and only if :

All the singular values are zero except σ_1.

$\sqrt{\sigma_1^2} \overset{?}{=} \sigma_1$

$\sigma_1 = \sigma_1$ ✓ , since $\sigma_1 > 0$

and this happens when the rank of X is $\underline{1}$.

Problem 2 :

2) Fact: $vec(AXB) = (B^T \otimes A) vec(X)$ ...①

Find:

$$\min_X \|AXB - Y\|_F^2 \quad ...②, \text{ where } A \in R^{m \times p}, \; X \in R^{p \times q}, \; B \in R^{q \times n}, \; Y \in R^{m \times n}$$

Sol.: Review Basics:

$vec(AXB) \equiv$ Vectorization; $AXB \equiv [m \times p][p \times q][q \times n] \equiv [m \times n]$

Kronecker Product: multiply each element of first Matrix with the second Matrix.

Ex: $A \otimes B = [mp \times pq]$

Frobenus: $\|A\|_F$; It's the same as Eclidean But for Martices.

$$\|A\|_F = \sqrt{\sum_i \sum_j |a_{ij}|^2}$$

$$\|A\|_F = Tr(A^T A)$$

Note: $Vec(A + B) = Vec(A) + Vec(B)$

Now: Starting with the objective Function:

$$\min_X \|AXB - Y\|_F^2 = Tr((AXB - Y)^T (AXB - Y)) \quad ...③$$

Using Vectorization:

$$Vec(AXB - Y) = Vec(AXB) - Vec(Y)$$
$$= (B^T \otimes A) \, vec(X) - Vec(Y) \quad ...④$$

Subs. ④ in ②:

$$\min_X \|(B^T \otimes A) \, vec(X) - Vec(Y)\|_F^2$$

Take gradient and set to zero: From the Optimal least square solution

$$vec(X) = ((B^T \otimes A)^T (B^T \otimes A))^{-1} (B^T \otimes A)^T \; vec(Y)$$
$$= (BB^T \otimes A^T A)^{-1} (B^T \otimes A^T) \; vec(Y)$$

Since $X \equiv [m \times n] \rightarrow$ reshape $vec(X)$ to X. #

Problem 3: USArrests Dataset

1) PCA Approach: (Max. the variance of the data)

- Standarize the data.
- SVD
- k largest eigenvalues.
- Projection Matrix W from k.
- Transform the original dataset X via W to obtain a k-dimensional feature subspace Y.

⇒ Classes: Target

Features: Data

2) Data completion Using Proximal Gradient Descent

$$\frac{1}{2}\|P_\Omega(X - Z)\|_F^2 + \|Z\|_*$$

P_Ω → Projection Matrix.

(simulation)

- Mask, Generate a Matrix with missing entries.
- Applies a Proximal Gradient descent Method to find a Matrix Z that approximates X while min the obj function.

Results: After Implementation → we got a monotically decreasing objective Function with Iterations.

Problem 5: Logistic Regression: (Binary Classificati

$x_i \in R^{100}$, label is ± 1, then the obj.:

$$\min_{w} \sum_{i=1}^{m} \log(1 + \exp(-y_i w^T x_i))$$

while if the label is $\{1, 0\}$, the obj. is:

$$\min_{w} \sum_{i=1}^{m} \log(1 + \exp(w^T x_i)) - y_i w^T x_i$$

1) Gradient Descent Method:

$m = 100$

$$\text{step size} = \frac{1}{L} = \frac{1}{\text{Largest sing. value}} = \frac{1}{\frac{1}{4}\|XX^T\|_2}$$

$$= \frac{4}{\|X\|_2^2} \quad \text{* (Lipchitz).}$$

→ Apply both obj. functions to prove they are the same.

Result: The values and the chart shows us that the two objectives are exactly the same.

2) When it's $\{1, 0\}$, $P(y=1 \mid x, w) = \dfrac{1}{1 + \exp(-w^T x)}$,

Then the max. likelihood function is:

$P^y (1-P)^{1-y}$

→ Prove $\boxed{P^* = y}$. If we use MSE: $\min_P (y-P)^2$, $y=1$

$w \to 0$ in P very close to 0. Then if we optimize using GDM, ??

Sol.: Proof: As we did in class:

$P(y \mid x, w)$

→ $\log P(y \mid x, w) = y \log(P) + (1-y)\log(1-P)$

obj. → $\max_P y \log(P) + (1-y)\log(1-P)$

→ $\min_P -y\log(P) - (1-y)\log(1-P)$

Der. → $\frac{\partial}{\partial P}(-y\log(P) - (1-y)\log(1-P)) = \dfrac{-y}{P} + \dfrac{1-y}{1-P}$

Problem 5 : Section 2 : Completing :

der. = 0 → $\frac{-y}{p} + \frac{1-y}{1-p} = 0$

→ $y(1-p) = (1-y)p$

$\therefore$ $y = p$ # So, the optimal $p^* = y$ #

Now : MSE

$$\min_p \frac{1}{2}(y-p)^2$$

der. → $\frac{\partial}{\partial p}\left(\frac{1}{2}(y-p)^2\right) = -(y-p)$

GDM → $p = p - \alpha \frac{\partial}{\partial p}$

$\therefore$ → $p = p + \alpha(y-p)$

when : $y=1$, $p=0$

update → $p = p + \alpha \cdot 1$

with this large gradient, it's unproper so most likely it'll diverge instead if the step size wasn't set carefully.

This happens because MSE can't handle the small values of p ~~=~~ $(p=0)$ Unlike cross entropy as we seen before.

Section 3 : Newton Method :

After Implementing, it shows that Newton's is much faster and the convergence rate is way quicker. This is because it uses both Gradient and Hessian. #

Section 4 : SGD :

After Implementing, it shows that SGD is much faster but with slower convergence rate.

Problem 6: Kernel SVM.

1) $K(i,j) = \dfrac{\langle x_i, x_j \rangle}{\|x_i\|_2 \|x_j\|_2}$, Then K defines a proper kernel.

Sol.:

In order to be a proper kernel, it should satis. Two conditions:

1) Cosine Similarity:

$$k(i,j) = \frac{\langle x_i, x_j \rangle}{\|x_i\|_2 \|x_j\|_2} = \cos(\theta)$$

from ⇒ Definition of dot product.

So, First condition Holds

2) Positive Definite Matrix: (PSD)

$$\forall w \in \mathbb{R}^n, \; w^T k w \geq 0$$

$$w^T \left\langle \frac{x_i}{\|x_i\|_2}, \frac{x_j}{\|x_j\|_2} \right\rangle w$$

$$w^T \langle \tilde{x}_i, \tilde{x}_j \rangle w$$

$$= \left\langle \sum_{i=1}^{n} w \tilde{x}_i, \sum_{j=1}^{n} w \tilde{x}_j \right\rangle$$

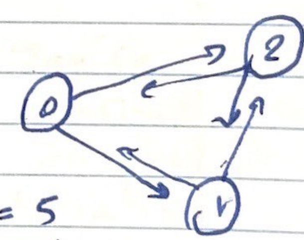
≥ 0 for any vector

∴ PSD #

∴ $k(i,j)$ is a proper kernel #.

2) Using Python (rbf_kernel, linear_kernel), the # of support vectors have been counted for each class.

3 classes → means 3 Pairs



pairs: [(0,1), (0,2), (1,2)]

Conclusion: support vectors:

pair (0,1) : [4 1] → support vectors = 5

pair (0,4) : [4 2] → " " = 6

pair (1,2) : [10 9] → " " = 19

Thank you So much!!

Q.7. :

1) My favourite book: Rich Dad Poor Dad.
This book has taught me a lot Business lesso
and how to live a good life. It's so Insightful
and enrichful.

My favorite travel destination: Lebanon.
I have done an Internship there Two years
ago. It's absolutely the best country ever you'd
visit. Kind pepple, Diversity, Gorgeous Places.

2) My favorite person: My father, he's absolutely
my role model. His lessons during my entire
life have been a great Footprint on my professional
and Social life. I will never forget his
favors.

3) My favorite Restaurant: Texas Roadhouse.
It's authentic, gorgoeus Place to visit, kind
Staff, and very delicious food.
My favourite offers: Fried Pickles and Bone-
In Ribeye.

4) My favorite Part: Is how you're so ~~energet~~
energetic, alway Positive and sharing the
Positivity with us. How you discuss different
Interesting topics with us. Additionally, for
Academic perspectives, how you teach us
both Math aspects and Coding for ML.
5) My favorit Algorithm is: SVM and kernel SVM
It's like Magikal thing, how it project into
low dimension space. I really loved it. #.
Thank you.