भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

Regression and Gradient Descent

Assignment -2 - CS3390

Kartik Srinivas - ES20BTECH11015
September 11, 2022

Index

1 Problem 1

We use the **epigraph trick in convex optimization**, we bound the modulus above by a value t_i and then we solve the optimization problem with t. Instead of minimizing the function directly, we find the lowest point of the upper bound.

$$P_1 = \min_{w_1, w_2, \ldots} \sum_{i=1}^{i=m} |\mathbf{w}^T x_i - y_i|$$

$$P_2 = \min_{\mathbf{t},\mathbf{w}} \sum_{i=1}^{i=m} t_i \quad \text{Subject to Constraints C}$$

$$C = \{\forall\ i|\ t_i \geq |\mathbf{w}^T x_i - y_i|\}$$

Note that, C is equivalent to the following set of constraints

$$G = \{\forall\ i|\ t_i \geq \mathbf{w}^T x_i - y_i \ \ \&\ \ \mathbf{w}^T x_i - y_i \geq -t_i\}$$

Now the constraints are all linear in format, and the objective is also linear, the problem can be seen as a linear program. For a more formal way of writing the linear program,

$$P = \min_{\mathbf{t},\mathbf{w}} \mathbf{1}^T \mathbf{t}$$

$$\vec{\mathbf{w}}^T \begin{bmatrix} \vec{\mathbf{x_1}} & \vec{\mathbf{x_2}} & \ldots & \vec{\mathbf{x_m}} \end{bmatrix} \leq \begin{bmatrix} t_1 + y_1 & t_2 + y_2 & \ldots \end{bmatrix}$$

$$\vec{\mathbf{w}}^T \begin{bmatrix} \vec{\mathbf{x_1}} & \vec{\mathbf{x_2}} & \ldots & \vec{\mathbf{x_m}} \end{bmatrix} \geq \begin{bmatrix} -t_1 + y_1 & -t_2 + y_2 & \ldots \end{bmatrix}$$

2 Problem 2

2.1 First direction

The flow of thought utilizes the fact that row independent matrices have Right Inverses. The following statements are equivalent

$$XX^T \text{ is invertible} \tag{1}$$
$$\therefore X \text{ has independent rows} \tag{2}$$
$$\therefore X \text{ has a right inverse} \tag{3}$$
$$\therefore Xb = d \text{ has a solution for every } b \in \mathbb{R}^d \tag{4}$$
$$\therefore \text{columns of X span the space} \tag{5}$$

2.1.1 Right invertibility

Proof of (3)
X has independent Rows, so $||\alpha^T X||^2 = 0$ if and only if $\alpha = 0$. From this it follows that $\alpha^T X^T X \alpha = 0$ if and only if $\alpha = 0$ which then means $\alpha^T X^T X = 0$ if and only if $\alpha = 0$. But $X^T X$ is square, and $X^T X$ has independent rows, so $X^T X$ is now invertible, therefore, the right inverse $= X^T(XX^T)^{-1}$
Proof of (4)
Consider b = Rd, where R is the right inverse of X

2.2 Second direction

$$X \text{ columns spans the space} \tag{6}$$
$$\therefore X \ \text{ has independent Rows} \tag{7}$$
$$\therefore X^T \ \text{ has Independent columns} \tag{8}$$
$$\therefore XX^T \ \text{ is square and has columns independent} \tag{9}$$
$$\therefore XX^T \text{ is invertible} \tag{10}$$

3 Problem - 3

The approach is to split the minimization problem into a bunch of smaller minimization problems

$$W^T = \begin{bmatrix} \mathbf{w_1}^T \\ \mathbf{w_2}^T \\ . \\ . \\ . \\ \mathbf{w_n}^T \end{bmatrix}$$

$$\phi(X) = \begin{bmatrix} \phi(\mathbf{x}) & \phi(\mathbf{x_2}) & \dots & \phi(\mathbf{x_m}) \end{bmatrix}$$

$$Y = [\mathbf{y_1}, \mathbf{y_2}, \dots, \mathbf{y_m}]$$

$$ERM = \min_{\mathbf{w} \in \mathbb{R}^{n \times d}} \sum_{i=1}^{i=m} ||(W^T \phi(x)_i - Y_i)||^2$$

The trick is to now just split the summation into several parts and minimize each one separately,

$$ERM = \min_{\mathbf{w} \in \mathbb{R}^{n \times d}} \sum_{i=1}^{i=m} \sum_{j=1}^{j=n} (\mathbf{w_j}^T \phi(x_i) - y_{ji})^2 \tag{11}$$

Now we exchange the summations and bring the outer summation over the w_i.

$$ERM = \min_{\mathbf{w} \in \mathbb{R}^{n \times d}} \sum_{j=1}^{j=n} \sum_{i=1}^{i=m} (\mathbf{w_j}^T \phi(x_i) - y_{ji})^2 \tag{12}$$

Since each term contains only one $\mathbf{w_j}$, the minimization can be done **individually**

$$ERM = \sum_{j=1}^{j=n} \boxed{\min_{\mathbf{w_j} \in \mathbb{R}^d} \sum_{i=1}^{i=m} (\mathbf{w_j}^T \phi(x_i) - y_{ji})^2} \tag{13}$$

The boxed part is equivalent to solving a single linear regression problem. This completes the proof

4 Problem 4

The explained variance has been calculated after predicting **all the four co-ordinates of the bounding box.**

4.1 No Bias and Identity Feature map :-

```
explained Variance with no bias and identity feature map
(fit_intercept = false)  =  -0.3276472406903836
```

4.2 Bias and Polynomial Kernel

```
Model explained variance with
polynomial mapping =  0.3106797763374498
```

5 Problem 5

w^* is the Baye's optimal parameter, because the samples y, are peint picked up form the posterior likelihood defined in the problem

$$p_*(y|x) = \mathcal{N}(w_*^T x, 1)$$

The loss used is square loss so the Baye's optimal function will assume the mean of the distribution. This can also be verified empirically, the coef_ parameter of the Linear regression model matches the chosen w_* in the program . To see the code please go to [Sri13] (Made public post assignment deadline)

$$f_*(x) = w_*^T x$$

For higher values of d, with m = 60,000 the time for convergence is very

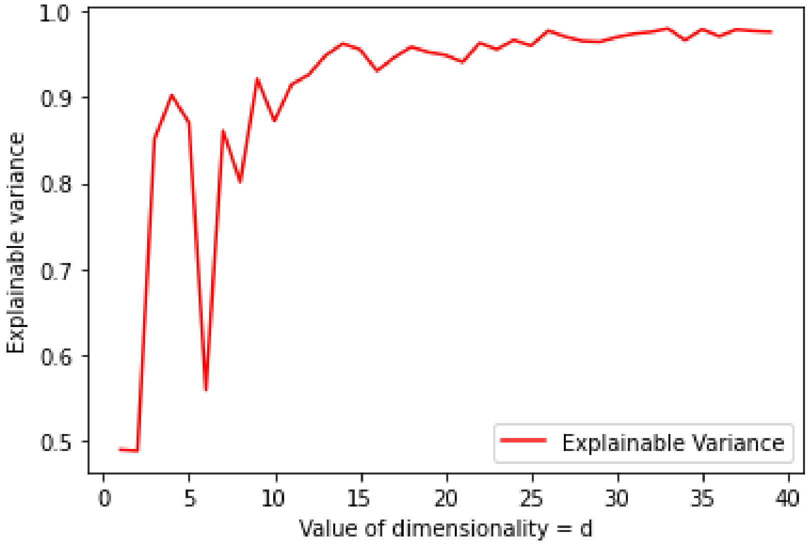

Figure 1: Explained Variance vs d

high.

6 Problem 6

The perceptron gives the same value perhaps because the same classifier is separating the points in both the cases (i.e the separating line just extends to become a separating plane in 3 dimensions?)

6.1 Perceptron

```
Score using Perceptron and identity map =  0.6825396825396826

Score using perceptron and polynomial kernel =  0.6825396825396826
```

6.2 Logistic Regression

```
score using unbiased identity feature map and
Logistic regression =  0.5238095238095238

Polynomial-Kernel -> Fails to converge
```

7 Problem 7

7.1 Analytical solution

$$\boldsymbol{\nabla}\big((v^T A v) - 2b^T v + c\big) = (A + A^T)v - 2b$$

The quadratic form is usually convex (provided A is symmetric positive definite)

$$\boldsymbol{\nabla}_v F = 0 \rightarrow Av = b$$

Analytical solution

$$v = A^{-1}b$$

Inverse exists as A is symmetric positive definite (non zero eigen values and $\det(A) = \prod \lambda_j$

7.2 Plots

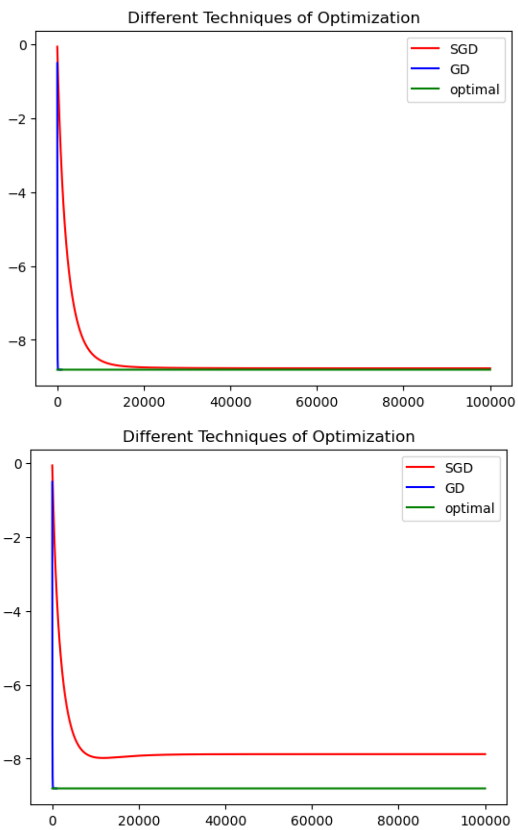

Figure 2: First plot has 0.1ϵ noise while the second has 0.5ϵ Noise

References

[Sri13] Kartik Srinivas. Cs3390 - machine learning. https://github.com/kartiksrinivas007/CS3390-Machine-Learning, 2013.