Assignment 2: CS 335 & CS 337

Richeek Das : 190260036

22nd September 2021

1.1 CS337: Theory

1.

1-vs-rest perceptron	**1-vs-1 perceptron**
1-vs-rest perceptron trains **K** binary classifiers when there are K classes. Due to the much lesser number of classifiers required, 1-vs-rest is usually faster for training.	1-vs-1 perceptron trains a binary classifier for every pair of classes. Therefore there are $\frac{\boldsymbol{K(K-1)}}{\boldsymbol{2}}$ binary classifiers for K classes.
1-vs-rest uses the entire dataset for training each of the sub-binary problems. This is an issue for large datasets and slow models (models that take a lot of time to train).	On the other hand 1-vs-1 models divide the datasets into one dataset for each class vs every other class (that is $\frac{K(K-1)}{2}$ overlapping sets).

2. If (f, y) gets misclassified as y' our perceptron algorithm will updates the weights of the perceptron.

$$score(f, y) = \sum_i f_i w_{y_i} \tag{1}$$
$$= f^T w_y \tag{2}$$

assuming $f \in \mathbb{R}^{n_f \times 1}$, $w_y \in \mathbb{R}^{n_f \times 1}$. $w \in \mathbb{R}^{n_f \times \mathrm{K}}$, where K is the number of classes.

If (f, y) gets misclassified as y': $w_y = w_y + f$ and $w_{y'} = w_{y'} - f$.

$$score(f, y) = f^T(w_y + f) \tag{3}$$
$$= f^T w_y + f^T f \tag{4}$$
$$= f^T w_y + \|f\|_2^2 \tag{5}$$
$$\geq f^T w_y \tag{6}$$

And,

$$score(f, y') = f^T(w_{y'} - f) \tag{7}$$
$$= f^T w_{y'} - f^T f \tag{8}$$
$$= f^T w_{y'} - \|f\|_2^2 \tag{9}$$
$$\leq f^T w_{y'} \tag{10}$$

with equality only when $f \in \mathbf{0}$.

3. We will show that the gradient descent update rule and the perceptron update rule are essentially the same:

$$\mathcal{L}(f, y) = \max(0, -yf^T w) \tag{11}$$

We can do case wise differentiation of the loss function:

$$\frac{\partial \mathcal{L}(f,y)}{\partial w} = \begin{cases} 0, & \text{for } yf^T w > 0 \\ -yf, & \text{for } yf^T w < 0 \\ undefined, & \text{for } yf^T w = 0 \end{cases} \tag{12}$$

Therefore the gradient descent rule is:

$$w = w - \sum_{\substack{f,y \in \\ dataset}} \nabla_w \mathcal{L}(f, y) = w - \sum_{\substack{f,y \in \\ wrong \\ pred}} (-yf) \tag{13}$$

This translates to the same expression as the perceptron update rule. For a particular weight of a particular class, we term +1 as the "class" and -1 as the "rest-of-the-classes". We get the following expression:

$$w = w + \sum_{\substack{f,y \in \\ wrong \\ pred \\ y=true}} f \tag{14}$$

$$w = w + \sum_{\substack{f,y \in \\ wrong \\ pred \\ y=wrong}} (-f) \tag{15}$$

Now, we notice that the gradient descent update rule for the aforementioned **hinge loss** is same as the perceptron update rule. We had previously proved that perceptron update rule reaches a convergence. Hence the result follows.

4. For the standard perceptron update rule (positive example):

$$(w + f)^T u = w^T u + f^T u \tag{16}$$

$$\geq w^T u + \gamma \tag{17}$$

where γ is the margin of separation.

$$\|w + f\|^2 = \|w\|^2 + 2w^T f + \|f\|^2 \tag{18}$$

$$\leq \|w\|^2 + \|f\|^2 \tag{19}$$

$$\leq \|w\|^2 + 1 \tag{20}$$

since $w^T f < 0$ (misclassification). Assume feature vectors are normalised.

Therefore, $u^T w^k \geq k\gamma$ and $\left\|w^k\right\|^2 \leq k$. Therefore by Cauchy-Schwartz inequality:

$$\sqrt{k} \geq \left\|w^k\right\| \geq u^T w^k \geq k\gamma \tag{21}$$

$$\implies k \leq \frac{1}{\gamma^2} \tag{22}$$

Now for the modified update rule:

$$(w + \frac{1}{2}f)^T u \geq w^T u + \frac{1}{2}\gamma \tag{23}$$

$$\left\| w + \frac{1}{2}f \right\|^2 \leq \|w\|^2 + \frac{1}{4} \tag{24}$$

Therefore, $u^T w^k \geq \frac{k\gamma}{2}$ and $\left\|w^k\right\|^2 \leq \frac{k}{4}$. Again using Cauchy-Schwartz inequality:

$$\frac{\sqrt{k}}{2} \geq \left\|w^k\right\| \geq u^T w^k \geq \frac{k\gamma}{2} \tag{25}$$

$$\implies k \leq \frac{1}{\gamma^2} \tag{26}$$

Therefore, the upper bound on the number of iterations under the modified algorithm **stays the same**. Hence we proved this result for 2 class setting. We can provide an intuitive argument for multiclass case:

Multiclass case: If we assume that w starts from 0, it seems like we are building w by adding and subtracting f at every step. Here if we uniformly scale every f it won't change anything. The magnitude of w will change, but all we need is the **sign** of $yf^T w$ for every datapoint to be +ve. If we scale w it won't affect the sign. Hence the number of iterations needed to converge solely depends on the relative magnitudes of f. Hence **the number of iterations needed to converge, stays the same.**

5. Here the points and their corresponding true predictions are:

$$[(0,0) : -1, (1,0) : -1, (0,1) : -1, (1,1) : 1]$$

where, a -1 classification means $class0$ and +1 classification means $class1$ ($class0$ and $class1$ refers to the boolean classes of true and false).

Plotting these points its easy to see that the separating line is the perpendicular bisector of the line joining the two points: $\left[\left(\frac{1}{2}, \frac{1}{2}\right), (1,1)\right]$. Therefore we can calculate γ that is margin of separation. It turns out to be $\mathbf{\frac{1}{2} \times \frac{1}{2} \times \sqrt{2} = \frac{1}{2\sqrt{2}}}$.

The upper bound on the number of iterations:

$$M \leq \frac{\|u^*\|_2^2 r^2}{\gamma^2} \tag{27}$$

We know $\gamma = \frac{1}{2\sqrt{2}}$. It is obvious $r = \sqrt{2}$. The best separating line is of the form $y = -x + c$. We also know that it passes through $\left(\frac{3}{4}, \frac{3}{4}\right)$. Therefore $c = \frac{3}{2}$. Hence, $u^* = \left[-1, \frac{3}{2}\right]$. $\|u^*\|_2^2 = 1 + \frac{9}{4} = \frac{13}{4}$.

Therefore the iteration upper bound $= \frac{\frac{13}{4} \times 2}{\left(\frac{1}{2\sqrt{2}}\right)^2} = 52$. **Therefore we can upper bound the number of iterations by 52**.

2.1 CS337: Theory

1. For a simple demonstration let's take a single observation of a variable X: $\mathbb{P}(X|\mu, \sigma^2) \sim \mathcal{N}(\mu, \sigma^2)$. Assume $\mu \sim \text{Laplace}(0, \lambda)$ and a known variance of σ^2. We can find the posterior distribution from this:

$$f(X, \mu, \sigma^2|\lambda) \propto f(X|\mu, \sigma^2) \cdot f(\mu|\lambda) \tag{28}$$

$$\propto \frac{1}{2\pi\sigma} \exp\left(-\frac{(x-\mu)^2}{\sigma^2}\right) \times \frac{1}{2\lambda} \exp\left(-\frac{|\mu|}{\lambda}\right) \tag{29}$$

MAP estimate is the μ which maximises this posterior distribution. We take the log-likelihood of this posterior and discard all the terms without μ.

$$\log\big(f(X,\mu,\sigma^2|\lambda)\big) = -\frac{\|x-\mu\|_2^2}{\sigma^2} - \frac{|\mu|_1}{\lambda} \tag{30}$$

Thus the maximum of this quantity will be MAP estimate and coincides with the LASSO problem if we reparameterize λ:

$$\text{arg}_\mu \min \|x-\mu\|_2^2 + \lambda'|\mu|_1 \tag{31}$$

2. This can be explained by taking a look at the level curves of the L1-norm and L2-norm terms. The least squares contours look like hyperellipsoids so they are likely to intercept the constraint region at the extremas. If we take a look at the extremas of the LASSO constraint its easy to notice that its a square with the vertices on the axes. Hence the Least square loss is likely to intercept at these extremas with certain coefficients remaining larger and others being forced to zero. Therefore for LASSO and not Ridge we would expect coefficients to be eliminated entirely and the sparsity follows.

3. There is no general closed form solution for the LASSO problem. But we do get a **closed form when the features are orthogonal**. We show this as follows:

The Least Squares Solution:

$$\hat{w}_{LS} = (X^TX)^{-1}X^Ty$$

which is a fixed quantity for a given X and y.

The LASSO Problem:

$$\text{arg}_w \min \frac{1}{2}\|y-Xw\|_2^2 + \gamma\|w\|_1 \tag{32}$$

$$\implies \text{arg}_w \min \left(\frac{1}{2}y^Ty - y^TXw + \frac{1}{2}w^Tw + \gamma\|w\|_1\right) \tag{33}$$

$$\implies \text{arg}_w \min \left(-y^TXw + \frac{1}{2}\|w\|_2^2 + \gamma\|w\|_1\right) \tag{34}$$

If $\boldsymbol{X}$ is orthogonal, $\boldsymbol{\hat{w}_{LS}}$ reduces to $\boldsymbol{X^Ty}$:

Therefore substituting this value in the last expression and expanding the w vector we get:

$$\text{arg}_w \min \sum_i -\hat{w}_{LS_i}w_i + \frac{1}{2}w_i^2 + \gamma|w_i| \tag{35}$$

Therefore our summation is composed of terms which can be individually minimised.

Our minimisation problem now reduces to:

$$\mathcal{E}_i = -\hat{w}_{LS_i}w_i + \frac{1}{2}w_i^2 + \gamma|w_i| \tag{36}$$

Since there is a modulus term we need to be careful with the minimisation problem. Notice when $\hat{w}_{LS_i} > 0$, we should have $w_i > 0$ otherwise we can flip the sign and get a lower value. Similarly when $\hat{w}_{LS_i} < 0$, we should have $w_i < 0$. Therefore we can divide the problem into cases and get differentiable counterparts:

Case 1: $\hat{w}_{LS_i} > 0$ and $w_i > 0$,

$$\mathcal{E}_i = -\hat{w}_{LS_i}w_i + \frac{1}{2}w_i^2 + \gamma w_i \tag{37}$$

We differentiate and set this to zero. We get, $w_i = \hat{w}_{LS_i} - \gamma$. We also have the constraint $w_i > 0$ in place so, $\hat{w}_{LS_i} > \gamma$ is needed.

Case 2: $\hat{w}_{LS_i} \leq 0$ and $w_i \leq 0$,

$$\mathcal{E}_i = -\hat{w}_{LS_i} w_i + \frac{1}{2} w_i^2 - \gamma w_i \tag{38}$$

We differentiate and set this to zero. We get, $w_i = \hat{w}_{LS_i} + \gamma$. We also have the constraint $w_i \leq 0$ in place so, $\hat{w}_{LS_i} \leq -\gamma$ is needed.

Therefore, we can combine the above two results as:

$$w_{\text{lasso}_i} = \text{sgn}(\hat{w}_{LS_i}) \left(|\hat{w}_{LS_i}| - \gamma\right)^+ \tag{39}$$

2.2 CS337: Lab

1. We complete the `ista()` function in the notebook and get the following result:

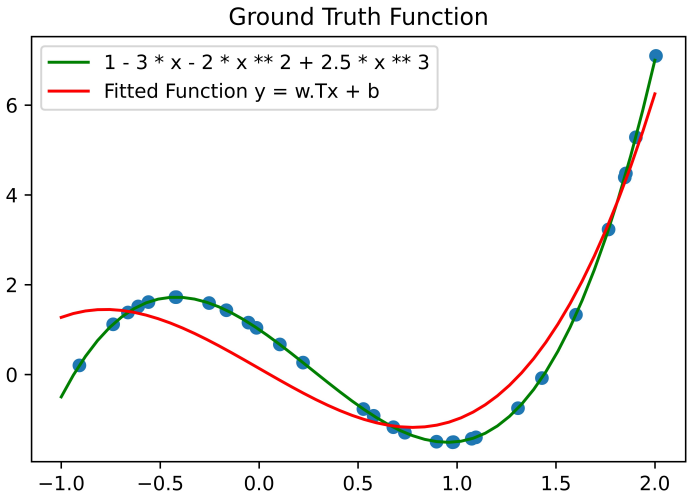

Figure 1: Curve fitted by lasso under the setting: `epochs=100000, lr=1e-3, lamda = 0.15`. We achieve `w` and `b`: `array([-1.2675, 0, 1.3983, 0, -1.2675]), 0.1347`

3. Here we analyse how LASSO performs feature selection using the Soft Thresholding function. We compare the weights observed from LASSO and Ridge regression in the following figure.

In the figure below we notice that in LASSO the 2nd and 4th weight values are exactly 0. For Ridge regression the 2nd and 4th values achieve relatively lower absolute weights but are not zero. They have finitely small values.

So in this case it is apparent LASSO is performing some sort of hard feature selection while Ridge performs soft feature selection (it reduces the value of a particular feature but almost never achieves exact 0 value). We have already explained in `Q2.1.2` why LASSO achieves sparser solutions. This is exactly what we observe here. By feature selection we mean LASSO zeroes out the 2nd and 4th feature and considers only the 1st, 3rd and 5th features as important for fitting the curve.

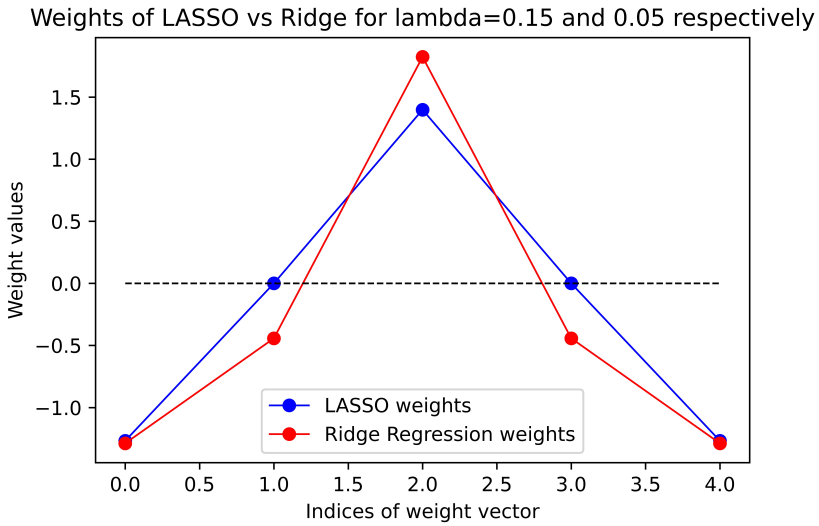

Figure 2: Comparison of weights observed from LASSO and Ridge Regression

3.1 CS337: Theory

1.

a. Increasing the value of λ in lasso regression: Increases bias and decreases variance. Increasing λ can be intuitively visualised as increasing our prior belief in w (Laplacian). Thus our strong prior belief won't be easily changed by small alterations in the dataset. But having a strong initial belief, will force us to have high bias towards that belief.

b. Increasing model complexity by adding more features of high degree: This is expected to lead to a reduction in bias but increase in variance. Having higher degree features corresponds to having more degrees of freedom to tune our model to fit the given data. Since it fits the data well, small alterations in data can lead to drastic changes in the trained model, hence the higher variance.

c. Reducing dimension by choosing only those subset of features which are of more importance: This can be viewed as a variance reduction method, since there is reduction in dimensionality and hence decrease in required model complexity. But this may also lead to increased bias from eliminating some of the relevant features.

2. Here $f(\hat{x})$ approximates $f(x)$, $x \in$ Test Set. We can write the MSE as:

$$\mathbb{E}\left[\left(f(\hat{x}) - f(x)\right)^2\right] = \mathbb{E}\left[f(\hat{x})^2\right] + f(x))^2 - 2\mathbb{E}\left[f(\hat{x})\right] f(x) \tag{40}$$

$$= \mathbb{E}\left[f(\hat{x})^2\right] - \mathbb{E}^2\left[f(\hat{x})\right] + \mathbb{E}^2\left[f(\hat{x})\right] + f(x))^2 - 2\mathbb{E}\left[f(\hat{x})\right] f(x) \tag{41}$$

$$= Variance\left(f(\hat{x})\right) + \left(\mathbb{E}(f(\hat{x})) - f(x)\right)^2 \tag{42}$$

$$= Variance\left(f(\hat{x})\right) + Bias^2\left(f(\hat{x}), f(x)\right) \textbf{ [Proved.]} \tag{43}$$

3.2 CS337: Lab

2: Bias variance trade-off for OLS, Ridge and Lasso regression methodologies.

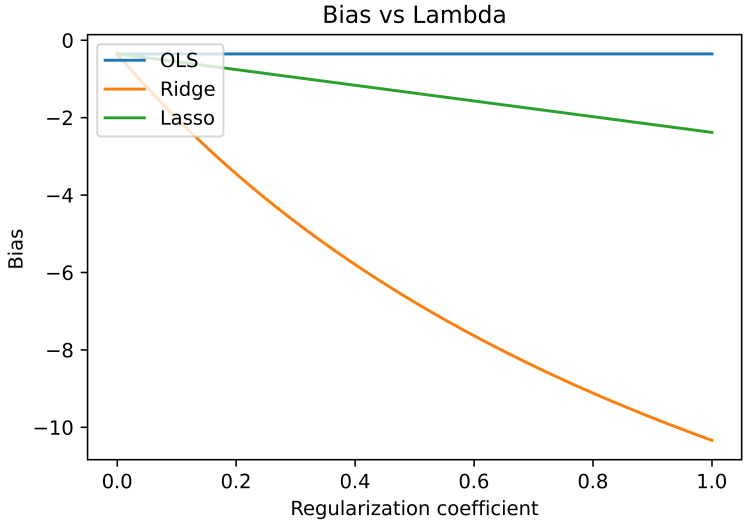

Figure 3: Variation in bias with change in regularisation coefficient

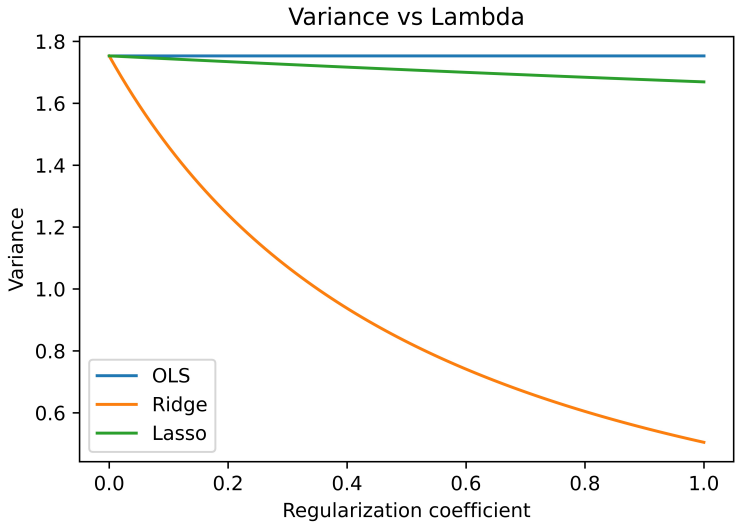

Figure 4: Variation in variance with change in regularisation coefficient

We see what we would naturally expect. With increase in the regularisation coefficient we see a **decrease in variance** and **increase in the absolute value of bias**. Also notice ridge regularization has a Gaussian Prior which converges pretty fast. Whereas LASSO has a Laplacian prior which is a heavy tailed distribution. So the effect of regularisation with respect to change in regularisation coefficient is much more prominent in the case of Ridge regression (the variance quickly decreases and bias quickly increases).