

Machine Learning Homework 2

Megi Dervishi

May 21, 2020

Exercise 1

- (a) Overfitting is especially a big problem when the training data size is low and the number of features to learn is very large compared to the data. By regularizing you avoid the risk of overfitting and hence have a better accuracy in the test data. So by choosing a good λ regularization significantly reduces the variance of the regression model without a huge increase in the bias of it. In this case the number of features are 784 while the training data has 900 images and the testing data has 250 images.
- (b) I used the proximal gradient descent algorithm also known as ISTA for finding the θ_λ for the ℓ_1 norm and applied directly Gradient Descent for the ℓ_2 norm. For both graphs the increase of λ results in the increase of error. This is normal because if λ becomes too high then the bias of the model increases which results in lower accuracy. Also note that for the ℓ_1 norm the error increases much faster than for the ℓ_2 norm.

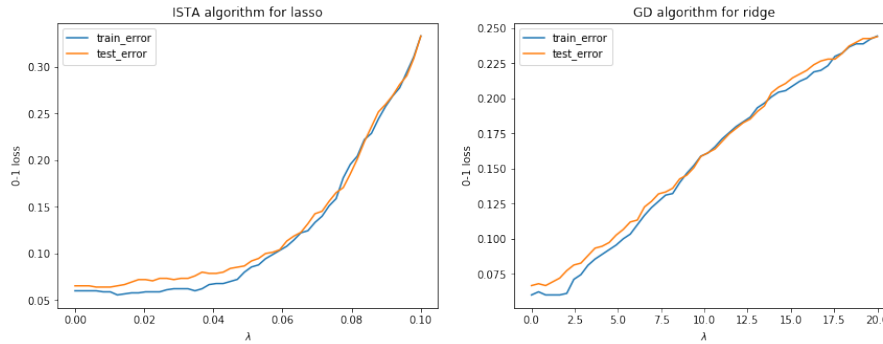


Figure 1: Error for the ISTA algorithm with $\|\cdot\|_1$ penalty (left) and GD algorithm with $\|\cdot\|_2^2$ penalty (right).

- (c) In order to best tune λ I performed K - fold cross validation which resulted with the following best lambdas:

$$\lambda_{lasso} = 0.011 \quad \lambda_{lasso} = 0.588$$

- (d)

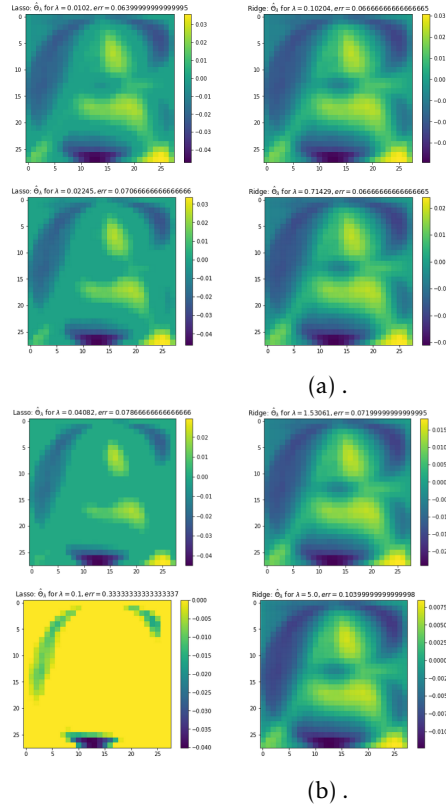
Exercise 2

- (a) By Baysian Law we get the result (1). By law of total probability we find $\mathbb{P}[X = x]$ and substitute it back into (1) which gives the final result (2):

$$\mathbb{P}[Y = 1|X] = \frac{\mathbb{P}[X = x|Y = 1] \mathbb{P}[Y = 1]}{\mathbb{P}[X = x]} = \frac{\mathcal{N}(\mu_1, \Sigma)(x) \cdot \pi}{\mathbb{P}[X = x]} \quad (1)$$

$$\begin{aligned} \mathbb{P}[X = x] &= \mathbb{P}[X = x|Y = 1] \mathbb{P}[Y = 1] + \mathbb{P}[X = x|Y = -1] \mathbb{P}[Y = -1] \\ &= \mathcal{N}(\mu_1, \Sigma)(x) \pi + \mathcal{N}(\mu_{-1}, \Sigma)(x) (1 - \pi) \end{aligned}$$

$$\frac{\mathcal{N}(\mu_1, \Sigma)(x) \cdot \pi}{\mathbb{P}[X = x]} = \frac{1}{1 + \frac{\mathcal{N}(\mu_{-1}, \Sigma)(x) \cdot (1 - \pi)}{\mathcal{N}(\mu_1, \Sigma)(x) \cdot \pi}} \quad (2)$$



By substituting the normal density function and simplifying we have (3). Now we want to transform (3) into something of the form $\exp(-\beta_0 - \langle \beta, x \rangle) = \exp(-\beta_0 - \text{tr}(\beta^T x))$:

$$\begin{aligned}
 \frac{\mathcal{N}(\mu_{-1}, \Sigma)(x) \cdot (1 - \pi)}{\mathcal{N}(\mu_1, \Sigma)(x) \cdot \pi} &= \exp\left(-\frac{1}{2}(x - \mu_{-1})^T \Sigma^{-1}(x - \mu_{-1}) + \frac{1}{2}(x - \mu_1)^T \Sigma^{-1}(x - \mu_1) + \log\left(\frac{1 - \pi}{\pi}\right)\right) \quad (3) \\
 &= \exp\left(\frac{1}{2}(\mu_{-1}^T \Sigma^{-1} x + x^T \Sigma^{-1} \mu_{-1}) + \frac{1}{2}(-\mu_1^T \Sigma^{-1} x - x^T \Sigma^{-1} \mu_1) + \frac{1}{2}(\mu_1^T \Sigma^{-1} \mu_1 - \mu_{-1}^T \Sigma^{-1} \mu_{-1}) + \log\left(\frac{1 - \pi}{\pi}\right)\right) \\
 &= \exp\left(\frac{1}{2}(\mu_{-1}^T \Sigma^{-1} x + \mu_{-1}^T \Sigma^{-1} x) + \frac{1}{2}(-\mu_1^T \Sigma^{-1} x - \mu_1^T \Sigma^{-1} x) - \underbrace{\left(\frac{1}{2}(\mu_{-1}^T \Sigma^{-1} \mu_{-1} + \mu_1^T \Sigma^{-1} \mu_1) - \log\left(\frac{1 - \pi}{\pi}\right)\right)}_{\beta_0}\right) \\
 &= \exp((\mu_{-1}^T \Sigma^{-1} + \mu_1^T \Sigma^{-1})x - \beta_0) \\
 &= \exp(-(\mu_1^T \Sigma^{-1} - \mu_{-1}^T \Sigma^{-1})x - \beta_0) \\
 &= \exp(-\beta^T x - \beta_0)
 \end{aligned}$$

Since $-\beta^T x$ is a value then it is equal to $\text{tr}(-\beta^T x)$. We finally conclude that indeed:

$$\mathbb{P}[Y = 1|X] = \frac{1}{1 + \exp(-\beta_0 - \langle \beta, X \rangle)}$$

(b) We have that:

$$\mathcal{L}(\beta_0, \beta) = \prod_{j=1}^n \mathbb{P}[Y = y_j|X = x_j] = \prod_{j=1}^n \frac{1}{1 + \exp(-y_j \beta_0 - y_j \langle \beta, x_j \rangle)} \quad (4)$$

The reason behind the second equality is because we notice that $\mathbb{P}[Y = -1|X] = \frac{1}{1 + \exp(\beta_0 + \langle \beta, X \rangle)}$ and by (a) we can generalize $\mathbb{P}[Y = j|X = x_j]$ to the above expression. We have to maximize (4) in other words find

$\arg \max_{\beta \in \mathbb{R}^d} \log(\mathcal{L}(\beta_0, \beta))$. Since \log is an increasing function we can take the log of the likelihood. Hence:

$$\begin{aligned} \arg \max_{\beta \in \mathbb{R}^d} \log(\mathcal{L}(\beta_0, \beta)) &= \arg \max_{\beta \in \mathbb{R}^d} \sum_{j=1}^n \log\left(\frac{1}{1 + \exp(\beta_0 + \langle \beta, X \rangle)}\right) \\ &= \arg \max_{\beta \in \mathbb{R}^d} - \sum_{j=1}^n \log(1 + \exp(\beta_0 + \langle \beta, X \rangle)) \\ &= \arg \min_{\beta \in \mathbb{R}^d} \frac{1}{n} \sum_{j=1}^n \log(1 + \exp(\beta_0 + \langle \beta, X \rangle)) \end{aligned}$$

If we let $\beta_0 = 0$ then this corresponds to the solution of logistic regression with $\lambda = 0$. Note(since $\frac{1}{n}$ is a constant adding it to argmin does not change).

- (c) For the algorithm that I implemented I based myself on the following: <https://eigenfoo.xyz/lda/>.
- (d) False positive means that you predict positive when the correct answer is false and False negative means that you predict negative when the correct answer is true. A classification error simply describes the ratio between the number of wrongly predicted values and the total number of values. The confusion matrix shows the ways in which your model was confused. In other words it gives more insights on which label was wrongly predicted i.e. when do you have false positive or false negative. By having these extra information you could pin-point better where the error comes from and perhaps improve it.

Exercise 3

- (a) Since we are doing unsupervised learning I have reloaded the data into one and reshuffled it. The dataset is composed by 1650 of 28 by 28 pixels images which are normalized. Hence the input space is $\mathcal{X} \in [0, 1]^{784}$ and $\mathcal{Y} \in \{0, 1, 2\}$ i.e. the image will have a label $Y_i = 0$ if it shows "A", 1 if it shows "B" and 2 if it shows "C". In figure a y
- (b) You get an error of 20% which may seem as if it is a lot. However when looking at the confusion matrix we observe that the model mostly confuses the "B" and "C" and is always good at separating them between "A".