CS 337, Fall 2023
Overfitting and Regularization

Scribes: Sabyasachi Samantaray*, Aditya Nemiwal*, Chilukuri Abhiram*,
Shah Param Kaushik, Kommineni Sai Lokesh, Mandala Praneeth Goud
Edited by: Ashwin Ramachandran

August 8, 2023

1 Notation

The following are the definitions of symbols used in the scribe:

- d is the number of features or dimensions of the data
- n is the total number of data points
- $\mathbf{w} \in \mathbb{R}^{d+1}$ is the vector of parameters that the model aims to learn.
- $\mathbf{y} \in \mathbb{R}^n$ is the vector of target or output values in the training data.
- $\phi \in \mathbb{R}^{n \times (d+1)}$ is the feature matrix; each row in the matrix corresponds to a data point, and each column corresponds to a specific feature.

2 Overfitting

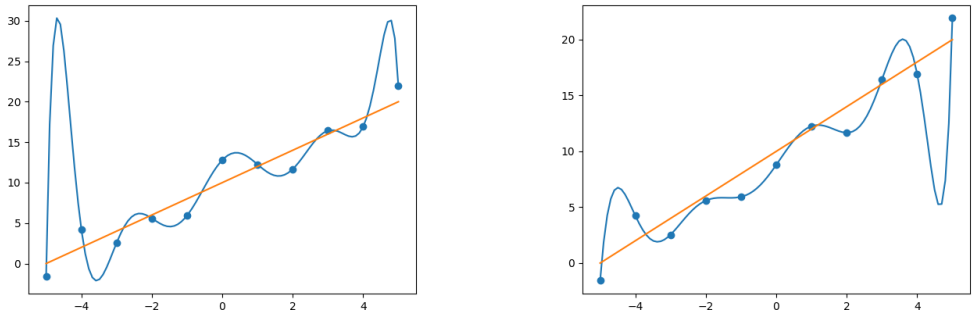
It is tempting to assume that having large number of parameters (making complex model) in our hypothesis class would fit the training data perfectly. But this would fail to predict new unseen data miserably. This is termed **overfitting**.
Consider the line y = 2x+10, let's say we sampled few points from this true line, and this process had some noise involved(Figure 1). In the figure 1a, we try to fit a polynomial regression model with degree 10. Our predictor function $h_{\mathbf{w}}(\mathbf{x})$ is given as:

$$h_{\mathbf{w}}(\mathbf{x}) = w_0 + w_1 x + w_2 x^2 + \cdots + w_{10} x^{10}$$

It is clearly evident from the image, that the training error is 0, the model perfectly fits the training data, but the test error is fairly high due to the huge variance of the predictor function. Additionally, on small perturbation(here, changing the the data point(0,13) to (0,8), 1b), has vastly changed the shape of the curve fit, because of several degrees of freedom.

Ideally we would want to hit the sweet spot between a simple fit (Underfitting, suffering from high bias) and a complex fit (Overfitting, suffering from high variance).

*Larger credit to these scribes for the final notes.



(a) Original (b) Perturbation

Figure 1: Overfitting due to complex modelling

2.1 Combatting Overfitting

Consider the problem of polynomial regression. Our predictor function $h_{\mathbf{w}}(\mathbf{x})$ is given as:

$$h_{\mathbf{w}}(\mathbf{x}) = w_0 + w_1 x + w_2 x^2 + \cdots + w_d x^d$$

The hyperparameter we need to tune here is d (the degree of the polynomial). To find the best d, a possible **coarse strategy** would be to tune d and evaluate the model on a development set to pick a good value of d. This is not a very efficient approach as it leads to higher training times and hence increased computational cost.

3 Regularization

Instead of tuning, regularization modifies the loss function to explicitly constrain the model complexity. The general regularised optimisation equation looks like:

$$L_{reg}(\mathbf{w}, D_{train}) = L_{MSE}(\mathbf{w}, D_{train}) + \lambda R(\mathbf{w})$$

where the first term, $L_{MSE}(\mathbf{w}, D_{train})$ is the measure of fit to the training data set and the second term $\lambda R(\mathbf{w})$ is the regulariser term, with $\lambda \geq 0$. The $R(\mathbf{w})$ is a measure of the model's complexity. This can help alleviate overfitting caused by the first term. It does this by penalising/shrinking weights in the weight vector making some of them equal to near zero, Thus even if the model is a high-degree polynomial, minimizing the L_{reg} yields many (near) zero weights, thereby shrinking the model complexity. Two examples of these shrinkage-based regularizations are discussed in the sections below.

Why this works? When the coefficients are very large and the model is highly complex, the errors on dev set or test set are large due to large variances in the predictor function. This can be alleviated by penalising the weight terms(weight norm), thereby making values of $\mathbf{w}$ small, and hence the errors are not every large!

3.1 Ridge Regression

Ridge regression (also called L2-Normalised Regression) is a regularization technique to combat overfitting. The penalty term $R(\mathbf{w})$ here is a Euclidean norm, given as $R(\mathbf{w}) = ||\mathbf{w}||^2$. The optimisation objective function and the optimal weights for L2-regularized regression can be written as:

$$\mathbf{w}_{L_2} = \arg\min_{\mathbf{w}}(||\mathbf{y} - \phi\mathbf{w}||_2^2 + \lambda||\mathbf{w}||_2^2)$$

The above equation is equivalent to the following constrained optimization problem.

$$\mathbf{w}_{L_2} = \arg\min_{\mathbf{w}} ||\mathbf{y} - \phi\mathbf{w}||_2^2 \text{ s.t } ||\mathbf{w}||_2^2 \leq t^2$$

Note: Usually, we don't include w_0 in the penalty term R($\mathbf{w}$). This goes well with the intuition that the intercept doesn't depend on the feature vectors and hence penalising w_0 would lead to a poor fit.

Infact, this optimisation problem has a closed form solution just as the non regularised objective function.

$$\nabla L_{ridge}(\mathbf{w}) = 0$$

$$-2\phi^T(\mathbf{y} - \phi\mathbf{w}) + 2\lambda\mathbf{w} = 0$$

$$\mathbf{w}_{ridge} = (\phi^T\phi + \lambda I)^{-1}\phi^T\mathbf{y}$$

Note: The best thing about the closed form solution is that such a closed form solution always exists, unlike the unregularized loss function solution, because here the term $\phi^T\phi + \lambda I$ is always invertible(provided the regularization parameter $\lambda > 0$).

A matrix $\mathbf{X} \in \mathbb{R}_{n\times n}$ is positive definite if for any $\mathbf{v} \neq 0$, $\mathbf{v}^T\mathbf{X}\mathbf{v} > 0$. And a positive definite matrix is always invertible. Since $\mathbf{v}^T\mathbf{X}\mathbf{v} > 0$ for all $\mathbf{v} \neq 0$, this implies $\mathbf{X}\mathbf{v} \neq 0$ for all $\mathbf{v} \neq 0$, which is the definition of an invertible matrix.

In our case, consider a non zero vector $\mathbf{v}$,

$$\mathbf{v}^T(\phi^T\phi + \lambda I)\mathbf{v} = \mathbf{v}^T\phi^T\phi\mathbf{v} + \lambda\mathbf{v}^T\mathbf{v} = (\phi\mathbf{v})^T\phi\mathbf{v} + \lambda\mathbf{v}^T\mathbf{v} = ||\phi\mathbf{v}||_2^2 + \lambda||\mathbf{v}||_2^2$$

The above expression is strictly positive for positive values of λ.

3.2 Lasso Regression

Also known as L1-normalised regression, Lasso (Least Absolute Shrinkage & Selection Operation) Regression is a similar technique to Ridge regression, but instead of using Euclidean L2 norm for the penalty, we use the L1 norm. So $R(\mathbf{w}) = ||\mathbf{w}||_1$, and the total loss function can be expressed as:

$$\mathbf{w}_{L_1} = argmin_{\mathbf{w}}(||\mathbf{y} - \phi\mathbf{w}||_2^2 + \lambda||\mathbf{w}||_1)$$

The equivalent constrained form for the above is:

$$\mathbf{w}_{L_1} = argmin_{\mathbf{w}}(||\mathbf{y} - \phi\mathbf{w}||_2^2) \text{ s.t } ||\mathbf{w}||_1 \leq t$$

As the L1 norm isn't differentiable, there is no closed form solution for the above optimisation problem. Other methods which can be used to solve for the optimal weights:

- Quadratic Programming
- Iterative optimisation algorithms (e.g. Gradient Descent)

3.3 Comparison

L1 Regularization yields sparse weight vectors compared to L2 Regularization. This can be intuitively understood from an example.

Referring to the figure 2, assume the constrained version of the regularization methods, and the predictor with only two weights β_1 and β_2. Say, at $\hat{\beta}$, the MSE is minimised. So around that we draw the contours(ellipses) of the loss function, and the points at which a contour touches the boundary of enclosed area

by the constraints (rhombus in Lasso, and circle in Ridge) gives the final optimal weights.

There will be a large number of cases, where the contour will touch the square at one of its corners (one weight resulting in zero) as compared to touching the circle on the axes. By extrapolation, we can see that in higher dimensions, the cases where multiple weights are zero will occur much more frequently in Lasso regression due to the sharp boundaries. So Lasso regression will prove to be useful in cases when there are outliers present in the training data and using Ridge regression could result in non-zero weights of undesired features present in the set of basis functions. Lasso can effectively ignore these features by setting their weights to zero. Ridge, with its more gradual constraints, might still assign non-zero weights to these features.

Note: It is not that always sparsity helps. Infact, for neural networks L2 regularization is most widely used (due to its smoothness). But for many benchmark tasks (feature selection tasks), L1 regularization is preferred and encourages models that utilize a smaller subset of features, which can be valuable for interpretability and reduces computational complexity.

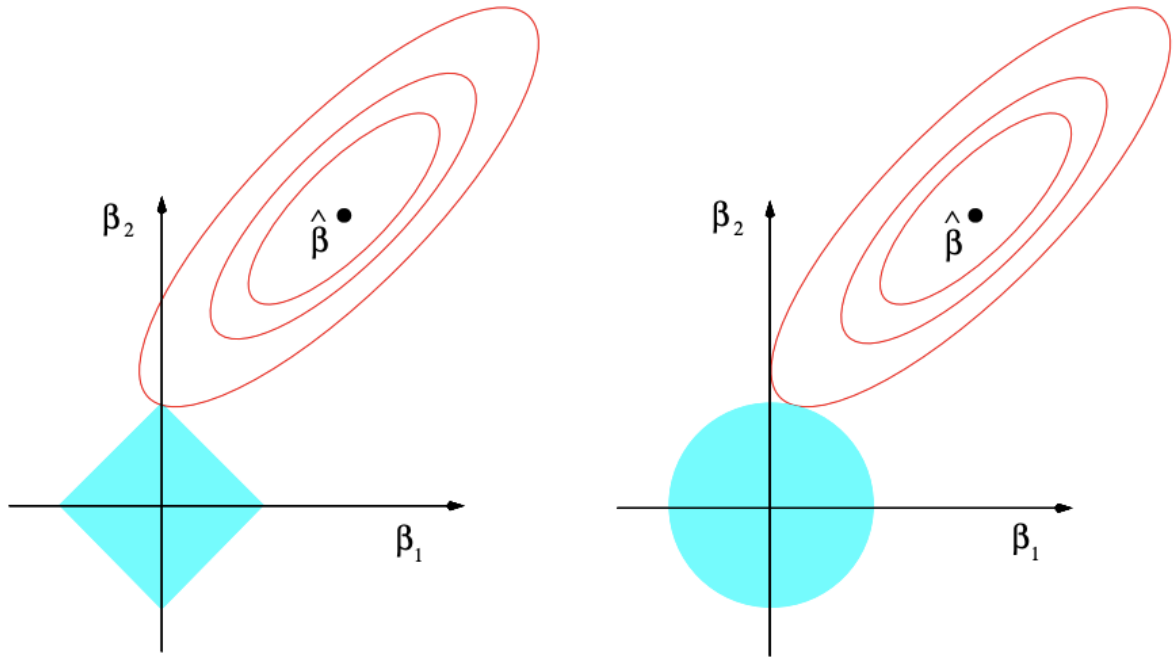


FIGURE 3.11. *Estimation picture for the lasso (left) and ridge regression (right). Shown are contours of the error and constraint functions. The solid blue areas are the constraint regions* $|\beta_1| + |\beta_2| \leq t$ *and* $\beta_1^2 + \beta_2^2 \leq t^2$*, respectively, while the red ellipses are the contours of the least squares error function.*

Figure 2: Comparison of L1 and L2 Regression methods. Figures reproduced from The Elements of Statistical Learning (Trevor Hastie, Robert Tibshirani and Jerome Friedman. Second Edition. 2009)