ML - Lecture Notes - 18

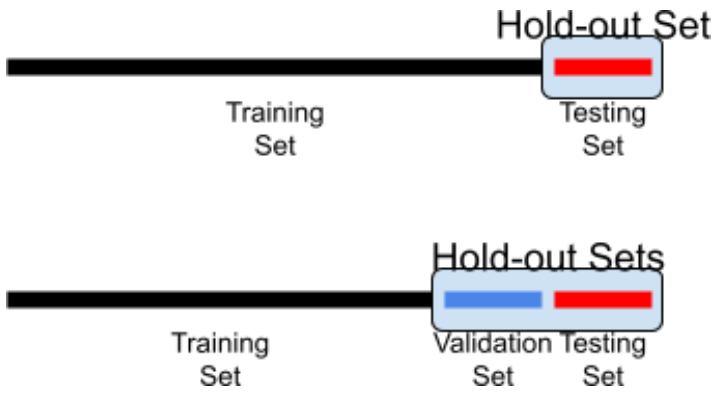
Sagnik Rout, Krishanu Kanta

March 2023

Overview

Hold-out sets

Instead of using an entire dataset for training, different sets, called validation and test sets, are separated or set aside (and, thus, the hold-out name) from the entire dataset. The model is trained only on what is termed the training dataset.



Reason:- we want to test and validate using data that the model has not seen; otherwise, it will be just memorisation.

Bias

In statistics, bias is the difference between the expected value of an estimator and its estimand. Alternatively, in other words, the difference between the predicted and actual values. Bias does not come from the ML/AI algorithms; it comes from people.

Expected value

$E(X)$ or $E(X = x)$ or expected value is the theoretical probability-weighted mean of the random variable X. For discrete, it is the sum of the values of X multiplied by its corresponding probability. And for the continuous variable sum is replaced by the integral (with the density $f(x)$).

$$E(X) = \sum x_i P(X = x_i) \qquad (discrete)$$

$$E(X) = \int_{-\infty}^{\infty} x f(x) dx \qquad (continuous)$$

The expected values are sums/integrals, so whatever we are allowed to do with constants and brackets for sums and integrals, we are also allowed to do with expected values. That is why if a and b are constants, $E[aX + b] = aE(X) + b$. Oh, and $E(X)$ itself is also a constant — it is not random after it has been calculated — so $E(E(X)) = E(X)$.

Variance

Variance is the square of the standard deviation. It is one of the crucial descriptors of the shape of a distribution. When there is low variance, we have more certainty about the prediction.Replacing X with $(X - E(X))^2$ in the $E(X)$ formula gives a variance of a distribution.

$$V(X) = E[(X - E(X))^2] = \sum [x - E(X)]^2 P(X = x)$$

Alternatively,

$$V(X) = E[(X - E(X))^2] = E[X^2 - 2XE(X) + E(X)^2]$$

$$= E(X^2) - 2E(X)E(X) + [E(X)]^2$$

$$= E[(X)^2] - [E(X)]^2$$

The expected values are sums/integrals, so whatever we are allowed to do with constants and brackets for sums and integrals, we are also allowed to do with expected values. That is why if a and b are constants, $E[aX + b] = aE(X) + b$. Oh, and $E(X)$ itself is also a constant — it is not random after it has been calculated — so $E(E(X)) = E(X)$.

Estimands and Estimators

Estimands are the things we want to estimate that are often represented by θ. Estimands θ are parameters, so they are (unknown) constants: $E(\theta) = \theta$ and $V(\theta) = 0$. Estimators are the formulas or functions used to estimate the estimand. A hat on θ often indicates them:- $\hat{\theta}$
So $E()$ of the random variable $X = (\hat{\theta} - \theta)$.

$$E(X) = E((\hat{\theta} - \theta)) = E(\hat{\theta}) - E(\theta) = E(\hat{\theta}) - E(\theta) = E(\hat{\theta}) - \theta$$

This quantity is called bias in statistics.

$$Bias = E(\hat{\theta}) - \theta$$

An unbiased estimator is one where

$$E(\hat{\theta}) = \theta$$

Bias Variance trade-off

If we have two models (estimators) with the same bias, the one with the smaller variance is always preferred (efficiency).
Using $X = (\hat{\theta} - \theta)$ into our variance formula: $V(X) = E[(X)^2] - [E(X)]^2$
it becomes $V(\hat{\theta} - \theta) = E[(\hat{\theta} - \theta)^2] - [E(\hat{\theta} - \theta)]^2$
using $V(\hat{\theta} - \theta) = V(\hat{\theta})$ as $V(\theta) = 0$

$$V(\hat{\theta}) = E[(\hat{\theta} - \theta)^2] - [E(\hat{\theta}) - E(\theta)]^2$$

using $E(\theta) = \theta$

$$E[(\hat{\theta} - \theta)^2] = [E(\hat{\theta}) - \theta]^2 + V(\hat{\theta})$$

using Bias $= E(\hat{\theta}) - \theta$

$$E[(\hat{\theta} - \theta)^2] = [Bias]^2 + V(\hat{\theta}) = Bias^2 + Variance$$

Bias Variance trade-off

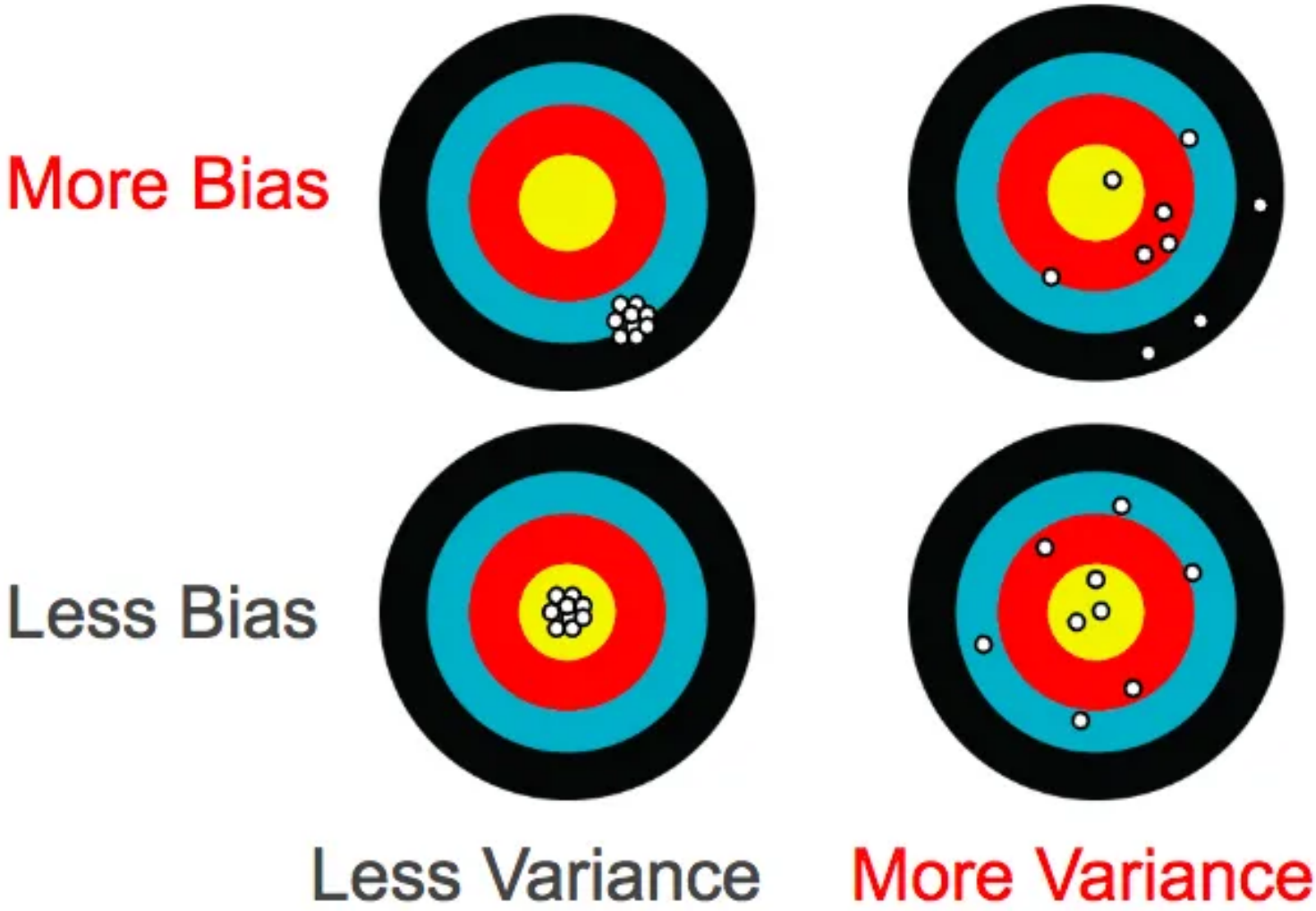
The value $E[(\hat{\theta} - \theta)^2]$ is nothing but mean squared error (MSE).

$$MSE = Bias^2 + Variance$$

A better model has a lower MSE. The absolute best model has a zero MSE: it makes no mistakes. That means it also has no bias and no variance. However, the reality is far from perfect.

One model can have a low bias but high variance, while the other can have low variance but high bias, and yet both can have the same MSE.

Bias Variance trade-off

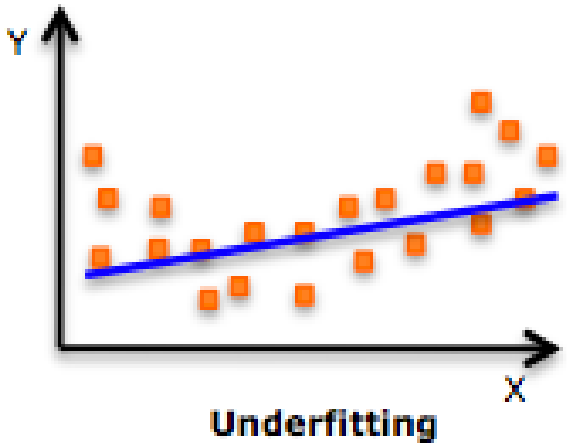


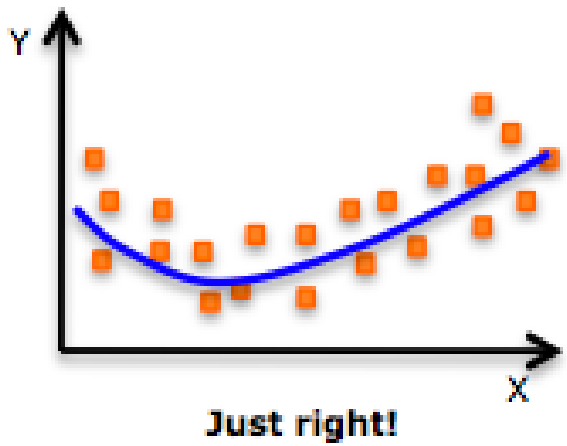
As you can infer from this representation, we ideally want a model with less variance and less bias.

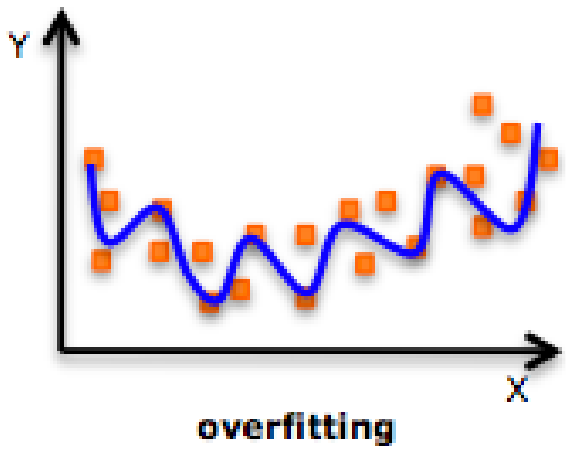
Bias Variance trade-off

We need more or more relevant data to get a better model else there is a limit to the extent to which the model can be good.

Underfitting and Overfitting are discussed well in earlier Lecture Notes. Here we will go through an example:-







Bias Variance trade-off

For the given data set instead of a linear relation; a polynomial relation can fit the data better. Thus, we can get a better training MSE than the true model's test MSE, but in doing so, we fit all the noise along with the signal. This is called overfitting. It is from making the model more complex. It has excellent MSE but high variance.

Knowing this, we develop a penalty for complexity in the scoring function. This will give us more complexity only if it improves the fit by at least a certain amount. This is called regularisation.

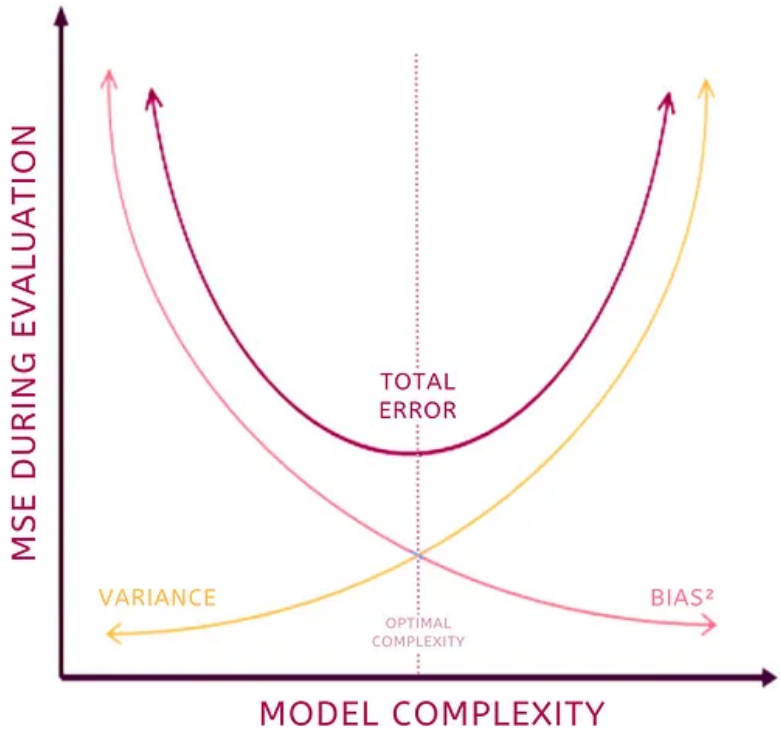
If we regularise too much, demanding less complexity than necessary, we get an oversimplified model with a high bias with an excellent MSE. This is called underfitting.

This trade-off is called the bias-variance trade-off.

Bias Variance trade-off

Learnt from Bias-Variance trade-off :-

- When you get an excellent model performance score, it can be due to underfitting or overfitting. Only validation and testing can tell.
- Training and actual performance is not the same (can vary alot).
- The goldilocks model is where we can only improve bias by hurting standard deviation proportionately more, and vice versa. That is where we stop. You made things as good as they can be!



Why Regularization?

The performance of a machine learning model degrades due to overfitting or underfitting. Underfitting happens if our model is simple or needs to be trained longer.

Over fitting happens due to a complex model that is learning noise. Here Regularization comes into the picture as it stops over fitting.

Before coming into regularisation, we need to know how to detect overfitting. So to do this, we use a technique named Cross-Validation.

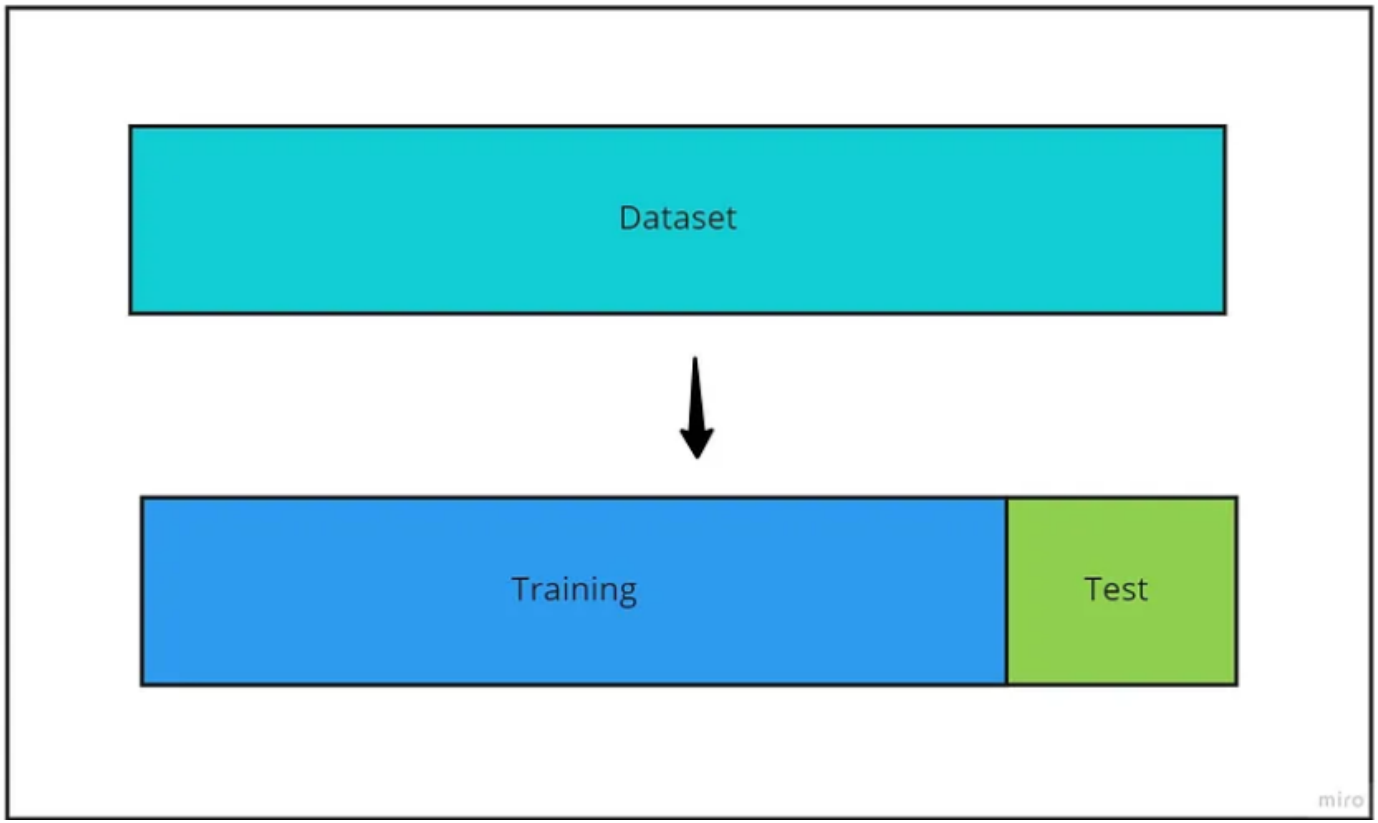
Cross-validation

Cross-validation is a technique which helps us evaluate a machine-learning model and test its performance. It has many types. It can assess the quality of a predictive model and how the model can generalise for an independent data set that the model is yet to see.

To validate, we often use a technique which is simple to use and is a very common one.

Hold-out technique

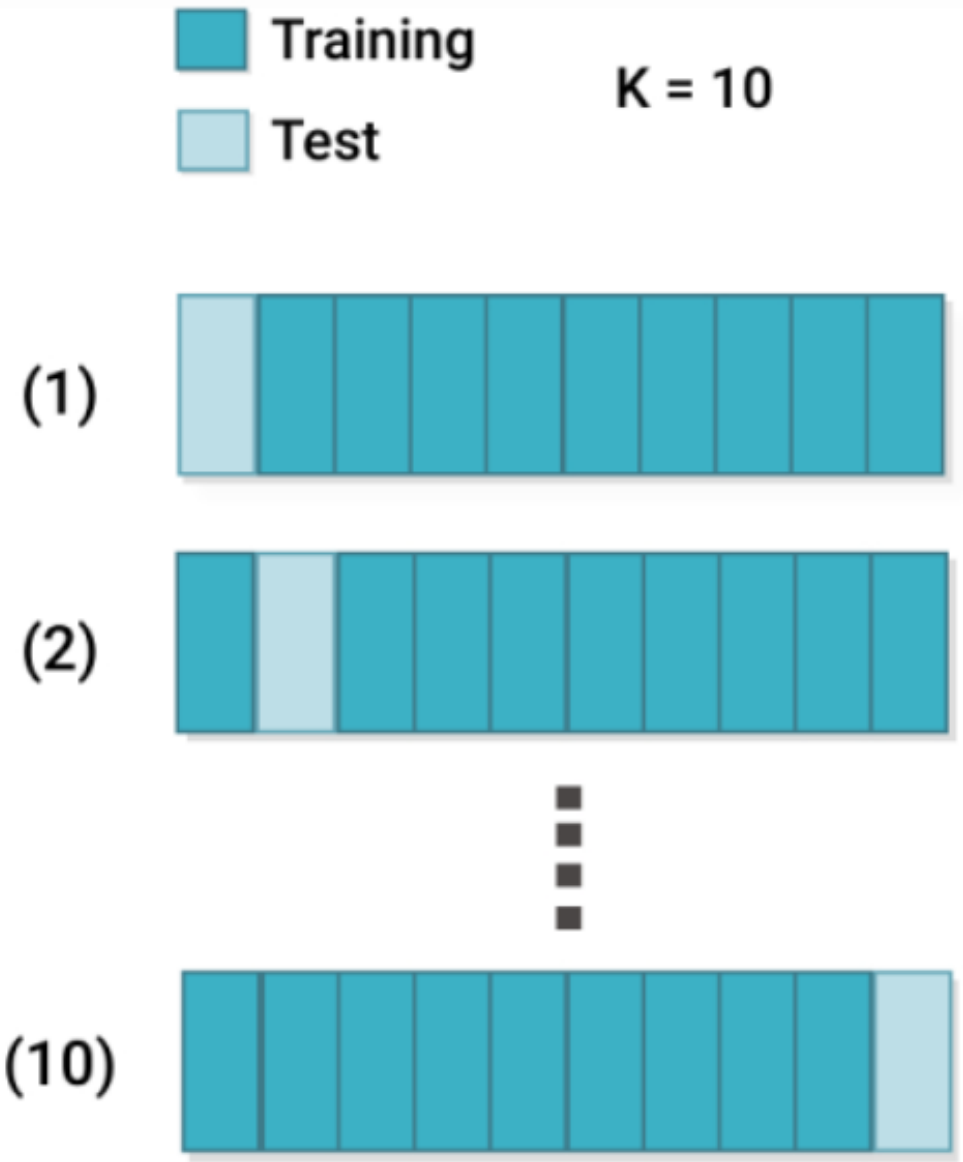
The algorithm of the hold-out technique break the dataset into the training set and the test set. Usually, 80% of the dataset is assigned to the training set, and 20% is to the test set, but the splitting depends on us. Then we train the model on the training set and validate it on the test set.



k-fold technique

We also use another k-fold technique: This minimises the disadvantages of the hold-out method.
We choose a number of k folds. Usually, k is 5 or 10, but it can be any number less than the length. Then we split the data set into k equal parts.
We choose k – 1 folds as the training set. The remaining fold will be the test set.
We then train it on the training set. On each cross-validation iteration, we must train a new model independently of the model trained on the previous iteration. We will validate using the test set and save the result.

k-fold technique



What is Regularisation?

The literal meaning of what we understand by regularisation is to make something regular. In the ML context, we make something regular by adding information and creating a solution that prevents overfitting. We use regularisation by adding some bias into our model to prevent it from overfitting. The "something" we are making regular in our ML context is the "objective function", something we try to minimise during the optimisation problem. To get our model's "best" implementation, we use an optimisation algorithm to identify the set of inputs that maximises – or minimise – the objective function. In ML we want to minimise the objective function to lower the error. So we call the objective function "The loss function". We mainly use the gradient descent algorithm to implement the best objective function.

What is Regularisation?

Before discussing the techniques properly, we discuss a bit about linear regression.
A linear model is represented as:

$$\hat{y}_i = \sum_{j=1}^{n} (x_{ij} * \beta_j) + \beta_0, \quad i = 1, \ldots k$$

β_j = weights,
β_0 = the bias term,
$\hat{Y}_i$ =the predicted value for the ith data point x_{ij}. In the above equation, n is the number of features, and k is the number of data points.

What is Regularisation?

The linear regression model computes the unknown weights and bias terms by minimising the sum of squared error (SSE) between y_i the ith observation and $\hat{Y}_i$ the value predicted by the model for the ith data point.

$$SSE = \sum_{i=1}^{k} (\hat{y}_i - y_i)^2 = \sum_{i=1}^{k} \left(\sum_{j=1}^{n} (x_{ij} * \beta_j + \beta_0) - y_i \right)^2$$

What is Regularisation?

Let X be a $k \times n+1$ matrix consisting of x_{ij} and an extra column of 1's.

$$X = \begin{bmatrix} x_{11} & \dots & x1n & 1 \\ \vdots & \ddots & \vdots & \\ x_{k1} & \dots & x_{kn} & 1 \end{bmatrix}$$

The observations are represented by y, a vector of length k.

$$y = \begin{bmatrix} y_1 \\ \vdots \\ y_k \end{bmatrix}$$

What is Regularisation?

The vector of length $n+1$ represents the unknown weights β_j and β_0.

$$\beta = \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_n \\ \beta_0 \end{bmatrix}$$

The expression for SSE can be rewritten in a matrix form as:

$$\hat{y} = X\beta$$

$$SSE = (y - \hat{y})^T (y - \hat{y}) = \left(y^T y - 2\beta^T X^T y + \beta^T X^T X \beta \right)$$

$$\beta = \left(X^T X \right)^{-1} X^T y$$

Regularisation Techniques

There are two types of regularisation techniques:

- Lasso Regularization(L1)
- Ridge Regularization(L2)

If we use the conditions of lasso and ridge together, we call it Elastic Net Regularization.
We also use a regularisation technique that is called Dropout Technique. We will not discuss this technique in detail. However, the idea of the dropout regularisation technique is to randomly make zero some elements of the input tensor with probability p, where p is a hyperparameter. We use this technique in neural networks.

Regularisation Techniques

The cost function of the two regularisation techniques:

L1

$$\sum^{N}(y_i - \sum^{M}(x_{ij}\beta_j)^2 + \lambda \sum^{M}(|\beta_j|))$$

L2

$$\sum^{N}(y_i - \sum^{M}(x_{ij}\beta_j)^2 + \lambda \sum^{M}(\beta_j^2))$$

Lasso Regularization or L1 regularisation: It shrinks the parameters towards 0. We assign a feature with a 0 weight; we multiply the feature values by 0, which returns 0, eradicating the significance of that feature. This means we can eliminate the less important features, simplifying our model.

We are penalising the absolute value of the weights. Lambda is the tuning parameter. RSS is the shrinkage quantity,

$$\text{RSS} = \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2$$

$$\sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j| = \text{RSS} + \lambda \sum_{j=1}^{p} |\beta_j|$$

Regularisation Techniques

This variation differs from ridge regression only in penalising the high coefficients. It uses $|\beta_j|$ (modulus) instead of squares of β. In L1, β_j can be 0 if the respective feature did not contribute enough to the overall model.
Ridge or L2 regularisation: The difference from lasso here is the shrinkage quantity; we use the L2 norm.

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}\beta_j^2 = \mathrm{RSS} + \lambda\sum_{j=1}^{p}\beta_j^2$$

λ is the tuning parameter by which we know how much it penalises the flexibility of our model. The increase in flexibility of a model is represented by an increase in its coefficients. These coefficients must be small if we want to minimise the above function.

Regularisation Techniques

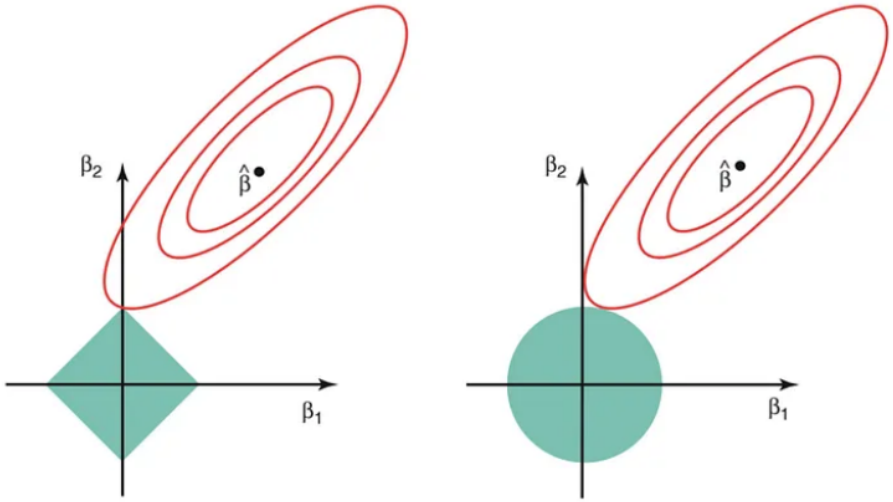
When $\lambda = 0$, the penalty term has no effect, and the estimates produced by ridge regression will be equal to least squares. However, as $\lambda \to \infty$, the impact of the shrinkage penalty grows, and the ridge regression coefficient estimates will approach zero. As can be seen, selecting a good value of λ is critical. Cross-validation comes in handy for this purpose.

Some comparison

We think of ridge regression as solving an equation where the summation of squares of coefficients is less than or equal to s. Lasso can be interpreted as an equation where the summation of the modulus of coefficients is less than or equal to s, and s is a constant for each shrinkage factor λ. These equations are also referred to as constraint functions. The ridge regression is expressed by $\beta_1^2 + \beta_2^2 \leq s$, and for the lasso, the equation becomes $|\beta_1| + |\beta_2| \leq s$.

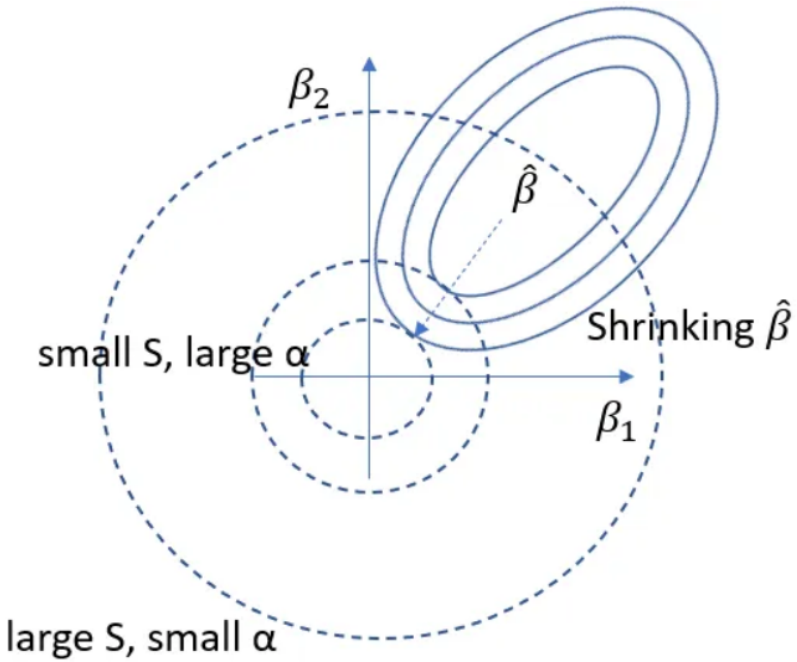
Some comparison

This is the SSE contours for L2 and L1 regularisations:



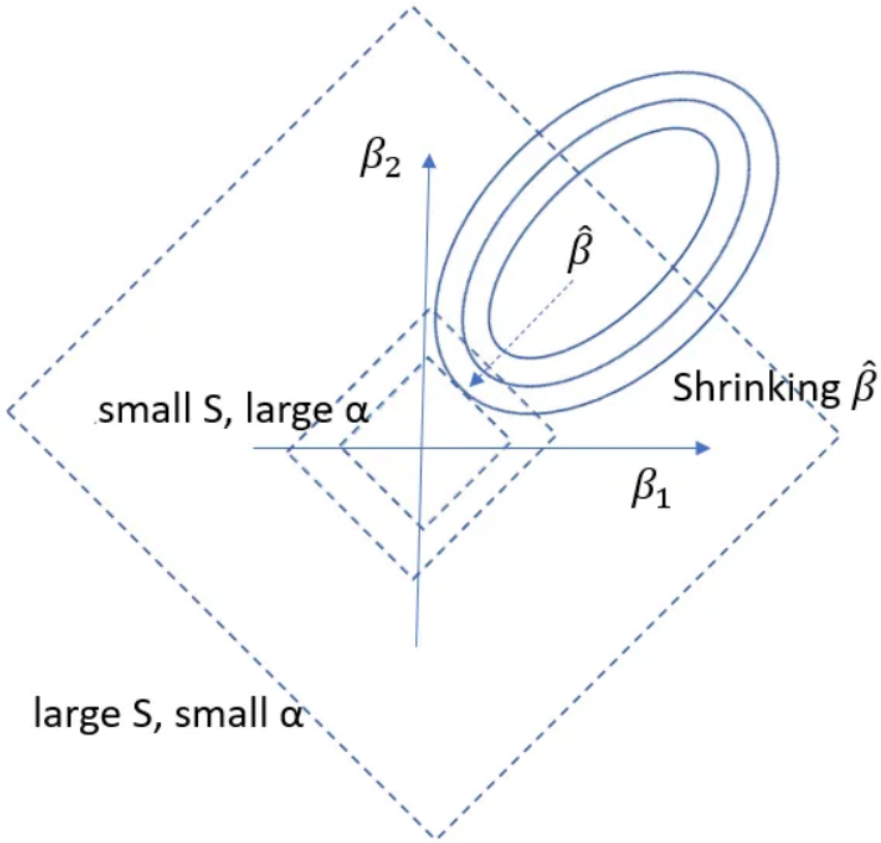
Some comparison

The weightage shrinkage for L2 Regularization is given below.



Some comparison

The weightage shrinkage for L1 Regularization is given below.



Some comparison

Since ridge regression has a circular constraint with no sharp points, this intersection will not generally occur on an axis. So the ridge regression coefficient estimates will be exclusively non-zero. However, the lasso constraint has corners at each of the axes, so the ellipse will often intersect the constraint region at an axis. When this occurs, some of the coefficients become zero.

References

- Is There Always a Tradeoff Between Bias and Variance? — by Cassie Kozyrkov — Feb, 2023 — Towards Data Science (medium.com)
- What is bias? The amazing thing about AI is just how... — by Cassie Kozyrkov — Towards Data Science
- Were 21% of New York City residents really infected with the novel coronavirus? — by Cassie Kozyrkov — Towards Data Science
- Machine learning — Is the emperor wearing clothes? — by Cassie Kozyrkov — Medium
- Why is MSE $= Bias^2 + Variance$?. Introduction to "good" statistical... — by Cassie Kozyrkov — Towards Data Science
- Overfitting, Underfitting, and Regularization — by Cassie Kozyrkov — Feb, 2023 — Towards Data Science
- The Bias-Variance Tradeoff, Explained — by Cassie Kozyrkov — Feb, 2023 — Towards Data Science
- Overfitting and Underfitting (kharshit.github.io)

References

- Subhankar Sir's Lecture CS458
- https://builtin.com/data-science/l2-regularization
- https://towardsdatascience.com/regularization-in-machine-learning-6fbc4417b1e5
- https://towardsdatascience.com/understanding-8-types-of-cross-validation-80c935a4976d
- https://towardsdatascience.com/cross-validation-705644663568
- https://towardsdatascience.com/regularization-for-machine-learning-67c37b132d61
- https://towardsdatascience.com/regularization-what-why-when-and-how-d4a329b6b27f
- https://towardsdatascience.com/regularization-in-machine-learning-76441ddcf99a
- https://towardsdatascience.com/regularization-in-machine-learning-7f85f64ff8e8