Ridge and Lasso

ML as optimization problem : Ridge & Lasso

- Every machine learning problem is basically an **optimization problem**
- That is, you wish to find either a maximum or a minimum of a specific function
- The function that you want to optimize is usually called the **loss function** (or **cost function**)
- The loss function is defined for each machine learning algorithm you use, and this is the main metric for evaluating the accuracy of your trained model.

An example of our dataset:

House size (X)	House price (Y)
50	102
70	127
32	65
68	131
93	190

An Example

For the house price prediction example, after the model is trained, we are able to predict new house prices based on their features. For each house price we predict, denoted as $\hat{Y}_i$, and the actual house price Y_i we can calculate the loss by:

$$l_i = (\hat{Y}_i - Y_i)^2$$

This is the most basic form of a loss for a specific data-point, That is used mostly for **linear regression** algorithms. The loss function as a whole can be denoted as:

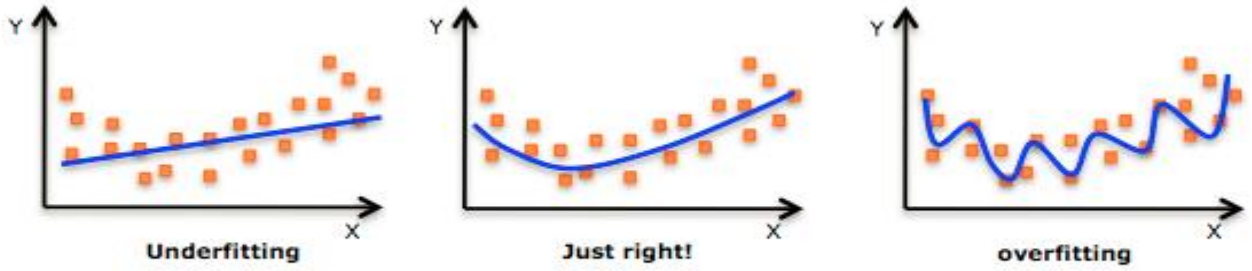
$$L = \sum(\hat{Y}_i - Y_i)^2$$

Which simply defines that our model's loss is the sum of distances between the house price we've predicted and the **ground truth**. This loss function, in particular, is called **quadratic loss** or **least squares**. We wish to **minimize** the loss function (L) as much as possible so the prediction will be as close as possible to the ground truth.

If you followed me up until now, you are familiar with the basic concept of every practical machine learning problem. **Remember,** every machine learning algorithm defines its own loss function according to its goal in life.

Contd…

- Having more features may seem like a perfect way for improving the accuracy of our trained model (reducing the loss) – because the model that will be trained will be more flexible and will take into account more parameters. On the other hand, we need to be extremely careful about overfitting the data. As we know, every dataset has noisy samples. The inaccuracies can lead to a low-quality model if not trained carefully. The model might end up memorizing the noise instead of learning the trend of the data.
- A visual example of a non-linear over-fitted model:



Avoiding Overfitting

- The main concept behind avoiding over-fit is simplifying the models as much as possible. Simple models do not (usually) over-fit. On the other hand, we need to pay attention the to gentle trade-off between overfitting and under-fitting a model.
- One of the most common mechanisms for avoiding over-fit is called regularization.

Concept of Regularization

- Model Ready
- Predicted the output
- Why study regularization?
- Is it required?

“Everything should be made simple as possible, but not simpler – Albert Einstein”

Concept of Regularization

- In regularization, what we do is normally we keep the same number of features, but reduce the magnitude of the coefficients.

- This approach involves fitting a model involving all *p* predictors. However, the estimated coefficients are shrunken towards zero relative to the least squares estimates. This shrinkage (also known as *regularization*) has the effect of reducing variance.
  Depending on what type of shrinkage is performed, some of the coefficients may be estimated to be exactly zero.

Ridge

Regularized machine learning model, is a model that its loss function contains another element that should be minimized as well.

Let's see an example:

$$L = \sum(\hat{Y}_i - Y_i)^2 + \lambda \sum \beta^2 \quad \textbf{(Shrinkage penalty)}$$

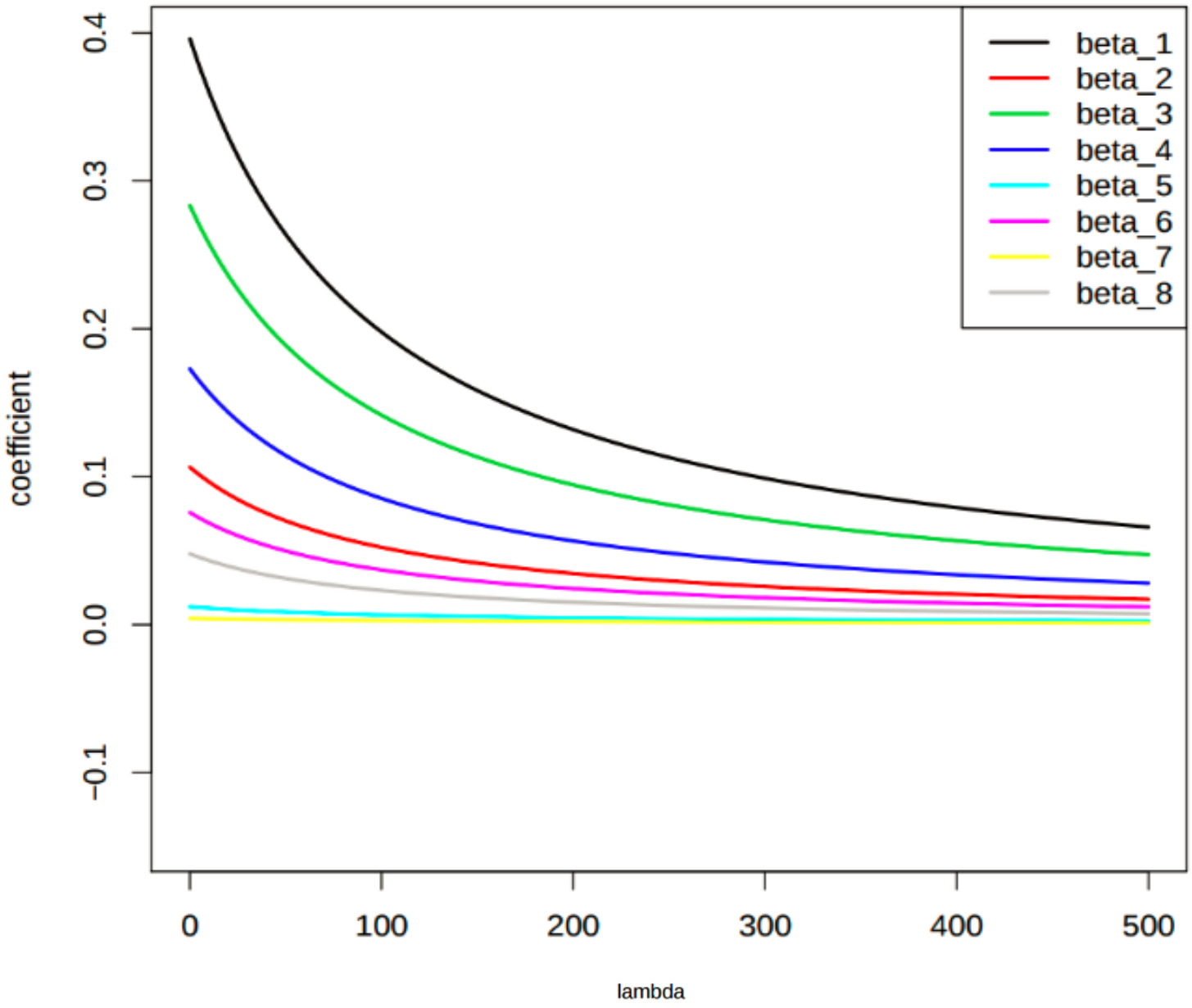
- This loss function includes two elements.
- The first one is the one you've seen before – the sum of distances between each prediction and its ground truth.
- The second element though, a.k.a the regularization term, might seem a bit bizarre. It sums over squared β values and multiplies it by another parameter λ.
- The reason for doing that is to "punish" the loss function for high values of the coefficients β.
- As aforesaid, simple models are better than complex models and usually do not over-fit.
- Therefore, we need to try and simplify the model as much as possible. Remember that our goal of the iterative process is to minimize the loss function. By punishing the β values we add a constraint to minimize them as much as possible.
- There is a gentle trade-off between fitting the model, but not overfitting it.

- This approach is called **Ridge regression**.

Ridge

- Ridge regression is an extension for linear regression. It's basically a regularized linear regression model. The λ parameter(>=0) is a scalar that should be learned as well, using a method called **cross validation**

- A super important fact we need to notice about ridge regression is that it enforces the βcoefficients to be lower, but it **does not** enforce them to be zero. That is, it will not get rid of irrelevant features but rather **minimize their impact on the trained model**.


beta_1
beta_2
beta_3
beta_4
beta_5
beta_6
beta_7
beta_8
coefficient
0.4
0.3
0.2
0.1
0.0
−0.1
0
100
200
300
400
500
lambda

Advantage of Ridge

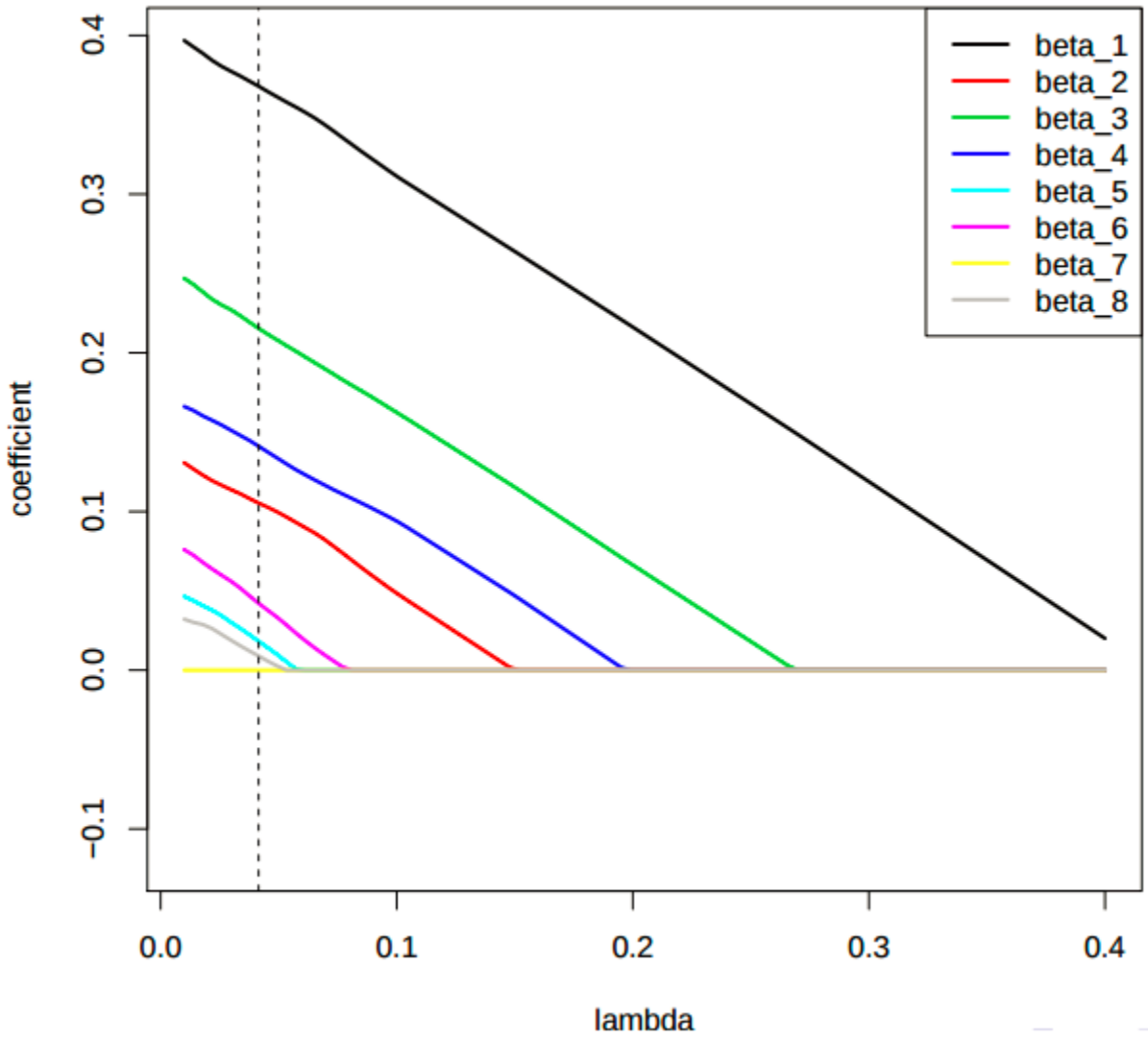
- It shrinks the parameters, therefore it is mostly used to prevent multicollinearity.
- It reduces the model complexity by coefficient shrinkage
- It uses L2 regularization technique

Lasso(least absolute shrinkage and selection operator)

- Lasso is another extension built on regularized linear regression, but with a small twist. The loss function of Lasso is in the form:

$$L = \sum(\hat{Y}i - Yi)2 + \lambda \sum |\beta|$$

- The only difference from Ridge regression is that the regularization term is in absolute value.
- But this difference has a huge impact on the trade-off we've discussed before.
- Lasso method overcomes the disadvantage of Ridge regression by not only punishing high values of the coefficients β but actually setting them to zero if they are not relevant.
- Therefore, you might end up with fewer features included in the model than you started with, which is a huge advantage.


beta_1
beta_2
beta_3
beta_4
beta_5
beta_6
beta_7
beta_8
coefficient
0.4
0.3
0.2
0.1
0.0
−0.1
0.0
0.1
0.2
0.3
0.4
lambda

Advantage of Lasso

- It uses L1 regularization technique
- It is generally used when we have more number of features, because it automatically does feature selection.