Open in app

Towards Data Science
610K Followers

You have **1** free member-only story left this month. Upgrade for unlimited access.

Wenwei Xu · Aug 22, 2019 · 4 min read ★

What’s the difference between Linear Regression, Lasso, Ridge, and ElasticNet in sklearn?

Image by Free-Photos from Pixabay

What’s the difference between them?

Lasso, Ridge and ElasticNet are all part of the Linear Regression family where the x (input) and y (output) are assumed to have a linear relationship. In sklearn, LinearRegression refers to the most ordinary least square linear regression method without regularization (penalty on weights) . The main difference among them is whether the model is penalized for its weights. For the rest of the post, I am going to talk about them in the context of scikit-learn library.

Linear regression (in scikit-learn) is the most basic form, where the model is not penalized for its choice of weights, at all. That means, during the training stage, if the model feels like one particular feature is particularly important, the model may place a large weight to the feature. This sometimes leads to overfitting in small datasets. Hence, following methods are invented.

Lasso is a modification of linear regression, where the model is penalized for the sum of absolute values of the weights. Thus, the absolute

Open in app

$$\frac{1}{2m}\sum_{i=1}(y-Xw)^2+alpha\sum_{j=1}\left|w_j\right|$$

As you see, Lasso introduced a new hyperparameter, *alpha*, the coefficient to penalize weights.

Ridge takes a step further and penalizes the model for the sum of squared value of the weights. Thus, the weights not only tend to have smaller absolute values, but also really tend to penalize the extremes of the weights, resulting in a group of weights that are more evenly distributed. The objective function becomes:

$$\sum_{i=1}^{n}(y-Xw)^2+alpha\sum_{j=1}^{p}{w_j}^2$$

ElasticNet is a hybrid of Lasso and Ridge, where both the absolute value penalization and squared penalization are included, being regulated with another coefficient l1_ratio:

$$\frac{1}{2m}\sum_{i=1}^{m}(y-Xw)^2+alpha*ratio*\sum_{j=1}^{p}\left|w_j\right| \ +0.5*alpha*(1-ratio)*\sum_{j=1}^{p}{w_j}^2$$

Are your data Scaled yet?

As you can see in these equations above, the weights penalization are summed together in the loss function. Suppose we have a feature *house_size* in the 2000 range, while another feature *num_bedrooms* in the range of 3, then we would expect that the weight for house_size may be naturally smaller than the weight for *num_bedrooms*. In such case, penalizing each feature’s weight the same way becomes inappropriate. Hence, it is important to scale or normalize the data before entering them to the models. **A quick note, the default setting in sklearn for these model set ‘normalize’ to false.** You will either want to turn the ‘normalize’ to ‘on’, or use ScandardScaler to scale the data. Typically, use ScandardScaler is a good practice because you may want to scale your testing data using the same scale.

When to use which?

There are a few things to remember:

(1) **sklearn’s algorithm cheat sheet suggests you to try Lasso, ElasticNet, or Ridge when you data-set is smaller than 100k rows.** Otherwise, try SGDRegressor.

(2) **Lasso and ElasticNet tend to give sparse weights (most zeros)**, because the l1 regularization cares equally about driving down big weights to small weights, or driving small weights to zeros. If you have a lot of predictors (features), and you suspect that not all of them are that important, Lasso and ElasticNet may be really good idea to start with.

Open in app

relevant for predictions, try Ridge as a good regularized linear regression method.

(4) **You will need to scale your data before using these regularized linear regression methods.** Use StandardScaler first, or set 'normalize' in these estimators to 'True'.

Buy me a coffee

Sign up for The Variable

By Towards Data Science

Every Thursday, the Variable delivers the very best of Towards Data Science: from hands-on tutorials and cutting-edge research to original features you don't want to miss. Take a look.

Get this newsletter

Emails will be sent to jiaren.kjr@gmail.com.
Not you?