

LASSO Regression

Apply regularization with LASSO regression.

Chapter Goals:

- Learn about sparse linear regression via LASSO

A. Sparse regularization

While ridge regularization uses an L2 norm penalty term, another regularization method called LASSO ([https://en.wikipedia.org/wiki/Lasso_\(statistics\)](https://en.wikipedia.org/wiki/Lasso_(statistics))) uses an L1 norm (<http://mathworld.wolfram.com/L1-Norm.html>) for the weights penalty term. Specifically, LASSO regularization will find the optimal weights to minimize the following quantity:

$$\alpha ||w||_1 + \sum_{i=1}^n (\mathbf{x}_i \cdot w - y_i)^2$$

where $||w||_1$ represents the L1 norm of the weights.

LASSO regularization tends to prefer linear models with fewer parameter values. This means that it will likely zero-out some of the weight coefficients. This reduces the number of features that the model is actually dependent on (since some of the coefficients will now be 0), which can be beneficial when some features are completely irrelevant or duplicates of other features.

In scikit-learn, we implement LASSO using the `Lasso` (https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Lasso.html#sklearn.linear_model.Lasso) object, which is part of the `linear_model` module. Like the `Ridge` object, it takes in the model's α value with the `alpha` keyword argument (default is 1.0).

The code below demonstrates how to use the `Lasso` object on a dataset with 150 observations and 4 features.



```
1 # predefined dataset
2 print('Data shape: {}'.format(data.shape))
3 print('Labels shape: {}'.format(labels.shape))
4
5 from sklearn import linear_model
6 reg = linear_model.Lasso(alpha=0.1)
7 reg.fit(data, labels)
8 print('Coefficients: {}'.format(repr(reg.coef_)))
9 print('Intercept: {}'.format(reg.intercept_))
10 print('R2: {}'.format(reg.score(data, labels)))
```



Output

1.273s

Data shape: (150, 4)

Labels shape: (150,)

Coefficients: array([0. , -0. , 0.40830957, 0.])

Intercept: -0.534699558318563

R2: 0.895831189504504

In the example above, note that a majority of the weights are 0, due to the LASSO sparse weight preference.

There is also a cross-validated version in the form of the `LassoCV` (https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LassoCV.html#sklearn.linear_model.LassoCV) object, which works in essentially the same way as the `RidgeCV` object.

Time to Code!

The coding exercise in this chapter uses the `Lasso` object of the `linear_model` module (imported in backend) to complete the `lasso_reg` function.



The function will fit a LASSO regression model to the input data and labels. The α hyperparameter for the model is provided to the function via the `alpha` input argument.

Set `reg` equal to `linear_model.Lasso` initialized with `alpha` for the `alpha` keyword argument.

Call `reg.fit` with `data` and `labels` as the two input arguments. Then return `reg`.

```
1 def lasso_reg(data, labels, alpha):
2     # CODE HERE
3     pass
```



Solution



```
1 def lasso_reg(data, labels, alpha):
2     reg = linear_model.Lasso(alpha=alpha)
3     reg.fit(data, labels)
4     return reg
5
6
```

[← Back](#)

Ridge Regression

[Next →](#)

Bayesian Regression



Mark as Completed

43% completed, meet the [criteria](#) and claim your course certificate!

[Buy Certificate](#)



Report
an Issue



Ask a Question

(https://discuss.educative.io/tag/lasso-regression__data-modeling-with-scikit-learn__machine-learning-for-software-engineers)

