Penalized Regression in R

by **Jason Brownlee** on July 25, 2014 in **Uncategorized**

3 2 13

In this post you will discover 3 recipes for penalized regression for the R platform.

You can copy and paste the recipes in this post to make a jump-start on your own problem or to learn and practice with linear regression in R.

Penalized Regression
Photo by Bay Area Bias, some rights reserved

Each example in this post uses the longley dataset provided in the datasets package that comes with R. The longley dataset describes 7 economic variables observed from 1947 to 1962 used to predict the number of people employed yearly.

Ridge Regression

Ridge Regression creates a linear regression model that is penalized with the L2-norm which is the

sum of the squared coefficients. This has the effect of shrinking the coefficient values (and the complexity of the model) allowing some coefficients with minor contribution to the response to get close to zero.

```
# load the package
library(glmnet)
# load data
data(longley)
x <- as.matrix(longley[,1:6])
y <- as.matrix(longley[,7])
# fit model
fit <- glmnet(x, y, family="gaussian", alpha=0, lambda=0.001)
# summarize the fit
summary(fit)
# make predictions
predictions <- predict(fit, x, type="link")
# summarize accuracy
rmse <- mean((y - predictions)^2)
print(rmse)
```

Learn about the **glmnet** function in the glmnet package.

Least Absolute Shrinkage and Selection Operator

Least Absolute Shrinkage and Selection Operator (LASSO) creates a regression model that is penalized with the L1-norm which is the sum of the absolute coefficients. This has the effect of shrinking coefficient values (and the complexity of the model), allowing some with a minor affect to the response to become zero.

```
# load the package
library(lars)
# load data
data(longley)
x <- as.matrix(longley[,1:6])
y <- as.matrix(longley[,7])
# fit model
fit <- lars(x, y, type="lasso")
# summarize the fit
summary(fit)
# select a step with a minimum error
best_step <- fit$df[which.min(fit$RSS)]
# make predictions
predictions <- predict(fit, x, s=best_step, type="fit")$fit
# summarize accuracy
rmse <- mean((y - predictions)^2)
print(rmse)
```

Learn about the **lars** function in the lars package.

Elastic Net

Elastic Net creates a regression model that
has the effect of effectively shrinking coeffi
coefficients to zero (as in LASSO).

```
# load the package
```

```
library(glmnet)
# load data
data(longley)
x <- as.matrix(longley[,1:6])
y <- as.matrix(longley[,7])
# fit model
fit <- glmnet(x, y, family="gaussi
# summarize the fit
summary(fit)
# make predictions
predictions <- predict(fit, x, type="link")
# summarize accuracy
rmse <- mean((y - predictions)^2)
print(rmse)
```


Learn about the **glmnet** function in the glmnet package.

Summary

In this post you discovered 3 recipes for penalized regression in R.

Penalization is a powerful method for attribute selection and improving the accuracy of predictive models. For more information see Chapter 6 of Applied Predictive Modeling by Kuhn and Johnson that provides an excellent introduction to linear regression with R for beginners.

About Jason Brownlee

The editor-in-chief at MachineLearningMastery.com. Jason is a husband, father, researcher, author, professional programmer and a machine learning practitioner. Learn more about him.

View all posts by Jason Brownlee →

‹ Linear Regression in R

Non-Linear Regression in R ›

5 Responses to *Penalized Regression in R*

Hrvoje July 25, 2014 at 10:53 pm #

REPLY

Nice article, but first and third code are the same 🙂 What's the difference?

jasonb July 26, 2014 at 7:40 am #

REPLY

Almost. Note the value of alpha (the elastic net mixing parameter).
A great thing about the glmnet function is that it can do ridge, lasso and a hybrid of both. In the first example, we have used glmnet with an alpha of 0 which results in ridge regression (only L2). If alpha was set to 1 it would be lasso (only L1). Note in the third example that alpha is set to 0.5, this is the elastic net mixture of L1 and L2 at a 50% mixing.
I hope that is clearer.

Hrvoje July 26, 2014 at 11:47 pm #

REPLY

It' s much clearer now. Tnx a lot 🙂

TropoSco August 2, 2014 at 7:50 pm #

REPLY

Thanks for the post,

I was wondering if you knew the differences (computational and statistical performances) between using the lars package and the glmnet one with alpha=1 for performing a LASSO regression ?

Thank you for your time and keep up the good work!

JOY May 12, 2015 at 10:04 pm #

REPLY

nice article.I want to know how to use R for regression analysis.you are following the step by step method to do it to my e-mail.I have R package already on my Laptop

Leave a Reply

Name (required)

Email (will not be published) (required)

Website

SUBMIT COMMENT

Welcome!

Hi, my name is Jason and welcome to Machine Learning Mastery where we make programmers awesome at machine learning.

If you have any questions or you just want to chat about machine learning, send me an email.

Free Guide to the Best Resources

Discover the best videos, books, courses and MUCH MORE in my Machine Learning Resource Guide.

Get your copy right now!

Download For Free

Are You a Developer?

Do You want to FINALLY get started with

POPULAR

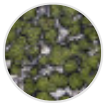
A Tour of Machine Learning Algorithms
NOVEMBER 25, 2013

How to Run Your First Classifier in Weka
FEBRUARY 17, 2014

Tutorial To Implement k-Nearest Neighbors in Python From Scratch
SEPTEMBER 12, 2014

How To Implement Naive Bayes From Scratch in Python
DECEMBER 8, 2014

Machine Learning for Programmers: Leap from developer to machine learning practitioner
AUGUST 17, 2015



Discover Feature Engineering, How to Engineer Features and How to Get Good at It
SEPTEMBER 26, 2014

Find Your Machine Learning Tribe: Get Started And Avoid Getting The Wrong Advice
AUGUST 31, 2015

Best Machine Learning Resources for Getting Started
NOVEMBER 27, 2013

4 Self-Study Machine Learning Projects
JANUARY 3, 2014

Feature Selection with the Caret R Package
SEPTEMBER 22, 2014

Looking for something? Search!

Search...

Privacy | Contact | About