A comparison between data thinning and traditional data splitting methods

Hansen Zhang, Ethan Ancell
hansez@uw.edu, ancell@uw.edu

Advisor: Emanuela Furfaro
efurfaro@uw.edu

May 2024

1 Introduction

"Double Dipping" is the act of using the same data to select a regression model and test/validate the model [Kriegeskorte 2009, Neufeld 2023]. Following on our LASSO regression, if we wish to perform inference for the selected coefficients, it becomes problematic because we cannot do inference using the same data that we selected our variables with. This will usually result in invalid inference.

We can solve this in two different ways. The first is splitting, which is the most common way to overcome double dipping and involves partitioning or "splitting" our data into two sets of observations: a training set and a test set. The other is thinning, a newer method for creating independent training and test sets based upon a resampling technique and membership within certain classes of distributions. This method works especially well when there are relatively few observations.

In this paper, we will be applying data thinning and subsequent LASSO variable selection on the Introduction to Statistical Learning (ISLR) 'Hitters' data set to take a look at how it works. Then, we will conduct a simulation study on the efficiency of inference for regression coefficients for LASSO selected thinned versus split data through an evaluation of their average 95 % confidence interval lengths.

2 Methodology

2.1 Review of data splitting methods

Data splitting is a universal and important task in creating train and test splits when carrying out statistical learning. In order to make sure the model is generalized and not overfit to some data's random noise, we must have a data set to train on, and another set containing new data to test the accuracy of the model's prediction against. Hence, data splitting is an integral aspect of evaluating and improving prediction accuracy.

2.1.1 Data splitting methodology

Data splitting is the process of separating observed data (their variables) randomly into two groups. This split can take on any proportion, but the convention is that greater than 50% of the data is used on training. The most popular splits are 80/20, 75/25 or 50/50. A simple example of data splitting using the 'mt cars' data set from base r is as follows:

Listing 1: Example R code

```
data(mtcars)

smp_size <- floor(0.75 * nrow(mtcars))
set.seed(123)
train_ind <- sample(seq_len(nrow(mtcars)), size = smp_
    size)

train <- mtcars[train_ind, ]
test <- mtcars[-train_ind, ]
```

2.1.2 Data thinning

Data thinning is a new alternative to data splitting developed by Witten et al. Rather than splitting separate observations into distinct groups, we instead use convolution of random variables split individual variables, say X, into independent random variables, Xi^1 and Xi^2, each of which contributes its part to training or testing. We wish to make a comparison between data splitting and data thinning for multiple linear regression. Note that:

$$\begin{aligned} Y_i &= f(X_1, X_2, \dots, X_p) + \epsilon_i \\ &= \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + ... + \beta_p X_{ip} + \epsilon_i \end{aligned}$$

where Y_i is the observed outcome and X_{ij} the realization of the jth predictor of the ith data point:

$$d_i = (X_1, X_2, \dots, X_p, Y)$$

Notice, we can rewrite the above linear relationship as

$$Y_i = \beta^T X_i = \begin{bmatrix} \beta_1 & \beta_2 & ... & \beta_p \end{bmatrix} \begin{bmatrix} 1 \\ X_{i1} \\ X_{i2} \\ \vdots \\ X_{ip} \end{bmatrix} + \epsilon_i$$

The additional random noise, ϵ_i, is also a random variable, which we will assume is normally distributed with mean 0:

$$\epsilon \sim N(0, \sigma^2) \perp X_i$$

Thus, by the sum of independent normal variables, we have that:

$$Y_i = \beta^T X_i + \epsilon \sim N\left(\beta^T X_i, \sigma^2\right)$$

From Table 2 (Neufeld et al, 2023), we have that:

$$Y_i^{(tr)} | Y_i = y_i \sim N(\epsilon y_i, \epsilon(1-\epsilon)\sigma^2)$$

$$Y_i = Y_i^{(tr)} + Y_i^{(te)} \implies Y_i^{(te)} = y_i - Y_i^{(tr)}$$

Then it follows that:

$$Y_i^{(tr)} \sim N(\epsilon\beta^T X_i, \epsilon\sigma^2)$$

$$Y_i^{(te)} \sim N((1-\epsilon)\beta^T X_i, (1-\epsilon)\sigma^2)$$

where

$$Y_i^{(tr)} \perp Y_i^{(te)}$$

In the section 2.3, we apply this in R.

2.1.3 Data thinning for regression

By the derived results in the previous section, and the knowledge that Y_i^{tr} and Y_i^{te} by the definition of convolution, we can apply the method to R.

2.1.4 Efficiency of Inference on Beta

Does thinning or splitting result in more efficient inference?

To test this, we run a simulation in R.

Steps (for split and thinned, separately):

1. Fix arbitrary sparse beta (the vector of coefficients) and fix an arbitrary X
2. Sample an observed Y = (Y1, Y2, . . . , Yn) by simulating additive random noise
3. Perform variable selection using train set
4. Perform inference on only the LASSO-selected beta variables (95 % CI) using test set
5. Repeat steps 2-5 many times

2.2 Review of variable selection

Variable selection is an important method of making an n-dimensional multiple regression model more interpretable. Essentially, we want to select for and include only the predictors (variables) which are most well-correlated with our outcome variable. By doing so, the fitted regression function can have fewer variables and belong to a smaller dimension.

- LASSO

  Linear models are generally easiest to fit, but may not match the true, theoretical form of the function that we attempt to estimate. The advantage of a linear model lies in the fact that they require fewer parameters, and are therefore more interpret-able. The most common technique for fitting a linear relationship is least squares. However, there shrinkage methods which can generally make a model more interpretable. We will focus particularly on LASSO.

  The LASSO coefficient minimizes the quantity

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

  This is simply the residual sum of squares (RSS) with an additional penalty term called the L1 regularization.

  Through this method, LASSO performs variable selection in other words, it penalizes coefficient size by a factor λ. When $\lambda = 0$, the error function simply becomes that of the Least Squares method. The large the value of λ, the fewer parameters we will end up with.

2.3 Dataset introduction

The dataset we used for our tutorial is the 'Hitters' dataset from the ISLR2 library. Our choice of of outcome variable is the hitter's salary. The predictors are the other 19 variables.

Listing 2: Example R code

```
Hitters = na.omit(Hitters) #data frame
x <- model.matrix(Salary ~ ., Hitters)[, -1] #all
    variables except Salary
y <- log(Hitters$Salary) #Salary
n <- nrow(x)

# Before we do anything, we need to know sigma^2
sigma2_estimate <- mean(lm(y ~ x)$residuals^2)
```

```
# Now I want to separate the information into y_train and
      y_test,
# similar to what we do with data splitting.
eps <- 0.5

# Simulate Y_tr conditional on the observed y
y_train <- rnorm(n = n, mean = eps * y, sd = sqrt(eps *
    (1-eps) * sigma2_estimate))
y_test <- y - y_train
```

3 Results

All code used in our simulation can be found at https://github.com/hansez/thinvssplit
After running the simulation in R using 1000 repetitions, 50 predictors and 200 observations, we have that:

When doing selective inference naively (not using splitting or thinning) yields average coverage of 0.89, which disagrees with our nominal coverage rate of 0.95. (This is evidence that we have invalid inference if we don't split or thin) With thinning and splitting, the average coverage rate was 0.9466 and 0.9427 respectively (valid inference)

However, the average lengths of the confidence intervals from thinning and splitting differed greatly at 0.45 and 0.55 respectively. (We get more efficient inference with thinning.)

4 Conclusion

When our target of inference depends on our data, we need to use a information division technique like splitting or thinning to achieve valid inference. However... The results imply that data thinning is superior to data splitting when it comes to inferential efficiency.

5 References

James G, Witten D, Hastie T, Tibshirani R. An Introduction to Statistical Learning : With Applications in R. Springer; 2013.2.

Neufeld A, Dharamshi A, Gao LL, Witten D. Data thinning for convolution-closed distributions. arXiv.org.doi:https://doi.org/10.48550/arXiv.2301.072763.

Kriegeskorte N, Simmons WK, Bellgowan PSF, Baker CI. Circular analysis in systems neuroscience: the dangers of double dipping. Nature Neuroscience. 2009;12(5):535-540. doi:https://doi.org/10.1038/nn.2303