Selected Topics In Data Science

Bruce Campbell

2021-01-26

Chapter 1

Preface

This is the first installment on my promise to elucidate less popular topics in statistics and machine learning. I wrote this as a way to solidify my understanding of some of the topics that are treated here. Hopefully others will find value here.

Chapter 2

Introduction

"Where must we go, we who wander this wasteland, in search of our better selves." -The First History of Man

This is a living book. It's under development. We are using the **bookdown** package [Xie, 2020] in this book, which was built on top of R Markdown and **knitr** [Xie, 2015].

Chapter 3

On Model Averaging

Recall that we can break down model error into the bias an variance $bias(\hat{Y}) = E[\hat{Y} - E[Y]]$

If we are averaging models $i = 1, \cdots, k$ then

$\text{MSE}\left(\hat{Y}_i\right) = \left\{\text{bias}\left(\hat{Y}_i\right)\right\}^2 + \text{var}\left(\hat{Y}_i\right)$

Chapter 4

Sensitivity Analysis and Shapley Values

Global sensitivity analysis measures the importance of input variables to a function. This is an important task in quantifying the uncertainty in which target variables can be predicted from their inputs. Sobol indices [Saltelli and Sobol', 1995] are a popular approach to this. It turns out that there's a relationship between Sobol indices and Shapley values. We explore this relationship here and demonstrate their effectiveness on some linear and non-linear models.

4.1 Relationship between Sobol indices and Shapley values

Shapley values are based on $f(x) - E[f(x)]$ while Sobol indices decompose output variance into fractions contributed by the inputs. The Sobol index is a global measure of feature importance while Shapley values focus on local explanations although we could combine local Shapley values to achieve a global importance measure. Sobol indices are based on expectations and can be used for features not included in the model / function of interest. In

this way we could query for important features correlated with those that the model does use.

4.2 CRAN sensitivity package

```
library(ggplot2)
library(pander)
if(!require(sensitivity)){
    install.packages("sensitivity")
    library(sensitivity)
}
```

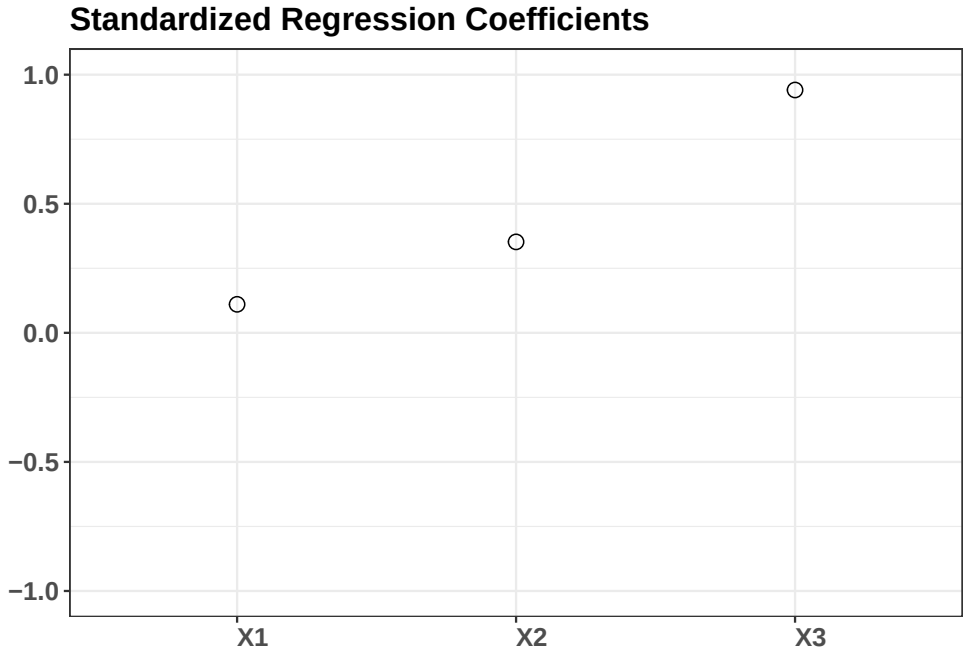
Standardized Regression Coefficients (SRC), or the Standardized Rank Regression Coefficients (SRRC), which are sensitivity indices based on linear or monotonic assumptions in the case of independent factors.

```
n <- 100
X <- data.frame(X1 = runif(n, 0.5, 1.5),
                X2 = runif(n, 1.5, 4.5),
                X3 = runif(n, 4.5, 13.5))

# linear model : Y = X1 + X2 + X3

y <- with(X, X1 + X2 + X3)


Z <- src(X, y, rank = FALSE, logistic = FALSE, nboot = 0, conf = 0.95)

pander(Z$SRC,caption = "Standardized Regression Coefficients ")
```

Table 4.1: Standardized Regression Coefficients

	original
X1	0.1104
X2	0.3524
X3	0.9407

```
ggplot(Z, ylim = c(-1,1))+ggtitle("Standardized Regression Coefficients")
```

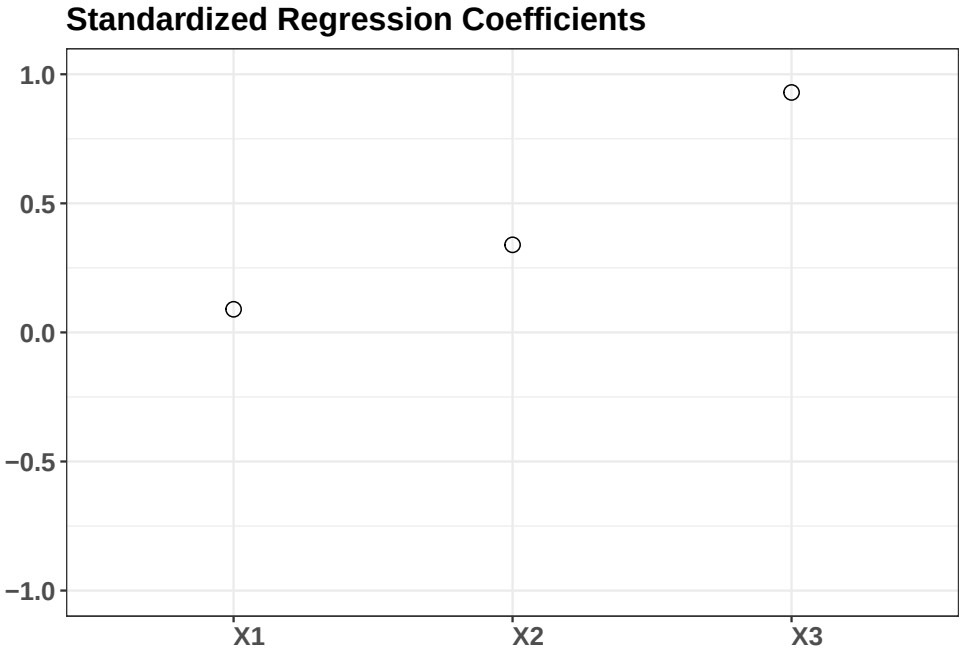



```
y <- with(X, X1 + X2 + X3)
y <- y + rnorm(nrow(X),0,1/2)
df<- data.frame(cbind(X,y))

Z <- src(X, y, rank = FALSE, logistic = FALSE, nboot = 0, conf = 0.95)

pander(Z$SRC,caption = "Standardized Regression Coefficients ")
```

Table 4.2: Standardized Regression Coefficients

	original
X1	0.08958
X2	0.3392
X3	0.9295

```
ggplot(Z, ylim = c(-1,1))+ggtitle("Standardized Regression Coefficients")
```




```
#lm.fit = lm(y ~ X1+X2+X3,data = df)
#summary(lm.fit)
#attach(df)
#plot(y, X1+X2+X3)
```

We see how the importance of X3 is ranked above X2 and likewise X2 is more important than X1. This is by design of the simulated data set. The standardized regression coefficients (beta coefficients) are calculated from that has been standardized, let's normalize and calculate the regression to

see if indeed that is the case.

```
dfs<- data.frame(scale(df,center = TRUE,scale = TRUE))
lm.fit = lm(y ~ X1+X2+X3,data = dfs)
summary(lm.fit)
```

```
##
## Call:
## lm(formula = y ~ X1 + X2 + X3, data = dfs)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.60361 -0.09522 -0.00626  0.08943  0.40125
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 3.054e-16  1.769e-02   0.000        1
## X1          8.958e-02  1.793e-02   4.998 2.61e-06 ***
## X2          3.392e-01  1.781e-02  19.047  < 2e-16 ***
## X3          9.295e-01  1.790e-02  51.942  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1769 on 96 degrees of freedom
## Multiple R-squared:  0.9696, Adjusted R-squared:  0.9687
## F-statistic:  1022 on 3 and 96 DF,  p-value: < 2.2e-16
```

We see that the values are very close.

4.3 Partial Correlation Coefficients

```
x <- pcc(X, y, nboot = 100)
print(x)
```

```
##
```

```
## Call:
## pcc(X = X, y = y, nboot = 100)
##
## Partial Correlation Coefficients (PCC):
##     original          bias  std. error min. c.i. max. c.i.
## X1 0.4543671  0.0041092337 0.080517838 0.2860613 0.6075130
## X2 0.8892416 -0.0011730747 0.018469869 0.8569707 0.9281357
## X3 0.9826698 -0.0001722503 0.003255498 0.9766680 0.9903750
```

4.4 Sobol indices for deterministic function and for model

```
y.fun <- function(X) {

  X1<- X[,1]
  X2<- X[,2]
  X3<- X[,3]

  X1+X2+X3
}

yhat.fun<-function(X,lm)
{
  X1<- X[,1]
  X2<- X[,2]
  X3<- X[,3]

  yhat <- predict(lm.fit,data.frame(X1=X1,X2=X2,X3=X3))
  return(yhat)
}

nboot = 1000
```

```
x <- sobol(model = y.fun, X[1:50,], X[51:100,], order = 3, nboot = nboot)
S.sobol <- x$S
pander(S.sobol)
```

	original	bias	std. error	min. c.i.	max. c.i.
X1	1.302	0.04949	0.8788	-0.5695	2.849
X2	1.494	0.04884	0.8287	-0.4242	3.045
X3	1.527	0.03317	0.3645	0.76	2.196
X1*X2	-1.453	-0.05783	0.8793	-2.996	0.4286
X1*X3	-1.453	-0.05783	0.8793	-2.996	0.4286
X2*X3	-1.453	-0.05783	0.8793	-2.996	0.4286
X1*X2*X3	1.453	0.05783	0.8793	-0.4286	2.996

```
#yhat.fun(data.frame(X1=1,X2=2,X3=3),lm.fit)

x <- sobol(model = yhat.fun,X[1:50,], X[51:100,], order = 3, nboot = nboot)
S.sobol <- x$S
pander(S.sobol)
```

	original	bias	std. error	min. c.i.	max. c.i.
X1	1.064	0.04465	0.7367	-0.476	2.395
X2	1.064	0.04445	0.7332	-0.4372	2.384
X3	1.324	0.01343	0.1234	1.043	1.533
X1*X2	-1.078	-0.04514	0.7354	-2.409	0.4652
X1*X3	-1.078	-0.04514	0.7354	-2.409	0.4652
X2*X3	-1.078	-0.04514	0.7354	-2.409	0.4652
X1*X2*X3	1.078	0.04514	0.7354	-0.4652	2.409

Chapter 5

Random Effects and Mixed Models

5.1 Crossed versus nested random effects.

How do they differ and how are they specified correctly in lme4 and in JAGS / Stan?

5.2 Very Large Number of RE's

https://arxiv.org/abs/1610.08088

Chapter 6

Propensity Score Matching

6.1 Caliper

Putting constraints on matching can reduce bias [Lunt, 2013].

Matching on the propensity score is widely used to estimate the effect of an exposure in observational studies. However, the quality of the matches can be affected by decisions made during the matching process, particularly the order in which subjects are selected for matching and the maximum permitted difference between matched subjects (the "caliper"). This study used simulations to explore the effects of these decisions on both the imbalance of covariates and the closeness of matching, while allowing the numbers of potential matches and strengths of association between the confounding variable and the exposure to vary. It was found that, without a caliper, substantial bias was possible, particularly with a relatively small reservoir of potential matches and strong confounder-exposure association. Use of the recommended caliper reduced the bias considerably, but bias remained if subjects were selected by increasing or decreasing propensity score. A tighter caliper led to greatly reduced bias and closer matches, although some subjects could not be matched. This study suggests that a narrow caliper can improve the performance of propensity score matching. In situations where it is impossible to find appropriate matches for all exposed subjects, it is better to

select subjects in order of the best available matches, rather than increasing or decreasing the propensity score.

Chapter 7

Introduction to Normalizing Flows

7.1 Variational Inference With NF

Variational inference now lies at the core of large-scale topic models of text (Hoffman et al., 2013), provides the state-of-the-art in semi-supervised classification (Kingma et al., 2014), drives the models that currently produce the most realistic generative models of images (Gregor et al., 2014; Rezende et al., 2014), and are a default Proceedings of the 32 nd International Conference on Machine Learning, Lille, France, 2015. JMLR: W&CP volume 37. Copyright 2015 by the author(s). tool for the understanding of many physical and chemical systems. Despite these successes and ongoing advances, there are a number of disadvantages of variational methods that limit their power and hamper their wider adoption as a default method for statistical inference. It is one of these limitations, the choice of posterior approximation, that we address in this paper

[http://proceedings.mlr.press/v37/rezende15.pdf]

Generative modeling loosely refers to building a model of data, for instance p(image), that we can sample from. This is in contrast to discriminative

modeling, such as regression or classification, which tries to estimate conditional distributions such as p(class | image).

7.2

Bibliography

Mark Lunt. Selecting an Appropriate Caliper Can Be Essential for Achieving Good Balance With Propensity Score Matching. *American Journal of Epidemiology*, 179(2):226–235, 10 2013. ISSN 0002-9262. doi: 10.1093/aje/kwt212. URL https://doi.org/10.1093/aje/kwt212.

Andrea Saltelli and I.M. Sobol'. Sensitivity analysis for nonlinear mathematical models: Numerical experience. *Matematicheskoe Modelirovanie*, 7, 01 1995.

Yihui Xie. *Dynamic Documents with R and knitr*. Chapman and Hall/CRC, Boca Raton, Florida, 2nd edition, 2015. URL http://yihui.org/knitr/. ISBN 978-1498716963.

Yihui Xie. *bookdown: Authoring Books and Technical Documents with R Markdown*, 2020. URL https://github.com/rstudio/bookdown. R package version 0.21.