Aidan Gogarty

A Short Logistic Regression on a Spam/Nonspam Email Dataset

Introduction

We seek to perform a logistic regression on the Spam dataset, with 'type' as our predicted variable, and the special characters and capital letters as our covariates.

Firstly we load the dataset and restrict ourselves to the variables which we are interested in, namely columns 49-58. Our variable to be predicted is already a factor and all other covariates are already numeric, so there is no need to change anything here.

Next, we fit a logistic regression to the dataset. However, we get a warning.

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

We see that several of our fitted probabilites are effectively zero. Below is a table of values from our model, as well as 95% Confidence Interval values for the Odds.

Table for Logistic Regression Model Values

	Estimate	Std. Error	Z Score	P Value	LB Odds	Odds	UB Odds
(Intercept)	-1.677	0.070	-23.835	0.000	0.162	0.187	0.215
charSemicolon	-1.055	0.412	-2.562	0.010	0.153	0.348	0.794
charRoundbracket	-1.441	0.251	-5.733	0.000	0.143	0.237	0.391
charSquarebracket	-3.878	1.085	-3.574	0.000	0.002	0.021	0.181
charExclamation	1.312	0.110	11.931	0.000	2.981	3.714	4.628
charDollar	10.586	0.601	17.622	0.000	11895.368	39573.080	131650.290
charHash	0.355	0.144	2.459	0.014	1.070	1.427	1.903
capitalAve	0.056	0.022	2.533	0.011	1.012	1.057	1.105
capitalLong	0.014	0.002	8.377	0.000	1.010	1.014	1.018
capitalTotal	0.000	0.000	1.895	0.058	1.000	1.000	1.000

We see that we have enormous odds for the charDollar variable, as well as odds for the charExclamation variable which are comparatively larger than those of other covariates, though tiny compared to the odds of charDollar.

It seems that complete separation might be occurring, and this issue could be due to our X values being continuous, and our Y variable being a binomial categorical variable. For classification purposes, this may not be an issue, but we would also like to do some inference on our coefficients, especially those of charExclamation and charDollar.

To be more certain about what is happening, we will fit a bias-reduction logistic regression from the brglm package and compare it to our previous model.

Table for Bias-Reduction Logistic Regression Model Values

	Estimate	Std. Error	Z Score	P Value	LB Odds	Odds	UB Odds
(Intercept)	-1.673	0.070	-23.860	0.000	0.163	0.188	0.216
charSemicolon	-0.892	0.353	-2.527	0.011	0.202	0.410	0.830
charRoundbracket	-1.436	0.250	-5.737	0.000	0.144	0.238	0.392
charSquarebracket	-3.813	1.075	-3.547	0.000	0.003	0.022	0.190
charExclamation	1.307	0.110	11.913	0.000	2.965	3.695	4.604
charDollar	10.555	0.599	17.621	0.000	11574.242	38351.059	127075.595
charHash	0.332	0.140	2.365	0.018	1.053	1.393	1.843
capitalAve	0.054	0.022	2.498	0.012	1.011	1.056	1.103
capitalLong	0.014	0.002	8.440	0.000	1.010	1.014	1.018
capitalTotal	0.000	0.000	1.776	0.076	1.000	1.000	1.000

We see that the values for the bias-reduction model are quite similar to those of the standard logistic regression model above, and on doing some exploratory data analysis, we find that:

- the amount of spam emails containing at least one dollar character (charDollar > 0) is 61%, whereas for nonspam emails it is only 10%.
- the amount of spam emails at least one exclamation mark (charExclamation > 0) is 83%, whereas for nonspam emails it is only 27%.
- for values of charDollar and charExclamation together at 0.1 or above (charDollar $>= 0.1$ & charExclamation $>= 0.1$), we get 38% of spam emails, and 1% of nonspam emails.

It seems that our models are indeed accurate. We will proceed, comparing both models for Goodness-of-Fit and AUC-Optimisation.

Questions

The variables charDollar and charExclamation

The variables charDollar and charExclamation do indeed heavily affect the probability of the email being spam.

Inferential problems which arise are that we have p-values effectively at 0 for both charDollar and charExclamation, and this could cause us to question the accuracy of the coefficients for both variables. The extremely high value of the coefficient for charDollar also raises questions as to the inference potential of this variable. However, we have seen in our EDA that the presence of both characters is highly indicative of spam. A further consideration might be that, given that the nonspam emails come from the personal emails of workers in a location in the USA, dollar signs and exclamation marks might also be expected to be present in these personal emails.

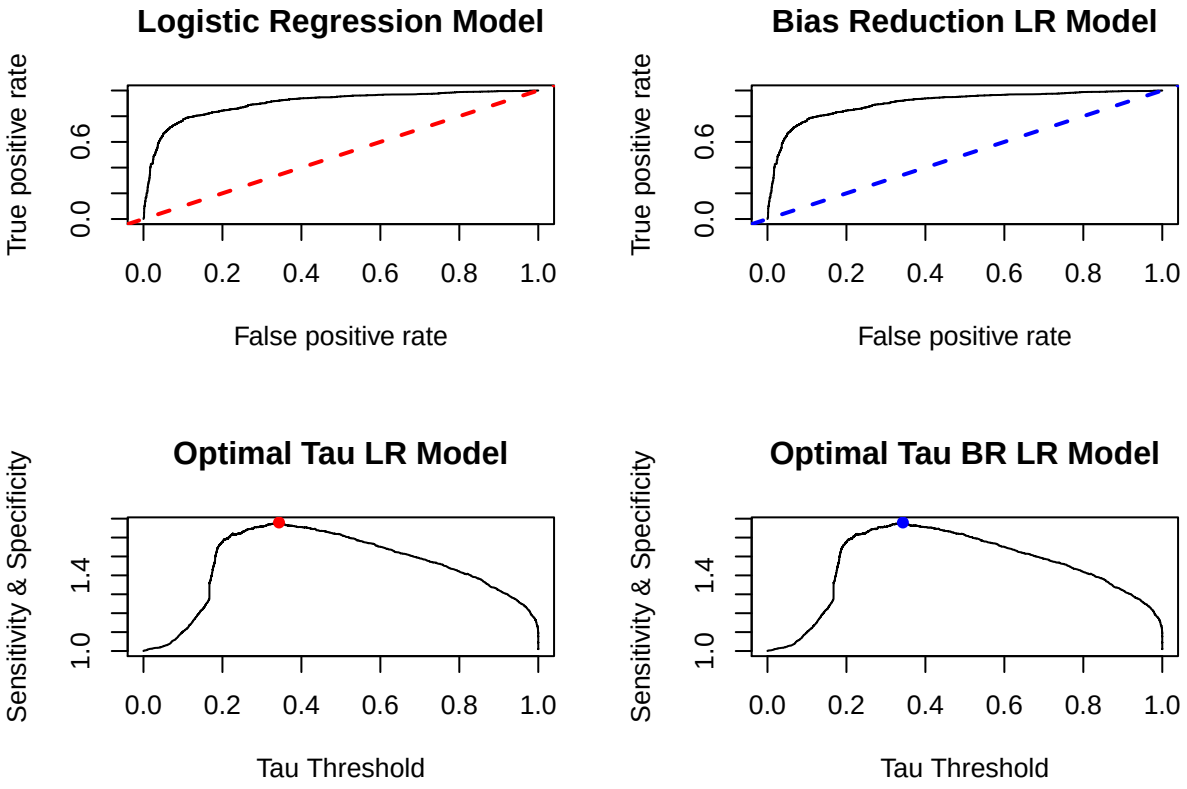
Is the model a good fit? The deviance test result for the original logistic model is 0.2579881 and for the Bias Reduction logistic model is 0.2541234. Neither test is statistically significant, meaning there is evidence that both models might be a good fit for the data.

Evaluating using ROC, AUC and Tau

Tables of predictions using both models give the following initial results:

Model	Tau	Sensitivity	Specificity	PPV	NPV	Accuracy	FDR	FPR
Logistic Regression	0.5	0.6668505	0.9487088	0.8942308	0.8140966	0.8376440	0.1057692	0.0512912
Bias-Reduction LR	0.5	0.6640927	0.9490674	0.8945022	0.8129032	0.8367746	0.1054978	0.0509326

Plotting the Area Under Curve with tau = 0.5, we can see that both graphs are very similar.



By optimising our value of tau in order to maximise the sum of sensitivity and specificity, we get the above graphs for each model. The AUC value for our Logistic Regression model is 0.90204, whereas the AUC value for our bias-reduction model is 0.90239. The ideal tau values are 0.3432798 and 0.3430328, respectively.

Finally, our improved accuracy gives the following new table of values.

Model	Tau	Sensitivity	Specificity	PPV	NPV	Accuracy	FDR	FPR
LR	0.3432798	0.7859901	0.8923960	0.8260870	0.8650904	0.8504673	0.1739130	0.1076040
BR-LR	0.3430328	0.7848869	0.8931133	0.8268449	0.8645833	0.8504673	0.1731551	0.1068867

Final Comments Although the initial high fitted values and warnings from our models may have caused us to question their accuracy, we have seen through our EDA as well as our ROC/AUC visualisations, tau-optimisation, and predictive performance measures, that the models are indeed good, and that we can draw strong conclusions from them about the influence of the presence of certain characters on an email being classified as spam.

Appendix

```
knitr::opts_chunk$set(echo = FALSE, warning = FALSE)
# loading data
library(kernlab)
data(spam)
spam2 <- spam[,49:58]
str(spam2)

# fitting model

model <- glm(type ~., data = spam2, family = "binomial")

# model coefficients and confidence intervals

model.summary <- summary(model)
model.coefs <- round(model.summary$coefficients,3)

w <- coef(model)

se <- model.coefs[,2]

wLB <- w - (2 * se)
wUB <- w + (2 * se)

CIs <- cbind(wLB,w,wUB)
oddsCIs <- round(exp(CIs),3)


d <- data.frame(model.coefs, oddsCIs)




knitr:::kable(d,
              col.names = c("Estimate", "Std. Error", "Z Score", "P Value",
                            "LB Odds", "Odds", "UB Odds"))

# fitting bias-reduction logistic regression;
#table of coefficients and confidence intervals

library(brglm)
br.model <- brglm(type ~., data = spam2, family = "binomial", method = "brglm.fit")

br.model.summary <- summary(br.model)
br.model.coefs <- round(br.model.summary$coefficients,3)

br.w <- coef(br.model)

br.se <- br.model.coefs[,2]

br.wLB <- br.w - (2 * br.se)
br.wUB <- br.w + (2 * br.se)
```

```r
br.CIs <- cbind(br.wLB, br.w, br.wUB)
br.oddsCIs <- round(exp(br.CIs),3)


br.d <- data.frame(br.model.coefs, br.oddsCIs)


knitr:::kable(br.d,
              col.names = c("Estimate", "Std. Error", "Z Score", "P Value",
                            "LB Odds", "Odds", "UB Odds"))

# exploratory data analysis; percentages of charDollar and charExclamation

library(dplyr)

spam.list <- spam2 %>% filter(type == "spam")
nonspam.list <- spam2 %>% filter(type == "nonspam")


dollar.spam <- spam.list %>% filter(charDollar > 0)
spam.dollar.percent <- round(nrow(dollar.spam) / nrow(spam.list), 2) * 100

dollar.nonspam <- nonspam.list %>% filter(charDollar > 0)
nonspam.dollar.percent <- round(nrow(dollar.nonspam) / nrow(nonspam.list), 2) * 100


exclam.spam <- spam.list %>% filter(charExclamation > 0)
spam.exclam.percent <- round(nrow(exclam.spam) / nrow(spam.list), 2) * 100

exclam.nonspam <- nonspam.list %>% filter(charExclamation > 0)
nonspam.exclam.percent <- round(nrow(exclam.nonspam) / nrow(nonspam.list), 2) * 100


dollar.and.exclam.spam <- spam.list %>% filter(charDollar >= 0.1 & charExclamation >= 0.1)
spam.dollar.and.exclam.percent <- round(nrow(dollar.and.exclam.spam) / nrow(spam.list), 2) * 100

dollar.and.exclam.nonspam <- nonspam.list %>% filter(charDollar >= 0.1 & charExclamation >= 0.1)
nonspam.dollar.and.exclam.percent <- round(nrow(dollar.and.exclam.nonspam)/nrow(nonspam.list),2)*100

# deviance test of both models

dev <- deviance(model)
br.dev <- deviance(br.model)

mod.mat <- model.matrix(model)
br.mod.mat <- model.matrix(br.model)

NOglm <- nrow(unique(mod.mat))
NObrglm <- nrow(unique(br.mod.mat))

length.glm <- length(model$coefficients)
length.brglm <- length(br.model$coefficients)
```

```r
DevTest <- 1 - pchisq(dev, NOglm - length.glm)
brDevTest <- 1 - pchisq(br.dev, NObrglm - length.brglm)

# tau and predictive performance tables

tau <- 0.5

p <- fitted(model)
br.p <- fitted(br.model)
pred <- ifelse( p > tau, 1, 0)
br.pred <- ifelse( br.p > tau, 1, 0)
tab <- table(spam2$type, pred)
br.tab <- table(spam2$type, br.pred)

TN <- tab[1,1]
FP <- tab[1,2]
FN <- tab[2,1]
TP <- tab[2,2]

brTN <- br.tab[1,1]
brFP <- br.tab[1,2]
brFN <- br.tab[2,1]
brTP <- br.tab[2,2]

nam <- "Logistic Regression"
t.val <- 0.5
sen <- TP/(TP + FN)
spe <- TN/(FP + TN)
ppv <- TP/(TP + FP)
npv <- TN/(TN + FN)
acc <- (TP + TN) / (TP + FP + TN + FN)
fdr <- FP/(FP + TP)
fpr <- FP/(FP + TN)

lr.df <- data.frame(nam,t.val,sen,spe,ppv,npv,acc,fdr,fpr)

names(lr.df) <- c("Model", "Tau", "Sensitivity","Specificity","PPV","NPV","Accuracy","FDR","FPR")

br.nam <- "Bias-Reduction LR"
br.t.val <- 0.5
br.sen <- brTP/(brTP + brFN)
br.spe <- brTN/(brFP + brTN)
br.ppv <- brTP/(brTP + brFP)
br.npv <- brTN/(brTN + brFN)
br.acc <- (brTP + brTN) / (brTP + brFP + brTN + brFN)
br.fdr <- brFP/(brFP + brTP)
br.fpr <- brFP/(brFP + brTN)

br.df <- data.frame(br.nam,br.t.val, br.sen, br.spe, br.ppv, br.npv, br.acc, br.fdr, br.fpr)

names(br.df) <- c("Model", "Tau","Sensitivity","Specificity","PPV","NPV","Accuracy","FDR","FPR")

val.tab <- rbind(lr.df, br.df)
```

```r
knitr::kable(val.tab)

# plotting ROC/AUC

library(ROCR)


model.predict <- prediction(fitted(model), spam2$type)
model.perf <- performance(model.predict, "tpr", "fpr")


br.model.predict <- prediction(fitted(br.model), spam2$type)
br.model.perf <- performance(br.model.predict, "tpr", "fpr")

# plotting tau optimisation

par(mfrow = c(2,2))

plot(model.perf, main = "Logistic Regression Model")
abline(0,1, col = "red", lty = 2, lwd = 2)

plot(br.model.perf, main = "Bias Reduction LR Model")
abline(0,1, col = "blue", lty = 2, lwd = 2)

auc1 <- performance(model.predict, "auc")

sens1 <- performance(model.predict, "sens")
spec1 <- performance(model.predict, "spec")

tau1 <- sens1@x.values[[1]]
sensSpec1 <- sens1@y.values[[1]] + spec1@y.values[[1]]
best1 <- which.max(sensSpec1)
plot(tau1, sensSpec1, type = "l", main = "Optimal Tau LR Model",
     xlab = "Tau Threshold", ylab = "Sensitivity & Specificity")
points(tau1[best1], sensSpec1[best1], pch = 16, col = "red", cex = 1)

auc2 <- performance(br.model.predict, "auc")


sens2 <- performance(br.model.predict, "sens")
spec2 <- performance(br.model.predict, "spec")

tau2 <- sens2@x.values[[1]]
sensSpec2 <- sens2@y.values[[1]] + spec2@y.values[[1]]
best2 <- which.max(sensSpec2)
plot(tau2, sensSpec2, type = "l", main = "Optimal Tau BR LR Model",
     xlab = "Tau Threshold", ylab = "Sensitivity & Specificity")
points(tau2[best2], sensSpec2[best2], pch = 16, col = "blue", cex = 1)

optimal1 <- round(as.numeric(auc1@y.values),5)
optimal2 <- round(as.numeric(auc2@y.values),5)

opt.tau1 <- as.numeric(tau1[best1])
```

```r
opt.tau2 <- as.numeric(tau2[best2])


# table of optimised predictive performance values

opt.p <- fitted(model)
opt.br.p <- fitted(br.model)
opt.pred <- ifelse( opt.p > opt.tau1, 1, 0)
opt.br.pred <- ifelse( opt.br.p > opt.tau2, 1, 0)
opt.tab <- table(spam2$type, opt.pred)
opt.br.tab <- table(spam2$type, opt.br.pred)

opt.TN <- opt.tab[1,1]
opt.FP <- opt.tab[1,2]
opt.FN <- opt.tab[2,1]
opt.TP <- opt.tab[2,2]

opt.brTN <- opt.br.tab[1,1]
opt.brFP <- opt.br.tab[1,2]
opt.brFN <- opt.br.tab[2,1]
opt.brTP <- opt.br.tab[2,2]

opt.nam <- "LR"
tau1.val <- opt.tau1
opt.sen <- opt.TP/(opt.TP + opt.FN)
opt.spe <- opt.TN/(opt.FP + opt.TN)
opt.ppv <- opt.TP/(opt.TP + opt.FP)
opt.npv <- opt.TN/(opt.TN + opt.FN)
opt.acc <- (opt.TP + opt.TN) / (opt.TP + opt.FP + opt.TN + opt.FN)
opt.fdr <- opt.FP/(opt.FP + opt.TP)
opt.fpr <- opt.FP/(opt.FP + opt.TN)

opt.lr.df <- data.frame(opt.nam,tau1.val,opt.sen,opt.spe,opt.ppv,opt.npv,
                        opt.acc,opt.fdr,opt.fpr)

names(opt.lr.df) <- c("Model", "Tau", "Sensitivity","Specificity","PPV","NPV","Accuracy","FDR","FPR")

opt.br.nam <- "BR-LR"
tau2.val <- opt.tau2
opt.br.sen <- opt.brTP/(opt.brTP + opt.brFN)
opt.br.spe <- opt.brTN/(opt.brFP + opt.brTN)
opt.br.ppv <- opt.brTP/(opt.brTP + opt.brFP)
opt.br.npv <- opt.brTN/(opt.brTN + opt.brFN)
opt.br.acc <- (opt.brTP + opt.brTN) / (opt.brTP + opt.brFP + opt.brTN + opt.brFN)
opt.br.fdr <- opt.brFP/(opt.brFP + opt.brTP)
opt.br.fpr <- opt.brFP/(opt.brFP + opt.brTN)

opt.br.df <- data.frame(opt.br.nam,tau2.val,opt.br.sen, opt.br.spe, opt.br.ppv, opt.br.npv,
                        opt.br.acc, opt.br.fdr, opt.br.fpr)

names(opt.br.df) <- c("Model", "Tau","Sensitivity","Specificity","PPV","NPV","Accuracy","FDR","FPR")

opt.val.tab <- rbind(opt.lr.df, opt.br.df)
```

```r
knitr::kable(opt.val.tab)
```