Logistic and Poisson Regression

William Chiu

01/11/2017

Overview

The paper demonstrates a close relationship between logistic and poisson regression. Due to this relationship, modelers may use either regression methods to predict the probability of a binary event (e.g., default versus non-default) or predict the number of events (e.g., number of defaults).

The paper will first show the algebraic derivation of the *logit* and *log odds*. The derivation depends only on the definition of probability. Using log odds definition, the paper will show that log odds can be modeled using either logistic regression or poisson regression. Finally, the paper will compare the results from the two regression methods on a test data set.

Deriving Logit

Using the definition of probability, prove that

$$log\left(\frac{p}{1-p}\right) = log\left(\frac{A}{B}\right)$$

where p is the probability of the event, A is the number of events, and B is the number of non-events. The left-hand side is often called the *logit* and the right-hand is often called *log odds*. After seeing the algebraic proof, you will be convinced that they are one and the same.

Assume that the probability of the event (p) is defined as follows

$$p = \frac{A}{A+B}$$

Using this definition of probability and some algebra, we can prove the equivalence of logit and log odds. First, take the log of both sides of the equation

$$log(p) = log\left(\frac{A}{A+B}\right)$$

Subtract $log(1-p)$ from both sides of the equation

$$log(p) - log(1-p) = log\left(\frac{A}{A+B}\right) - log(1-p)$$

Using the log rule, the left-hand side of the equation is the logit

$$log\left(\frac{p}{1-p}\right) = log\left(\frac{A}{A+B}\right) - log(1-p)$$

Substitute the definition of p into the right-hand side

$$log\left(\frac{p}{1-p}\right) = log\left(\frac{A}{A+B}\right) - log\left(1 - \frac{A}{A+B}\right)$$

Substitute 1 with $\frac{A+B}{A+B}$ as follows

$$log\left(\frac{p}{1-p}\right) = log\left(\frac{A}{A+B}\right) - log\left(\frac{A+B}{A+B} - \frac{A}{A+B}\right)$$

Cancel out the A as follows

$$log\left(\frac{p}{1-p}\right) = log\left(\frac{A}{A+B}\right) - log\left(\frac{B}{A+B}\right)$$

Expand the right-hand side using the log rule

$$log\left(\frac{p}{1-p}\right) = log(A) + log(A+B) - log(B) - log(A+B)$$

Do some subtraction and apply the log rule to get the derivation

$$log\left(\frac{p}{1-p}\right) = log(A) - log(B)$$

$$log\left(\frac{p}{1-p}\right) = log\left(\frac{A}{B}\right)$$

Log Odds, Logistic Regression, Poisson Regression

In logistic regression, we believe that log odds is a linear combination of the regressors and their corresponding parameters

$$log\left(\frac{A}{B}\right) = \beta_0 + \beta_1 X_1 + \ldots + \beta_k X_k$$

where the parameters $(\beta_0, \beta_1, \ldots, \beta_k)$ are estimated by maximizing the binomial log likelihood function. With some algebra, we can demonstrate that the parameters are unchanged whether we model log odds or the log of the number of events

$$log\left(\frac{A}{B}\right) = \beta_0 + \beta_1 X_1 + \ldots + \beta_k X_k$$

$$log(A) - log(B) = \beta_0 + \beta_1 X_1 + \ldots + \beta_k X_k$$

$$log(A) = \beta_0 + \beta_1 X_1 + \ldots + \beta_k X_k + log(B)$$

Using the last equation, we can estimate the parameters using poisson regression by maximizing the poisson log likelihood function.

Notice that the parameter for $log(B)$ is exactly one. This is called the **offset** term.

Example in R

Let's test our hypothesis that the parameters are the same whether we use logistic regression or poisson regression.

Here's some toy data

```
library(AER)
data(CreditCard)

CreditCard <- CreditCard[,1:4]

pander::pander(head(CreditCard))
```

card	reports	age	income
yes	0	37.67	4.52
yes	0	33.25	2.42
yes	0	33.67	4.5
yes	0	30.5	2.54
yes	0	32.17	9.787
yes	0	23.25	2.5

```
pander::pander(table(CreditCard$card))
```

no	yes
296	1023

Split the data set between training and test data sets using the 70-30 rule.

```
library(caret)

set.seed(1234)

train <- createDataPartition(CreditCard$card, p=0.7, list = FALSE)

CreditCard <- CreditCard[train,]  ### training data

testData <- CreditCard[-train,]

dim(CreditCard)   ### training data
```

```
## [1] 925   4
```

```
dim(testData)
```

```
## [1] 287   4
```

We can fit a logistic regression to predict yes.

```
logit.mod <- glm(card ~ ., data=CreditCard, family=binomial)

pander::pander(logit.mod, caption="")
```

	Estimate	Std. Error	z value	Pr(>|z|)
reports	-1.392	0.1349	-10.32	5.685e-25
age	-0.001002	0.009945	-0.1008	0.9197
income	0.2881	0.07354	3.917	8.949e-05
(Intercept)	1.016	0.3437	2.956	0.003121

Fitting a poisson regression is complicated because we need to **aggregate** the data to get the number of yes and number of no for each unique combination of regressor values. How we choose to aggregate the data is important. Since poisson regression uses the log link function, we cannot have number of no ever equal to zero.

To help us aggregate the data, create 4 quantiles for each regressor and calculate the mean regressor value for each bin.

```
CreditCardBin <- CreditCard

CreditCardBin$reports_bin <- Hmisc::cut2(CreditCardBin$reports, g=4, levels.mean=TRUE, d
igits=6)
CreditCardBin$age_bin <- Hmisc::cut2(CreditCardBin$age, g=4, levels.mean=TRUE, digits=6)
CreditCardBin$income_bin <- Hmisc::cut2(CreditCardBin$income, g=4, levels.mean=TRUE, dig
its=6)

CreditCardBin$reports <- as.numeric(as.character(CreditCardBin$reports_bin))
CreditCardBin$age <- as.numeric(as.character(CreditCardBin$age_bin))
CreditCardBin$income <- as.numeric(as.character(CreditCardBin$income_bin))

pander::pander(head(CreditCardBin))
```

	card	reports	age	income	reports_bin	age_bin	income_bin
1	yes	0	35.08	5.784	0.00000	35.0797	5.78435
2	yes	0	35.08	2.582	0.00000	35.0797	2.58189
3	yes	0	35.08	5.784	0.00000	35.0797	5.78435
4	yes	0	28.18	2.582	0.00000	28.1799	2.58189
6	yes	0	22.75	2.582	0.00000	22.7518	2.58189
7	yes	0	28.18	3.415	0.00000	28.1799	3.41474

Now we can calculate number of *yes* and number of *no* by each bin mean.

```
library(sqldf)

CreditCardCount <- sqldf("
              SELECT SUM(CASE WHEN card='yes' THEN 1 ELSE 0 END) AS card_yes
                ,SUM(CASE WHEN card='no' THEN 1 ELSE 0 END) AS card_no
                ,reports
                ,age
                ,income
              FROM CreditCardBin
              GROUP BY reports, age, income ")

pander::pander(head(CreditCardCount))
```

card_yes	card_no	reports	age	income
72	20	0	22.75	1.865
53	7	0	22.75	2.582
32	4	0	22.75	3.415
10	2	0	22.75	5.784
36	13	0	28.18	1.865
55	4	0	28.18	2.582

Now we can fit a poisson regression with an offset term.

```
poisson.mod <- glm(card_yes ~  reports + age + income +
                     offset(log(card_no)), data=CreditCardCount, family=poisson)

pander::pander(poisson.mod, caption="")
```

	Estimate	Std. Error	z value	Pr(>|z|)
reports	-0.9677	0.05155	-18.77	1.237e-78
age	0.004375	0.00441	0.9919	0.3212
income	0.2187	0.0248	8.821	1.138e-18
(Intercept)	1.021	0.147	6.94	3.919e-12

Here are the parameter estimates from logistic regression

```
pander::pander(logit.mod, caption="")
```

	Estimate	Std. Error	z value	Pr(>|z|)
reports	-1.392	0.1349	-10.32	5.685e-25

	Estimate	Std. Error	z value	Pr(>|z|)
age	-0.001002	0.009945	-0.1008	0.9197
income	0.2881	0.07354	3.917	8.949e-05
(Intercept)	1.016	0.3437	2.956	0.003121

The parameter estimates from logistic regression and poisson regression are not the same but similar. Differences could be attributed to our choice of bins. By aggregating the data set, we lose information from the regressors. However, this is not necessarily a bad thing:

1. By aggregating the data set, we reduce the number of records in the data set – this is computationally faster than using the original unaggregated data set
2. Regressors may suffer from outliers – by binning the regressor we reduce the variability in the regressor

Depending on the memory of your computer and/or the existence of noise in the regressors, poisson regression may be a competitive alternative to logistic regression.

Probability Predictions from Logistic and Poission Regression

The parameter estimates are different, but are the probability predictions similar? We are going to run the predict function on the **unaggregated** data set to see if predictions between the two models are similar.

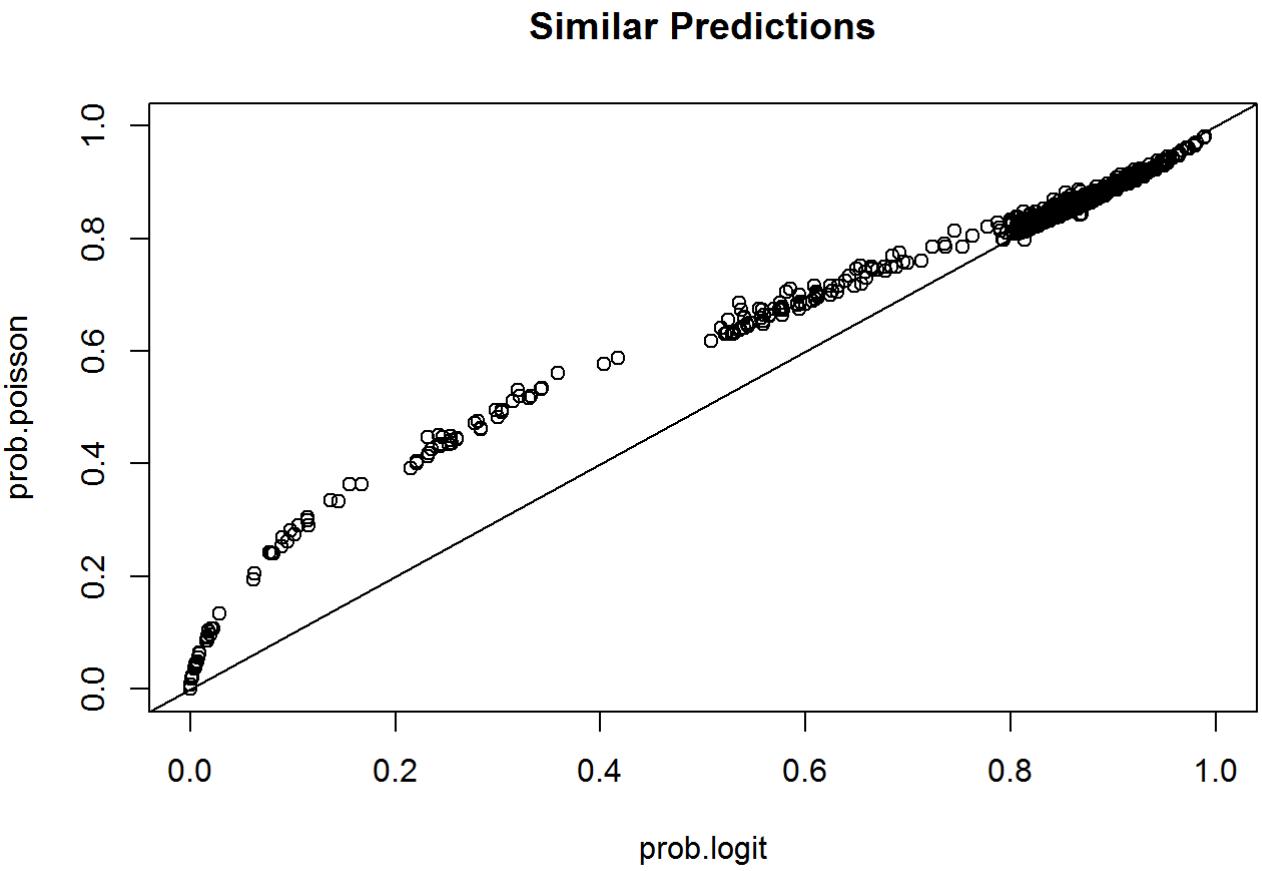
Since the poisson model expects a column called `card_no`, we will add it to the unaggregated data set and provide a constant value of 1. For logistic regression, we can directly compute the probability. For poisson regression, we need to convert the log odds to a probability.

```
CreditCard$card_no <- 1

logit.prob.pred <- predict(logit.mod, newdata=CreditCard, type="response")

poisson.link.pred <- predict(poisson.mod, newdata=CreditCard, type="link")
poisson.prob.pred <- gtools::inv.logit(poisson.link.pred)

prob.predictions <- data.frame(prob.logit=logit.prob.pred, prob.poisson=poisson.prob.pre
d)

pander::pander(head(prob.predictions))
```

	prob.logit	prob.poisson
1	0.9072	0.8979
2	0.8429	0.8449
3	0.9071	0.8959
4	0.8477	0.8468
6	0.8472	0.8415

	prob.logit	prob.poisson
7	0.8937	0.8817

```
with(prob.predictions, plot(prob.logit, prob.poisson, main="Similar Predictions",
                            xlim=c(0,1), ylim=c(0,1)))
abline(0,1)
```



The plot reveals, that although the parameter estimates are not exactly the same, the probability predictions are similar. We could also compute an R^2 to quantify the similarity between logistic and poisson regression.

```
similar <- caret::R2(prob.predictions$prob.logit, prob.predictions$prob.poisson
                     , formula="traditional")

similar2 <- caret::R2(prob.predictions$prob.logit, prob.predictions$prob.poisson
                     , formula="corr")
```

Probability predictions from both models are so similar that regressing predictions from logistic regression against predictions from poisson regression generates $R^2 = 91.85\%$.[1]

Binomial Logit Model

A seldomly used fact is that logistic regression can model a dependent (response) variable that is binary or count. This should not be surprising because log odds is the ratio of counts. Here's an example using the aggregated data set.

```
binomial.logit.mod <- glm(cbind(card_yes, card_no) ~ reports + age + income,
                          data=CreditCardCount, family=binomial)

pander::pander(binomial.logit.mod, caption="")
```

	Estimate	Std. Error	z value	Pr(>|z|)
reports	-0.9571	0.08024	-11.93	8.422e-33
age	0.003166	0.01046	0.3027	0.7621
income	0.2326	0.07016	3.315	0.0009168
(Intercept)	1.006	0.3491	2.882	0.003956

The parameter estimates for binomial logistic regression and poisson regression are nearly identical because the parameters were estimated on the same aggregated data. Here are the poisson estimates.

```
pander::pander(poisson.mod, caption="")
```

	Estimate	Std. Error	z value	Pr(>|z|)
reports	-0.9677	0.05155	-18.77	1.237e-78
age	0.004375	0.00441	0.9919	0.3212
income	0.2187	0.0248	8.821	1.138e-18
(Intercept)	1.021	0.147	6.94	3.919e-12

Here are the binary logistic regression estimates using unaggregated data.

```
pander::pander(logit.mod, caption="")
```

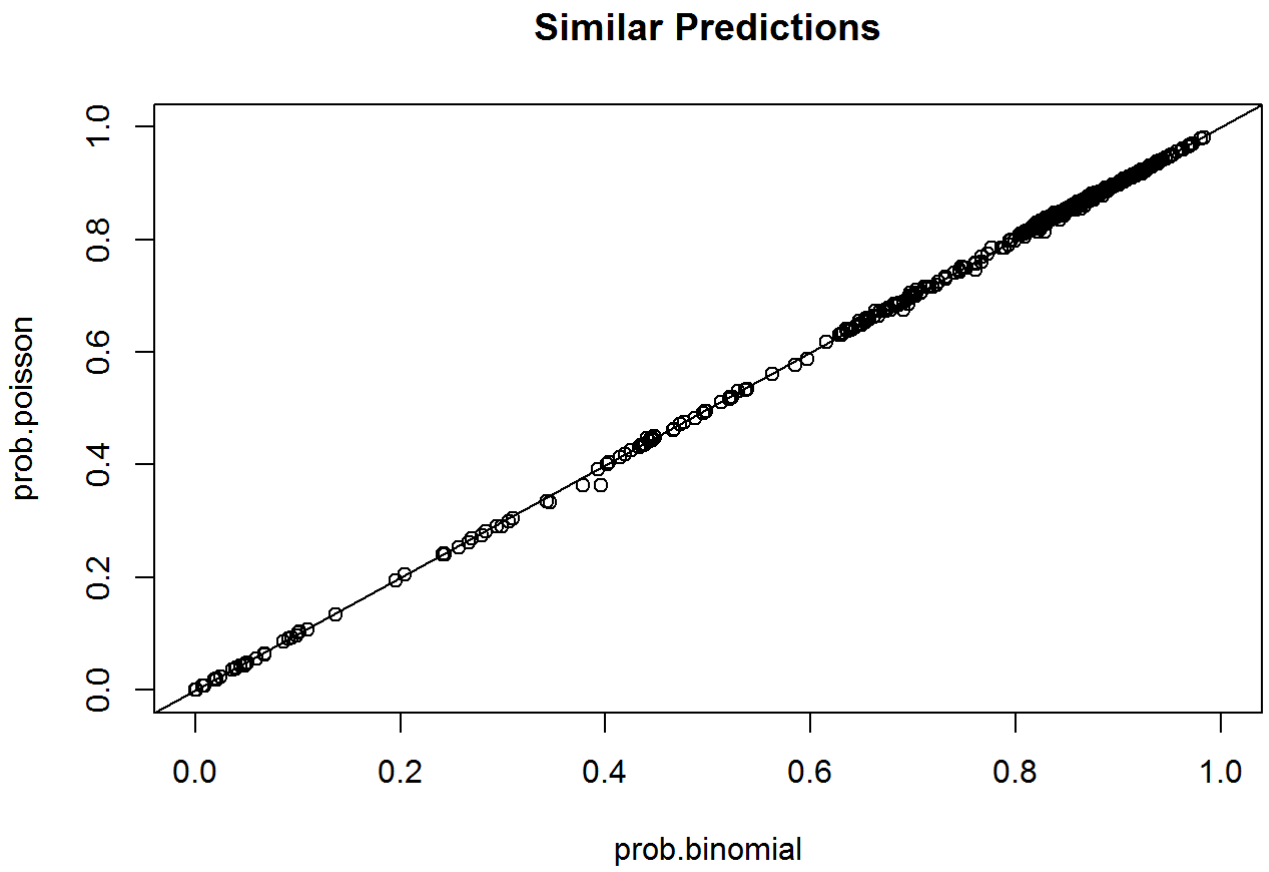
	Estimate	Std. Error	z value	Pr(>|z|)
reports	-1.392	0.1349	-10.32	5.685e-25
age	-0.001002	0.009945	-0.1008	0.9197
income	0.2881	0.07354	3.917	8.949e-05
(Intercept)	1.016	0.3437	2.956	0.003121

Probability predictions from the binomial logit model and poisson model are nearly identical. Using the unaggregated data set,

```
binomial.prob.pred <- predict(binomial.logit.mod, newdata=CreditCard, type="response")

prob.predictions <- data.frame(prob.logit=logit.prob.pred, prob.poisson=poisson.prob.pre
d,
                               prob.binomial=binomial.prob.pred)

pander::pander(head(prob.predictions))
```

	prob.logit	prob.poisson	prob.binomial
1	0.9072	0.8979	0.8981
2	0.8429	0.8449	0.8421
3	0.9071	0.8959	0.8965
4	0.8477	0.8468	0.8446
6	0.8472	0.8415	0.8404
7	0.8937	0.8817	0.8824

```
with(prob.predictions, plot(prob.binomial, prob.poisson, main="Similar Predictions",
                            xlim=c(0,1), ylim=c(0,1)))
abline(0,1)
```



Poisson Model for Rates

Up until now, we used the following specification for poisson regression, where $log(B)$ is the offset term.

$$log(A) = \beta_0 + \beta_1 X_1 + \ldots + \beta_k X_k + log(B)$$

We chose this specification to show that log odds can be modeled using either logistic regression or poisson regression. However, seldomly do we care about the log odds, we care about the predicted probability much more. We can specify a different poisson regression that directly models probability rather than log odds.

$$log(p) = \alpha_0 + \alpha_1 X_1 + \ldots + \alpha_k X_k$$

$$log\left(\frac{A}{A+B}\right) = \alpha_0 + \alpha_1 X_1 + \ldots + \alpha_k X_k$$

$$log(A) = \alpha_0 + \alpha_1 X_1 + \ldots + \alpha_k X_k + log(A+B)$$

The offset term is now $log(A+B)$, or the log of the total number of observations in the cell. However, now we have a problem, probability should always be between 0% and 100%, but the new specification allows p to be any real number greater than or equal to 0. If you are strictly predicting probability, then the new poisson model specification may be a poor choice. If you are predicting a rate that ranges from 0 to +Inf, then this may be a good choice.

```
poissonRate.mod <- glm(card_yes ~  reports + age + income +
                     offset(log(card_no + card_yes)), data=CreditCardCount, family=poiss
on)

pander::pander(poissonRate.mod, caption="")
```

	Estimate	Std. Error	z value	Pr(>|z|)
reports	-0.3115	0.05143	-6.058	1.382e-09
age	0.000358	0.00441	0.08119	0.9353
income	0.03875	0.02747	1.411	0.1583
(Intercept)	-0.2897	0.1461	-1.983	0.04734

The probability predictions from the log-odds poisson and rate poisson are similar The new poisson model requires a default value for both `card_yes` and `card_no`. They will both be set to 0.5.

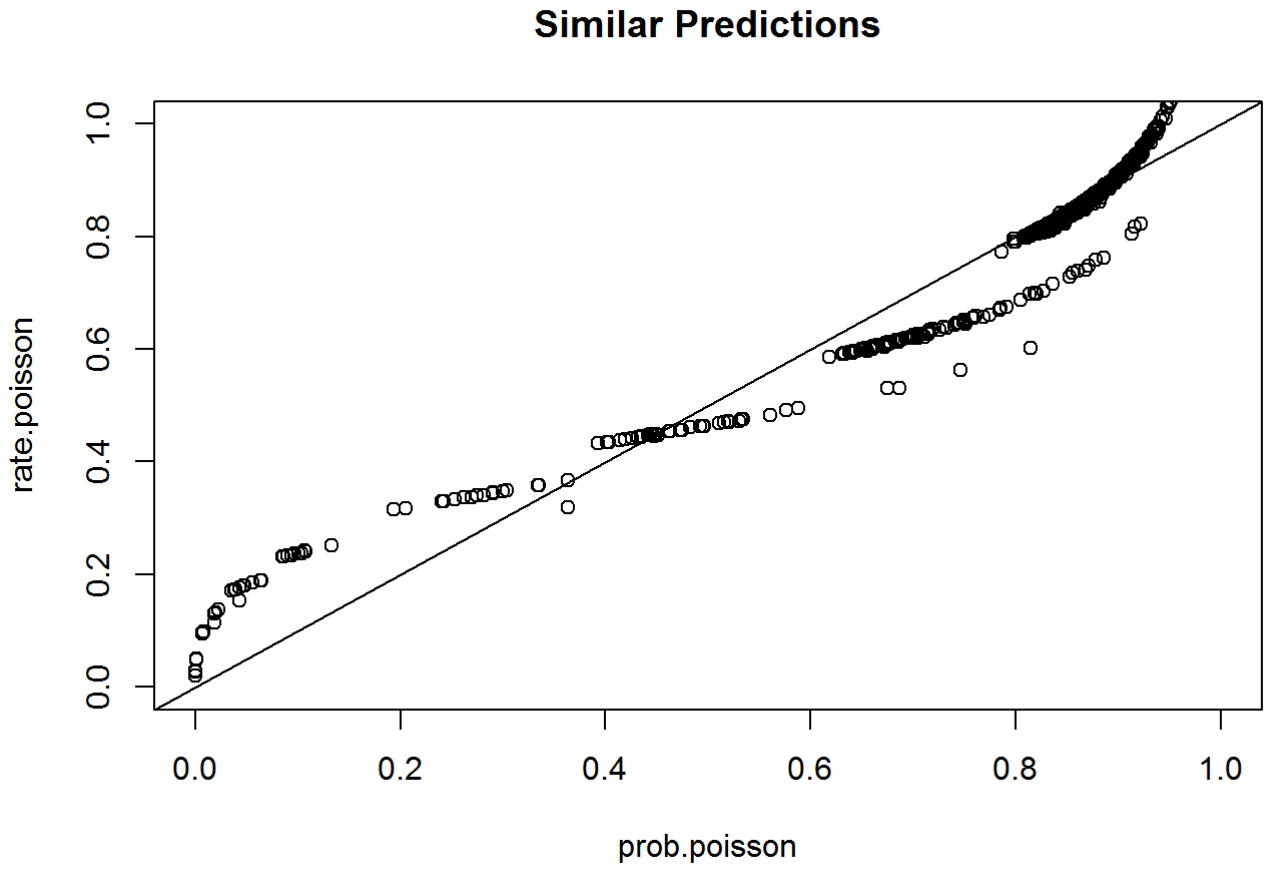
```
CreditCard$card_no <- 0.5
CreditCard$card_yes <- 0.5

poissonRate.link.pred <- predict(poissonRate.mod, newdata=CreditCard, type="link")
poissonRate.prob.pred <- exp(poissonRate.link.pred)

prob.predictions$rate.poisson <- poissonRate.prob.pred

pander::pander(head(prob.predictions), caption="")
```

	prob.logit	prob.poisson	prob.binomial	rate.poisson
1	0.9072	0.8979	0.8981	0.9039
2	0.8429	0.8449	0.8421	0.8319
3	0.9071	0.8959	0.8965	0.9019
4	0.8477	0.8468	0.8446	0.835
6	0.8472	0.8415	0.8404	0.8315
7	0.8937	0.8817	0.8824	0.8814

```
pander::pander(summary(prob.predictions), caption="")
```

prob.logit	prob.poisson	prob.binomial	rate.poisson
Min. :0.0000003	Min. :0.0000507	Min. :0.0000567	Min. :0.01991
1st Qu.:0.8139007	1st Qu.:0.8199471	1st Qu.:0.8171781	1st Qu.:0.80648
Median :0.8489203	Median :0.8480541	Median :0.8459902	Median :0.83575
Mean :0.7751351	Mean :0.7930543	Mean :0.7922007	Mean :0.78835
3rd Qu.:0.8819971	3rd Qu.:0.8775204	3rd Qu.:0.8762855	3rd Qu.:0.87101
Max. :0.9896980	Max. :0.9816660	Max. :0.9833133	Max. :1.23761

```
with(prob.predictions, plot(prob.poisson, rate.poisson, main="Similar Predictions",
                            xlim=c(0,1), ylim=c(0,1)))
abline(0,1)
```




```
similarRate <- caret::R2(prob.predictions$rate.poisson, prob.predictions$prob.poisson
                     , formula="traditional")

similarRate2 <- caret::R2(prob.predictions$rate.poisson, prob.predictions$prob.poisson
                     , formula="corr")
```

R^2 between log-odds poisson (converted to probability) and rate poisson is 94.17% or 94.24%.

Out-of-Sample Performance

We compared the in-sample probability predictions between 4 models:

1. Binary logistic regression
2. Poisson regression (log-odds)
3. Binomial logitic regression
4. Poisson regression (rate)

Predictions between models are similar but not the same. Now we will examine each model's ability to predict the dependent variable, `card`.

For the first 3 models, set `card_no` to 1 in the test data set. Then run the `predict` function for each model.

```
testData$card_no <- 1

logit.prob.oos <- predict(logit.mod, newdata=testData, type="response")

poisson.link.oos <- predict(poisson.mod, newdata=testData, type="link")
poisson.prob.oos <- gtools::inv.logit(poisson.link.oos)

binomial.prob.oos <- predict(binomial.logit.mod, newdata=testData, type="response")
```

For the final model, set both `card_no` and `card_yes` to 0.5.

```
testData$card_no <- 0.5
testData$card_yes <- 0.5

poissonRate.link.oos <- predict(poissonRate.mod, newdata=testData, type="link")
poissonRate.prob.oos <- exp(poissonRate.link.oos)
```

Now we can examine the out-of-sample performance of each model using ROC cruves.

```
library(pROC)

roc.logit <- roc(testData$card ~ logit.prob.oos)
roc.poisson <- roc(testData$card ~ poisson.prob.oos)
roc.binomial <- roc(testData$card ~ binomial.prob.oos)
roc.poissonrate <- roc(testData$card ~ poissonRate.prob.oos)

plot(roc.logit, legacy.axes=TRUE, print.auc=TRUE, print.auc.x=0.2,
     print.auc.y=0.4, main="Out-of-Sample Performance")
```

```
##
## Call:
## roc.formula(formula = testData$card ~ logit.prob.oos)
##
## Data: logit.prob.oos in 65 controls (testData$card no) < 222 cases (testData$card ye
s).
## Area under the curve: 0.7656
```

```
plot(roc.poisson, add=TRUE, print.auc=TRUE, col='blue',
     print.auc.x=0.2, print.auc.y=0.35)
```

```
##
## Call:
## roc.formula(formula = testData$card ~ poisson.prob.oos)
##
## Data: poisson.prob.oos in 65 controls (testData$card no) < 222 cases (testData$card y
es).
## Area under the curve: 0.7622
```

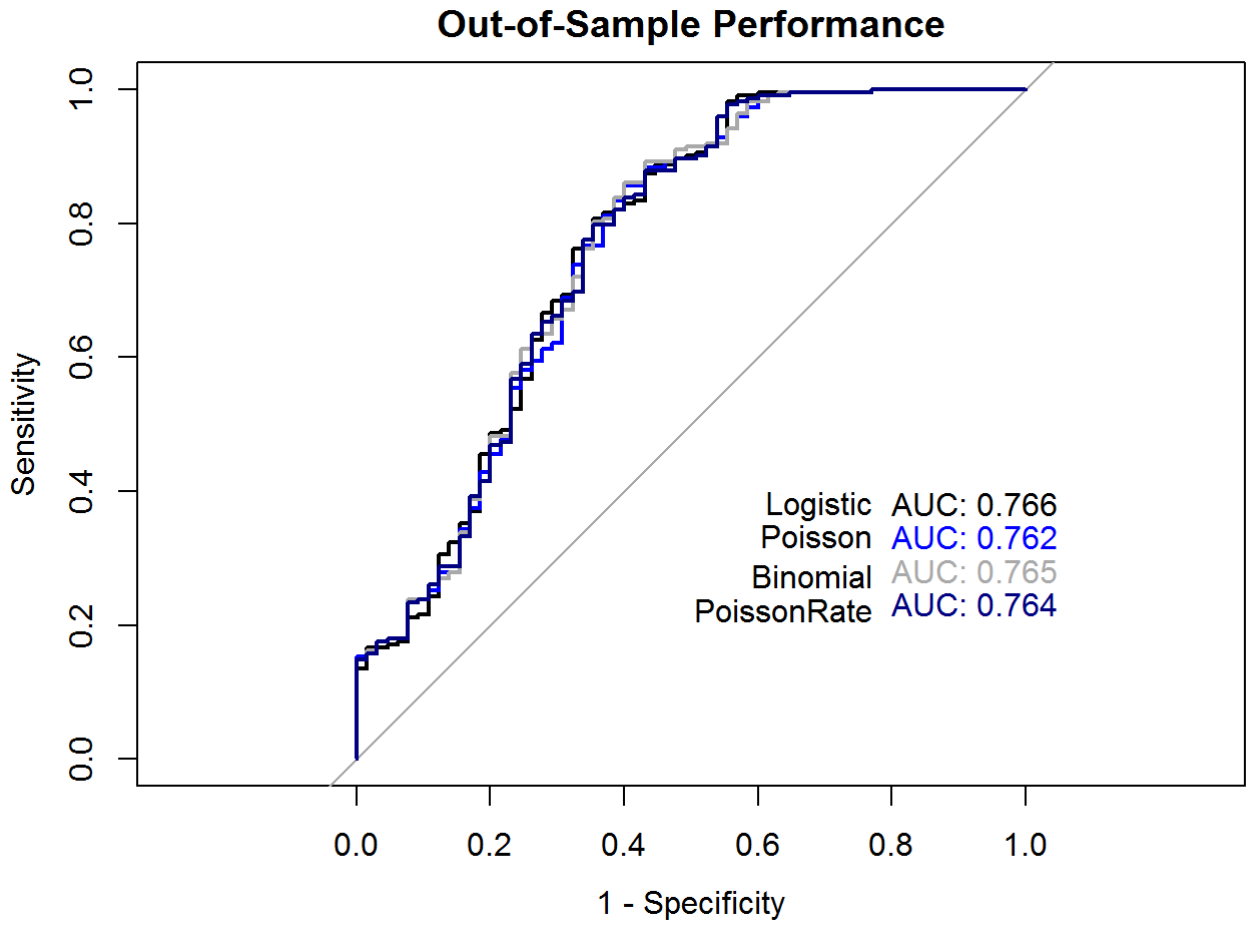
```
plot(roc.binomial, add=TRUE, print.auc=TRUE, col='darkgray',
     print.auc.x=0.2, print.auc.y=0.3)
```

```
##
## Call:
## roc.formula(formula = testData$card ~ binomial.prob.oos)
##
## Data: binomial.prob.oos in 65 controls (testData$card no) < 222 cases (testData$card
yes).
## Area under the curve: 0.7653
```

```
plot(roc.poissonrate, add=TRUE, print.auc=TRUE, col='navy'
     , print.auc.x=0.2, print.auc.y=0.25)
```

```
##
## Call:
## roc.formula(formula = testData$card ~ poissonRate.prob.oos)
##
## Data: poissonRate.prob.oos in 65 controls (testData$card no) < 222 cases (testData$ca
rd yes).
## Area under the curve: 0.7643
```

```
text(x=0.2, y=0.38, "Logistic", pos=2)
text(x=0.2, y=0.33, "Poisson", pos=2)
text(x=0.2, y=0.27, "Binomial", pos=2)
text(x=0.2, y=0.22, "PoissonRate", pos=2)
```



Binary logistic regression has the best out-of-sample performance, but only by a small amount compared to the other 3 models.

Conclusion

Logistic regression and poisson regression are similar. The paper demonstrated an algebraic relationship between probability, logit, and log odds. Using the definition of log odds, we demonstrated that the parameters of the model can be estimated using either logistic or poisson regression.

Although the parameter estimates are not identical between logistic and poisson regression (due to the use of binning in poisson regression), the probability predictions between the two models are similar.

---

1. There are multiple definitions of R^2. The traditional formula penalizes for deviating from the 45-degree black line. The correlation formula adjusts the slope and intercept of the black line and returns an $R^2 = 96.99\%$.↩