Predictive models for P&C insurance

Himchan Jeong

University of Connecticut

Data Science in Action

10 April, 2018

Purpose of the Project

- Introduce current practice done by property and casualty (P&C) insurance company
- Suggest the more sophisticated predictive model which can outperform the benchmarks

Current Approches for Claim Modeling

- (1) Two-parts model for frequency and severity
- (2) Tweedie model

Pitfalls in Current Practices

- (1) Dependence between the frequency and the severity
- (2) Longitudinal property of data structure.
  - For example, if we observed a policyholder i for T_i years, then we have following observation $N_{i1}, N_{i2}, \ldots, N_{iT_i}$, which may not be identically and independently distributed.

Possible Alternatives for the Benchmarks

- For dependence between the frequency and severity
    - Set $\mathbb{E}\left[\overline{C}|N\right] = e^{X\beta+N\theta}$
- For longitudinal property
    - Random effects model
- Non-traditional approaches
    - Neural network
    - Regression for each group classified by decision tree

Data Description

- Here I use a public dataset on insurance claim, provided by Wisconsin Propery Fund.
  (https://sites.google.com/a/wisc.edu/jed-frees/)
- It consists of 5,677 observation in traning set and 1,098 observation in test set.
- It is a longitudinal data with more or less 1,234 policyholder, followed for 5 years.
- Since the dataset includes information on multi-line insurance, here I used building and contents (BC), inland marine (IM), and new motor vehicle (PN) claim information.

Observable Policy Characteristics used as Covariates

Categorical variables	Description		Proportions
TypeCity	Indicator for city entity:	Y=1	14 %
TypeCounty	Indicator for county entity:	Y=1	5.78 %
TypeMisc	Indicator for miscellaneous entity:	Y=1	11.04 %
TypeSchool	Indicator for school entity:	Y=1	28.17 %
TypeTown	Indicator for town entity:	Y=1	17.28 %
TypeVillage	Indicator for village entity:	Y=1	23.73 %
NoClaimCreditBC	No BC claim in prior year:	Y=1	32.83 %
NoClaimCreditIM	No IM claim in prior year:	Y=1	42.1 %
NoClaimCreditPN	No PN claim in prior year:	Y=1	10.96 %

Continuous variables		Minimum	Mean	Maximum
CoverageBC	Log coverage amount of BC claim in mm	0	37.05	2444.8
lnDeductBC	Log deductible amount for BC claim	0	7.14	11.51
CoverageIM	Log coverage amount of IM claim in mm	0	0.85	46.75
lnDeductIM	Log deductible amount for IM claim	0	5.34	9.21
CoveragePN	Log coverage amount of PN claim in mm	0	0.16	25.67

Summary Statistics for Frequency

		Minimum	Mean	Variance	Maximum
FreqBC	number of BC claim in a year	0	0.88	37.31	231
FreqIM	number of IM claim in a year	0	0.06	0.1	6
FreqPN	number of PN claim in a year	0	0.16	0.92	19

In terms of frequency, IM has relatively moderate dispersion of the number of claim per year, whereas BC has very wide range. Usually, dataset used to calibrate two-parts GLM in practice rarely contains a policy which has more than six claims in a year. So we may need a different methodology for modelling such unusual high frequency.

Summary Statistics for Frequency (Cont'D)

Table 1: Distribution of frequency per claim type

Count	BC	IM	PN
0	3993	5441	5360
1	997	182	155
2	333	40	51
3	136	6	33
4	76	4	19
5	31	2	16
6	19	2	13
7	19	0	7
8	16	0	4
9	5	0	4
$>$9	52	0	15

Summary Statistics for Severity

		Minimum	Mean	Variance	Maximum
log(yAvgBC)	(log) avg size of BC claim in a year	5.17	8.76	1.86	16.37
log(yAvgIM)	(log) avg size of IM claim in a year	4.09	8.45	2.23	13.09
log(yAvgPN)	(log) avg size of PN claim in a year	3.56	7.63	1.22	10.71

Entertained Models

- Independent Two-parts [$\mathbb{E}\left[C|n\right] = \exp(X\beta)$]:
  Poisson-Gamma GLM, ZIP-Gamma GLM, neural network
- Dependent Two-parts [$\mathbb{E}\left[C|n\right] = \exp(X\beta + n\theta)$]:
  Poisson-Gamma GLM, ZIP-Gamma GLM, neural network
- One-part [$\mathbb{E}\left[S\right] = \exp(X\eta)$]:
  Tweedie GLM, neural network

Fitting Frequency in Neural Network

```r
library(nnet)
ltFreq <- train$FreqBC
adjltFreq <- (ltFreq - min(ltFreq)) /
             (max(ltFreq)-min(ltFreq))
nnet.freq.fit<- nnet(adjltFreq~.,data=train[-c(1,2,3,6,13,14
                 )], size=10,linout=TRUE,trace=FALSE)
nnet.freq.pred <- predict(nnet.freq.fit, newdata = test
                 [-c(1,2,3,6,13,14)])*(max(ltFreq)
                 -min(ltFreq))+min(ltFreq)
```

Fitting Average Severity in Neural Network

```
trainp <- subset(train,log(yAvgBC)>0)
ltClaim <- log(trainp$yAvgBC)
adjltClaim <- (ltClaim - min(ltClaim))/
              (max(ltClaim)-min(ltClaim))
nnet.avgsev_dep.fit<- nnet(adjltClaim~.,data=trainp[-c(1,2,3,
                      6,13)],size=10,linout=TRUE,trace=FALSE)
nnet.avgsev_indep.fit <- nnet(adjltClaim~.,data=trainp
                        [-c(1,2,3,6,13,14)],size=10
                        ,linout=TRUE,trace=FALSE)
```

Fitting Average Severity in Neural Network

```
trainp <- subset(train,log(yAvgBC)>0)
ltClaim <- log(trainp$yAvgBC)
adjltClaim <- (ltClaim - min(ltClaim))/
              (max(ltClaim)-min(ltClaim))
nnet.avgsev_dep.fit<- nnet(adjltClaim~.,data=trainp[-c(1,2,3,
                      6,13)],size=10,linout=TRUE,trace=FALSE)
nnet.avgsev_indep.fit <- nnet(adjltClaim~.,data=trainp
                        [-c(1,2,3,6,13,14)],size=10
                        ,linout=TRUE,trace=FALSE)
```

Fitting Aggregate Claim in Neural Network

```
tClaim <- log(train$ClaimBC+1)
adjtClaim <- (tClaim - min(tClaim))/(max(tClaim)-min(tClaim))
nnet.tsev.fit <- nnet(adjtClaim~.,data=train[-c(1,2,3,6,
                     13,14)],size=5,linout=TRUE,trace=FALSE)
```

Retrieving Prediction from fitted NN Model

When I got (a few) negative predictive values for frequency and total claim, I rounded up those values to 0.

```
nnet.freq.pred <- predict(nnet.freq.fit, newdata =
                  test[-c(1,2,3,6,13,14)])*
                  (max(ltFreq)-min(ltFreq))+min(ltFreq)
nnet.freq.pred <- nnet.freq.pred *(nnet.freq.pred >0)

nnet.tsev.pred <- exp(predict(nnet.tsev.fit, newdata =
                  test[-c(1,2,3,6,13,14)])*
                  (max(tClaim)-min(tClaim)))-1
nnet.tsev.pred <- nnet.tsev.pred * (nnet.tsev.pred > 0)
```

Retrieving Prediction from fitted NN Model (Cont'D)

In case of dependent two-part NN, we need to use n as a covariate so we need to first estimate n with fitted frequency model.

```
glm.avgsev_indep.pred <- exp(predict(glm.avgsev_indep,
                          test[-c(1,2,3,6,13,14)]))
test.avg.pois <- cbind(test[-c(1,2,3,6,13,14)],
                  glm.freq.pois_pred)
colnames(test.avg.pois)[9] <- "FreqBC"
glm.avgsev.pois_pred <- exp(predict(glm.avgsev_dep,
                         test.avg.pois))
```

Validation Measures for Model Comparison

- Mean Squared Error
- Gini Index
  - Gini index is equal to $2\times$ the area between the line of equality and the Lorenz curve drawn below.
  - In Lorenz curve, x-coordinate stands for cumulative proportion for number of policyholders, whereas y-coordinates stands for cumulative proportion of actual loss ordered by estimated premium.
  - Therefore, it measures the ability of differentiation of risk per model.

Gini Indices for BC Claim

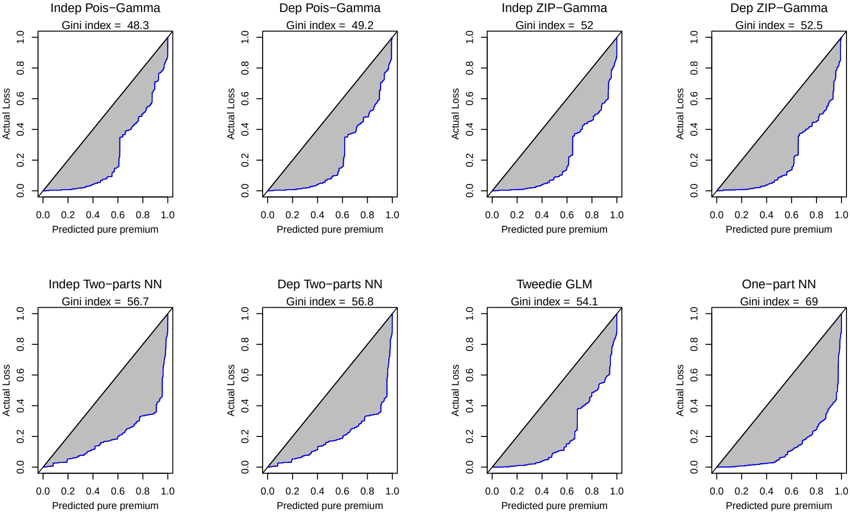

Figure 1: The Lorenz curve and the Gini index values for BC claim

Gini Indices IM Claim

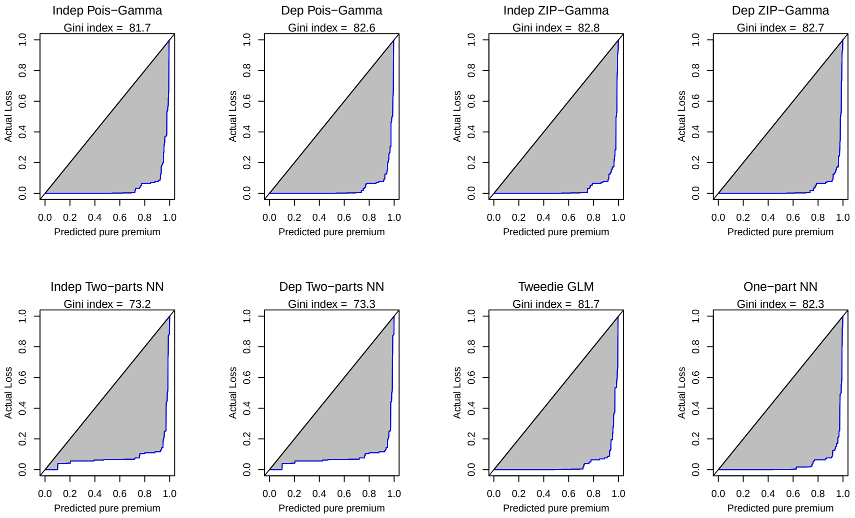

Figure 2: The Lorenz curve and the Gini index values for IM claim

Gini Indices PN Claim

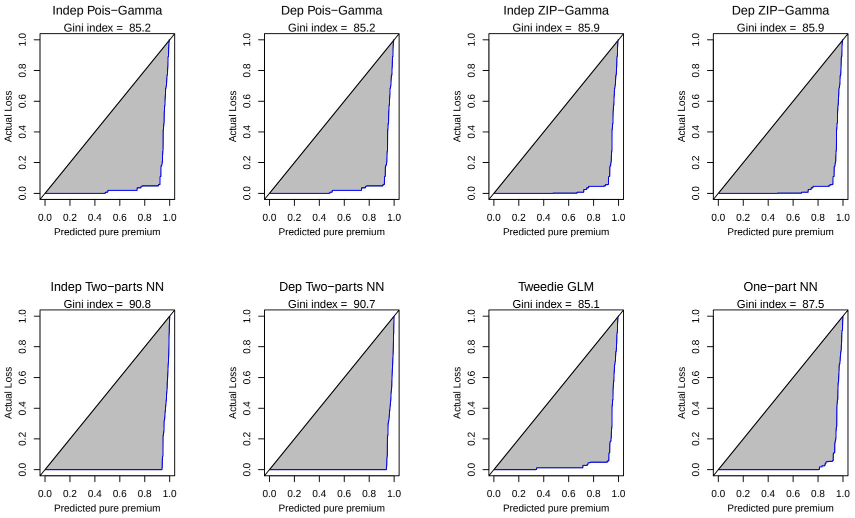

Figure 3: The Lorenz curve and the Gini index values for PN claim

MSEs for all Type of Claim per Model

```
MSEs
```

```
##                          BC        IM       PN
## Indep Pois-Gamma 314562.2 12541.825 4349.914
## Dep Pois-Gamma   183466.2  6647.310 4349.914
## Indep ZIP-Gamma  305939.2  7983.752 3655.829
## Dep ZIP-Gamma    203612.1  6939.829 3672.773
## Indep-2P NN      142480.4  6720.742 4070.260
## Dep-2P NN        142480.2  6720.774 4070.372
## 1P NN            141360.4  6684.799 3983.987
## Tweedie GLM      182278.8 30998.432 4401.668
```

Analysis of the Results

- According to the MSE and Gini indices of given models, in BC claim one part neural network outperforms the other models, whereas two-part dependent GLM was the best for IM and PN claim.
- Note that the difference of performance between neural network and traditional GLM was greater when observed claim had a lot of outlier.
- Therefore, we may consider using neural network for predictive modeling of non-trivial dataset, whereas traditional GLM still works well with trivial dataset.

Need of Compromise between two Objectives

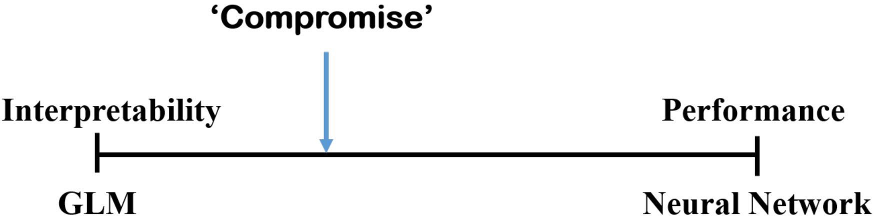

Figure 4: Interpretability and Performance

Future Works for this Project

- Now we have two categories of benchmarks; one is GLM (for interpretability) and the other is neural network.
- Following work should be refining current GLM by incorporating longitudinal property or more sophisticated distrubutional assumption.