Modeling Mortgage Defaults - Part 2: Logistic Transition Models

Objective

To explore the model creation process using real mortgage data to predict a 6-year probability of default.

This module is an extension of Part 1: Logistic Regression. In that module we discussed many of the fundamentals of model creation, including understandging your data source, variable selection, and model choice, with the logistic regression as our primary workhorse. In this module we extend the logistic framework (inherently a single-period format) to a multi-period model.

Model Description

The output of a single logistic regression is the probability of an event occuring sometime within a fixed interval. We can think of arranging several logistic regressions in series. We can look at the probability of an event happening during the first interval and conditioning the event happening in the second interval on it not having happened in the first period, and so on... We can think about the models being linked together, each subsequent model conditioned on no failure (or in our case, default) in any prior period.

How does this compare to running a single-interval regression over the larger interval. An analogy is trying to approximate a curve by a single line joining two endpoints vs. a piece-wise linear approximation, where each 'piece' is another regression.

The intent of this additional complexity is increased precision. The covariate loadings can be different over each interval. There is no reason to assume that you should use a borrower's FICO in the same way to predict a default within years 1 & 2, than to predict a default between years 3 & 4. We want our model to be sensitive to the changing impact of various covariates and we may want a model that allows us to do that.

In this module we'll divide our attention to three 2-year periods, giving us a total of 6 years. How do we decide on intervals of 2 years? We would do some sensitivity testing to identify intervals of relatively stationary betas. In order to find these regions we would need to run numerous regressions over different start & end points trying to identify regions where the coefficients change. We would use statistical tests to help demarcate when coefficients are different. This is a laborious exercise, and economic intuition and product knowledge can be useful. Knowledge of various turning points in a default profile, for example that IO products typically have default spikes at reset periods could help create appropriate intervals.

The 'Normal' Use of Logistic Transition Models

We must admit that creating a logistic transition on data that isn't stationary is not common practice. If we are interested in survival dynamics and the impact of time and changing coefficients other models exist to better accomplish those tasks. We explore one of those models, the hazard rate model, in Part 3. However, adapting this model to a non-stationary setting is interesting, though cumbersome, and yields great intuition for the model in Part 3.

Logistic transition models are used in industry to model stationary transition rates. An example of this can arise when looking at mortgage delinquency and liquidation timelines. Though delinquency roll-rates are certainly influenced by the macroeconomy and the housing cycle (which determine, for example, ease of selling a foreclosed property), there is a large aspect that is independent of the macroeconomy and depends on the servicer's effort. By controlling the degree of 'touch', a servicer can alter transition rates, helping more customers to cure or be modified, or pushing others through the foreclosure process faster. In this servicing context there are many transitions of interest - loans can roll into further delinquency, they can be cured,

they can be modified, or they can be liquidated through a variety of different disposition methods. After we establish that the transition rates do not change over time (a complicated process, similar to determining intervals of stationarity, as described above), we can draw a graph of transition probabilities and estimate each transition seperately. Assuming we have set up the matrix correctly, we can obtain our monthly forecast by simply using matrix multiplication. For a one-period lookahead, we multiply our current allocation (a vector), by the transition matrix A to obtain next month's vector. For a two-period lookahead, we would multiply our current allocation by A^2, etc.

An Introduction to Competing Risks

When we estimate model there are some additional considerations. There are 2 absorbing states in the context of our model, default and prepayment. A loan can remain as it is, current, or it can exit the pool through default or prepayment. If we were to only model exits from the pools via default then the number of loans remaining after interval 1 would be higher than actual since we modeled only part of the exits. Apply period 2 defaults to this higher number would be incorrect because we woud not expect the remaining pool to be so large.

To accurately model defaults over time we therefore also need to model the other absorbing states. This type of model is called a competing risks model where an observation can experience multiple transitions (though they don't need to be absorbing) and we are interested in a later accumulation to a certain state. As a result, we also need to estimate prepayment models for each interval. Modeling three intervals requires us to estimate 3 x 2 models, since we have 2 possible exit states, where each model is dedicated to calculating the probability of reaching a certain state.

It's helpful to highlight something we didn't pay much attention to in Part 1, mainly because it wasn't overly relevant, but that will become more relevant in Part 3. When estimating a model concerned with a particular event type we essentially ignore the other events types regardless of what they are, we can do this our context because the events are mutually exclusive. When estimating our 6-year default probability model in Part 1 we were unconcerned with all other event types - prepayment, lack of reporting, or any other reason for dropoff. In this way the models look at the likelihood of each event type seperately. In our current context exits due to prepayment won't be distinguished from other non-defaults in our default model, and defaults will be treated as other non-prepays in our prepayment model.

Plan

In this module we will set up a logistic transition framework for predicting defaults, and explore the sensitivity to changes in the covariates across the different intervals.

We estimate the 3 defaults models and 3 prepayment models. We connect the models and calculate the overall default rate by conditioning on the outcome over the prior interval.

$$P(D \in (a,b])) = P(D \in (a,b] | NoD \in (0,a])$$

Where D = Default. And we sum this across all three intervals, (0,2], (2,4], (4,6].

```
LogTrain <- extractCovariates(LogTrain)
LogTest <- extractCovariates(LogTest)

#We estimate the models and output them into a list.
Models <- TrainModels(LogTrain)

#Let's look at the default coefficients for the transition model
rbind(coef(Models[[1]]),coef(Models[[3]]),coef(Models[[5]]))
```

```
##      (Intercept)      OCLTV     CSCORE_B      SATO    HPI.Y1.2
## [1,]  12.0301935 0.04461832 -0.008511760 1.7957033 -16.6676451
## [2,]   2.0291295 0.03952537 -0.013907729 0.9918423  -0.7295392
## [3,]   0.6329959 0.05137701 -0.008840453 0.4287963  -3.0477161
```

Interpreting the Coefficients

The default betas vary significantly between the three periods. The effect of SATO on default probability decreases over time. This makes sense, and we may in fact expect a similar relationship to hold for all origination characteristics, they grow stale and less predictive over time. The impact of HPI on the 3 intervals is very different. This is likely due to the time period we selected for analysis. Home prices during the first two years after origination, for most of the loans, grew significantly. We censored the variable to not exceed 100% (only looking for downward movement). There was very little downward HPI movement during those first two years, meaning there was very little variance in the covariate making the coefficient very sensitive which is what we see.

The change in loadings of OCLTV and CSCORE_B are slightly confusing. To add context let's think of the first two-year interval as primarily one of home price growth and the next two as a worsening decline. The Credit Score has a greater impact over years 3&4 than 1&2, and then decreases signficantly for years 5&6. A possible story we could test is that though we expect the associated effect of origination characteristics to dampen over time, in a downward economic environment (which incidentally affected the majority of the loans in our dataset in years 3&4) the predictive power of a credit score is higher than in an upward or flat economic environment.

We should view these betas in conjunction with those estimated from the prepayment models to make sure any intuitions we gain are consistent across the models.

Model Sensitivity Analysis

With our current setup we can do something we were unable to do in Part 1, we can look at the effect of a given HPI drop happening in the first, second, or third interval. Below we look at default probability curves for a 30% HPI drop, occuring in the first, second, and third interval. We also compare this to the output of our model from Part 1.

```
Logistic2 <- glm(RESPONSE_LOG ~ OCLTV + CSCORE_B + SATO + HPI_CHANGE,
    data = LogTrain, family = "binomial")

# generate new data
FICO_Log2 <- expand.grid(OCLTV = 75, CSCORE_B = seq(600, 750,
    length.out = 100), SATO = 0, HPI_CHANGE = 0.7, HPI.Y1.2 = NA,
    HPIY1to4 = NA, HPIY1to6 = NA, model = "Logistic2")
FICO_LogTrans_HPIEnd <- expand.grid(OCLTV = 75, CSCORE_B = seq(600,
    750, length.out = 100), SATO = 0, HPI_CHANGE = NA, HPI.Y1.2 = 1,
    HPIY1to4 = 1, HPIY1to6 = 0.7, model = "LogTrans-HPI Change End")
FICO_LogTrans_HPIBeg <- expand.grid(OCLTV = 75, CSCORE_B = seq(600,
    750, length.out = 100), SATO = 0, HPI_CHANGE = NA, HPI.Y1.2 = 0.7,
    HPIY1to4 = 0.7, HPIY1to6 = 0.7, model = "LogTrans-HPI Change Start")
FICO_LogTrans_HPIMid <- expand.grid(OCLTV = 75, CSCORE_B = seq(600,
    750, length.out = 100), SATO = 0, HPI_CHANGE = NA, HPI.Y1.2 = 1,
    HPIY1to4 = 0.7, HPIY1to6 = 0.7, model = "LogTrans-HPI Change Mid")

# add responses
FICO_Log2["RESPONSE"] <- 100 * predict.glm(Logistic2, type = "response",
    newdata = FICO_Log2)
```

```
FICO_LogTrans_HPIBeg["RESPONSE"] <- LogTransitionProb(Models,
    FICO_LogTrans_HPIBeg)
```

```
## The cumulative default rate is  15.1602%
```

```
FICO_LogTrans_HPIEnd["RESPONSE"] <- LogTransitionProb(Models,
    FICO_LogTrans_HPIEnd)
```

```
## The cumulative default rate is  2.7302%
```

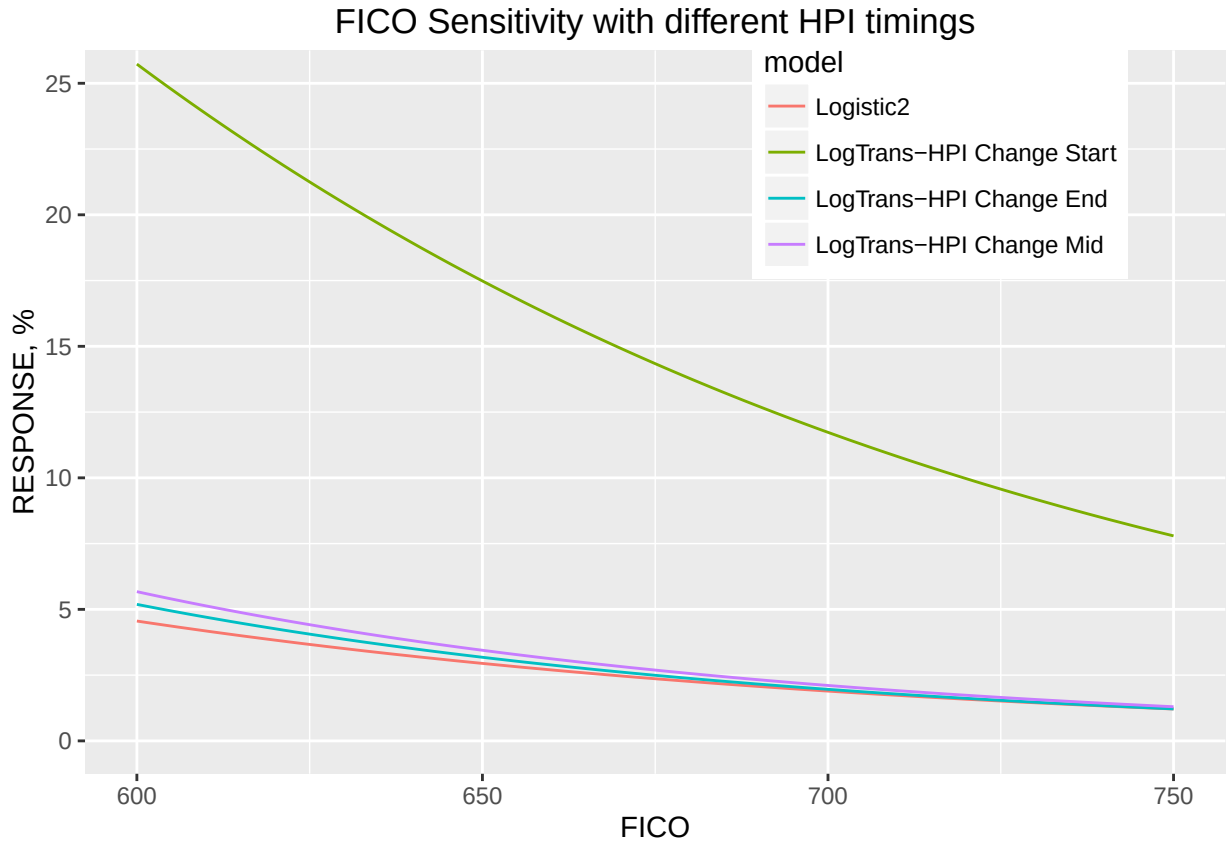
```
FICO_LogTrans_HPIMid["RESPONSE"] <- LogTransitionProb(Models,
    FICO_LogTrans_HPIMid)
```

```
## The cumulative default rate is  2.9565%
```

```
Output <- rbind(FICO_Log2, FICO_LogTrans_HPIBeg, FICO_LogTrans_HPIEnd,
    FICO_LogTrans_HPIMid)

suppressWarnings(library(ggplot2))

ggplot(Output, aes(x = CSCORE_B, y = RESPONSE, colour = model,
    group = model)) + geom_line() + ggtitle("FICO Sensitivity with different HPI timings") +
    coord_cartesian(ylim = c(0, 25)) + ylab("RESPONSE, %") +
    xlab("FICO") + theme(legend.position = c(0.75, 0.85))
```



How should we interpret these sensitivities? Most of the curves are clustered together except for the curve corresponding to an HPI drop happening in the first interval (recall, this is an interval of increasing national home prices). There does not seem to be a great effect on default probability from the decrease in HPI

happening in (2,4] or (4,6]. The impact of the same HPI drop occuring in the first interval is interesting. The default probability of a loan experiencing a severe price decline in an environment of house price growth is drastically increased.

Helper functions

```
# populate additional variables
extractCovariates <- function(df) {
    # if missing DEFAULT.CODE set to 999, and DEFAULT.DATE to
    # 2015-12-01
    df[is.na(df$DEFAULT.CODE), "DEFAULT.DATE"] <- "2015-12-01"
    df[is.na(df$DEFAULT.CODE), "DEFAULT.CODE"] <- 999

    # format as dates
    df$ORIG_DTE <- as.Date(df$ORIG_DTE)
    df$DEFAULT.DATE <- as.Date(df$DEFAULT.DATE)

    df$Y12DEFAULT <- ifelse(df$RESPONSE == 1, 1, 0)
    df$Y34DEFAULT <- ifelse(df$RESPONSE == 2, 1, 0)
    df$Y56DEFAULT <- ifelse(df$RESPONSE == 3, 1, 0)
    df$Y12PREPAY <- ifelse((df$DEFAULT.CODE == 1) & (df$DEFAULT.DATE <=
        (df$ORIG_DTE + 2 * 365)), 1, 0)
    df$Y34PREPAY <- ifelse((df$DEFAULT.CODE == 1) & (df$DEFAULT.DATE >
        (df$ORIG_DTE + 2 * 365)) & (df$DEFAULT.DATE <= (df$ORIG_DTE +
        4 * 365)), 1, 0)
    df$Y56PREPAY <- ifelse((df$DEFAULT.CODE == 1) & (df$DEFAULT.DATE >
        (df$ORIG_DTE + 4 * 365)) & (df$DEFAULT.DATE <= (df$ORIG_DTE +
        6 * 365)), 1, 0)
    df$HPIY1to4 <- df$HPI.Y1.2 * df$HPI.Y3.4
    df$HPIY1to6 <- df$HPIY1to4 * df$HPI.Y5.6
    return(df)
}

TrainModels <- function(df) {

    # CHANGED FUNCTIONS
    MODELDefaultY1to2 <- glm(Y12DEFAULT ~ OCLTV + CSCORE_B +
        SATO + HPI.Y1.2, data = df, family = "binomial")
    MODELDefaultY3to4 <- glm(Y34DEFAULT ~ OCLTV + CSCORE_B +
        SATO + HPIY1to4, data = subset(df, ORIG_DTE + 365 * 2 <
        DEFAULT.DATE), family = "binomial")
    MODELDefaultY5to6 <- glm(Y56DEFAULT ~ OCLTV + CSCORE_B +
        SATO + HPIY1to6, data = subset(df, ORIG_DTE + 365 * 4 <
        DEFAULT.DATE), family = "binomial")

    MODELPrepayY1to2 <- glm(Y12PREPAY ~ OCLTV + CSCORE_B + SATO +
        HPI.Y1.2, data = df, family = "binomial")
    MODELPrepayY3to4 <- glm(Y34PREPAY ~ OCLTV + CSCORE_B + SATO +
        HPIY1to4, data = subset(df, ORIG_DTE + 365 * 2 < DEFAULT.DATE),
        family = "binomial")
    MODELPrepayY5to6 <- glm(Y56PREPAY ~ OCLTV + CSCORE_B + SATO +
        HPIY1to6, data = subset(df, ORIG_DTE + 365 * 4 < DEFAULT.DATE),
        family = "binomial")
```

```
    Models <- list()
    Models[[1]] <- MODELDefaultY1to2
    Models[[2]] <- MODELPrepayY1to2
    Models[[3]] <- MODELDefaultY3to4
    Models[[4]] <- MODELPrepayY3to4
    Models[[5]] <- MODELDefaultY5to6
    Models[[6]] <- MODELPrepayY5to6

    return(Models)
}

LogTransitionProb <- function(Models, data = Models[[1]]$data) {

    DefProb1 <- predict.glm(Models[[1]], data, type = "response")
    DefProb2 <- predict.glm(Models[[3]], data, type = "response")
    DefProb3 <- predict.glm(Models[[5]], data, type = "response")

    PPYProb1 <- predict.glm(Models[[2]], data, type = "response")
    PPYProb2 <- predict.glm(Models[[4]], data, type = "response")
    PPYProb3 <- predict.glm(Models[[6]], data, type = "response")

    EffectiveDef1 <- DefProb1
    EffectiveDef2 <- max(0, 1 - DefProb1 - PPYProb1) * DefProb2
    EffectiveDef3 <- max(0, 1 - DefProb1 - PPYProb1) * max(0,
        1 - DefProb2 - PPYProb2) * DefProb3

    cat("The cumulative default rate is ", paste(round(100 *
        mean(EffectiveDef1 + EffectiveDef2 + EffectiveDef3),
        4), "%", sep = ""))

    invisible(100 * (EffectiveDef1 + EffectiveDef2 + EffectiveDef3))
}
```