Enhancement of Heterogeneity Testing for IRB Models

Statistical Power Analysis

Andrija Djurovic

www.linkedin.com/in/andrija-djurovic

Model Heterogeneity

- A typical step in building credit risk models is discretizing the model output into ratings, pools, or buckets.
- Practitioners generally follow established principles for this discretization.
- These principles result in specific characteristics, some of which are mandatory, while others vary by model and are desirable but not essential.
- Monotonicity and heterogeneity are typically regarded as mandatory characteristics.
- In this context, heterogeneity refers to adequate differentiation in risk profiles across ratings, pools, or buckets. It is commonly tested in Probability of Default (PD), Loss Given Default (LGD), and Exposure at Default (EAD) models, often using tests like the two-proportion test and t-test.
- Heterogeneity is assessed during the initial model validation and monitored over time.

Practical Challenges in Heterogeneity Testing

- A common approach to heterogeneity testing relies on observed (realized) average values between adjacent grades, pools, or buckets.
- Additionally, heterogeneity is often tested at the overall modeling dataset level during initial model validation and at a specific reference date or across a few consolidated reference dates during periodic model validation.
- Given the different possible levels of testing, practitioners often face challenges in assessing overall heterogeneity. A common approach is to define a percentage of pairs for which the specified test fails.
- Analyzing the common approach used in practice, a few questions arise:
  - Is having a unified threshold for each portfolio type or rating scale/pools sufficient?
  - Is it sufficient to assess heterogeneity using only observed averages?
  - If the model is well-calibrated, what are the conclusions of heterogeneity testing?
  - If the model is well-calibrated, what is the probability of observing differences between adjacent grades, pools, or buckets?
  - What statistical methods can help practitioners support heterogeneity analysis?
- The following slides present a simplified simulation using statistical power analysis to support heterogeneity testing by providing the probabilities of identifying the difference between the PDs of adjacent rating grades in the case of a well-calibrated model. Building on the results from the power analysis, the final slide presents the probabilities of failure for different numbers of rating pairs in the PD model. Practitioners are encouraged to adjust the simulation setup to reflect specific assumptions and to combine the effect of heterogeneity shortfall with the potential impact of the model's lack of predictive ability.
- A similar approach can be used to calculate the probabilities of monotonicity disruption for different numbers of rating, pool, or bucket pairs.
- Enhancing the standard approach to heterogeneity testing can help practitioners better understand this aspect of model validation and support conclusions with additional valuable analysis.

Simulation Setup

Dataset

The dataset used for the following simulation is shown below:

```
##  rating   no nb    odr     pd  p.val
##     RG1 1500  6 0.0040 0.0057     NA
##     RG2 1920 14 0.0073 0.0105 0.1051
##     RG3 2925 36 0.0123 0.0169 0.0455
##     RG4 4515 95 0.0210 0.0310 0.0026
##     RG5 2535 90 0.0355 0.0530 0.0001
##     RG6 1365 83 0.0608 0.0793 0.0001
##     RG7   91  9 0.0989 0.1451 0.0741
##     RG8  148 26 0.1757 0.2590 0.0515
```

where:

- `rating` denotes the rating grade;
- `no` represents the number of obligors per rating grade;
- `odr` is the observed default rate per rating grade;
- `pd` is the calibrated PD estimate;
- `p.val` is the p-value of the heterogeneity test (test of two proportions) for adjacent rating pairs.

Statistical Power Analysis

- The following slides demonstrate the calculation of statistical power for adjacent rating grades under the assumption that the model is well-calibrated (i.e., the true PDs are equal to the calibrated PDs).
- In this context, statistical power provides insights into the probability of correctly identifying differences between PDs in adjacent pairs if the true PDs match the calibrated ones.
- Results are presented using an analytical solution and a Monte Carlo simulation.
- Finally, the statistical power results from the Monte Carlo simulation are used to calculate and graphically present the probability of heterogeneity failure for different numbers of rating pairs.

R Code

```r
#utils functions import
source("https://raw.githubusercontent.com/andrija-djurovic/adsfcr/refs/heads/main/mrm/utils.R")

#data import
url <- "https://raw.githubusercontent.com/andrija-djurovic/adsfcr/refs/heads/main/mrm/rs.csv"
rs <- read.csv(file = url,
               header = TRUE)

#statistical power - analytical solution
rs$power.a <- NA
for (i in 2:nrow(rs)) {
      rs$power.a[i] <- ht.power.a(n1 = rs$no[i-1],
                                  n2 = rs$no[i],
                                  p1 = rs$pd[i-1],
                                  p2 = rs$pd[i],
                                  alpha = 0.05)
      }
rs$power.a <- round(x = rs$power.a,
                    digits = 4)
#print rs data frame
rs
```

```
##   rating   no nb    odr     pd  p.val power.a
## 1    RG1 1500  6 0.0040 0.0057     NA      NA
## 2    RG2 1920 14 0.0073 0.0105 0.1051  0.4512
## 3    RG3 2925 36 0.0123 0.0169 0.0455  0.5775
## 4    RG4 4515 95 0.0210 0.0310 0.0026  0.9882
## 5    RG5 2535 90 0.0355 0.0530 0.0001  0.9970
## 6    RG6 1365 83 0.0608 0.0793 0.0001  0.9350
## 7    RG7   91  9 0.0989 0.1451 0.0741  0.6700
## 8    RG8  148 26 0.1757 0.2590 0.0515  0.6777
```

R Code cont.

```r
#statistical power - mc simulation
rs$power.s <- NA
for   (i in 2:nrow(rs)) {
      rs$power.s[i] <- ht.power.s(n1 = rs$no[i-1],
                                  n2 = rs$no[i],
                                  p1 = rs$pd[i-1],
                                  p2 = rs$pd[i],
                                  alpha = 0.05,
                                  sim = 1e4,
                                  seed = 2211 + i)
      }
rs$power.s <- round(x = rs$power.s,
                    digits = 4)
#print rs data frame
rs
```

```
##   rating   no nb    odr     pd  p.val power.a power.s
## 1    RG1 1500  6 0.0040 0.0057     NA      NA      NA
## 2    RG2 1920 14 0.0073 0.0105 0.1051  0.4512  0.4550
## 3    RG3 2925 36 0.0123 0.0169 0.0455  0.5775  0.5837
## 4    RG4 4515 95 0.0210 0.0310 0.0026  0.9882  0.9887
## 5    RG5 2535 90 0.0355 0.0530 0.0001  0.9970  0.9969
## 6    RG6 1365 83 0.0608 0.0793 0.0001  0.9350  0.9362
## 7    RG7   91  9 0.0989 0.1451 0.0741  0.6700  0.6618
## 8    RG8  148 26 0.1757 0.2590 0.0515  0.6777  0.6878
```

Python Code

```
import pandas as pd
import numpy as np
from scipy.stats import norm, binom
from statsmodels.stats.proportion import proportions_ztest
import requests
#utils functions import
url = "https://raw.githubusercontent.com/andrija-djurovic/adsfcr/refs/heads/main/mrm/utils.py"
r = requests.get(url)
exec(r.text)
#data import
url = "https://raw.githubusercontent.com/andrija-djurovic/adsfcr/refs/heads/main/mrm/rs.csv"
rs = pd.read_csv(filepath_or_buffer = url)
#statistical power - analytical solution
rs["power_a"] = np.nan
for i in range(1, len(rs)):
    rs.loc[i, "power_a"] = ht_power_a(n1 = rs.loc[i-1, "no"],
                                      n2 = rs.loc[i, "no"],
                                      p1 = rs.loc[i-1, "pd"],
                                      p2 = rs.loc[i, "pd"],
                                      alpha = 0.05)

rs["power_a"] = rs["power_a"].round(4)
#print rs data frame
rs
```

```
##    rating    no  nb     odr      pd   p.val  power_a
## 0     RG1  1500   6  0.0040  0.0057     NaN      NaN
## 1     RG2  1920  14  0.0073  0.0105  0.1051   0.4512
## 2     RG3  2925  36  0.0123  0.0169  0.0455   0.5775
## 3     RG4  4515  95  0.0210  0.0310  0.0026   0.9882
## 4     RG5  2535  90  0.0355  0.0530  0.0001   0.9970
## 5     RG6  1365  83  0.0608  0.0793  0.0001   0.9350
## 6     RG7    91   9  0.0989  0.1451  0.0741   0.6700
## 7     RG8   148  26  0.1757  0.2590  0.0515   0.6777
```

Python Code

```
#statistical power - mc simulation
rs["power_s"] = np.nan
for i in range(1, len(rs)):
    rs.loc[i, "power_s"] = ht_power_s(n1 = rs.loc[i-1, "no"],
                                      n2 = rs.loc[i, "no"],
                                      p1 = rs.loc[i-1, "pd"],
                                      p2 = rs.loc[i, "pd"],
                                      alpha = 0.05,
                                      sim = 10000,
                                      seed = 2211 + i)

rs["power_s"] = rs["power_s"].round(4)
#print rs data frame
rs
```

```
##   rating    no  nb     odr      pd   p.val  power_a  power_s
## 0    RG1  1500   6  0.0040  0.0057     NaN      NaN      NaN
## 1    RG2  1920  14  0.0073  0.0105  0.1051   0.4512   0.4585
## 2    RG3  2925  36  0.0123  0.0169  0.0455   0.5775   0.5933
## 3    RG4  4515  95  0.0210  0.0310  0.0026   0.9882   0.9875
## 4    RG5  2535  90  0.0355  0.0530  0.0001   0.9970   0.9973
## 5    RG6  1365  83  0.0608  0.0793  0.0001   0.9350   0.9408
## 6    RG7    91   9  0.0989  0.1451  0.0741   0.6700   0.6645
## 7    RG8   148  26  0.1757  0.2590  0.0515   0.6777   0.6940
```

Simulation Results