Paper 2731-2018

Evaluation of Different Approaches to Reject Inference: a case study in Credit Risk

Sergiu Luca, Desjardins

ABSTRACT

In the age of Internet of Things (IofT), variables with missing values are part of the norm. With the current available processing power and software that niftily capitalizes on this trend (SAS® being an example of such), hundreds of variables can be considered for a model. Even as such raw data is aggregated and combined in more elegant forms (ex. a ratio of a 12-month averages), the missing value problem tags along nonetheless. This issue is especially pernicious when it comes to the target/explained variable. In Credit Risk Modelling, this is the case for good/bad performance for rejected applications. Such applications are part of the adjudication scorecard dataset. Different approaches are currently in circulation to deal with this data inconvenience. Papers and book chapters discussing these approaches point oftentimes at the scarcity of empirical work related to this problem. The scarcity is indeed intrinsic to the problem. In this paper, data is mined from one such rare portfolio where performance on rejects is observed in order to evaluate empirically three popular approaches used for reject inference: Re-weighting, Re-classification and Fuzzy augmentation.

INTRODUCTION

There is a broad spectrum of opinions on the level of importance reject inference plays in the model development of an adjudication scorecard. The debate stems from the complexity of including a category of credit seekers for which the target performance variable is missing – a logical consequence of an application being rejected. On one side of this spectrum, it is considered that all the necessary and pertinent information for modeling resides with accepted applications (‘accepts’) – for which you observe performance (good or bad) – and, thus, there is little use of rejected applications (‘rejects’); on the other side, rejects comprise a unique information set without which no model will be *complete* due to selection (or sample) bias. This problem was designated a Missing Value Problem and was divided in three categories by (Little & Rubin, D.B., 1987).

MISSING VALUE PROBLEM

Fundamentally, lack of performance for rejects is a missing variable problem. Three types of missing depended variables are considered:

- Missing completely at random (MCAR)
- Missing at random (MAR)
- Missing not at random (MNAR)

On a more intuitive level, the first case covers the situation when there is no adjudication deliberation whatsoever. Applications randomly selected would fall under this category. If, however, for the same risk profile (according to risk behavior data), some applications are accepted while other are rejected, we have ourselves a case of MAR, the second one in order. The last case would be possible under completely manual adjudication or, if a scoring system is in place, under heavy override.

Considering that for the sample at hand, accept/ reject decisions closely followed a scoring system, the reject inference techniques tested aim at solving the MAR problem.

DATA SETUP AND METHODOLOGY

The sample contains 100 000 applications for additional credit on revolving products. This size ought to be more than sufficient to allow for robust results. A hold-out sample representing 30% (30 000) observation was randomly selected from among the total sample.

Key features of the sample data:

- Sample contains observed performance for all applicants regardless of acceptance status, also known as Through the Door (TTD) population.
- Adjudication decisions are based on score reflecting risk behavior of the applicant. Low side overrides are excluded from the sample.

(The following **variables** are part of the data base)

- **IndDef18 (or y):** Performance over 18 months following the decision on the application (0: good; 1: bad)
- **X1 to X40 :** 40 variables representing features of the credit risk behavior of the applicant
- **ACCEPTEE**: Flag to identify accepted (1) and rejected (0) applications

To select among the 40 independent variables, we use the stepwise selection. This method is available in SAS® via the SELECTION= option of the MODEL statement.

The performance of the model will be gauged through the prism of the classification precision / discriminant power. The Area under the Receiver Operating Curve (AuROC) is the aggregate used to quantify the classification precision. Its value varies normally between 0.5 and 1 (though, in theory, the domain of possible values spans from 0 to 1). It is useful to interpret the value of AuROC as the probability that the model will rank a randomly chosen event (i.e. bad) higher than a randomly chosen non-event (i.e. good). At the value 0.5, we have got ourselves a random model.

For an adjudication scorecard, the range of a competitive AuROC goes normally from 0.70 to 0.85. You would rarely encounter an up-and-running adjudication scorecard with AuROC outside this range. Values below and above are rare for different reasons. It is difficult to obtain robust (not over-fitted) models with AuROC above 0.85. A value of 0.70, in contrast, is quite a feasible value; however, a model with such performance will only be accepted under rare circumstances.

MODEL ESTIMATION ON TTD WITH OBSERVED PERFORMANCE FOR REJECTS

This model constitutes our benchmark, against which the performance of other models will be compared. Note that the AuROC of the model is presented and compared in the section **Comparative Analysis.**

The following code was used to fit the model on all applications (accepted and rejected) with known performance:

```
ODS OUTPUT     ROCAssociation = SASGF.E1_DEV_Roc(WHERE=(RocModel='Model'))
               ModelBuildingSummary = SASGF.E1_Model;

proc logistic data=C1_E4_S7030_DEV_WOE_40v NAMELEN=32;

MODEL IndDef18(EVENT='1') = X1-X40/
            selection=stepwise slentry = 0.001 slstay = 0.001;
ROC;
OUTPUT OUT=E1_DEV_estim p=phat xbeta=lnodds;
SCORE DATA=C1_E4_S7030_VLD_WOE_40v OUT=E1_VLD_estim;

run;
```

SLENTRY and SLSTAY specify the minimum significance level of a chi-square distributed statistic, in order to enter and stay in the model, respectively.

OUTPUT statement allows obtaining individual estimates (a column is added with values for each row) for the base sample.

SCORE statement serves a similar purpose as OUTPUT, but for ‘external’ samples (not used in the fitting of the model).

NAMELEN option of the LOGISITIC procedure set at 32 insures that names of the selected model variables in the ODS OUTPUT table MODELBUILDINGSUMMARY are complete (not truncated at 20 characters, as it is the default setting).

The ROC statement outputs the AuROC value. The table ROCASSOCIATION of the ODS OUTPUT is populated upon use of the ROC statement.

The PROC LOGISTIC is used under similar configuration for the rest of the cases when logistic regression is required.

Finally, for the training data, we obtain an AuROC of 0.7532. After 'scoring' the hold-out sample, the AuROC is evaluated at 0.754. The proximity of the values is a reassuring sign of lack of overfitting.

NO REJECT INFERENCE, ACCEPTED APPLICATIONS WITH PERFORMANCE

As it is evident by now, in normal circumstances, developers of adjudication scorecards do not have the luxury of having performance data for refused applications. Under this usual paradigm, one option is to use the accepted observations (for which performance is observed) to fit a model. Given that the performance variable is not a MCAR case, you should expect a weaker model when the development is based solely on Accepted applications.

It is indeed the case, although the difference is not as large as you might expect. When the model fitted on Accepts is applied to accepts, the AuROC is 0.7326. When it is applied to sample of TTD applications, the AuROC is 0.7496. It is the latter result that will be statistically compared with the 'benchmark' in the section Comparative Analysis further along.

REJECT INFERENCE

In this section, three approaches to reject inference are described. On a conceptual level, inference of performance on rejects makes use of accept/reject status and/or good/bad status. The three methods chosen cover all three possible combinations, such as:

1. Re-weighting (augmentation) makes use of accept / reject status
2. Re-classification makes use of both
3. Fuzzy Augmentation/Parceling makes use of good / bad status

RE-WEIGHTING (AUGMENTATION)

This approach consists essentially of assigning a weight to approved observations such as to obtain a distribution that takes into account the contribution of rejects. Re-weighting is an immediate and obvious denomination for this method. The name Augmentation is popular as well, and it is an insight into the fact that normally the weights are such that the weighted sample size increases. The weights take specifically the following form:

$$W_{aug} = \frac{A+R}{A},$$

where *A* stands for accepted application and *R* stands for rejected application. On an intuitive level, this is the inverse of the acceptance rate. Thus, on an individual level *j*, the weight takes the following form:

$$\boldsymbol{W_{aug}(j) = \frac{1}{P(a_j)}} \qquad [1],$$

where $P(a_j)$ is the estimate of the acceptance rate obtained as probability outcome of an accepted/rejected model.

Consequently, we start by fitting a model on TTD training data which will allow estimating the individual probability of being accepted. The resulting model has the AuROC equal to 82%.

```
proc LOGISTIC DATA= C1_E4_S7030_DEV_WOE_40v ;
```

```
      MODEL ACCEPTEE(EVENT='1') = X1-X40 /
            selection=stepwise slentry = 0.001    slstay = 0.001;
      ROC;
      OUTPUT OUT=E3_estim p=P_Accept;
      SCORE DATA=E2_DEV_ACCPT OUT=E3_DEV_ACCPT_augm;
   RUN;
```

The accepted data set is then scored, where each observation obtains the probability of being accepted. A new variable *vw* holds the individual weights.

```
   data     E3_DEV_ACCPT_augm_w;
      FORMAT vw 8.4;
      set E3_DEV_ACCPT_augm;

      vw=1/P_1;
   run;
```

In SAS, weights are easily integrated in the logistic regression through the WEIGHT statement.

The resulting model applied to the TTD training data obtains an AuROC equal to 0.7494; the value is 0.7509 for the hold-out sample.

RE-CLASSIFICATION

Re-classification makes use of the accept/reject model as well, but under a different form and for a different purpose. The goal pursued under Re-classification consists of adding to the observed bads a sub-set of rejects most likely to be bads (note, no goods are added). The accept/reject model serves the purpose of proxy identification of rejects most likely to be bads. Once identified, these rejects-turned-bads ('RTB') are assigned a weight so that their contribution to resulting total bads (observed + inferred bads) is controlled. An important rule of thumb is that the inferred bads should not account for more than the observed bads. A popular value for the proportion of inferred bads to total bads (p) is in fact 30%. In addition to 30%, we have tested a proportion of 50%. Given that the difference between the two is negligible, we have only kept the 30%. The computation of the proportion at hand is given by the following formula:

$$p = \frac{wx}{D_0 + wx},$$

where D_o represents the total observed bads, *x* stands for the number of RTB, and *w* is the calculated weight attributed to *x* in order to obtain *p* (in our case, 30%). Out of this formula, the weight to obtain the desired *p* is calculated:

$$\boldsymbol{w} = \left(\frac{\boldsymbol{p}}{\boldsymbol{1-p}}\right) * \left(\frac{\boldsymbol{D_o}}{\boldsymbol{x}}\right) \quad [2]$$

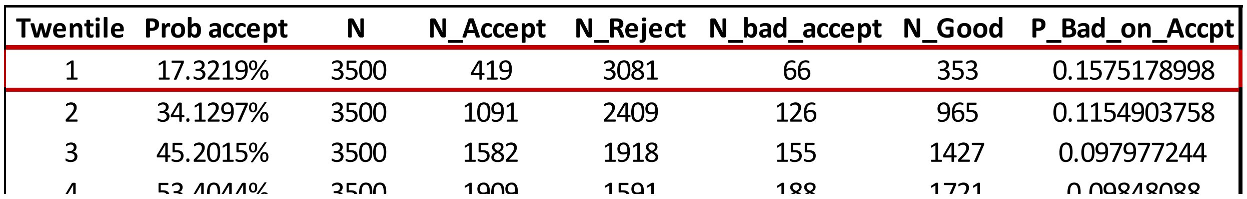
With this weight calculated, a model is fitted on data containing the accepted applications and the RTB weighted.

To apply this reject inference method in practice, we should start by stratifying the TTD sample into groups from less likely to more likely to be accepted. The RANK procedure comes in handy at this job. It is common practice to regroup observations into twentiles (20 groups of 5%).

```
/*a. 20 groups: lowest probability of acceptance to highest*/
   proc rank      data   = E3_estim
                  groups = 20
                  out    = E4a;
                  var      P_accept;
                  ranks    Groups;
   run;
   proc sql;
      create table E4a_table20      as
            select      DISTINCT
                        (Groups+1)          as Twentile
                        ,mean(P_accept)     as P_Accept FORMAT=PERCENT10.4
                        ,count(*)           as N
                        ,sum(ACCEPTEE)      as N_Accept
                        ,count(*) - calculated N_Accept
                                            as N_Reject
                        ,sum(ACCEPTEE*IndDef18)
                                            as N_Bad_Accept
                        ,calculated N_Accept - calculated N_Bad_Accept
                                            as N_Good
                        ,calculated N_Bad_Accept / calculated N_Accept
                                            as P_Bad_on_Accpt
            from        E4a
            GROUP BY    Groups
      ;
   quit;
```

SQL procedure offers a nifty way of aggregating results by group assigned by PROC RANK.

The following output table serves as a basis to identifying the target subset of rejects. The lowest twentile regroups accepts with highest probability of being rejected according to the accept/reject model fitted earlier. Judging by the relative frequency of observed bads among these accepts (bads 0.1575 or 15.75%), the rejects in the lowest twentile are good candidates for RTB's.



Twentile	Prob accept	N	N_Accept	N_Reject	N_bad_accept	N_Good	P_Bad_on_Accpt
1	17.3219%	3500	419	3081	66	353	0.1575178998
2	34.1297%	3500	1091	2409	126	965	0.1154903758
3	45.2015%	3500	1582	1918	155	1427	0.097977244
4	53.4044%	3500	1909	1591	188	1721	0.09848088

Table 1. Snippet from the Accept/Reject Re-classification table

It is useful to verify within the lowest twentile, if there are no pockets of concentrated risk (with significantly higher relative frequency of observed bads). It is not the case with the data at hand. We have thus chosen 3081 rejects as bad.

```
/* b. Data base for model fitting Accepted + rejects infered as bad */

   DATA      E4b;
      FORMAT Bad_Infer Flag_Rej0 2.;
      SET E4a;

      WHERE ACCEPTEE | GROUPS = 0;

      /* All rejects in the riskiest twintile are designated as defaults
      */
      IF GROUPS=0 & ACCEPTEE = 0
            THEN DO; Bad_Infer=1;           Flag_Rej0=1; END;
            ELSE DO; Bad_infer=IndDef18; Flag_Rej0=0; END;
```

```
RUN;

/* c. Set weights */
PROC SQL; select sum(ACCEPTEE*Bad_infer), sum(Flag_Rej0*Bad_infer)
         into   :Do, :Rd
         from E4b;
quit;
%PUT &Do. &Rd.;
```

With the number of RTD established, a weight value is calculated according to the Equation [2] above.

```
DATA     E4c;
   FORMAT vw_30pct vw_50pct 10.4;
   SET E4b;

   IF Flag_Rej0 THEN DO;
         vw_30pct = (0.3/(1-0.3))*(&Do./&Rd.);
         vw_50pct = (0.5/(1-0.5))*(&Do./&Rd.);
   END;
   ELSE DO; vw_30pct=1; vw_50pct=1; END;
RUN;
```

FUZZY AUGMENTATION / PARCELING

Fuzzy augmentation, perhaps more aptly called fuzzy parceling, is perhaps the most intuitive method of the three: a good/bad model fitted on Accepted applications with observed performance is used to obtain probability estimates for the rejected applications. These estimates become then the weights for rejects. In the resulting sample, each reject obtains two lines: one for the bad with corresponding probability of being bad as weight, while the second line is marked as good with the respective probability of being good assigned as weight. As by definition the sum of the two probabilities is equal to 1 ($P(B) = 1 - P(G)$), the reject continues to contribute as single observation despite its two-line expansion in the dataset.

```
   /*a. Score rejected with p_BAD based on accepts */
DATA E2_DEV_REJ ; SET C1_E4_S7030_DEV_WOE_40v; WHERE ~ACCEPTEE; RUN;

proc logistic INMODEL=E2_DEV_ACCPT_OUTMODEL;
   SCORE DATA=E2_DEV_REJ OUT=E2_DEV_REJ_ESTIM(RENAME=(P_1 = P_bad));
RUN;

   /*b. 2 obs per reject with p_BAD(hat) as weight */
DATA E2_DEV_REJ_ESTIM_fuzz
   (DROP=F_IndDef18 I_IndDef18 P_0 P_bad w);
   FORMAT vw BEST7.3;
   SET E2_DEV_REJ_ESTIM;

   /* First line: weight corresponding to NON-EVENT probability  */
   vw=P_0; IndDef18=0; OUTPUT;
   /* Second line: weight corresponding to EVENT probability  */
   vw=P_bad; IndDef18=1; OUTPUT;
RUN;
```

COMPARATIVE ANALYSIS

In order to compare properly the performance of the models obtained, it is useful to have a measure of statistical variance, which in turn can help establish if the observed difference is statistical significant.

The ROCCONTRAST statement in the PROC LOGISTIC serves well this purpose.

```
PROC LOGISTIC DATA=Results_5m_DEV;
   MODEL IndDef18 = P_E1_TTD P_E2_ACPT P_E3_AUGM
                    P_E4_recl P_E5_fuzz / NOFIT;

   ROC 'TTD'              P_E1_TTD ;
   ROC 'ACPT'             P_E2_ACPT ;
   ROC 'Augmentation'     P_E3_AUGM ;
   ROC 'Reclassification' P_E4_recl ;
   ROC 'Fuzz Parc'        P_E5_fuzz ;

   ROCCONTRAST reference('TTD') / estimate e;
RUN;
```

Table 2 contains the results obtained with ROCCONTRAST, which are in fact the results of our analysis.

The reject inference approach that yields the closest result to TTD (the benchmark) is Augmentation. In fact, it is the only method for which the AuROC is statistically identical to the AuROC of TTD at 5% significance level (p-value equal to 18.61%).

The results for the hold-out sample (**Table 3**) are more uniform. P-values are around 5% for all differences: slightly above 5% for Augmentation, and slightly below for the rest. Consequently, the main value added of the hold-out sample is to keep in check the risk of overfitting the model, while the results on the development sample serve as the main base for the conclusion.

DEVELOPMENT / TRAINTING (70%)

ROC Association Statistics							
	Mann-Whitney				Somers' D (Gini)		
ROC Model	Area	Standard Error	95% Wald Confidence Limits			Gamma	Tau-a
TTD	0.7532	0.00340	0.7465	0.7599	0.5064	0.5064	0.0637
ACPT	0.7496	0.00341	0.7430	0.7563	0.4993	0.4993	0.0628
Augmentation	0.7494	0.00341	0.7427	0.7561	0.4988	0.4989	0.0627
Reclassification	0.7517	0.00340	0.7451	0.7584	0.5035	0.5035	0.0633
Fuzz Parc	0.7487	0.00342	0.7420	0.7554	0.4973	0.4974	0.0626

ROC Contrast Estimation and Testing Results by Row						
Contrast	Estimate	Standard Error	95% Wald Confidence Limits		Chi-Square	Pr > ChiSq
ACPT - TTD	-0.00357	0.00108	-0.00569	-0.00144	10.8609	0.0010
Augmentation - TTD	-0.00379	0.00100	-0.00575	-0.00182	14.2784	0.0002
Reclassification - TTD	-0.00146	0.00110	-0.00362	0.000704	1.7480	0.1861
Fuzz Parc - TTD	-0.00454	0.00116	-0.00682	-0.00226	15.2122	<.0001

Table 2. Training sample, AuROC statistical comparative analysis

Validation / Hold-Out (30%)

ROC Association Statistics							
	Mann-Whitney				Somers' D (Gini)		
ROC Model	Area	Standard Error	95% Wald Confidence Limits			Gamma	Tau-a
TTD	0.7540	0.00515	0.7439	0.7641	0.5079	0.5080	0.0640
ACPT	0.7506	0.00514	0.7405	0.7607	0.5011	0.5012	0.0631
Augmentation	0.7509	0.00515	0.7409	0.7610	0.5019	0.5019	0.0632
Reclassification	0.7508	0.00520	0.7406	0.7610	0.5016	0.5016	0.0632
Fuzz Parc	0.7504	0.00516	0.7402	0.7605	0.5007	0.5008	0.0631

ROC Contrast Estimation and Testing Results by Row						
Contrast	Estimate	Standard Error	95% Wald Confidence Limits		Chi-Square	Pr > ChiSq
ACPT - TTD	-0.00339	0.00162	-0.00657	-0.00021	4.3527	0.0369
Augmentation - TTD	-0.00302	0.00150	-0.00595	-0.00008	4.0613	0.0439
Reclassification - TTD	-0.00318	0.00166	-0.00644	0.000072	3.6723	0.0553
Fuzz Parc - TTD	-0.00360	0.00176	-0.00705	-0.00015	4.1756	0.0410

Table 3. Hold-out sample, AuROC statistical comparative analysis

Normally, different ROC curves are plotted (through the PLOTS=ROC option); however, given the proximity of the values, it does not make sense to include it here.

CONCLUSION

The key contribution of the paper is to bring the debate on the supremacy of reject inference methodology on an empirical terrain. Three methods were selected that cover the three main conceptual frameworks. These methods are: Re-weighting, Re-classification, and Fuzzy Augmentation. As a result of comparing the classification performance of the full model (in this context the model fitted on the dataset with observed performance for accepts and rejects: TTD), four main conclusions seem to stand out:

- Even when the performance for rejects is MAR given the context (the scoring practice already in place), the difference between TTD and Accept models is statistically significant (0.7532 vs 0.7496)
- Only one method of the three comes close to the full model : Re-classification (the difference between the two AuROC values is statistically insignificant)
- Two other methods yield almost identical results to the Accept model.
- There seems to be little difference between the classification performances of the models.

REFERENCES

Anderson, R. (2007). *The Credit Scoring Toolkit : Theory and Practice for Retail Credit Risk Management and Decision Automation.* Oxford University Press.

Banasik, J., & Crook, J. (2004). Does reject inference really improve the performance of. *Journal of Banking & Finance, 28*, 857-874.

Feelders, A. (n.d.). An Overview of Model Based Reject Inference.

Little, R. J., & Rubin, D.B. (1987). Statistical Analysis with Missing Data.

Montrichard, D. (2009). Reject Inference Methodologies in Credit Risk Modeling.

Zeng, G., & Zhao, Q. (2014). A RULE OF THUMB FOR REJECT INFERENCE IN CREDIT SCORING. *Math. Finance Letters*.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Sergiu Luca
Desjardins
L[first name][a][t]g m a i l