A Reject Inference Primer – Methodology & Case Study

CSRSA Conference
June 7, 2016

Lamar Shahbazian
Senior Director, Product Management, FICO

Iana Tcharaktchieva
Manager, Retail Risk Modelling, Toronto Dominion
Agenda

• Introduction to reject inference
• Reject inference technique - Performance extrapolation
• Steps to successful reject inference
• Reject inference evaluation
The missing performance problem

- Modeling \( Y = f(X) \) for given credit applicant population
  - \( Y \) is binary ‘Good’ or ‘Bad’ risk variable

- What if \( Y \) is missing due to the existence of an underwriting screen?

- The performance of rejected applicants must be estimated prior to development of the risk model
Why use an inference technique?

- **Cost Effectiveness**
  - Cheaper than the alternative of booking rejects

- **Model Effectiveness**
  - Results in model that works better for “Through the Door” population
  - Adjusts for bias injected by prior screen
The crystal ball view of risk

# Trades 90+ Days Delinquent (at time of application)

<table>
<thead>
<tr>
<th># Trades 90+ Days Delinquent</th>
<th>Bad Rate after 24 months, if all applicants booked</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>2.5%</td>
</tr>
<tr>
<td>1</td>
<td>5.4%</td>
</tr>
<tr>
<td>2</td>
<td>8.7%</td>
</tr>
<tr>
<td>3</td>
<td>11.3%</td>
</tr>
<tr>
<td>4+</td>
<td>15.2%</td>
</tr>
</tbody>
</table>
Intelligent screen in place

# Trades 90+ Days Delinquent (at time of application)

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4+</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.5%</td>
<td>5.4%</td>
<td>8.7%</td>
<td>11.3%</td>
<td>15.2%</td>
</tr>
</tbody>
</table>

- **Bad Rate after 24 months:**
  - All Applicants
- **Reject Rate:**
  - 15.0%
  - 37.6%
  - 66.8%
  - 85.9%
  - 92.5%
Bias injected by intelligent screen

# Trades 90+ Days Delinquent (at time of application)

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4+</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bad Rate</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>after 24 months</td>
<td>2.5%</td>
<td>5.4%</td>
<td>8.7%</td>
<td>11.3%</td>
<td>15.2%</td>
</tr>
<tr>
<td>All applicants</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Booked accounts</td>
<td>1.8%</td>
<td>3.7%</td>
<td>4.0%</td>
<td>4.2%</td>
<td>3.9%</td>
</tr>
</tbody>
</table>

Bias injected by intelligent screen: Impact on new model development

• Booked account modeling data will exhibit
  – Lower predictive risk content for predictors constituting prior screen, e.g. historical delinquency, time on file
  – Reversals in Weight of Evidence patterns, particularly when approval rates are low or manual score overriding high

• New risk models built on such biased data will
  – Swap in a higher rate of severely delinquent and ‘new to credit’ segments of the population

**Weight of Evidence (WoE)**
A measure of a variable level’s ability to discriminate between two performance outcomes.
Suppose a classed predictor has $K$ levels. Let $m_{0j}$, where $j = 1, 2, ..., K$, be the weighted number of observations in level $k$ with performance outcome 0 and $m_0 = m_{01} + m_{02} + ... + m_{0K}$ be the total weighted number of observations with outcome 0. Define $m_{1k}$ and $m_1$ similarly for outcome 1. Then the WoE for level $k$ is

$$WoE_k = \ln \left( \frac{m_{1k}}{m_{0k}} \right)$$

$$\frac{m_{1k}}{m_1} \frac{m_{0k}}{m_0}$$
Audience Questions – True or False

• If a variable is not predictive on my booked population, I shouldn’t include it in my inference model

• I should never include scores or policy variables in my inference model

• If a variable has a pattern that isn’t monotonic, I shouldn’t include it in my inference model

• Every bin in the scorecard should have a minimum of 50 goods and bads

• When I’m done with my inference I will know for sure that it’s the best possible inference
• If a variable is not predictive on my booked population, I shouldn’t include it in my inference model

  FALSE

• I should never include scores or policy variables in my inference model

  FALSE

• If a variable has a pattern that isn’t monotonic, I shouldn’t include it in my inference model

  FALSE

• Every bin in the scorecard should have a minimum of 50 goods and bads

  FALSE

• When I’m done with my inference I will know for sure that it’s the best possible inference

  FALSE
Agenda

• Introduction to reject inference
• Reject inference technique - Performance extrapolation
• Steps to successful reject inference
Performance extrapolation using internal account data

- Historically, reject inference performed using predictive data from observation sample window and internal performance data
- Performance of rejects extrapolated from performance of internal accounts
Performance extrapolation using internal account data

- Develop risk score $S$ using booked account internal Good/Bad performance and observation time predictors $X$.

- Compute score for each rejected observation and convert into probability scale: $p(Y=1|S)=\exp(S)/(1+\exp(S))$.

- Use the $p(Y = 1)$ performance estimate for rejects in new model development.
Agenda

• Introduction to reject inference
• Reject inference technique - Performance extrapolation
• Steps to successful reject inference
Inference Steps

1. Generate key performance variables
2. Class variables for inference
3. Build Known Good/Bad model for inference
4. Estimate unknown performance
5. Generate post-inference performance variable and evaluation model
6. Evaluate inference

Credit Applicants

Yes

Approve?

No

Approved

Rejected

Known Goods
Y=1

Known Bads
Y=0

Unknown Goods
Y=?

Unknown Bads
Y=?
Step 1: Generate Key Performance Variables

1. Known Good/Bad Performance Variable
   - Known (Booked) performance categories: “0” and “1”
   - Inference categories: Scorecard Professional recognizes “2”, “3”, “4” as categories that may need to be inferred

\[ Y = \begin{cases} 
  1 & \text{“Good”} \\
  0 & \text{“Bad”} \\
  2 & \text{“Reject”} \\
  3 & \text{“Uncashed”} 
\end{cases} \]

2. Accept/Reject Performance Variable

<table>
<thead>
<tr>
<th>Accept/Reject Value</th>
<th>Description</th>
<th>Known Gd/Bd mapping</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Accept</td>
<td>0,1,3</td>
</tr>
<tr>
<td>0</td>
<td>Reject</td>
<td>2</td>
</tr>
</tbody>
</table>

3. Booked/Uncashed Performance Variable

<table>
<thead>
<tr>
<th>Booked/Uncashed Value</th>
<th>Description</th>
<th>Known Gd/Bd mapping</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Booked</td>
<td>0,1</td>
</tr>
<tr>
<td>0</td>
<td>Uncashed</td>
<td>3</td>
</tr>
<tr>
<td>nan</td>
<td>N/A</td>
<td>2</td>
</tr>
</tbody>
</table>
Step 2: Class Variables for Inference

- Classing appropriately is a key step for a successful inference – the analyst’s chance to identify and adjust for sample bias due to the prior screen.

- Variables highly correlated with prior scores and policies are likely to exhibit the most significant censoring.

- Review all Known G/B and Accept/Reject patterns side by side.
  - Severe screening can occur on small segments of data not revealed by summary I.V.

<table>
<thead>
<tr>
<th>Predictor</th>
<th>Perf: perfR</th>
<th>← Perf: xenoperfR0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level</td>
<td>Select</td>
<td>Merge &lt;&gt; Neu.</td>
</tr>
<tr>
<td>0</td>
<td>-Inf&lt; 1</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1&lt; 2</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>2&lt; 3</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>3&lt; 4</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>4&lt; Inf</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Other</td>
<td></td>
</tr>
</tbody>
</table>
Step 2: Class Variables for Inference

- Class all potential screen variables as well as standard predictors
  - Make an initial pass through classing using the Accept/Reject performance as the auto classing driver
- **Spread the Reject distributions out carefully**
  - Don’t be afraid to class finely, particularly in tails!
- Apply your expectations about performance via pattern constraints

<table>
<thead>
<tr>
<th>Predictor: var121 (Notes)</th>
<th>Perf: xenoperf90 →</th>
<th>← Perf: perfAR</th>
</tr>
</thead>
<tbody>
<tr>
<td>C A Level</td>
<td>Select</td>
<td>Merge</td>
</tr>
<tr>
<td>6 Total</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0 -Inf&lt;1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 1-&lt;2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2 2-&lt;3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3 3-&lt;4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4 4-Inf</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5 Other</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Class finely
Step 3: Build Known G/B Model to be used for inference

- You are building a model to estimate reject performance
  - You may need to override the biased data to impose your expectations
  - Ordinary model building rules don’t necessarily apply!

- Exhaust the known performance predictive content
  - Build a “fat” scorecard
  - Include variables from every predictive “family”

- Incorporate as much of the prior screen as possible
  - Previous scores, region, loan amount are all candidates
  - Give special consideration to:
    - Variables with a high Accept/Reject Information Value
    - Variables with a high Known Good/Bad Information Value
    - Variables that are multi-dimensional components of the Accept/Reject screen
Step 3: Build Known G/B Model to be used for inference

### Classing Summary

<table>
<thead>
<tr>
<th>#</th>
<th>Predictors</th>
<th>perfGB IV</th>
<th>perfAR IV</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>var52</td>
<td>0.300</td>
<td>1.028</td>
</tr>
<tr>
<td>2</td>
<td>var14</td>
<td>0.181</td>
<td>1.141</td>
</tr>
<tr>
<td>3</td>
<td>var99</td>
<td>0.165</td>
<td>0.123</td>
</tr>
<tr>
<td>4</td>
<td>var93</td>
<td>0.141</td>
<td>0.193</td>
</tr>
<tr>
<td>5</td>
<td>var66</td>
<td>0.132</td>
<td>0.452</td>
</tr>
<tr>
<td>6</td>
<td>var233</td>
<td>0.125</td>
<td>0.189</td>
</tr>
<tr>
<td>7</td>
<td>var92</td>
<td>0.112</td>
<td>0.077</td>
</tr>
<tr>
<td>8</td>
<td>var179</td>
<td>0.094</td>
<td>0.553</td>
</tr>
<tr>
<td>9</td>
<td>var112</td>
<td>0.085</td>
<td>0.108</td>
</tr>
<tr>
<td>10</td>
<td>var137</td>
<td>0.061</td>
<td>0.324</td>
</tr>
<tr>
<td>11</td>
<td>var79</td>
<td>0.057</td>
<td>0.081</td>
</tr>
<tr>
<td>12</td>
<td>var50</td>
<td>0.039</td>
<td>0.027</td>
</tr>
<tr>
<td>13</td>
<td>var259</td>
<td>0.028</td>
<td>0.193</td>
</tr>
<tr>
<td>14</td>
<td>var35</td>
<td>0.018</td>
<td>0.021</td>
</tr>
<tr>
<td>15</td>
<td>var5</td>
<td>0.002</td>
<td>0.115</td>
</tr>
</tbody>
</table>

High Accept/Reject Information Value
Step 3: Build known G/B model to be used for inference

Exhaust the known performance predictive content

- Build a “fat” scorecard
- Include variables from every predictive “family”
Step 4: The Scorecard Professional inference approach uses performance extrapolation to infer outcome

1. Develop Booked Good/Bad risk score $S$ using Booked Good/Bad performance and predictors $X$.
   - Rejected accounts are flagged as $Y=2$

2. In inference model, compute score $S$ (in ln(odds) scale) for each rejected observation and convert into probability scale:
   $$p(Y=1|S) = \frac{\exp(S)}{1+\exp(S)}$$

3. Using the “Apply Inference” operator, create a TTD Good/Bad performance variable, with values of $Y=2$ replaced with $p(Y = 1|S)$ performance estimates
   - The Scorecard Pro optimization algorithm allows for inferred performance to be utilized directly in the log likelihood optimization
Step 5: Generate Post-Inference Variable & Evaluation Model

- Generate the All Good/Bad performance variable
  - Use the “Apply Inference” operator
  - Known performances are assigned their original 0s and 1s
  - Inferred performances assigned $p(Y=1|\text{Score})$

- Copy and re-run classing using the All Good/Bad performance variable as the primary performance

- Build a Quick & Dirty Evaluation Model
  - Model should resemble a reasonable operational model
  - Drop any control or screening variables from the model
  - Post the quick & dirty score variable for the inference evaluation
Step 6: Evaluate inference

• Inference is an inexact science
  – Short of booking all applicants, the true performance is unknown

• Inference relies heavily on adherence to expectations
  – Domain expertise is useful

• No one criterion determines a successful inference
  – Absence of red flags increases confidence in the reconstruction

• To evaluate reject inference, one must first identify the prior application screen and understand its impact
Step 6: Evaluate inference
Recommended Diagnostics

1. Bias correction: reconstructed population predictive patterns

2. Reasonable reconstructed population odds
   • Do odds of each inferred performance category meet expectations?

3. Approximately linear ln(odds) to ‘benchmark’ score

4. Reasonable swap sets
   • Does size of swap set meet expectations?
   • A function of the approval rate and the intelligence of the prior screen
   • Typically in the realm of 5% - 15% in established application risk model developments
Reject Inference Evaluation Diagnostic
Step 1: Bias correction

For each predictor impacted by the prior applicant screen, check that:

- Information value increases over biased Booked Good/Bad predictive content
- Pattern reconstructs in expected direction
Reject Inference Evaluation Diagnostic
Step 2: Reasonable reconstructed population odds

Odds of inferred rejects
- Should be significantly lower than odds of booked accounts
- How much lower a function of the prior approval rate and the effectiveness of the prior screen

Booked Good/Bad

<table>
<thead>
<tr>
<th>Summary Stats:</th>
<th>Score Distribution:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Perfi</td>
<td>0</td>
</tr>
<tr>
<td>Number</td>
<td>242,046</td>
</tr>
<tr>
<td>Mean</td>
<td>6,453</td>
</tr>
<tr>
<td>Var</td>
<td>1,300</td>
</tr>
<tr>
<td>Probs(s)</td>
<td>0.978</td>
</tr>
<tr>
<td>Odds</td>
<td>44.31:1</td>
</tr>
</tbody>
</table>

Reject Inferred Good/Bad

<table>
<thead>
<tr>
<th>Summary Stats:</th>
<th>Score Distribution:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Perfi</td>
<td>0</td>
</tr>
<tr>
<td>Number</td>
<td>100,815</td>
</tr>
<tr>
<td>Mean</td>
<td>2,966</td>
</tr>
<tr>
<td>Var</td>
<td>1,737</td>
</tr>
<tr>
<td>Probs(s)</td>
<td>0.744</td>
</tr>
<tr>
<td>Odds</td>
<td>8.64:1</td>
</tr>
</tbody>
</table>
Reject Inference Evaluation Diagnostic
Step 3: Approximate linear ln(odds) to ‘benchmark’ score

- If a benchmark score that is not a component of the prior applicant screen exists, examine its relationship to reconstructed Through-the-Door ln(odds).
  - The relationship is expected to be approximately linear
  - Note: a score used in the prior application screen provides a biased basis of comparison
Reject Inference Evaluation Diagnostic
Step 4: Reasonable swap sets

- Use the score based on the Quick & Dirty Evaluation Model to generate an Acceptance rate report

- The size of the swap set will be a function of the operating point and the effectiveness of the prior screen
  - There’s more to swap the further away the approval rate is from either tail of the distribution
  - A bad prior screen will lead to a larger swap set
  - Availability of new data sources will lead to a larger swap set

- Typically, the marginally risky accounts around the score cut-off are being swapped

- The profile of the swap-ins should look less risky than the profile of the swap-out
The Acceptance Rate Report allows you to assess the swaps

Step 4: Reasonable swap sets

<table>
<thead>
<tr>
<th>Summary</th>
<th>newscore AOBm44</th>
<th>Accept</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Reject</td>
<td>Accept</td>
<td>Accept</td>
</tr>
<tr>
<td>Reject</td>
<td>26.2%</td>
<td>14.6%</td>
<td>40.8%</td>
</tr>
<tr>
<td>Accept</td>
<td>14.6%</td>
<td>44.7%</td>
<td>59.2%</td>
</tr>
<tr>
<td>Total</td>
<td>40.8%</td>
<td>59.2%</td>
<td>100.0%</td>
</tr>
</tbody>
</table>

### var121

<table>
<thead>
<tr>
<th>Interval</th>
<th>Frequency</th>
<th>Percent</th>
<th>Acceptance Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Previous</td>
<td>newscore AOBm44</td>
<td></td>
</tr>
<tr>
<td>-Inf &lt;- 1</td>
<td>65,780</td>
<td>63.1%</td>
<td>63.8%</td>
</tr>
<tr>
<td>1 - &lt; 2</td>
<td>9,613</td>
<td>44.9%</td>
<td>41.5%</td>
</tr>
<tr>
<td>2 - &lt; 3</td>
<td>2,496</td>
<td>30.5%</td>
<td>28.4%</td>
</tr>
<tr>
<td>3 - Inf</td>
<td>1,249</td>
<td>21.1%</td>
<td>17.3%</td>
</tr>
<tr>
<td>Other</td>
<td>0</td>
<td>0.0%</td>
<td>0.0%</td>
</tr>
</tbody>
</table>

Reasonable Size Swap set

Swapping in applicants with 0 major derogs

Swapping out applicants with major derogs
Identifying Prior Accepts: A more complex strategy

- When the application screen is based on a more complex strategy, identifying a comparable set of Prior Accepts and New Accepts for swap-set analysis requires replacing the prior score with the new score wherever the score occurs in the strategy.

- Could “filter” records coming to the Acceptance Rate report to isolate records where score is used to make the decision.
Conclusions

• A number of useful reject inference techniques exist as an alternative to booking rejects for new model development.

• However, inference is an inexact science.

• Keys to successful inference are:
  – Identification and understanding of the sample bias injected by the underwriting screen;
  – Applying domain-expert driven expectations about the unbiased relationship between predictors and risk;
  – Running qualitative diagnostics to ensure the absence of red flags

• Careful monitoring of the new scores and resulting swap sets is recommended
References


Thank You

Lamar Shahbazian
415 446 6321
Lshahbazian@fico.com

Iana Tcharaktchieva
416.982.6821
Iana.Tcharaktchieva@td.com