Scoring Models, Probability Transformations & Model Calibration Using SAS

Ilan Benamara
Krzysztof Dzieciolowski
Rogers Communications

October 16, 2013
SAS Data Mining Forum, Toronto
Questions

How can we estimate campaign’s Take-up-rate?

How can we transform predictive model propensity scores into probabilities?

What can we do next?
Predictive Modeling Process

Model
Training + Validation (Testing) on a sample

Deploy
As per rank-ordering of propensity scores

Validate
Performance of model propensities, e.g. lift

What’s missing?

Probability Estimates

Training sample may be biased!
Why Probabilities?

- Model scores are propensities of an event when estimated from a biased training sample
- Propensity scores can only be used for rank-ordering
- Scores **can not** be used to estimate
  - Expected Sales
  - Expected Profit
  - Optimization of offer allocation does not work

\[
\text{Expected Sales} = \sum_{i=1}^{N} p_i (1) + (1 - p_i)(0) \\
= \sum_{i=1}^{N} p_i \\
= NT
\]

\[
\text{Campaign Profit} = \sum_{i=1}^{N} (p_i R - C) = NTR - NC \quad [1]
\]

where,

- \( P1 = \) probability of purchase
- \( N = \) Total number of customers
- \( T = \) Overall Take Up Rate (fraction of target customers with desired event)
- \( NT = \) number of sales
- \( R = \) Revenue generated from an accepted offer
- \( C = \) Cost of making an offer
• **Bayesian Posterior Probabilities (SAS Decision Node)**

Posterior probabilities are adjusted for priors as follows. Let:

\[ t \] be an index for target values (classes)
\[ \text{oldPrior} \] be the old probability or implicit prior probability for target \( t \)
\[ \text{OldPost}(i,t) \] be the posterior probability based on \( \text{OldPrior}(t) \)
\[ \text{Prior}(t) \] be the new prior probability desired for target \( t \)
\[ \text{Post}(i,t) \] be the posterior probability based on \( \text{Prior}(t) \)

then,

\[
\text{Post}(i,t) = \frac{\text{OldPost}(i,t)\text{Prior}(t)}{\text{OldPrior}(t)} \quad [3]
\]

\[
\sum_j \frac{\text{OldPost}(i,j)\text{Prior}(j)}{\text{OldPrior}(j)}
\]
Logistic Regression intercept transformation

\[
\ln \left( \frac{\text{Post}(i,t)}{1 - \text{Post}(i,t)} \right) = \ln \left( \frac{\text{Post}(i,t)}{\text{OldPost}(i,t)} \times \frac{1 - \text{OldPost}(i,t)}{1 - \text{Post}(i,t)} \right) + \ln \left( \frac{\text{OldPost}(i,t)}{1 - \text{OldPost}(i,t)} \right)
\]

\[
= \gamma + \ln \left( \frac{\text{OldPost}(i,t)}{1 - \text{OldPost}(i,t)} \right)
\]

where

\text{Post}(i,t) = \text{probability of an event } t \text{ for } i\text{'th customer in the universe}

\text{OldPost}(i,t) = \text{probability of an event } t \text{ for } i\text{'th customer in the sample}

In order to estimate new prediction probabilities only an intercept of the old logistic regression model needs to be adjusted by a constant \( \gamma \) term.
Smooth curves!
Why Probability Smoothing?

Mapping scores’ ranks (e.g. deciles) to events empirical Take Up Rates (TUR) in $k$ bins, yields a step function:

$$map \ f: x_i \longrightarrow p_j \quad if \ x_i \ is \ in \ b_j \ bin$$

$$f(x_i) = p_j$$

where,

- $i = 1...n$ (customers)
- $j = 1...k$ (bins)

$p_j$ is a proportion of events in $j^{th}$ bin
Why Smoothing?

Heuristic Motivation:

- A step function is:
  - a “crude” approximation of “true” probabilities
  - Discontinued at bin’s limits
  - Constant in bins thus not reflecting our expectation of probabilities being monotonic in scores
  - Does not look “smooth”

- One way to improve approximation to “true” probabilities, is to smooth the step function via Cubic Monotonic Splines
Probability Smoothing
Methodology:

SAS Procedure: **PROC TRANSREG** *(transformation regression)*

Procedure which fits linear models, optionally with spline and other nonlinear transformations of y and x’s. [2]
Probability Smoothing
Methodology: Splines

Monotone cubic spline smoothing

A cubic smoothing spline consists of cubic polynomials *, one polynomial between each successive pair of knots, that have continuous second derivatives at the knots.

This means that each polynomial segment can connect with the next in such a way that their slopes and curvatures change continuously at the knots, resulting in a smooth transition from segment to segment through knots.

Monotonic spline is assumed here to be a non-decreasing spline function.

A kth order polynomial (we used k=3) in x is defined as

\[ y = \beta_0 + \beta_1 x + \beta_2 x^2 \ldots \beta_k x^k \]
Probability Smoothing
Methodology: contn’d

Sample SAS code

```sas
PROC TRANSREG DATA=inputDS;
  MODEL IDENTITY(Y) = MSPLINE(X / NKNOTS=9);
  OUTPUT OUT=outputDS PPREFIX=PRED_;
RUN;
```

Where,

- **MSPLINES** = Monotonic Spline Regressions
- **X** = model scores
- **Y** = estimated probabilities
Smoothed Step Function:
An artificial data example with 9 knots

Probability Calibration

<table>
<thead>
<tr>
<th>SUM_scores</th>
<th>SUM_prob</th>
<th>SUM_events</th>
<th>SUM_prob_smooth</th>
</tr>
</thead>
<tbody>
<tr>
<td>12,345.67</td>
<td>4321</td>
<td>4321</td>
<td>4321</td>
</tr>
</tbody>
</table>
Smoothened Step Function: An example with 49 knots

Step function is no longer monotonic. An approximation is closer with more knots but only in the top bins.
Smoothed Step Function:
An example with 99 knots

An approximation with more knots becomes “noisy” in the lower bins but works well in upper bins. More bins better?
Conclusions/next steps

- It appears that empirical mapping and splines are effective in providing non-decreasing properly calibrated probabilities.
- Methodology is easy to implement in SAS.
- It may be worthwhile to research how to:
  - compare a global transformation to posterior probabilities with monotonic splines
  - Construct optimal binning method
References:

