Proxy Modelling Using Machine Learning: A Case Study at Royal London

Nick Jackson, Royal London Mutual
Gaurang Mehta, Eva Actuarial and Accounting Consultants Limited

Agenda:
• Introduction and Motivations
• Background to Machine Learning ("ML") Methods
• Model Comparisons
• Lasso Regression – “Optimisation” Grid
• Initial Conclusions
• Q & A
Introduction and Motivations (1)

Royal London ("RL") is developing an all-risk model using Least Square Monte Carlo ("LSMC"): We currently use a "conventional" forward step-wise algorithm to perform our fit.

LSMC uses a very large number of outer scenarios, each with very few inner scenarios. R-squared to identify the next most important term; Refit the model; penalty function prevents over-fitting.

Introduction and Motivations (2)

Artificial Intelligence, Machine Learning and "Big Data" are concepts that are becoming increasingly prevalent and accepted throughout a wide spectrum of real-life applications.

This has become possible in recent times with the significant advances in computer technology, enabling the processing of the huge datasets now available. Examples range from computers beating humans at chess and (the more complex) Go, real-time travel updates ("Google maps") and translation services, Insurance pricing, through to medical diagnoses and driverless cars.

LSMC uses very large datasets and therefore feels like an appropriate problem to which these new cutting edge tools ought to be applied. This could lead to improved fitting, reduced scenario budgets and/or a new way of validating the existing more established fitting processes.

This presentation summarises the results of a Proof-of-Concept ("POC") Machine Learning tool applied to a dataset for one of RL’s larger with-profits funds. The objective is to produce an all-risk polynomial to determine the SCR and associated PDF. This initial POC focused on fitting statistics.
Background to Machine Learning Methods
(a) Models Explored

- Key Questions:
  - Model Selection – Which Model to use for Proxy Model calibration?
  - Model Calibration – Under fitting / Over fitting
  - Model Optimisation – Reduction of Cash-flow Bill?

- Approach Used:
  - Max polynomial power = 3
  - Feature engineering – Use of standardized data (Features and Losses)

---

Background to Machine Learning Methods
(b) Feature Engineering (FE) and Feature Importance (FI)

- FE - Creating new features from existing ones:
  - Standardised Data vs. Non-standardised Data
  - Introducing “domain expertise” via deciding interaction features
  - Dummy variables (e.g. Management Actions on or off)

- FI – Exclude unimportant features:
  - Is a filter and helps to mute unnecessary noise
  - Similar to well-known dimension reduction techniques such as PCA, but different
  - Makes models more parsimonious without compromising predictive accuracy
  - Improves performance
Background to Machine Learning Methods

(c) Bias and Variance Trade-offs

- Validation
  - Evaluation of residuals
  - How well model fits to data
  - No indication about model fit to unknown data

- Cross Validation
  - Involves removing part of training data and used for predictions.
  - Process repeated a number of times (4 in this example)
  - Trade-off: Bias vs. Variance
  - Full training dataset used in final fit

13 November 2018

(d) Understanding Losses Dataset

- Input Features, i.e. Risk Drivers (X1, X2,..): 34
- Training dataset, i.e. fitting points: 49.8k
- Validation dataset, i.e. validation scenarios: 385

<table>
<thead>
<tr>
<th>index</th>
<th>L</th>
<th>X1</th>
<th>X2</th>
<th>X3</th>
<th>X4</th>
<th>X5</th>
<th>X6</th>
<th>X7</th>
<th>X8</th>
<th>X9</th>
<th>X10</th>
</tr>
</thead>
<tbody>
<tr>
<td>count</td>
<td>49,800</td>
<td>49,800</td>
<td>49,800</td>
<td>49,800</td>
<td>49,800</td>
<td>49,800</td>
<td>49,800</td>
<td>49,800</td>
<td>49,800</td>
<td>49,800</td>
<td>49,800</td>
</tr>
<tr>
<td>mean</td>
<td>1.0000</td>
<td>0.0001</td>
<td>-0.0000</td>
<td>-0.0000</td>
<td>-0.0000</td>
<td>-0.0000</td>
<td>-0.0000</td>
<td>-0.0000</td>
<td>-0.0000</td>
<td>-0.0000</td>
<td>-0.0000</td>
</tr>
<tr>
<td>std</td>
<td>0.6928</td>
<td>0.6928</td>
<td>0.6928</td>
<td>0.6928</td>
<td>0.6928</td>
<td>0.6928</td>
<td>0.6928</td>
<td>0.6928</td>
<td>0.6928</td>
<td>0.6928</td>
<td>0.6928</td>
</tr>
<tr>
<td>min</td>
<td>-1.0077</td>
<td>-1.0000</td>
<td>-1.0001</td>
<td>-1.0000</td>
<td>-1.0000</td>
<td>-1.0000</td>
<td>-1.0000</td>
<td>-1.0000</td>
<td>-1.0001</td>
<td>-1.0000</td>
<td>-1.0001</td>
</tr>
<tr>
<td>25%</td>
<td>-0.5143</td>
<td>-0.5000</td>
<td>-0.5000</td>
<td>-0.5000</td>
<td>-0.5000</td>
<td>-0.5000</td>
<td>-0.5000</td>
<td>-0.5000</td>
<td>-0.5000</td>
<td>-0.5000</td>
<td>-0.5000</td>
</tr>
<tr>
<td>50%</td>
<td>-0.1994</td>
<td>-0.0000</td>
<td>-0.0000</td>
<td>-0.0000</td>
<td>-0.0000</td>
<td>-0.0000</td>
<td>-0.0000</td>
<td>-0.0000</td>
<td>-0.0000</td>
<td>-0.0000</td>
<td>-0.0000</td>
</tr>
<tr>
<td>75%</td>
<td>0.3045</td>
<td>0.5000</td>
<td>0.5000</td>
<td>0.5000</td>
<td>0.5000</td>
<td>0.5000</td>
<td>0.5000</td>
<td>0.5000</td>
<td>0.5000</td>
<td>0.5000</td>
<td>0.5000</td>
</tr>
<tr>
<td>max</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>index</th>
<th>L</th>
<th>X1</th>
<th>X2</th>
<th>X3</th>
<th>X4</th>
<th>X5</th>
<th>X6</th>
<th>X7</th>
<th>X8</th>
<th>X9</th>
<th>X10</th>
</tr>
</thead>
<tbody>
<tr>
<td>count</td>
<td>385</td>
<td>385</td>
<td>385</td>
<td>385</td>
<td>385</td>
<td>385</td>
<td>385</td>
<td>385</td>
<td>385</td>
<td>385</td>
<td>385</td>
</tr>
<tr>
<td>mean</td>
<td>1.0000</td>
<td>0.0070</td>
<td>0.0868</td>
<td>0.0120</td>
<td>1.0000</td>
<td>0.0070</td>
<td>0.0110</td>
<td>0.0158</td>
<td>0.0088</td>
<td>0.0244</td>
<td>0.0427</td>
</tr>
<tr>
<td>std</td>
<td>0.2463</td>
<td>0.1659</td>
<td>0.1860</td>
<td>0.0000</td>
<td>0.1397</td>
<td>0.1671</td>
<td>0.1394</td>
<td>0.1361</td>
<td>0.1361</td>
<td>0.2397</td>
<td>0.2397</td>
</tr>
<tr>
<td>min</td>
<td>0.0891</td>
<td>0.0000</td>
<td>-1.3186</td>
<td>-1.4229</td>
<td>1.0000</td>
<td>-0.6430</td>
<td>-0.8872</td>
<td>-0.5820</td>
<td>0.3710</td>
<td>-0.7461</td>
<td>-0.9319</td>
</tr>
<tr>
<td>25%</td>
<td>0.3209</td>
<td>-0.0898</td>
<td>- -</td>
<td>-1.0000</td>
<td>-0.0292</td>
<td>- -</td>
<td>- -</td>
<td>- -</td>
<td>- -</td>
<td>- -</td>
<td></td>
</tr>
<tr>
<td>50%</td>
<td>0.5149</td>
<td>- -</td>
<td>- -</td>
<td>-1.0000</td>
<td>- -</td>
<td>- -</td>
<td>- -</td>
<td>- -</td>
<td>- -</td>
<td>- -</td>
<td></td>
</tr>
<tr>
<td>75%</td>
<td>0.6196</td>
<td>0.0767</td>
<td>0.1558</td>
<td>0.0484</td>
<td>1.0000</td>
<td>-0.0225</td>
<td>0.0094</td>
<td>0.0310</td>
<td>0.0710</td>
<td>0.0737</td>
<td>0.0737</td>
</tr>
<tr>
<td>max</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
</tr>
</tbody>
</table>

13 November 2018
Background to Machine Learning Methods
(e) Applying Feature Importance to Training Data

- It’s a filtration step used as a proposing step
- Set all features → Select the best Subset → Learning algorithm → Performance
- Independent of any ML Algorithm
- Feature importance is one of the most versatile features of ML:
  - simplification of models & shorter training times
  - avoids the “curse” of dimensionality
  - enhances generalisation by reducing overfitting
  - Reduces subjectivity in selecting cross terms

- Top 7 covers 85% → 146 terms (Cross Terms)
- Top 10 covers 90% → 309 terms (Cross Terms)
- Top 20 covers 95% → 1784 terms (Cross Terms)

Model Comparisons
(a) Linear Model vs. Lasso Regression (Description)

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Linear Model</th>
<th>Lasso Regression</th>
</tr>
</thead>
<tbody>
<tr>
<td>RSS</td>
<td>$\sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij})^2$</td>
<td>$\sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij})^2 + \lambda \sum_{j=1}^{p}</td>
</tr>
<tr>
<td>Variable Selection</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Model Interpretation</td>
<td>Easy</td>
<td>Easier</td>
</tr>
<tr>
<td>Variance</td>
<td>High</td>
<td>Low</td>
</tr>
<tr>
<td>Bias</td>
<td>Low</td>
<td>High</td>
</tr>
</tbody>
</table>
### Model Comparisons

#### (a) Linear Model vs. Lasso Regression (Results)

<table>
<thead>
<tr>
<th></th>
<th>Linear Model</th>
<th>Lasso Regression</th>
</tr>
</thead>
<tbody>
<tr>
<td>Features used in fitting</td>
<td>34</td>
<td>34</td>
</tr>
<tr>
<td>Combination Terms</td>
<td>7769</td>
<td>7769</td>
</tr>
<tr>
<td>$R^2$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Abs. Max Value</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Std Deviation</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Key points:**
- Lasso performs materially better in comparison to Linear Model
- Reduces both Max. Absolute Error and Standard Deviation of residuals
- Same $R^2$ but materially different fitting results

#### (b) Lasso Regression – Importance of Feature Importance!!

**Feature Importance**

<table>
<thead>
<tr>
<th></th>
<th>Lasso Regression</th>
<th>Lasso Regression with FI (10 Features)</th>
<th>Lasso Regression with FI (20 Features)</th>
<th>Lasso Regression with FI (30 Features)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Features used in fitting</td>
<td>34</td>
<td>10</td>
<td>20</td>
<td>36</td>
</tr>
<tr>
<td>Terms (excluding intercept)</td>
<td>7769</td>
<td>309</td>
<td>1784</td>
<td>5459</td>
</tr>
<tr>
<td>Total Feature Importance</td>
<td>100%</td>
<td>90%</td>
<td>95%</td>
<td>99%</td>
</tr>
<tr>
<td>Average $R^2$</td>
<td>95%</td>
<td>94%</td>
<td>94%</td>
<td>93%</td>
</tr>
<tr>
<td>Abs. Max Value (Predicted “True” value)</td>
<td>64</td>
<td>87</td>
<td>57</td>
<td>61</td>
</tr>
<tr>
<td>Std Deviation (Predicted “True” value)</td>
<td>18</td>
<td>21</td>
<td>15</td>
<td>17</td>
</tr>
</tbody>
</table>

- FI leads to:
  - more manageable model
  - Improvement in fit
  - Reduction in run time
Model Comparisons
(b) Lasso Regression – Importance of Feature Importance (Results)

- 10 features covers 85% variation → Not enough;
- 34 features covers 100% variation and is an improved fit;
- 20 features covers 95% variation, leading to a further improvement still. This reflects less over-fitting;
- Optimum number of features is between 20 and 30.
- More to come once we review the remaining models…….

Approach comes from the same linear model family
Two widely used approaches – forward and backward stepwise algorithms.
Feature selection for backward regression by removing statistically unimportant features
Implementation:
- Step1: Starts with full polynomial
- Step2: Removes the statistically insignificant features (AIC, $R^2$, MSE, etc.)
- Step3: Repeats step 2 iteratively
- Step4: Stops when no further features can be removed without any statistical significance
Model Comparisons
(c) Backward Stepwise vs. Lasso Algorithm (Results)

<table>
<thead>
<tr>
<th></th>
<th>BSM (with FI)</th>
<th>Lasso (with FI)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Features used in fitting</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>Cross Validation</td>
<td>4-Fold</td>
<td>4-Fold</td>
</tr>
<tr>
<td>Training Data</td>
<td>35k</td>
<td>35k</td>
</tr>
<tr>
<td>AVG. $R^2$</td>
<td>94.02%</td>
<td>94.26%</td>
</tr>
<tr>
<td>Abs. Max Value</td>
<td>73</td>
<td>58</td>
</tr>
<tr>
<td>(Predicted value – True value)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Std Deviation</td>
<td>20</td>
<td>16</td>
</tr>
<tr>
<td>(Predicted value – True value)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Key Points:**
- Lasso performs better even after applying Feature Importance
- Why?

Model Comparisons
(d) Random Forest Algorithm (Description)

- Widely used as a “classification” algorithm
- Also used for regression purposes
- Works on averaging several noisy but unbiased models & reduces variance
Random Forest (with FI)  
Lasso Regression (with FI)

Features used in fitting: 20
Cross Validation: 4-Fold 4-Fold
Training Data: 35k 35k

<table>
<thead>
<tr>
<th></th>
<th>Random Forest (with FI)</th>
<th>Lasso Regression (with FI)</th>
</tr>
</thead>
<tbody>
<tr>
<td>AVG. $R^2$</td>
<td>85.68%</td>
<td>94.26%</td>
</tr>
<tr>
<td>Abs. Max Value</td>
<td>423</td>
<td>58</td>
</tr>
<tr>
<td>Std Deviation</td>
<td>102</td>
<td>16</td>
</tr>
</tbody>
</table>

Key Points:
- Random Forest leads to increased standard deviations and Max absolute error
- Random Forest is more appropriate for classification problems than regression problems

Model Comparisons
(d) Random Forest Algorithm vs. Lasso Algorithm (results)

Model Comparisons
(e) Neural Network Algorithm vs. Lasso Algorithm (Description)

- A class of non-linear statistical models
- Impressive results for some real-life examples
  - Google search
  - Cancer research
  - Driverless cars
- Generally implemented using back propagation, where error term is distributed back up through layers by modifying weights at each node.
**Model Comparisons**

(e) Neural Network Algorithm vs. Lasso Algorithm (Results)

<table>
<thead>
<tr>
<th></th>
<th>Neural Network (with FI)</th>
<th>Lasso Regression (with FI)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cross Validation</td>
<td>4-Fold</td>
<td>4-Fold</td>
</tr>
<tr>
<td>Training Data</td>
<td>35k</td>
<td>35k</td>
</tr>
<tr>
<td>AVG. ( R^2 )</td>
<td>94.8%</td>
<td>94.26%</td>
</tr>
<tr>
<td>Abs. Max Value</td>
<td>379</td>
<td>58</td>
</tr>
<tr>
<td>Std Deviation</td>
<td>83</td>
<td>16</td>
</tr>
</tbody>
</table>

Key Points:
- Neural Network algorithm leads to increased standard deviations and Max absolute error in this application.
- Neural Network algorithm may require further tuning of hyper-parameters for better results.

**Lasso Regression: “Optimisation” Grid**

Refresher: This model gives the best results of the models examined

- 10 features covers 85% variation → Not enough;
- 34 features covers 100% variation and is an improved fit;
- 20 features covers 95% variation, leading to a further improvement still. This reflects less over-fitting;
- Optimum number of features is between 20 and 30.
- What if we also vary the simulation budget?
Lasso Regression: “Optimisation” Grid
Number of Features vs. Size of Training Dataset

Key Points:
• Increasing number of features improves fit (up to a point)
• Increasing training data set improves fit
• Parameter tuning can reduces/optimises the cash-flow bill
• Sweet-spot here is 35k Sims and 20 Features.

Initial Conclusions

(a) Technical:
• Jury is still out there - there is no single “best” approach (“Horses for Courses!”);
• Analysis of training data is equally important before selecting any approach;
• Use of feature engineering and feature importance are the two key ML techniques which reduce complexity of the existing proxy model and / or improve its accuracy;
• Consider Bias-Variance trade-off, i.e. beware of under/over-fitting; and
• Further technical investigation areas identified, e.g. Auto-encoders for Regression techniques and Stacking/Hyper-parameter optimisation under RF/NN algorithms.

(b) Business:
• Recognising methodology developments in current practice, leading to improved proxy model fits;
• Reduced LSMC simulation budget – cheaper (and quicker) results; and
• Validation of the selected proxy model fit using alternative models.
Q & A

• Questions?

• For further details on ProxyML Software write to gaurang.mehta@evact.co.uk