IFRS9 Survival Analysis with an Application in Apache Spark

Credit Scoring and Credit Control XV Conference

Hristiana Vidinova
August 2017
Introduction
Introduction

Expected credit losses

Macroeconomic modelling

Probability of default

IFRS 9 aspects
IFRS 9
Expected credit losses

B5.5.3: …an entity shall measure the loss allowance for a financial instrument at an amount equal to the 
*lifetime* expected credit losses if the credit risk on that 
financial instrument has increased significantly since 
initial recognition.

5.5.11. If reasonable and supportable forward-looking 
information is available without undue cost of effort, 
an entity cannot rely solely on past due information 
when determining whether credit risk has increased 
significantly since initial recognition.

B5.5.17. An entity shall measure expected credit 
losses of a financial instrument in a way that reflects:

a) an unbiased and probability-weighted amount that 
is determined by evaluating a range of possible 
outcomes;

b) the time value of money; and

c) reasonable and supportable information that is 
available without undue cost or effort at the 
reporting date about past events, current conditions 
and forecasts of future economic conditions.
Goal

- Produce ECL forecasts
- Account-classification
- Include macroeconomic data
- Build upon existing models
- Calibrate behavior scores
Methodology
Building blocks

- Survival model
- Maturity effects
- Error correction
Survival analysis

Account level model – PD prediction for each account (similarly for other events)
Models not if but when an account will enter default
• Better estimates of expected losses
• Strong framework for IFRS 9
Easily accommodates IFRS 9 framework

Expected credit losses

- Constant (or structural) account characteristics
- Time-varying account characteristics (e.g. maturity; current status; behaviour data and scores)
- Time-varying internal factors (policy changes etc.)
- Time-varying external factors (economics, etc.)
Time specifications

Continuous-time

Often proportional hazard assumption
Individual hazard functions differ proportionately based on a function of observed covariates
Parametric estimation
• Assume distribution
  • eg. Log-logistic hazard
• Maximum likelihood estimation
Non-parametric estimation
  Cox PH

Discrete time

Application of continuous-time models is not recommended to discrete survival data due to the large number of ties that result more natural in social and behavioural applications where time is measured discretely.
Discrete-time models can easily accommodate time-varying covariates. Do not require a hazard-related proportionality assumption. These models allow for unstructured, as well as structured estimation of the hazard function at each discrete time point.
Estimate

\[ P(y_{t+\tau}|y_t = 0, S_t, A_{t+\tau}, \{E_i\}_{i=-\infty}^{t+\tau}) \]

Where

- \( y_t \) - account status at time \( t \)
- \( S_t \) - behaviour score at time \( t \)
- \( A_t \) - age of account at time \( t \)
- \( E_t \) - economics variables at time \( t \)
Recursive representation

Let define

\[ p_{t,k,S,A,E} = P(y_{t+k}|y_{t+k-1} = 0, S_t, A_t, \{E_i\}_{i=-\infty}^{t+k}) \]

Then

\[ P(y_{t+\tau}|y_t = 0, S_t, A_t, \{E_i\}_{i=-\infty}^{t+\tau}) = \sum_{i=1}^{\tau} \left( p_{t,i,S,A,E} \prod_{j=1}^{i-1} (1 - p_{t,j,S,A,E}) \right) \]

Therefore it is enough to calculate \( p_{t,k,S,A,E} \) for \( k = 1, \ldots, \tau \)
Probability decomposition

We will focus on the following form

\[ p_{t,k,S,A,E} = h \left( g \left( p_{t,k,A}, p_{t,k,S}, p_{t,k,E} \right) \right) \]

Where

- \( p_{t,k,S} = P(y_{t+k} | y_{t+k-1} = 0, S_t) \) - account behaviour component
- \( p_{t,k,A} = P(y_{t+k} | y_{t+k-1} = 0, A_{t+k}) \) - account maturity component
- \( p_{t,k,E} = P(y_{t+k} | y_{t+k-1} = 0, \{E_i\}_{i=-\infty}^{t+k}) \) - economics environment component
- \( h(x) = \frac{1}{1+e^{-x}} \)
- \( g(x, y, z) = \alpha_1 + \alpha_2 x + \alpha_3 y + \alpha_4 z \)
Account behaviour component

Include calibrated behaviour score to behaviour probabilities

Standard scorecard development approach

• Fine/Coarse classing per score and forecasting window
• Incorporate interaction terms as well
• Logistic regression estimates
How to create a forecast

Modelling stage

Forecasting stage

$PD_{t+1} = f(behaviour \ score_t)$

Behaviour score needs to be forecasted
How to create a forecast
Multiple-horizon model

**Modelling stage**

$PD_{t+1} = f(\text{behaviour score}_t)$

$PD_{t+2} = f(\text{behaviour score}_t)$

$\ldots$

$PD_{t+s} = f(\text{behaviour score}_t)$

**Forecasting stage**

Behaviour score does not need to be forecasted
Multiple-horizon model

Modelling stage

$PD_{t+1} = f(\text{behaviour score}_t)$

$PD_{t+2} = f(\text{behaviour score}_t)$

$PD_{t+s} = f(\text{behaviour score}_t)$

Exploded panel
Account maturity component

Motivated by age-cohort analysis / EMV models
Account maturity component

Standard scorecard development approach

- Fine/Coarse classing per age of account
- Logistic regression estimates
- No problem in forecasting
Economics component
Error-correction models

Long-term dynamics

Equilibrium state – no inherent tendency to change

The system tends to return to this state after deviations

Entails a systematic co-movement among economic variables

Short-term dynamics

“Errors” in the short-term

Transitory deviations from the long-term relationship due to shocks
Economics component

Error correction equation:

$$\Delta(y_t) = \alpha + \beta \Delta x_t - \gamma (y_{t-1} - \beta x_{t-1})$$

$\gamma$ – speed of error correction

Include more lags
Account states

Marginal probabilities are estimated by using approach from previous section
Technical aspects
Account level models

- Millions observations
- Hundreds of variables
- 10-20 years of history
- Multiple-horizon forecast

- Huge datasets
- Difficult to process
Sample size

Time series
- 100 historic months

<table>
<thead>
<tr>
<th>Accounts</th>
<th>Observations</th>
<th>Exploded panel</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>32</td>
<td>704</td>
</tr>
<tr>
<td>100 000</td>
<td>32 mn</td>
<td>704 mn</td>
</tr>
</tbody>
</table>
What is Apache Spark?

1. Powerful **open source** engine for **large-scale** data processing
2. Started as a research project in **UC Berkeley AMPLab** in 2009
3. The **largest** Apache Foundation project
4. Sorts **100TB** of data in **23 minutes**
Spark Architecture
Why Spark?

1. Fast **in-memory** and disk computing
2. Runs **everywhere** – on a cluster or standalone
3. Write Spark applications in **Python, R, Java, Scala**
4. A rich stack of **libraries** – SQL, machine learning, graph analytics, streaming data
5. Batch jobs and **real-time** analytics
6. Lazy evaluations and Catalyst optimizer
7. It’s fault tolerant
Python application
Use
Process iterated for each macro scenario
Significant increase in risk

PD Curves

Remaining PD

Cumulative PD

Expected lifetime

Reporting

Originations
Significant increase in risk

PD Curves

Remaining PD

PD estimate at reporting date