Mathematics and Statistics for Credit Scoring

Chapter 2: Credit Scoring Models

Overview

In this chapter we shall:-

1. Define a scorecard;
2. Review logistic regression and maximum likelihood estimation;
3. Look at how to develop a credit scoring model using logistic regression;
4. Show how to use the model to calculate credit scores.

"Characteristics" and "variables": some terminology

In statistics, we sometimes refer to predictor variables as ***covariates***.

In credit scoring, the borrower variables are referred to as borrower ***characteristics*** (such as age, income, etc).

The word ***attribute*** is also used, but ambiguously: sometimes for a variable and sometimes for a level of a variable.

Additionally, in machine learning and data mining, a variable is also known as a ***feature***!

In this course, we will generally refer to ***predictor variables*** ...

... But you should be familiar with all possible terminology.

The Scorecard

We will link borrower characteristics to a credit score using a statistical model. This is the **scorecard**.

Traditionally, the credit score is a *linear* combination of weighted variable values. For m borrower characteristics we have

$$s = \beta_0 + \sum_{i=1}^{m} \beta_i x_i$$

where

- $x_1, \ldots, x_m$ are the borrower characteristics,
- $\beta_1, \ldots, \beta_m$ are weights on each characteristic, and
- β_0 is a constant term.

This formula is more easily expressed in vector notation:

$$s = \beta_0 + \boldsymbol{\beta} \cdot \mathbf{x}$$

where $\mathbf{x}$ and $\boldsymbol{\beta}$ are vectors of borrower characteristics and weights, respectively:

$$\mathbf{x} = (x_1, \ldots, x_m) \text{ and } \boldsymbol{\beta} = (\beta_1, \ldots, \beta_m).$$

The vector of weights $\boldsymbol{\beta}$ form a **scorecard** which is used to score a given individual.

Non-linear scorecards are also possible and we will cover them later in the course.

Outcome of a loan

The credit score is typically linked to the risk of default.

Or, it could be linked to whichever event we are specifically modelling, such as repayment delinquency or fraudulent transactions.

We use a binary outcome variable $y \in \{0,1\}$ to represent the outcome with $y = 1$ indicating the outcome we are interested in. Examples are given below.

$y = 1$	$y = 0$
Default	Non-default
Delinquency	Non-delinquency
Fraudulent transaction	Legitimate transaction
Positive event	Negative event
Bad customer	Good customer

In general, we will refer to an outcome as a positive or negative outcome.

Credit score and Probability of Outcome

We quantify the risk by assigning a probability of outcome to each credit score s using a link function

$$P(y = 0|s) = f_L(s).$$

Typically, we want increasing scores to reflect increasing creditworthiness, therefore the link function should increase with score:

$$\text{For all } s_1 < s_2, \;\; f_L(s_1) < f_L(s_2).$$

Also, of course, $0 \leq f_L(s) \leq 1$ for all s.

When the outcome of interest is default, then this gives **probability of default** (PD) as

$$P_D = P(y = 1|s) = 1 - f_L(s).$$

The PD is much used in the credit industry and by regulators.

Log-odds link function

It is natural to use a log-odds link function for scores and this is standard in the industry:

$$\mathrm{s} = \log\left(\frac{P(y=0|s)}{P(y=1|s)}\right)$$

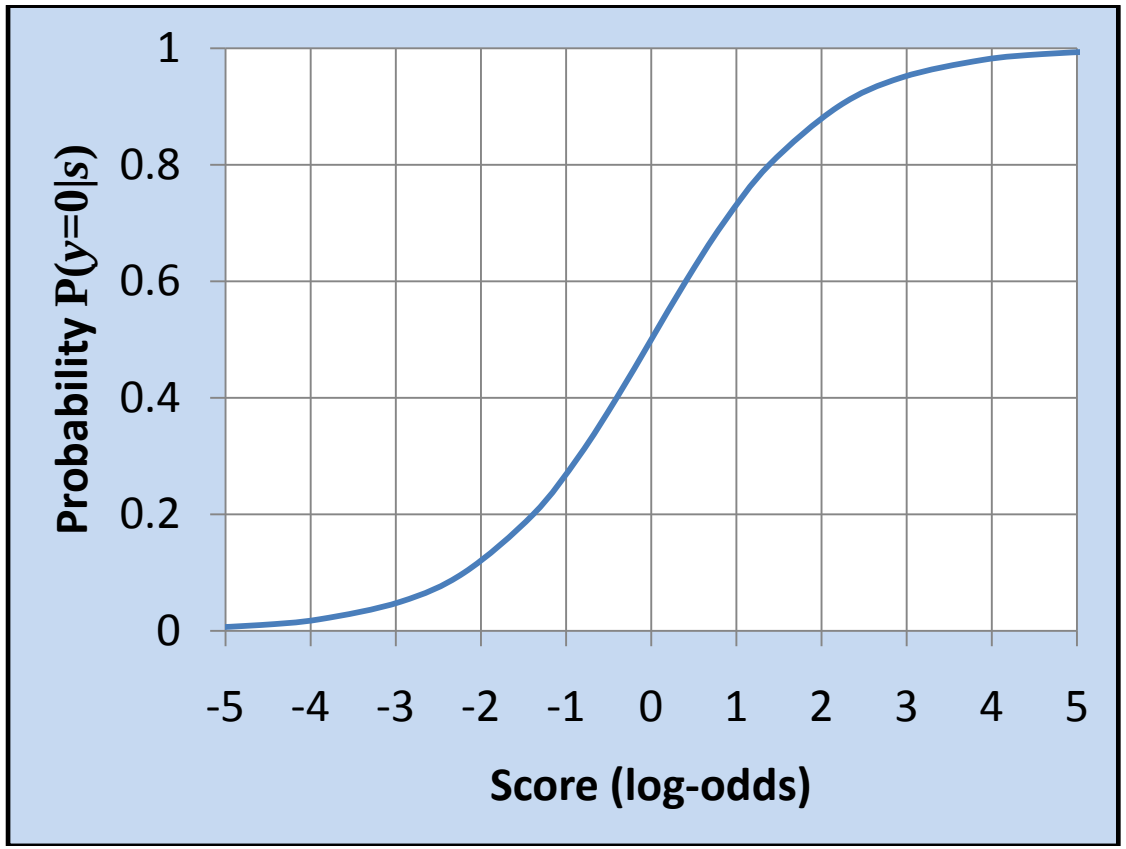
This gives us the link function

$$P(y=0|s) = f_{LO}(s) = \frac{1}{1+e^{-s}}.$$

The log-odds link function has two main advantages:

1. Has "greater resolution" at extreme probabilities.
2. Suitable for commonly used statistical methods (ie logistic regression).

The log-odds link function



Because the "bad" outcomes are rare, typically $P(y = 0|s) > 0.5$.
This implies that most scores will be greater than 0
(although this is not guaranteed!).

Credit score rescaling

Credit bureaus and banks usually adjust their credit scores so that they are positive integers within a given range.

For example, Experian scores are between 0 and 999.

This is a purely cosmetic step since non-mathematicians (the public, in general, or senior bank management) are not usually comfortable with negative real numbers.

The adjustment is usually a linear rescaling of the score given directly by the model.

Example 2.1.

Suppose we have a log-odds score and we are expecting PDs between 0.005 and 0.8. We may use this adjustment to get a score between 0 and 999:

$$s_{\text{cosmetic}} = \begin{cases} 0, & \text{if } s < -1.39 \\ \lfloor 150(s + 1.39) \rfloor, & \text{if } -1.39 \leq s < 5.29 \\ 999, & if\ s \geq 5.27 \end{cases}$$

since $\log\left(\frac{1-0.8}{0.8}\right) \approx -1.39$ and $\log\left(\frac{1-0.005}{0.005}\right) \approx 5.29$.

However, for our purposes, we are interested in the "raw" model scores so this is what we will work with for the remainder of the course.

Naive Bayes classifier

The Naive Bayes method follows naturally from the log-odds formulation of a scorecard *if we assume independence between the covariates* in the model, and using Bayes Rule.

By Bayes Rule,

$$\frac{P(y=0|\mathbf{x})}{P(y=1|\mathbf{x})} = \frac{P(\mathbf{x}|y=0)}{P(\mathbf{x}|y=1)}\frac{P(y=0)}{P(y=1)}$$
$$= \frac{P(\mathbf{x}|y=0)}{P(\mathbf{x}|y=1)}\frac{(1-p_1)}{p_1}$$

If the covariates in $\mathbf{x}$ are independent of one another then

$$\frac{P(y=0|\mathbf{x})}{P(y=1|\mathbf{x})} = \frac{1-p_1}{p_1}\prod_{j=1}^{m}\frac{P(x_j|y=0)}{P(x_j|y=1)}$$

and therefore, taking logs of both sides,

$$s = w_0 + \sum_{j=1}^{m} w(x_j)$$

where $w_0 = \log\left(\frac{1-p_1}{p_1}\right)$ is the log-odds of the negative event

and $w(x_j) = \log\left(\frac{P(x_j|y=0)}{P(x_j|y=1)}\right)$ is the **weights of evidence** (WOE) of a particular value of the jth predictor variable.

[*Note: WOE will crop up several times through the module.*]

The Naive Bayes classifier is a very simple method to produce a linear scorecard based only on the WOE of each covariate.

However, the assumption of independence is almost never met. For example, both *age* and *income* are common covariates to include in a scorecard: we would expect an association with increasing income with age, at least until retirement age.

Nevertheless, if we are careful about which variables we include, Naive Bayes may be sufficiently robust to be a good choice of classifier.

Logistic regression

If we use the log-odds interpretation of the credit score, we arrive at the most commonly used credit scoring model based on logistic regression:

$$\log\left(\frac{P(y=0|s)}{P(y=1|s)}\right) = s = \beta_0 + \boldsymbol{\beta}\cdot\mathbf{x}$$

How do we determine the vector of coefficients $\boldsymbol{\beta}$?

This is done by **training** on an existing data set of borrowers for which we already know their outcome.

> Formally a training data set is a sequence of characteristic/outcome pairs for some n observations:
>
> $$D_{\text{train}} = \left((\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2) \ \ldots, (\mathbf{x}_n, y_n)\right)$$

In the case of logistic regression, model fit to training data is acheived using maximum likelihood estimation.

Maximum Likelihood Estimation (MLE)

MLE is a general purpose method for parametric model estimation.
We will make use of it to estimate the logistic regression.

If we have a model with parametric structure $\boldsymbol{\theta}$, we can compute the **likelihood** that the model will generate a sequence of n observations $\mathbf{D} = (d_1, \ldots, d_n)$.

$$L(\boldsymbol{\theta}|\mathbf{D}) = P(\mathbf{D}|\boldsymbol{\theta})$$

The model which best fits the data is selected as the one which maximizes this likelihood.

$$\widehat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} L(\boldsymbol{\theta}|\mathbf{D})$$

If we *assume independence between the observations*, this then gives

$$\widehat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \prod_{i=1}^{m} P(d_i|\boldsymbol{\theta})$$

This MLE can be expressed more conveniently in terms of log-likelihoods (since $\log$ is monotonic on its argument):

$$\widehat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \sum_{i=1}^{m} \log P(d_i|\boldsymbol{\theta})$$

Remember:

- We do not know the true value of the parameter $\boldsymbol{\theta}$, but we want to estimate it.

- To distinguish the estimate from the true value, in our notation, we put a "hat" on the estimate: $\widehat{\boldsymbol{\theta}}$.

MLE has several nice asymptotic properties:

- Consistency
- Asymptotic normality
- Efficiency.

MLE for logistic regression

Consider the training data set D_{train} with n observations (borrowers).

Remember

- $\mathbf{x}_i$ denotes values for predictor variables for observation i.
- y_i denotes the outcome for observation i, either 0 or 1.

Then the likelihood of the outcome for each observation i is given by

$$\begin{array}{ll} P(y_i = 0|\mathbf{x}_i, \boldsymbol{\beta}) & \text{if } y_i{=}0, \\ 1 - P(y_i = 0|\mathbf{x}_i, \boldsymbol{\beta}) & \text{if } y_i{=}1 \end{array}$$

which is

$$P(y_i = 0|\mathbf{x}_i, \boldsymbol{\beta})^{1-y_i}\big(1 - P(y_i = 0|\mathbf{x}_i, \boldsymbol{\beta})\big)^{y_i}$$

giving log-likelihood for each observation:

$$(1 - y_i)\log P(y_i = 0|\mathbf{x}_i, \boldsymbol{\beta}) + y_i \log\big(1 - P(y_i = 0|\mathbf{x}_i, \boldsymbol{\beta})\big)$$

Assuming independence between observations, this gives the log-likelihood function for $\boldsymbol{\beta}$:

$$\log L(\boldsymbol{\beta}|D_{\text{train}}) = \sum_{i=1}^{n}(1-y_i)\log\left(\frac{1}{1+e^{-(\beta_0+\boldsymbol{\beta}\cdot\mathbf{x}_i)}}\right) + y_i \log\left(\frac{1}{1+e^{\beta_0+\boldsymbol{\beta}\cdot\mathbf{x}_i}}\right)$$

Differentiating by each coefficient in $\boldsymbol{\beta}$ and setting the derivative equal to zero to find the maxima gives

$$\sum_{i=1}^{n}\left(1-y_i-\left(\frac{1}{1+e^{-(\beta_0+\boldsymbol{\beta}\cdot\mathbf{x}_i)}}\right)\right)=0$$

and

$$\sum_{i=1}^{n} x_{ij}\left(1-y_i-\left(\frac{1}{1+e^{-(\beta_0+\boldsymbol{\beta}\cdot\mathbf{x}_i)}}\right)\right)=0$$

for each attribute j=1 to m.

These are non-linear equations that can be solved by computer intensive processes such as Newton-Raphson methods.

Standard errors on the MLE

Since $\widehat{\boldsymbol{\theta}}$ is only an estimate of the best model to explain the data, it is possible to derive standard errors $\hat{\mathbf{s}}$ on the estimates.

Asymptotic normality for MLE is such that

$$\frac{(\widehat{\theta}_j - \theta_j)}{\hat{s}_j} \to N(0,1) \text{ as } n \to \infty$$

where $\hat{\theta}_j$, θ_j and $\hat{s}_j$ are the jth components of $\widehat{\boldsymbol{\theta}}$, $\boldsymbol{\theta}$ and $\hat{\mathbf{s}}$ respectively and $N(0,1)$ is the standard normal distribution.

This property then allows us to generate:-

- Generate a hypothesis tests using the Wald chi-square statistic;
- Generate confidence intervals around the estimate.

Hypothesis Test

We test the hypothesis that an estimated coefficient is not zero against the null hypothesis that it is zero. That is, we testing if a parameter has a genuine effect in the model.

- Null hypothesis: $H_0: \theta_j = 0$
- Alternative hypothesis: $H_1: \theta_j \neq 0$

The Wald test says reject H_0 if $\frac{|\widehat{\theta}_j|}{\hat{s}_j} > z_{\alpha/2}$ for some significance level α, where $z_{\alpha/2} = \Phi^{-1}(1 - \alpha/2)$ and Φ is the CDF for the standard normal distribution.

Confidence intervals

The asymptotic normality property also allows us to compute confidence intervals (CIs):

$$P\left(\hat{\theta}_j - z_{\alpha/2}\hat{s}_j < \theta_j < \hat{\theta}_j + z_{\alpha/2}\hat{s}_j\right) \to 1 - \alpha$$

as $n \to \infty$.

This is a range of possible values of the parameter within a given confidence level $1 - \alpha$.

Note: the larger the confidence level, the broader the confidence interval.

Example 2.2

For large enough sample size, if we have an estimate of $\hat{\theta}$=2.3 with $\hat{s}$=0.2 then calculate the 95%CI for the estimate.

- $\alpha = 1 - 0.95 = 0.05$.
- Then $z_{\alpha/2} \approx 2$.
- So the CI is approximately $(2.3 - 2 \times 0.2, 2.3 + 2 \times 0.2) = (1.9, 2.7)$.

Likelihood Ratio Test of Goodness-of-Fit

The maximized likelihood gives a measure of how well the model fits the data (1=perfect fit, 0=no fit). The ratio of likelihoods between two models, A "nested" in B, can be used to test whether the fit of A improves on B.

Definitions

Suppose we have two models A and B with the same structure except A has more parameters than B:

$$\boldsymbol{\theta}_A = (\theta_1, \ldots, \theta_{m+r}) \text{ and } \boldsymbol{\theta}_B = (\theta_1, \ldots, \theta_m)$$

Then A *is nested in* B.

The *likelihood ratio statistic* is $\lambda = 2\log\left(\frac{L(\hat{\boldsymbol{\theta}}_A)}{L(\hat{\boldsymbol{\theta}}_B)}\right)$.

Theorem

λ approximates a chi-square distribution with r degrees of freedom.

Proof

- Follows immediately from asymptotic normality property for each of the r parameter estimates.
- The sum of the squares of these random variables then follows the chi-square distribution.

This can be used to compute a chi-square statistical significance test of model fit.

In particular, model B can be set to the null model (ie no parameters) to get a basic test of model fit for any logistic regression.

Example 2.3

The following logistic regression output was produced on a data set of 40,000 credit cards.

Likelihood Ratio = 1819 (p-value < 0.001)

Variable	Coefficient	Estimate	Standard error	Wald chi-square	P > chi-square
Intercept	β_0	-0.181	0.084	4.6	0.032
Age	β_1	+0.0353	0.0013	757.6	<0.001
Income (log)	β_2	-0.0164	0.0100	2.67	0.10
Residential phone	β_3	+0.622	0.030	430.8	<0.001
Home owner *		0			
Renter	β_4	-0.155	0.039	15.6	<0.001
Lives with parents	β_5	+0.256	0.045	32.1	<0.001
Months in residence	β_6	-0.00025	0.00011	5.4	0.020
Months in current job	β_7	+0.00210	0.00025	72.9	<0.001

* Notice that the Home owner category is set as base residency category and so has no coefficient estimate. We will discuss this in a later lecture.

We have used logistic regression to model the negative outcome (ie $y = 0$).

- This may seem odd given that the outcome of interest is the positive one (eg default).
- However, this model ensures the log-odds scores are the right way round: ie increasing scores imply increasing creditworthiness.
- There is no material difference. If we had modelled $y = 1$, the signs on the coefficient estimates would be reversed but everything else would be the same.

Interpretations:

- The estimates (highlighted) form the scorecard.
- Estimates greater than 0 indicate relative decrease in risk.
- Estimates less than 0 indicate relative increase in risk.
- Small p-values indicate coefficients that are statistically significantly different to zero (how small?).
- Large p-values indicate coefficients that have a good chance of actually being zero.

Sample results

Remember in the exercise in Chapter 1 we gave details of six borrowers. You were asked to select three to accept and three to reject.

Here the scores assigned by the model above are shown. The observations with the three lowest scores are rejected by the model. The actual outcome in each case is also shown. *How does your performance compare with the model?*

Age	Monthly Income (£)	Residential phone?	Residence type?	Months in residence	Months in current job	Score	Model accept or reject?	Actual outcome
22	1,145	Yes	Home owner	48	12	1.11	Reject	Good
46	15,500	Yes	Renter	48	192	2.14	Accept	Good
71	900	Yes	Renter	96	12	2.68	Accept	Good
32	5,000	Yes	Renter	48	168	1.61	Accept	Bad
25	1,385	Yes	Renter	12	0	1.05	Reject	Bad
43	3,145	No	Home owner	96	36	1.25	Reject	Bad

Example 2.4

Take the first borrower and apply the scorecard.

Variable	Value	Coefficient	Estimate	Value × Estimate
Intercept	n/a	β_0	-0.181	-0.181
Age	22	β_1	+0.0353	+0.777
Income (log)	log(1145) =7.04	β_2	-0.0164	-0.116
Residential phone	1	β_3	+0.622	+0.622
Home owner *	1		0	0
Renter	0	β_4	-0.155	0
Lives with parents	0	β_5	+0.256	0
Months in residence	48	β_6	-0.00025	-0.012
Months in current job	12	β_7	+0.00210	0.025
Score (sum)				**+1.115**

Example 2.4 continued

Compute the PD of the borrower.

Score = 1.115

Remember

$$\mathrm{s} = \log\left(\frac{P(y=0|s)}{P(y=1|s)}\right)$$

Therefore

$$P(y=1|s) = \frac{1}{1+e^{s}} \approx 0.25.$$

Exercise 2.1

The following logistic regression scorecard model was built on a data set of 20,000 personal loans. The outcome variable was non-default. Interpret the model to determine how each risk factor is associated with default.

Likelihood Ratio = 83.8 (p-value < 0.001)

Variable	Estimate	Standard error	Wald chi-square	P > chi-square
Intercept	+2.66	0.083	1028	<0.001
Age 18-29*	0			
Age 30-47	+0.14	0.08	3.0	0.08
Age 48+	+0.47	0.10	20.5	<0.001
Number of credit cards	+0.05	0.03	2.4	0.12
Self-employed	-0.47	0.07	44.0	<0.001
Months in current residence	+0.009	0.004	4.3	0.04

*Excluded category

Calculate the log-odds score for a 33 year old self-employed woman with no credit cards and 10 months in her current residence.

Logistic regression in R

Logistic regression is available in most statistics packages (eg SAS, Stata, R).

In particular, in R, logistic regression can be run using `glm`.

Example 2.5

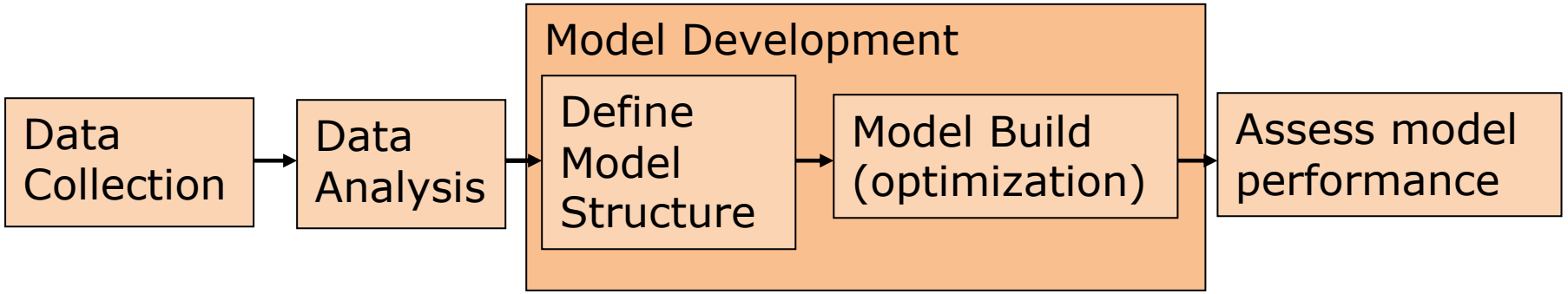
```
train <- read.delim("train.txt")
attach(train)

glm.out <- glm(good ~ age + income2_log + res_phone +
res_type_A + res_type_C + months_in_res + months_in_the_job,
               family = binomial("logit"))
print(summary(glm.out))
```

Model Development

Building a scorecard using a statistical model, such as logistic regression, is only one step in model development.

Credit scorecard development is a much larger process involving several absolutely essential steps (this is true for *applied statistics*, in general):-



We will be studying each of these steps in detail over the course of future lectures.

Overview of Chapter 2

Topics covered in this chapter were:-

1. Scorecards;
2. Log-odds credit scores;
3. Naive Bayes classifier;
4. Logistic regression;
5. Developing and using a simple scorecard using logistic regression.