Machine Learning MT 2013: Weeks 4-7, Practical 2

Lecturer: Phil Blunsom
Demonstrator: Hanno Nickau

Introduction

In this practical you will compete against your classmates to build the best performing model to predict the probability of a bank customer defaulting on a loan.

Banks play a crucial role in market economies. They decide who can get finance and on what terms and can make or break investment decisions. For markets and society to function, individuals and companies need access to credit. Credit scoring algorithms, which make a guess at the probability of default, are the method banks use to determine whether or not a loan should be granted.

You will build a Machine Learning system to predict which customers will experience a serious delinquency (payment at least 90 days late) within the next two years. You will enter your system into an online competition so that you can see how effective your approach is compared with your classmates.

Website

A competition website for this practical resides at:

`inclass.kaggle.com/c/oxford-cs-credit-scoring-competition-michaelmas-2013-edition`

You will need to setup an account in order to be able to submit the results from your system. This can be done at:

`inclass.kaggle.com/account/register`

The competition is restricted to Oxford University participants, so you will need to use an email address ending in `ox.ac.uk`.

Once you have registered you will be able to download the competition data (you will be asked to accept the competition rules the first time), and submit your results using the *Make a Submission* button. You can make up to ten submissions per day.

Data

The training and testing data for the competition can be downloaded through the web interface. This data was previously used in a public competition, see `www.kaggle.com/c/GiveMeSomeCredit` for more details, in particular the forum for that competition will give you many good ideas for improving your algorithms.

The data is in *comma separated values (csv)* format. Each row corresponds to a person and each column is an attribute. The first column is an id, you should discard this. The table below describes the remaining columns.

Column 1 *SeriousDlqin2yrs* is the response variable that your system needs to predict (t in the notation of the lecture slides), while the remaining columns are the features (x_i). In the test data the values for the response variables are missing, your system must predict them. The file `sampleEntry.csv` gives an example of the format for the submission file that your system must generate. The first column is the row index from the test file, while the second is your systems predicted probability of delinquency (class 1 in the training data).

Column	Variable Name	Description	Type
1	SeriousDlqin2yrs	Person experienced 90 days past due delinquency or worse	Y(1)/N(0)
2	RevolvingUtilizationOfUnsecuredLines	Total balance on credit cards and personal lines of credit except real estate and no installment debt like car loans divided by the sum of credit limits	percentage
3	age	Age of borrower in years	integer
4	NumberOfTime30-59DaysPastDueNotWorse	Number of times borrower has been 30-59 days past due but no worse in the last 2 years.	integer
5	DebtRatio	Monthly debt payments, alimony, living costs divided by monthly gross income	percentage
6	MonthlyIncome	Monthly income	real
7	NumberOfOpenCreditLinesAndLoans	Number of Open loans (installment like car loan or mortgage) and Lines of credit (e.g. credit cards)	integer
8	NumberOfTimes90DaysLate	Number of times borrower has been 90 days or more past due.	integer
9	NumberRealEstateLoansOrLines	Number of mortgage and real estate loans including home equity lines of credit	integer
10	NumberOfTime60-89DaysPastDueNotWorse	Number of times borrower has been 60-89 days past due but no worse in the last 2 years.	integer
11	NumberOfDependents	Number of dependents in family excluding themselves (spouse, children etc.)	integer

The training and test data contain occasional missing values (identified by the value *NA*). You should make a choice as to what to do with these values. Two naive options are to set all *NAs* to zero, or remove such rows from the training data. Of course you have no choice but to process these rows in the test data.

Tasks

Task 1: Submit a basic system To complete this practical you are required to build and submit at least one model for the competition. You may implement any machine learning algorithm and use any programming language you prefer. Alternatively you may make use of the Python code for logistic regression supplied for this practical (see below).

You can submit for scoring at most two submissions each day. You would be wise to develop a testing framework for your models, for example cross validation or splitting off some of the training data as held-out test data. You should also think about how you compare your models. The example includes code for calculating both the AUC and cross-entropy of test data.

Task 2: Explore feature transformations The feature representation of the supplied data is overly simplistic. Clearly the response variable is a non-linear function of these features. In order to build a strong system (and hopefully win the competition) you will need to transform the input data in some way to deal with these non-linearities. You should consider filtering out noisy rows, applying non-linear functions to the features (sqrt, log, tanh, sigmoid, etc.), and adding combinations of features that might be informative. Investigate at least two such transformations. In you report (see Assessment) you should describe why you chose the

transformations you did, and whether they had the desired effect.

Task 3: Win the competitions by exploring other algorithms and implementations (Optional) Once you have completed your own system implementation, you may download open-source implementations of machine learning algorithms from the web and see if you can maximise your competition score.

Scoring

When you submit a set of test results to the competition webpage they will be scored against the hidden response variables. The *Area Under the Curve (AUC)* metric is used to rank the submissions. During the competition the leader-board will display your best submission's AUC on a subset of the test data. At the end of the competition the results will be updated to display the AUC on the full test data.

Initially there are two benchmark submissions on the leader board. The one titled *Example Submission* is supplied by Kaggle and was produced using the RandomForest package in the R statistics language.[1] The second is a submission by the Lecturer using the supplied logistic regression code and some simple feature transformations.

Forum

The competition webpage includes a forum for discussing the practical. You can use this forum to ask questions relating to the practical, or to share insights with your classmates. You will also use this forum to submit your final report.

Code

Example Python code for a logistic regression model is available on the course website. As was the case for the first practical, this code is missing crucial lines that you must complete to make it function.

At the following URL you will find an archive of files for this practical which, when downloaded and unzipped, will provide a directory `practical_2`:

```
http://www.cs.ox.ac.uk/teaching/materials13-14/machinelearning/practical2.tar.gz
```

This directory contains a partially implemented logistic regression model and a directory containing the data for the practical (also available from the competition webpage). The data files with `_mod` in their names having been modified to set all *NAs* to zero.

Assessment

To complete this practical you must setup an account on the competition website and submit at least one set of test results from a machine learning system that you have implemented, either using the code provided or implemented from scratch. You must also submit a post to the competition forum describing your implementation, its strengths and weaknesses, and at least two data transformations you tried for Task 2 (approximately half a page of text in total).

To be signed off for this practical you must show your forum post, and identify your submission on the competition leader-board, to one of the practical demonstrators.

[1] `www.r-project.org`