M345S17 QUANTITATIVE METHODS IN RETAIL FINANCE 2019 COURSEWORK 2

Fraud scoring using Anomaly Detection

In this exercise you will use the data set of fraudulent credit card transactions that you used in CW1. This time you will use a multivariate kernel density estimator (KDE) for prediction. You will use the R statistical language for your work.

Tony Bellotti

TIMESCALE

Your coursework must be submitted by 4pm on Monday 25th February 2019.

INSTRUCTIONS

1. Use the training and test set of credit card transactions that you used in CW1.

2. When using the KDE with high dimensional data, there are computational problems. One is that R cannot handle the precision of the calculations (eg the normalizing term $\frac{1}{nh^m}$ can become very large). For this reason we will work with the log of the density. Following the material in Chapter 7 of the course notes, show that when the standard multivariate normal function
$$K_{\mathrm{NORM}}(\mathbf{z}) = (2\pi)^{-m/2}\exp(-\mathbf{z}\cdot\mathbf{z}/2)$$
is used as the kernel function, then
$$\log \hat{f}_{\mathrm{P}}(\mathbf{x};h) = c + \varphi(\mathbf{x};h) + \log\left[\sum_{i=1}^{n}\exp\left(\frac{-(\mathbf{x}-\mathbf{x}_i)\cdot(\mathbf{x}-\mathbf{x}_i)}{2h^2} - \varphi(\mathbf{x};h)\right)\right]$$
for some constant c (constant relative to $\mathbf{x}$) and where
$$\varphi(\mathbf{x};h) = \max_{i\in\{1,\cdots,n\}}\left(\frac{-(\mathbf{x}-\mathbf{x}_i)\cdot(\mathbf{x}-\mathbf{x}_i)}{2h^2}\right)$$

3. Show that $\varphi(\mathbf{x};h) \le \log \hat{f}_{\mathrm{P}}(\mathbf{x};h) - c \le \log n$ for any $\mathbf{x}$ and h.

4. Implement the formula from step 2 in R to compute the fraud score
$$s(\mathbf{x}; h) = \log \hat{f}_{\mathrm{P}}(\mathbf{x}; h) - c$$
for any new observation $\mathbf{x}$.
See Appendix A for coding hints.

5. Use your R implementation of $s(\mathbf{x}; h)$ to compute fraud scores for all observations in the test set, based only on the density estimate of legitimate transactions in the training data set. Use $h = 0.1$.

6. Construct a precision-recall (PR) curve and compute the area under the precision-recall curve (AUPRC), when applying these fraud scores to the test data set.

7. If an alarm rate of no more than 0.5% is required, what is the maximum recall that can be achieved using this model, based on the results on the test set?

8. How do the results using ANN and KDE compare? Which is the better approach, and why?

9. Your coursework must be submitted as:
   a) a paper copy of your solutions to the student office and
   b) an electronic submission, along with an R script giving the commands you used to complete the coursework, emailed to a.bellotti@imperial.ac.uk with subject heading “M345S17 CW2”. Your R script should include annotated comments describing what it is doing at each step.

Remember to include all results and the R code you used in your report.

APPENDIX A: HELP IN R

Although you can use the PRROC package again, do not use a package to implement the KDE. You need to code this yourself.

`for` loops

You can use `for` loops in R to implement the sum in the KDE and to calculate the fraud score for each test observation. The `for` loop has the following syntax:-

```
for (i in a:b) {
   … statements ….
}
```

It will cycle over values of `i` from `a` to `b`.
For example, this code will compute the factorial of x:

```
x <- 4 #(or other input)
fac <- 1
for (i in 1:x) {
   fac <- fac*i
}
fac
```

Warning! Implementing KDE with `for` loops is slow. As a guide, on my five year old laptop, it takes just over an hour for KDE to compute fraud scores for the whole data set. Therefore, I suggest that while you are writing and debugging your R code, you use a small subsample of your test data, just to check it is working right. Only apply to the whole test data when you are confident it is working correctly.