Resource-bounded Fraud Detection

Luis Torgo

LIAAD-INESC Porto LA / FEP, University of Porto
R. de Ceuta, 118, 6., 4050-190 Porto, Portugal
ltorgo@liaad.up.pt – http://www.liaad.up.pt/~ltorgo

Abstract. This paper describes an approach to fraud detection targeted at applications where this task is followed by a posterior human analysis of the signaled frauds. This is a frequent setup on fraud detection applications (e.g. credit card misuse, telecom fraud, etc.). In real world applications this human inspection is usually constrained by limited resources. In this context, standard fraud detection methods that simply tag each case as being (or not) a possible fraud are not very useful if the number of tagged cases surpasses the available resources. A much more useful approach is to produce a ranking of fraud that can be used to optimize the available inspection resources by first addressing the cases with higher rank. In this paper we propose a method that produces such ranking. The method is based on the output of standard agglomerative hierarchical clustering algorithms, resulting in no significant additional computational costs. Our comparisons with a state of the art method provide convincing evidence of the competitiveness of our proposal.

1 Introduction

Fraud detection is a hot topic in several research areas (e.g. [5, 13, 15]). Due to the intrinsic characteristics of fraudulent events it is often associated with other research topics like outlier detection, anomaly detection or change detection. The connecting feature among these topics is the interest on deviations from "normal" behavior. Depending on the characteristics of the fraud data available different data mining methodologies can be applied. Namely, when the available data includes information regards each observation being (or not) fraudulent, supervised classification techniques are typically used (e.g. [4]). On the contrary, in several domains such classifications do not exist and thus unsupervised techniques are required (e.g. [3]). Finally, we may have a mix of both types of data with a few labelled observations (e.g. resulting from past inspection activities) and a large set of unlabeled cases. These situations are often handled with semi-supervised approaches (e.g. [12]).

Several authors (e.g. [13]) have criticized the use of labelled data for fraud detection. These authors have noted that in most real world applications it is difficult to have reliable labels due to several factors like the cost of obtaining them, among others. In this paper, we assume the available data is not labelled. The method we propose produces a ranking of fraud probability for a set of

unlabeled observations. Many detection methods provide yes/no answers to this task. We claim that for real world applications of fraud detection systems this type of answer may lead to sub-optimal decisions. In effect, in most applications the detection systems are used to help in planning posterior inspection activities. These activities are typically constrained by a limited amount of resources (human or other). In this context, it is preferable to have a ranking of fraud instead of a set of cases predicted as fraudulent. Such ranking is much more flexible for the correct use of the available resources and will most probably lead to better results. Without these ranks and if the cases labeled as fraudulent are more than what the available resources allow to inspect, the user is left with the unguided task of deciding which ones to inspect. By providing a rank of fraud, the resources can be used on the cases that have a higher probability of fraud.

2 Outlier Ranking

The approach we propose for obtaining a ranking of fraud probability assumes that frauds are rare and that can be regarded as outliers from the bulk of "normal" data. Outlier detection is a well studied topic (e.g. [2]). Different approaches have been taken to this task. Distribution-based approaches (e.g. [6]) assume a certain parametric distribution of the data and signal outliers as observations that deviate from this distribution. The main drawbacks of these approaches lie on the constraints of the assumed distributions. Depth-based methods (e.g. [14]) are based on computational geometry and compute different layers of k-d convex hulls and then represent each data point in this space together with an assigned depth. In practice these methods are too inefficient for dealing with large data sets. Knorr and Ng [10] introduced distance-based outlier detection methods. These approaches generalize several notions of distribution-based methods but still suffer from several problems, namely when the density of the data points varies (e.g. [17]). Density-based local outliers [17] are able to find this type of outliers and are the appropriate setup whenever we have a data set with a complex distribution structure.

Clustering algorithms can also be used to identify outliers as a side effect of the clustering process (e.g. [11]). Most clustering methods rely on a distance metric and thus can be seen as distance-based approaches. However, iterative methods like hierarchical clustering algorithms (e.g. [7]) can also handle different density regions. In effect, if we take for instance agglomerative hierarchical clustering methods, they proceed in an iterative fashion by merging two of the current groups (which initially are formed by single observations) based on some criterion that is related to their proximity. This decision is taken locally, i.e. for each pair of groups, and takes into account the density of these two groups only. It is based on this observation that we plan to explore hierarchical clustering methods as a form of producing outlier rankings that are able to handle applications with both global and local outlier types.

2.1 Height-based Outlier Factors

Our proposal is based on an agglomerative hierarchical clustering method (e.g. [7]). These methods start with as many clusters as there are training observations and then go through an iterative process where at each stage two of the current clusters are merged to form a new grouping of the data. This merging process results in a tree-based structure usually known as a dendogram. The merging step is guided by the information contained on the distance matrix of all available data. Several methods can be used to select the two groups to be merged at each stage. For instance, the single linkage method selects the pair of groups which has the smallest distance between any of their members. One of our goals is to have an outlier detection method that can handle local outliers, i.e. being able to capture areas of high local density and spot nearby points that somehow break this density, though from a global perspective they could look near to these areas and thus would not be regarded as outliers. In this context, we have selected to work with the Ward's [19] agglomeration method. Merging according to this method is carried out by selecting the pair of groups which would result in a new group with minimal variance, i.e. maximally compact. This means that outliers tend to be selected later on this iterative process because, by definition, they are clearly separated from their neighborhood and thus will increase the variance of a group when joining it. Informally, the idea behind our proposal is to use the height (in the dendogram) at which any observation is merged into a group of observations as an indicator of its outlyingness. If an observation is really an outlier this should only occur at later stages of the merging process, i.e. the observation should be merged at a higher level than "normal" observations. More formally, we set the outlyingness factor of any observation as,

$$OF_H(x) = \frac{h}{N} \tag{1}$$

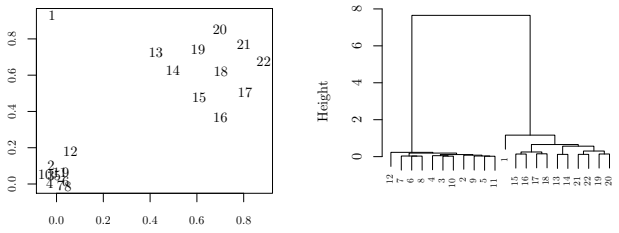
where h is the level of the hierarchy H at which the case is merged[1], and N is the number of training cases (which is also the maximum level of the hierarchy by definition of the hierarchical clustering process).

One of the main advantages of our proposal is that we can use a standard hierarchical clustering algorithm to obtain the OF_H values without any additional computational cost. This means our proposal as a time complexity of $O(N^2)$ and a space complexity of $O(N)$ [8]. We use the *hclust()* function of the statistical software environment R [16], which is based on Fortran code by F. Murtagh [9]. This function includes in its output a matrix (**merge**) that can be used to easily obtain the necessary values for calculating directly the value of OF_H according to Equation 1.

Figure 1.(a) shows an artificial data set with two marked clusters of observations with very different density. As it can be observed there are two clear outliers: observations 1 and 12. While the former can be seen as a global outlier, the latter is clearly a local outlier. In effect, it is only regarded as an outlier

[1] Counting from bottom up.

because of the high density of its neighbors, as it is in effect nearer observation 2 than, say the 14th from the 15th. However, as these two latter are in a less compact region their distance is not regarded as a signal of outlyingness. This is a clear example of a data set with both global and local outliers and we would like our method to clearly signal both 1 and 12 as observations with a high probability of being outliers.



(a) Data (b) Dendogram

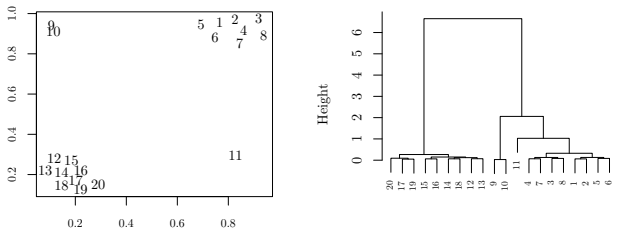
Fig. 1. An artificial example.

Figure 1.(b) shows the dendogram obtained by using an agglomerative (Ward) hierarchical clustering algorithm. As it can be seen, both 1 and 12 are the last observations to be individually merged into some cluster. As such, it does not come as a surprise that when running our method on this data we get the top 5 outliers shown on Table 1.

Table 1. Outlier ranking for the example of Figure 1.

Rank	CaseID	OF_H
1	1	0.9091
2	12	0.6818
3	17	0.5909
4	18	0.5909
5	19	0.5455

In spite of the success, this method has serious problems when facing compact groups of outliers. In effect, if we have a data set where there are a few outliers that are very similar to each other, they will be merged with each other very quickly (i.e. at a low level of the hierarchy) and thus will have a very low OF_H value in spite of being outliers. Figure 2 illustrates this problem. For this data set, the method ranks observations 9 and 10, which are clear outliers, as the least probable outliers (they are in effect the first to be merged by the Ward method).



(a) Data (b) Dendogram

Fig. 2. A problematic artificial example for our initial proposal.

The example of Figure 2 shows a clear failure of our initial proposal. The failure results from considering only the height at which individual observations are merged and not groups of observations. However, if there is a small group[2] of similar observations that is quite different from others, and thus will only be merged with other groups at later stages, our proposal will not consider this as a signal of outlyingness of the members of that group. Still, we should remark that the general idea of our proposal remains valid as long as we generalize it for these situations. We can do this by assigning a value similar to that of Equation 1 to all members of the smallest group of any merge that occurs along the hierarchical clustering process. However, we should reinforce this value with some size-dependent factor (i.e. the smallest the most probable that we are facing outliers). Formally, for each merge of a group g_s with a group g_l, where $|g_s| < |g_l|$, we set the outlier factor of the members of g_s as,

$$OF\left(g_s\right) = \left(1 - \frac{|g_s|}{N}\right) \times \frac{h}{N} \tag{2}$$

where $|g_s|$ is the cardinality of the smallest group, g_s, and h is the level of the hierarchy where the merge occurs. The OF value of the larger group g_l is set to zero. The value of OF ranges from zero to one, and it is maximum when a single observation is merged at the last level of the hierarchy.

Any observation can belong to several groups along its upwards path through the dendogram. As such, it will probably get several of these scores at different levels. We set the outlyingness factor of any observation as the maximum OF score it got along its path through the dendogram. By proceeding this way we are in effect taking care of local outliers, which at some merging stage might have got a very high score of OF because they are clear outliers with respect to some group that they have merged with, even though at higher levels of the hierarchy (i.e. seen more globally), they might not get such high OF values. This means that the outlyingness factor of an observation is given by,

[2] Such that it could make sense to talk about a set of outliers.

$$OF_H(x) = \max_{g \in G_x} OF(g) \tag{3}$$

where G_x is the set of groups in the dendogram to which x belongs.

Applying this method to the problematic example of Figure 2, we get the outlier ranking shown in Table 2. This is the expected result for this problem, which means that this new formulation is able to handle compact and small groups of outlier observations, like for instance observations 9 and 10 of this problem.

Table 2. Outlier ranking for the example of Figure 2 using our new proposal.

Rank	CaseID	OF_H
1	9	0.8100
2	10	0.8100
3	11	0.8075
4	15	0.6300
5	16	0.6300

We should remark that although we have always used our proposal in the context of a hierarchical clustering process using the Ward's agglomerative criterion, our proposal is not dependent on this criterion. In effect, the method could be applicable to any agglomerative criterion and/or distance metric. Still, we think the Ward's method is more adequate for finding local outliers.

3 Experimental Evaluation

In this section we present several experiments providing some insights on the effective behavior of our proposal. We compare our method with the state of the art in terms of obtaining degrees of outlyingness: the LOF method [17].

As we did not have access to real world data sets of fraud detection we decided to use some supervised regression data sets obtained from Torgo's repository [18]. These data sets can be used to provide an idea of how well does our method captures the most deviating observations and at the same time, as these are supervised regression problems with a continuous target variable, we can check whether the top ranked outliers correspond to unusual values of the target variable. This is somehow similar to the process it would be followed if the method were to be used in real world applications of fraud detection: first the method would provide a ranking and then, after human inspection, we would confirm the degree of fraud of the observations. The assumption we are making here is that observations that have unusual values on the input variables (so as to make them stand as outliers), will also have unusual values on the target variable. This assumption is reasonable provided there is some relationship between the target and input variables and the unknown regression surface is reasonably smooth. The data sets we have used are the following:

- *Boston Housing* (BH) - 506 cases described by 14 variables. The data concerns the task of predicting the median price of houses in different areas of Boston. The input variables describe some socio-economical features of the areas. The distribution of the prices has a normal-like shape around the value 22, with a few extremely high or low prices for some areas.
- *Abalone* (AB) - 4177 cases described by 9 variables. The data concerns the prediction of the number of rings in an abalone, which is supposed to be directly related to their age (the goal of the original application). The distribution of the number of rings also follows a normal-like distribution around the value 9, with a few small values and also some unusually high values.
- *Alga 1* (A1) - 200 cases described by 12 variables. This data concerns the prediction of the concentration values of a rare harmful alga species in 200 water samples drawn from different European rivers. The distribution is concentrated around values very near zero, with a few unusually high concentration values (known as algae blooms).
- *Machine-CPU* (MC) - 209 observations described by 7 variables. This data set concerns the task of predicting the relative CPU performance based on some hardware features. The target variable has a distribution centered on small values (around 50) with a few extreme values of performance.

Regards the compared methods, we have based our implementation of OF_H on the *hclust()* function of the statistical software environment R [16], as mentioned before. With respect to LOF we have used the implementation of this method available on the package *dprep* [1] of the same software environment. For the outlier ranking tasks we have eliminated the target variable information. We have used the two principal components to obtain a 2-D plot of all data points of each domain. In all graphs we report the proportion of variance of the original data that is explained by these 2-D summarization. For all data sets it is above 95% which means that the plots are a good spatial representation of the original data. For the larger data sets of have signalled the top 20 outliers according to each method, while for smaller we have only used the top 10.

Figure 3 shows the results for the Boston Housing domain. With a few exceptions in the case of LOF, most of the signalled points can easily be accepted as outliers. The solutions obtained by both methods are quite similar, with LOF being able to identify a few outliers at the bottom right corner that OF_H is not, while the opposite occurs for some outliers identified by our method (at the middle of the graph). Generally, both solutions seem reasonable with a slight advantage of our method.

Are the outliers signalled by both methods associated with unusual values of the target variable? In this case unusual values are very low or very high median house values. Figure 4 shows a continuous approximation (obtained with a kernel density estimator) of the distribution of the values of the target variable using: i) all data set; ii) only the top 20 outliers according to our method; and iii) the top 20 according to LOF. The concrete values are also shown by adding two rugs at the top and bottom of the graph. This figure confirms the similarity of the solutions of both methods.

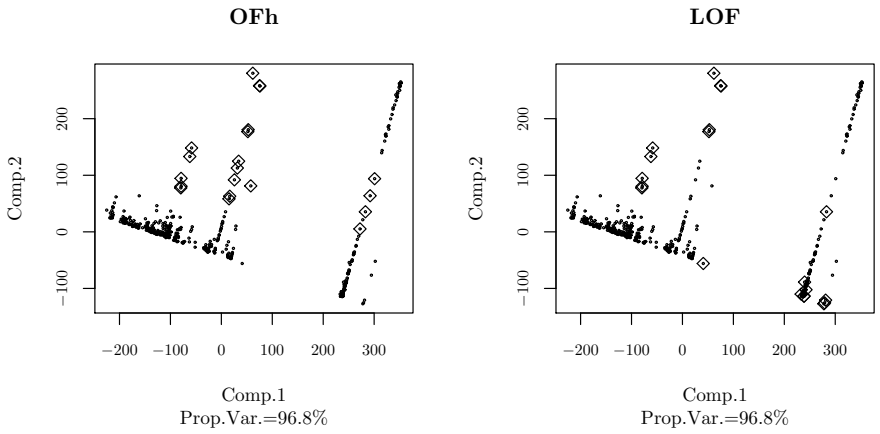


Fig. 3. The top 20 outliers for the Boston Housing domain.

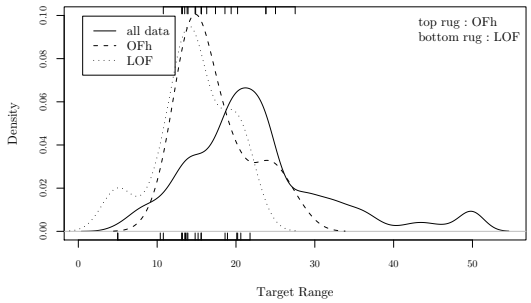


Fig. 4. Boston target variable distribution for all data set and for the outliers.

Figure 5 shows the results for the Abalone domain. In this case it is interesting to note the quite different focus of the methods. LOF focus on local outliers, i.e. cases that deviate slightly from the main bulk of data. This is not surprising as that is the main goal of this method. Our proposal, on the contrary seems to be focused on other type of outliers that deviate largely from the main bulk of data and are located in less dense regions. Both methods provide correct indications with respect to finding outliers (though LOF seems to be making a few mistakes). We have checked that our method also ranks high the outliers signalled by LOF, though not including them on the top 20. It is interesting to check whether these two different groups of outliers also correspond to different target variable values. Figure 6 confirms this. The outliers signalled by our method have unusually high number of rings, whilst LOF seems to be more focused on extreme low values.

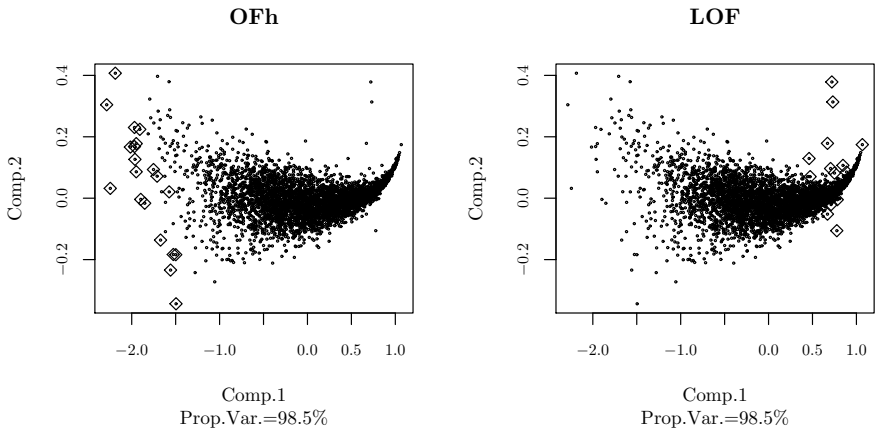


Fig. 5. The top 20 outliers for the Abalone domain.

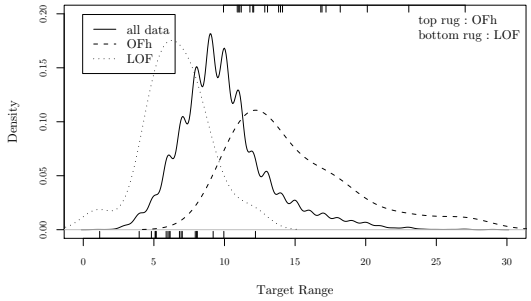


Fig. 6. Abalone target variable distribution for all data set and for the outliers.

Figure 7 shows the solutions for the Alga 1 domain. Once again both methods do a good job at spotting the most deviating observations. Still, we can claim a slight advantage of our proposal as a few outliers signalled by LOF can hardly be regard as such. With respect to the corresponding target variable distribution (c.f. Figure 8) the advantage of our method is not so clear, as LOF includes two larger algae blooms on its top 10 outliers.

Finally, Figure 9 shows the rankings for the Machine CPU domain. We should start by referring that due to the fact that the two principal components only use two of the original variables of the domain, several data points get plotted on the same place as they have equal values on these variables. That is the explanation for apparently less than 10 outliers being plotted in the graphs. In this domain we observe a clear advantage of our proposal as LOF misses the most obvious outliers. This is confirmed when looking at the distribution

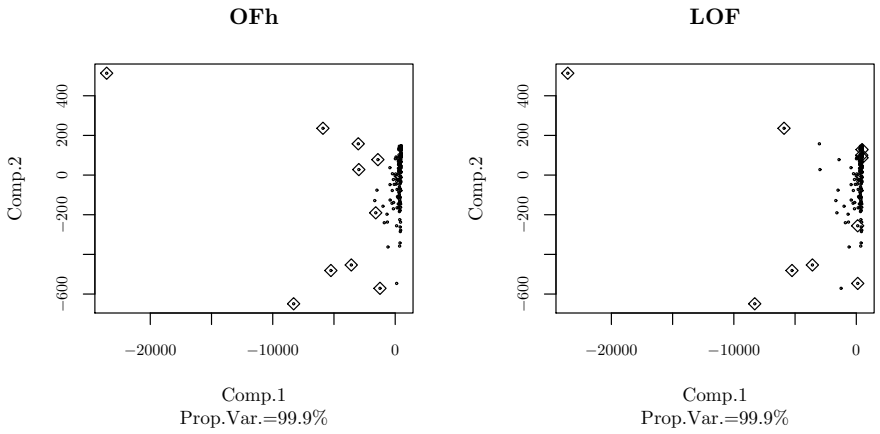


Fig. 7. The top 10 outliers for the Alga 1 domain.

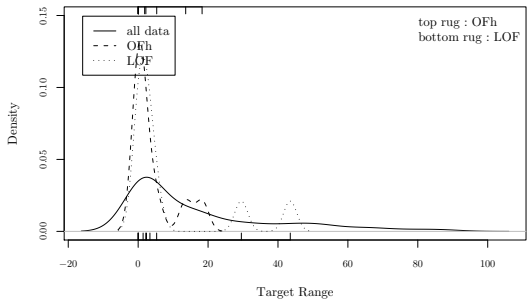


Fig. 8. Alga 1 target variable distribution for all data set and for the outliers.

data (Figure 10), where we see that the outliers ranked higher by our method effectively correspond to the most extreme values of CPU performance.

4 Conclusions

In this paper we have presented an outlier ranking method that can be applied to fraud detection problems allowing a resource-aware planning of any posterior inspection activities, which is a key requirement of several business activities.

Compared to other existing approaches the most distinguishing feature of our work is the fact that it relies on the output of common hierarchical clustering algorithms, thus not requiring additional computational efforts to obtain a ranking of outliers.

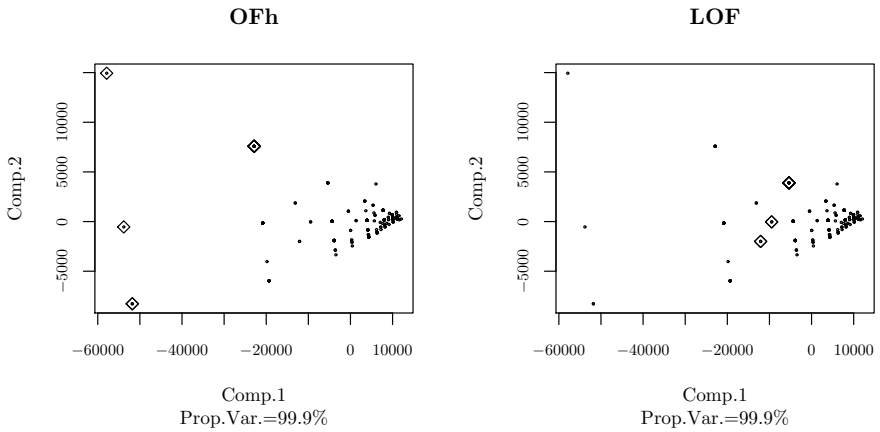


Fig. 9. The top 10 outliers for the Machine CPU domain.

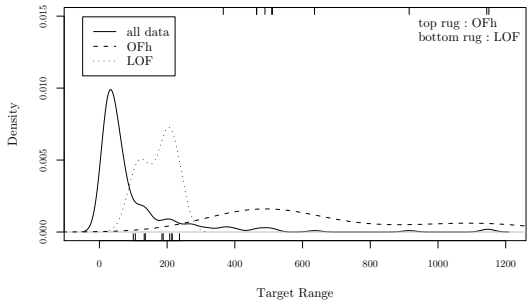


Fig. 10. Machine CPU target variable distribution for all data set and for the outliers.

The initial set of experiments that we have presented, comparing our method to a state of the art alternative (LOF), confirm the validity of our proposal. In effect, we have always found the results of our method to be equivalent and in some cases even slightly superior to the results of LOF.

Further work should extend the analysis of the method both experimentally as well as theoretically. We also plan to make a more deep analysis of the computational requirements of our proposal.

References

1. Edgar Acuna and Caroline Rodriguez. *dprep: Data preprocessing and visualization functions for classification.* R package version 1.0.

2. Victoria Hodge ; Jim Austin. A survey of outlier detection methodologies. *Artificial Intelligence Review*, 22:85–126, 2004.
3. R.J. Bolton and D. J. Hand. Unsupervised profiling methods for fraud detection. In *Credit Scoring and Credit Control VII*, 2001.
4. S. Ghosh and D. Reilly. Credit card fraud detection with a neural network. In *Proc. of the 27th Annual Hawaii Intern. Conf. on System Science.*, volume 3 of *DSS/Knowledg-Based Systems*, pages 621–630, 1994.
5. Richard J. Bolton ; David J. Hand. Statistical fraud detection: A review. *Statistical Science*, 17(3):235–255, 2002.
6. D. Hawkins. *Identification of Outliers*. Chapman & Hall, 1980.
7. L. Kaufman and P. Rousseeuw. *Finding Groups in Data: an introduction to cluster analysis*. Wiley Series in Probability and Mathematical Statistics, 1990.
8. F. Murtagh. Complexities of hierarchic clustering algorithms: state of the art. *Computational Statistics Quarterly*, 1:101–113, 1984.
9. F. Murtagh. Multidimensional clustering algorithms. *COMPSTAT Lectures 4, Wuerzburg: Physica-Verlag*, 1985.
10. E. Knorr ; R. Ng. Algorithms for mining distance-based outliers in large datasets. In *Proc. 24th Int. Conf. Very Large Data Bases, VLDB*, pages 392–403, 1998.
11. R. Ng and J. Han. Efficient and efective clustering method for spatial data mining. In *Proc. of VLDB'94*, 1994.
12. K. Nigam, A. McCallum, S. Thrun, and T. Mitchell. Text classication from labeled and unlabeled documents using em. *Machine Learning*, 39:103–134, 2000.
13. C. Phua, V. Lee, K. Smith, and R. Gayler. A comprehendive survey of data mining-based fraud detection research. *Artificial Intelligence Review*, (submitted), 2005.
14. F. Preparata and M. Shamos. *Computational Geometry: an introduction*. Springer-Verlag, 1988.
15. Tom Fawcett ; Foster Provost. Adaptive fraud detection. *Data Mining and Knowledge Discovery*, 1(3):291–316, 1997.
16. R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, 2007. ISBN 3-900051-07-0.
17. M. Breunig ; H. Kriegel ; R. Ng ; J. Sander. Lof: identifying density-based local outliers. In *ACM Int. Conf. on Management of Data*, pages 93–104, 2000.
18. L. Torgo. Repository of regression data sets, `http://www.liaad.up.pt/~ltorgo/Regression/DataSets.html`.
19. J. Ward. Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, 58(236), 1963.