Credit-Card Fraud Profiling Using a Hybrid Incremental Clustering Methodology

Marie-Jeanne Lesot, Adrien Revault d'Allonnes

Credit-Card Fraud Profiling Using a Hybrid Incremental Clustering Methodology

Marie-Jeanne Lesot and Adrien Revault d'Allonnes

LIP6, Université Pierre et Marie Curie-Paris 6, UMR7606
4 place Jussieu
Paris cedex 05, 75252, France

Abstract. This paper addresses the task of helping investigators identify characteristics in credit-card frauds, so as to establish fraud profiles. To do this, a clustering methodology based on the combination of an incremental variant of the linearised fuzzy c-medoids and a hierarchical clustering is proposed. This algorithm can process very large sets of heterogeneous data, i.e. described by both categorical and numeric features. The relevance of the proposed approach is illustrated on a real dataset containing next to one million fraudulent transactions.

Keywords: Incremental clustering; Hybrid Clustering; Bank Fraud; Credit Card Security.

1 Introduction

With the generalisation of credit and debit cards as modes of payment, credit-card frauds in e-commerce and other mail-order or distant transactions have become a major issue for all banks and card-issuers. For instance, according to the 2011 annual Banque de France report [1], whereas the overall 2010 fraud rate in France is as low as 0.074%, corresponding to an amount of € 368.9 million, the frauds in domestic card-not-present payments (i.e. made online, over the phone or by post) represent 0.26% of this type of transaction, about three and a half times more. These frauds represent 62% of all fraud cases in terms of value.

As a consequence, the analysis and automatic detection of fraudulent transactions has become a largely studied field in the machine learning community [2–4], in particular in the case of e-commerce. This task is both essential from an application point of view, as mentioned above, and scientifically highly challenging, because of its difficulty, part of which is due to the quantity of data that must be processed and the extreme class imbalance.

From a machine-learning standpoint two problems should be separated, namely fraud detection and fraud characterisation. The former aims at predicting whether or not a given transaction should be accepted, so as to decline tentative frauds as they take place. Its objective is, therefore, to differentiate fraudulent and genuine transactions and it is, thus, part of the supervised-learning framework. As such, it should be formulated as a discrimination task in a highly imbalanced

two-class setting. This particular problem can use the card history to identify frauds as transactions that differ from the card-holder habits.

The second machine-learning problem, fraud characterisation, endeavours to identify distinct fraudster profiles which can then be conceived as operational procedures and used as investigative tools in the apprehension of fraudsters or to assist in fraud detection. The objective is, therefore, to identify, in a set of frauds, distinct subtypes of frauds exhibiting similar properties. It is part of the unsupervised-learning framework and is essentially a clustering task applied to the set of all fraudulent transactions. It should be noted that, in this case, card history is of no use, since fraudster profiles are independent from card-holder habits and, therefore, frauds need only be compared to other frauds and not to legitimate transactions.

In this paper, we consider the latter type of approach and propose a hybrid incremental clustering methodology to address this task. This method exhibits the following characteristics: first, it ensures a rich description of the identified fraudster profiles and allows a multi-level analysis, through the *hierarchical structure* of the extracted clusters it yields. Second, the method allows the processing of large datasets. Indeed, even if frauds represent a small minority of all transactions, they are still very numerous. This is the reason why we propose to combine a hierarchical clustering step, to organise the identified clusters in a dendrogram but with a very high computational cost, to a preliminary data-decomposition step, through an efficient partitioning step. For the partitioning step, we propose an *incremental approach* which processes the dataset in smaller subsets. Third, the method can deal with *heterogeneous data*, i.e. data whose features can be either categorical or numeric. Indeed, apart from the amount which is a number, transactions are, for instance, described by the country where they take place or the general category of transacted product. Finally, on a more general level, the method is *robust*, that is, it does not suffer from its random initialisation.

In the next section, we describe the methodology proposed to address this task. Section 3 then presents the experimental results obtained on real data.

2 Proposed Methodology

Clustering data that are both in vast amounts and of a hybrid nature imposes constraints on candidate algorithms. In this section, we outline the main approaches dealing with these issues and we describe in more detail the linearised Fuzzy c-Medoids [5] on which the proposed methodology relies. We then detail the proposed methodology.

2.1 Related Work

Clustering Large Data Sets Very large datasets, having become more and more common, have given rise to a large diversity of scalable clustering algorithms.

One way to tackle the problem is to make existing algorithms go faster with specific optimisations. For instance, acceleration of the k-means method and its variants can be achieved using improved initialisation methods to reduce the number of iterations [6, 7]. For the Partitioning Around Medoids (PAM [8]) approach, CLARANS [9] or the linearised fuzzy c-medoids algorithm [5], detailed below, alleviate computational costs by updating medoids in their vicinity.

Another approach, incremental clustering, iteratively applies a clustering algorithm on data subsamples which are processed individually. The samples are extracted from the dataset (e.g. randomly) or they can be imposed by a temporal constraint, when the data is available as time goes by. One variant proposes to build a single sample, guaranteeing its representativeness, so that the results of a single application of a clustering algorithm can be considered as meaningful for the entire dataset [10].

In the general case, however, each sample is clustered and these partial results are then merged into the final partition of the dataset [11]. This fusion can be performed progressively by including in the clustering step of a given sample the results from the previous steps: a sample is summarised by the extracted clusters and this summary is processed together with the next sample [12–14]. The incremental variant of DBSCAN [15] proposes an efficient strategy to determine the region of space where the cluster structure identified in the previous samples should be updated and then locally applies the DBSCAN algorithm. Alternatively, the fusion can be performed at the end, when all samples have been processed, for instance by applying an additional clustering step to the centres obtained from each sample. BIRCH [16], for example, incrementally performs a precluster-ing step to build a compact representation of the dataset, based on structured summaries that optimise memory usage along user-specified requirements. The centres obtained after scanning the whole dataset then undergo a clustering process. CURE offers a compromise between hierarchical and partitioning clustering, by both using cluster representative points and applying cluster fusion [17]. Both approaches, progressive and closing fusions, can also be combined [18].

Clustering Heterogeneous Data Hybrid data, data described by both numeric and categorical attributes, define another case where specific clustering algorithms are required. Such data rule out the usage of all mean-centred clustering techniques, in particular the very commonly applied k-means and its variants.

Two main approaches can be distinguished for this problem: so-called relational methods which rely on the pairwise dissimilarity matrix (e.g. based on the pairwise distances) and not on vector descriptions of the data. This type of approach includes, in particular, hierarchical clustering methods, density-based methods [19] as well as relational variants of classic algorithms [20, 21].

On the other hand, medoid-based methods [8, 5] constitute variants of the mean-centered methods that do not define the cluster representative as the average of its members, but as its medoid, that is, the data point that minimises the possibly weighted distance to cluster members.

Linearised Fuzzy *c*-Medoids The linearised fuzzy c-medoids algorithm, denoted *l-fcmed* in the following, combines several properties of the previously listed algorithms: it can process data that are both in vast amounts and of a hybrid nature [5]. Indeed, it belongs to both accelerated techniques and medoid-based methods. Moreover, being a fuzzy variant of such algorithms, it offers properties of robustness and independence from random initialisation.

More formally, if x_i, $i = 1, \ldots, n$ are the data points, c the desired number of clusters, v_r, $r = 1, \ldots, c$ the cluster centres and u_{ir} the membership degree of datum x_i to cluster r, the algorithm alternatively updates the membership degrees and the cluster centres using the following equations:

$$u_{ir} = \left[\sum_{s=1}^{c} \left(\frac{d(x_i, v_r)}{d(x_i, v_s)} \right)^{\frac{2}{m-1}} \right]^{-1} \qquad v_r = \underset{k \in N(v_r)}{\operatorname{argmin}} \sum_{i=1}^{n} u_{ri}^m d(x_k, x_i) \tag{1}$$

where m is the so-called fuzzifier, d a suitable metric and $N(v_r)$ the neighbourhood of centre v_r. The latter is defined as the p data maximising membership to cluster r.

The membership degrees are, thus, updated as in the fuzzy c-means and the cluster centres as the data points that minimise the weighted distance to cluster members. To reduce the computational cost, *l-fcmed* searches for a suitable medoid update close to each current medoid, in $N(v_r)$, instead of computing the minimum over the whole dataset. Both updates are iterated until medoid positions stabilise.

The *l-fcmed* parameters are c, the number of clusters, m, the fuzzifier, and p, the neighbourhood size. The algorithm also depends on the chosen metric d.

2.2 Global Architecture

To cluster fraudulent transactions, we propose a two-step methodology, illustrated in Figure 1, inspired from the existing approaches described above: before performing a hierarchical clustering, because of its high computational cost, we operate a segmentation using a partitioning algorithm. Because of its advantages, listed in the previous section, we choose to use the linearised fuzzy c-medoids, or rather we propose an incremental extension to further limit computational strains, which we describe in the following.

The second step then uses a hierarchical clustering method to generalise the obtained clusters. Its output dendrogram allows the data analyst to choose the desired level of compromise between homogeneity and generality.

2.3 Incremental Partitioning Step

Following the classic incremental methodology, instead of performing the partitioning clustering on the whole dataset, we operate *l-fcmed* iteratively on randomly selected samples of size n_l. As detailed below, we propose to introduce two substeps to improve its efficiency in the considered global architecture: medoid selection and unaffected fraud allocation.

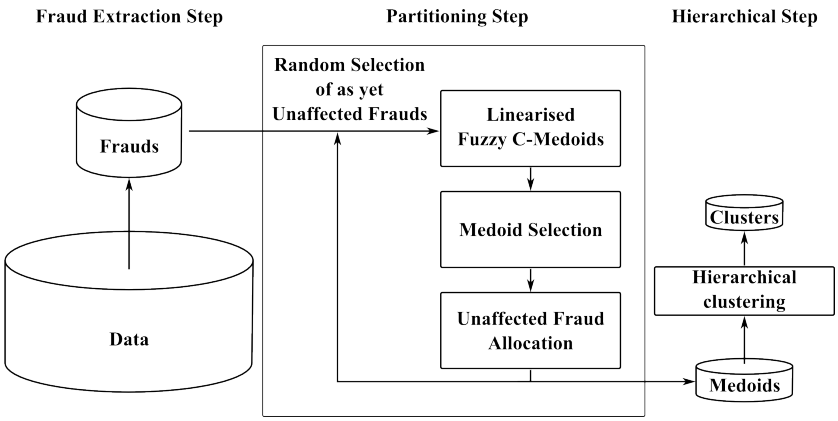

Fig. 1. Global architecture of the proposed methodology

Medoid selection The aim of the partitioning step is to purposefully build an over-segmentation to summarise the data while minimising the loss of information, as it is a preliminary step to the hierarchical clustering step. We thus force a compactness constraint on the over-segmented clusters, in order to keep only the most homogeneous, discarding the rest.

To select the clusters C_r, $r = 1, \ldots, c$, whose medoids are highly representative of their assigned data, we propose to keep those of sufficient size and exhibiting a very high homogeneity level. The latter is evaluated by a measure of the dispersion of members of C_r that could, for instance, be relative to the cluster diameter, $diam(C_r) = \max_{x_i, x_j \in C_r} d(x_i, x_j)$. The selection criterion can thus be formalised as:

$$size(C_r) = |C_r| > \tau \qquad \text{and} \qquad disp(C_r) \leq \xi \tag{2}$$

where τ is the minimal acceptable size and ξ a user-set compactness threshold.

All data in the discarded clusters is then put back into the general pool of transactions to be clustered. They thus become candidates for the random sample selection of following iterations of the partitioning step.

Unaffected Fraud Allocation Before iterating to the next sample, we scan the data that are yet to be clustered, so as to add unaffected frauds to the identified clusters. This cluster augmentation has a double advantage: first, it avoids the discovery in subsequent iterations of clusters similar to the selected ones, i.e. it avoids cluster duplication or redundancy. It therefore simplifies the posterior fusion step. Moreover, it further alleviates computational costs by reducing the size of the frauds to cluster in following iterations.

This is done by selecting, from the pool of unclustered data, those frauds which can be allocated to the selected clusters without degrading their quality,

that is, frauds which are sufficiently close to the medoid and are in the allowed dispersion. Formally an unaffected fraud x is assigned to cluster C_r if:

$$d(x, v_r) < disp(C_r) \tag{3}$$

This step can be seen as similar to the extension step performed by [10] but, in our case, it remains a tentative extension, that is, only performed under the condition that it does not deteriorate cluster dispersion.

Having built a compact and homogeneous partition of a subset of the data through over-segmentation, medoid selection and cluster augmentation, the process is repeated until no further cluster meets the required standards.

2.4 Hierarchical Step

Once all homogeneous clusters satisfying the constraints have been created, a hierarchical clustering with complete linkage is operated on the resulting medoids. Because we have heavily reduced the volume of data with our partitioning, this step is computationally acceptable. The resulting hierarchy offers a progressive agglomeration of clusters and allows for the selection of a suitable compromise between cluster density and number of clusters. The selection of this compromise is made by the visual examination of the hierarchy dendrogram.

3 Experimental Results

3.1 Data and Experimental Setup

We applied the proposed hybrid incremental clustering methodology to a dataset made of 958 828 fraudulent transactions. These correspond to transactions that were rejected by the actual card-holders who, in this way, label the data and identify the transactions to be considered as frauds.

Each fraud in the dataset is described as a combination of numeric and categorical features. The first type of attributes includes the amount of the transaction in euros, a positive real number. The second type includes the country where the transaction took place and the merchant category code of the product.

The distance d between two transactions t_1 and t_2, represented as vectors of their features, is defined as the average of the distances on each attribute, i.e. $d(t_1, t_2) = 1/q \sum_{i=1}^{q} d_i(t_{1i}, t_{2i})$, where d_i is the distance for attribute A_i. This is either $d_i = d_{cat}$, if A_i is categorical, or $d_i = d_{num}$ if it is numeric. Each is defined as follows:

$$d_{cat}(x, y) = \begin{cases} 1 & \text{if } x \neq y \\ 0 & \text{otherwise} \end{cases} \qquad d_{num}(x, y) = \frac{|x - y|}{\max(x, y)}$$

The distance for categorical attributes d_{cat} is binary: it equals 0 if the two values to be compared are identical, and 1 in all other cases. The distance for numeric attributes is defined as a relative gap: the assumption behind this is that a

difference of 2€ in amount, for instance, should not have the same impact if the compared amounts are around 5€ or if they are closer to 1 000€.

Regarding the parameters , we use the following setup: each sample contains $n_l = 50\,000$ randomly selected data. *l-fcmed* is initialised randomly and applied with the high value $c = 4\,000$ because we want an over-segmentation. We use the common value $m = 2$ for the fuzzifier. The size of the neighbourhood around the medoids in which the following is selected is $p = \lfloor 50\,000/4\,000 \rfloor$.

Medoid selection, as presented in equation 2, depends on cluster size and dispersion. The minimal size required is set at $\tau = 10$. Since it bears on all attributes, we write the dispersion metric $disp(C)$ as the vector of its attributes' dispersions. For categorical attributes, the number of different values represents dispersion; for numeric attributes, dispersion is best represented by their standard-deviation normalised by their mean value. Formally supposing that for attribute A we write its value in x as $A(x)$, and its average value in cluster C as $\overline{A_C} = \sum_{x \in C} A(x)/|C|$, we may write these dispersions as:

$$disp_{cat}(C) = |\{A(x) | x \in C\}| \qquad disp_{num}(C) = \frac{1}{|C|} \frac{\sqrt{\sum_{x \in C} (A(x) - \overline{A_C})^2}}{\overline{A_C}} \quad (4)$$

In this way, $disp(C)$ is the vector of all dispersions and ξ is also a vector giving local thresholds, which we set at $\xi_{cat} = 1$, limiting the categorical attributes to a single value in the selected clusters, and $\xi_{num} = 0.01$.

The cut threshold, for the hierarchical clustering, is set to 0.5, based on a visual inspection of the dendrogram (see Figure 4) to achieve an acceptable compromise between the final number of clusters and their homogeneity.

3.2 Incremental Partitioning Step: Results

The evolution of the incremental partitioning step is illustrated on Figures 2 and 3. Figure 2 shows the number of frauds assigned at each iteration, before the allocation of unaffected frauds. It should be observed that this number, starting at 18 373, represents 36.7% of the processed 50 000 points. This small proportion of selected data illustrates how the compactness constraint rejects a large amount of clusters and data. After this initial high value, the amount of selected fraudulent transactions decreases very rapidly. The curve presents an obvious asymptote around 500, i.e. around less than 1% of the considered 50 000 lines of data. This, in itself, is a satisfactory justification for stopping the partitioning step after the illustrated 50 iterations.

The left graph in Figure 3 shows the cumulative number of clusters selected at each iteration. At the first iteration, 394 clusters are selected as being quality clusters. By the end of the process 4 214 have been identified. The inflexion of the curve indicates that the number of newly discovered quality clusters decreases with the number of iterations, another explanation for stopping after fifty iterations of the partitioning step.

The right graph in Figure 3 shows the cumulative number of assigned data after reallocation at each iteration, i.e. the total number of transactions that have

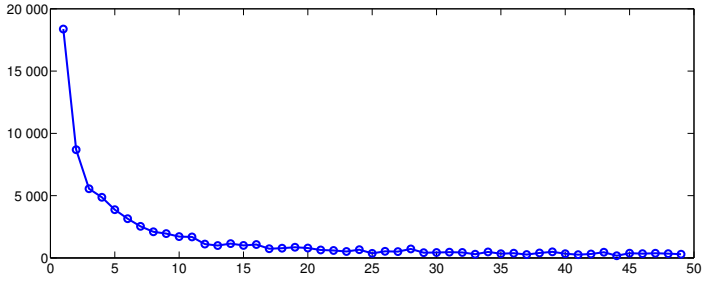

Fig. 2. Evolution of the incremental partitioning step: number of assigned transactions, at each iteration.

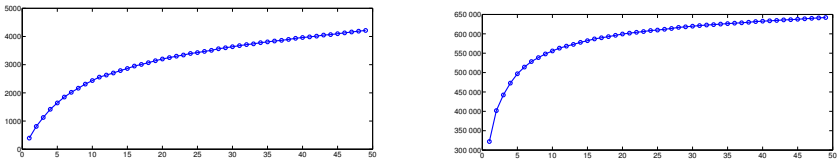

Fig. 3. Evolution of the incremental partitioning step: (left) cumulative number of selected clusters, at each iteration, (right) cumulative number of assigned data after reassignment at each iteration.

been assigned to any of the created clusters. 322 115 transactions are assigned at the very first iteration, which indicates a high redundancy in the considered data: many transactions are identical or very close one to another in the considered description space, and thus fulfil the strict homogeneity condition imposed on the assignment step. At each further iteration, the number of assigned transactions drastically decreases, the last iterations bringing very little gain. This curve, thus, also argues in favour of ending iterations of the partitioning.

As a result of these different choices, the incremental partitioning step produces, in the end, 4 214 clusters containing 642 054 frauds.

3.3 Hierarchical Step: Results

Figure 4 shows the dendrogram obtained after applying a hierarchical clustering to the medoids obtained in the partitioning step. The same distance is used for medoids as for transactions.

Visual inspection of the dendrogram prompts a cut above a cost of 0.5. Indeed, cutting the dendrogram at 0.5 yields 156 clusters, which represents a reasonable compromise between cluster homogeneity and number of clusters. The resulting clusters have a small diameter and their number, 156 as compared to

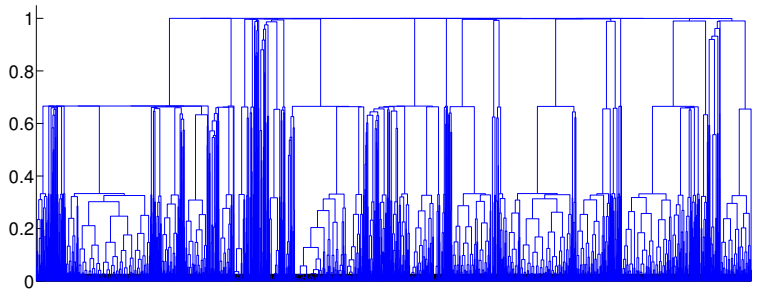

Fig. 4. Dendrogram of the hierarchical clustering step applied to the medoids obtained in the partitioning step.

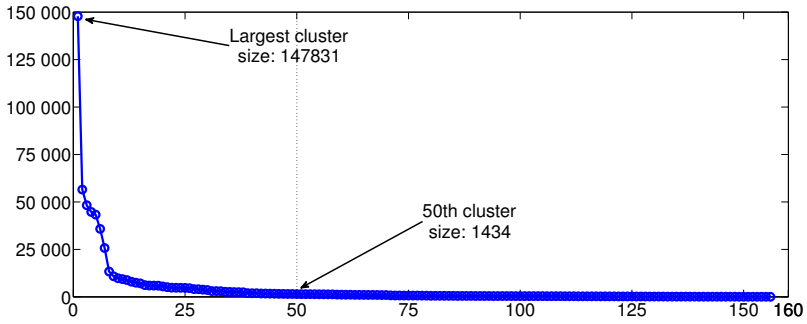

Fig. 5. Sizes of the final clusters.

the original 958 828 transactions, is a cognitively manageable amount of data to study for the human analyst.

3.4 Final Results

To take a closer look at the resulting clusters, we start off by studying the distribution of cluster sizes globally.

Figure 5 shows the cluster sizes in decreasing order and also shows a high disparity. In particular, the greatest cluster represents 23% of affected frauds on its own, whereas the second largest is only one third of the largest. More generally, the fifty largest clusters cover 92% of all affected frauds. In the following we successively comment the hundred and six smallest and the fifty largest clusters.

Analysis of the Smaller Clusters On average the smaller clusters, the ones after the first fifty, only contain 471 frauds, i.e. 7‰ of the affected frauds. It

could, therefore, be argued that they contain too little information and that it not necessary to study them further. Indeed, the fraud profiles they are associated to may seem too anecdotal. However, some of these clusters, or groups of clusters, may still warrant the analyst's attention because of some striking characteristics.

In this way, 9 clusters, representing 1 745 transactions, are each composed of exact replicas of a single transaction, that is, identical amount, country and activity. Even if the largest of these clusters only has 535 frauds, their being identical marks them as potential parts of a particular modus operandi, which any analyst might wish to investigate further.

Another trigger which might tingle an analyst's curiosity is exhibited by cluster 147, which has a mean amount of 914€, an oddly high value compared to the average fraud value of about 112€. Moreover, all its transactions are linked to the same country and activity. This activity only appears in this cluster. For these reasons, even if cluster 147 only has 72 transactions, a closer study of these transactions seems appropriate to find other similarities, such as the identity of the seller or the dates on which they took place, for instance.

Analysis of the Larger Clusters The fifty largest clusters represent the most notable fraudster profiles. Indeed, they are singularly homogeneous: just two of them, clusters 26 and 44, are not described by a single country and activity.

If we take a closer look at these two atypical clusters, we see that cluster 26 associates 4 546 frauds to three distinct merchant activities. Of these three activities, two are heavily outnumbered by the dominant one, the latter having 4 402 frauds, or 96.8% of the cluster, when the other two represent 80 and 64 transactions, that is, 1.8% and 1.4% respectively. Furthermore, these two minority activity types do not appear in any other cluster. The dominant activity does appear in five other clusters but with different countries, whereas cluster 26 only has one country. This cluster's homogeneity is also apparent in its amount distribution, since the range it covers is $[10, 13.16]$.

The other mixed cluster, cluster 44, has 1 670 fraudulent transactions and two countries. Once again, one of the countries is dominated by the other, representing just 104 individuals or 6.2% of the cluster population, and is only present in this cluster. These compactness anomalies are, therefore, slight and explainable.

If we turn back to the general population of large clusters, the 96% with only one activity and one country, we see that compactness does not constrain size. Indeed, the largest of all clusters, with 147 831 frauds, equivalent to 23% of all affected frauds, belongs to this category. Regarding the amounts, this cluster spreads over the $[1, 480]$ interval, a reasonable size compared to the global span of the data over $[0, 10\,276]$. The left part of Figure 6 shows the histogram of its amounts and also shows how the distribution is concentrated on small values.

Another illustration of the first fifty clusters' quality, is given by cluster 12. This cluster, still defined over a single country and activity, has 8 828 frauds with amounts on the interval $[0.99, 83.73]$. The right part of Figure 6 offers a view on their dispersion. Looking at this distribution, we see that it could be subdivided into homogeneous subintervals, probably the ones given by the partitioning step,

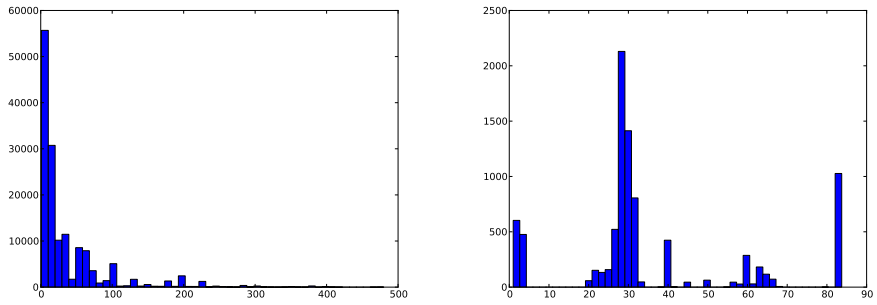

Fig. 6. Histogram of the amounts (left) for the largest cluster (population: 147 831; bin-width: 9.58€), (right) for 12^{th} largest cluster (population: 8 828; bin-width: 1.65€).

later joined by the hierarchical clustering. Cluster 12, thus, illustrates the use of this fusion step: instead of studying individually all 93 original clusters – the clusters fused during the hierarchical step to form cluster 12 – the analyst can focus on a generalised view, yet still be able to identify potentially interesting subgroups. The analyst may yet explore these subgroups by choosing to cut the dendrogram at a lower value. This inspection can be made locally by concentrating on the branch which actually contains the interesting clusters. This granularity refocusing ability, local or not, emanating from the cluster hierarchy, is an added benefit and justification for the proposed global architecture of the clustering method.

4 Conclusion

In this paper, we proposed a methodology for the identification of the characteristics of credit-card frauds, through the identification of distinct fraud profiles. It is based on the combination of an incremental variant of the fuzzy c-medoids with hierarchical clustering, and it is thus able to process very large heterogeneous data. We illustrated the relevance of the proposed approach on a real dataset describing next to one million online fraudulent transactions.

Ongoing work aims at enriching the interpretation of the obtained profiles, in particular by the construction of typical transactions representing each fraud profile, so as to ease their characterisation. To that aim, the use of fuzzy prototypes is considered, in order to underline the specificity of each profile as opposed to the others.

Acknowledgments

This work was supported by the project eFraudBox funded by ANR-CSOSG 2009. The authors also thank Nizar Malkiya for his help in implementing and testing the methodology.

References

1. Banque de France: Annual Report of the Observatory for Payment Card Security. http://www.banque-france.fr/observatoire/telechar/gb/2010/rapport-annuel-OSCP-2010-gb.pdf (2010)
2. Bolton, R.J., Hand, D.J.: Statistical fraud detection: a review. Statistical science **17**(3) (2002) 235–255
3. Phua, C., Lee, V., Smith, K., Gayler, R.: A comprehensive survey of data mining-based fraud detection research. Artificial Intelligence Review (2005)
4. Laleh, N., Azgomi, M.A.: A taxonomy of frauds and fraud detection techniques. Information Systems, Technology and Management Communications in Computer and Information Science **3** (2009) 256–267
5. Krishnapuram, R., Joshi, A., Nasraoui, O., Yi, L.: Low complexity fuzzy relational clustering algorithms for web mining. IEEE Transactions on Fuzzy Systems **9**(4) (2001) 595–607
6. Cheng, T.W., Goldgof, D., Hall, L.: Fast fuzzy clustering. Fuzzy sets and systems **93** (1998) 49–56
7. Altman, D.: Efficient fuzzy clustering of multi-spectral images. In: Proc. of the IEEE Int. Conf. on Fuzzy Systems, FUZZ-IEEE'99. (1999)
8. Kaufman, L., Rousseeuw, P.: Finding groups in data, an introduction to cluster analysis. John Wiley & Sons, Brussels, Belgium (1990)
9. Ng, R., Han, J.: Efficient and effective clustering methods for spatial data mining. In: Proc. of the 20th Very Large DataBases Conference, VLDB'94. (1994) 144–155
10. Hathaway, R., Bezdek, J.: Extending fuzzy and probabilistic clustering to very large data sets. Computational statistics & data analysis **51** (2006) 215–234
11. Hore, P., Hall, L., Goldgof, D.: A cluster ensemble framework for large data sets. Pattern Recognition **42** (2009) 676–688
12. Farnstrom, F., Lewis, J., Elkan, C.: Scalability for clustering algorithms revisited. SIGKDD Explorations **2**(1) (2000) 51–57
13. Hore, P., Hall, L., Goldgof, D.: Single pass fuzzy c means. In: Proc. of the IEEE Int. Conf. on Fuzzy Systems, FUZZ-IEEE'07. (2007) 1–7
14. Hore, P., Hall, L., Goldgof, D., Cheng, W.: Online fuzzy c means. In: Proc. of NAFIPS'08. (2008) 1–5
15. Ester, M., Kriegel, H.P., Sander, J., Wimmer, M., Xu, X.: Incremental clustering for mining in a data warehousing environment. In: Proc. of the 24th Very Large DataBases Conference, VLDB'98. (1998) 323–333
16. Zhang, T., Ramakrishnan, R., Livny, M.: Birch: an efficient data clustering method for very large databases. In: Proc. of the ACM Int. Conf on Management of Data, SIGMOD'96, ACM Press (1996) 103–114
17. Guha, S., Rastogi, R., Shim, K.: CURE: an efficient clustering algorithm for large databases. In: Proc. of the ACM Int. Conf on Management of Data, SIGMOD'98. (1998) 73–84
18. Bradley, P., Fayyad, U., Reina, C.: Scaling clustering algorithms to large databases. In: Proc. of KDD'98, AAAI Press (1998) 9–15
19. Sander, J., Ester, M., Kriegel, H.P., Xu, X.: Density-based clustering in spatial databases: the algorithm DBSCAN and its application. Data Mining and Knowledge Discovery **2**(2) (1998) 169–194
20. Hathaway, R., Bezdek, J.: Nerf *c*-means: non euclidean relational fuzzy clustering. Pattern Recognition **27** (1994) 429–437
21. Hathaway, R., Bezdek, J., Davenport, J.: On relational data versions of *c*-means algorithms. Pattern Recognition Letters **17** (1996) 607–612