Combining Social Network Analysis with Semi-supervised Clustering: a case study on fraud detection

João Botelho
Instituto Superior Técnico
Av. Rovisco Pais 1
1049-001 Lisboa, Portugal
+351 218 419 407
joao.botelho@ist.utl.pt

Cláudia Antunes
Instituto Superior Técnico
Av. Rovisco Pais 1
1049-001 Lisboa, Portugal
+351 918 358 590
claudia.antunes@ist.utl.pt

ABSTRACT

At time of crisis, when fraud permanently frightens the basis of modern societies, the existence of effective tools to prevent it, or just to identify it in time, is critical. However, the detection of fraud is naturally impaired (among other issues) by the difficulty on labelling data, due to the cost of identifying and attest fraud. Moreover, the inability to incorporate domain knowledge in the mining process makes classifiers to use inadequate attributes to distinguish entities, ignoring most of existing relevant information, like entities' social relations. In this paper, we propose a new methodology for enriching semi-supervised clustering with information collected through the analysis of social networks. The methodology is then applied on tax fraud detection and assessed by measuring the impact of this enrichment in the accuracy of semi-supervised clustering methods.

Categories and Subject Descriptors

H.2.8 [**Database Applications**]: Data Mining

General Terms

Algorithms, Performance, Experimentation, Standardization, Theory.

Keywords

Semi-supervised clustering, Social network analysis, Fraud detection

1. INTRODUCTION

Fraud detection is a hard problem that introduces a significant number of challenges [1]. From these, the unbalanced nature of the datasets is one of the most studied [2], and the results achieved by classifiers adapted to deal with this problem are becoming acceptable.

However, supervised techniques require the existence of representative labelled datasets (*training set*) and the correct characterization of their instances to succeed. Indeed, these issues represent two other important problems. For example, classifiers trained with traditional techniques for fraud detection or churn prediction on telecommunications, usually present accuracy levels significantly below the average. Actually, both domains share two particularities: the difficulty on correctly labelling data and the fact that agents in those domains are humans.

The difficulty on labelling data resides on the fact that, on both cases, only manually validated cases can be assigned as positive instances. For example, in fraud detection any non-caught fraud just can be labelled as non-fraud. This means that an unknown amount of instances are incorrectly classified, which difficult the training process.

The quantity of labelled instances may become the most difficult issue to overcome, since it depends on the availability and ability of domain experts to label existing data. In this context, semi-supervised clustering may be an important tool, due to its ability to deal with a reduce amount of labelled data.

On the other hand, the fact that agents in those domains are humans, introduces some additional knowledge: humans are recognized both as social entities and to be socially influenced.

Acknowledging the need for methods that deal with a small amount of labelled data and the social nature of its instances, we propose a new methodology – *S2C+SNA*, based on the combination of semi-supervised clustering and social network analysis.

While semi-supervised clustering deal effectively with reduced amounts of labelled instances for creating a classifier, social network analysis contribute to derive new attributes for characterizing agents, which means that they make possible to introduce more information and domain knowledge on describing instances.

This paper focuses on the application of the proposed methodology to the problem of detecting fraud using small fractions of labelled data. After an overview of semi supervised clustering and its main algorithms, social network analysis is shortly summarized. The new methodology is described in section 4, illustrating its application in fraud detection. A case study on the same domain is performed in section 5. The paper concludes with a critical analysis of achieved results and presenting some guidelines for future work.

2. SEMI-SUPERVISED CLUSTERING

Clustering has played an important role in data analysis for decades, since it can identify major patterns or trends without any supervisory information, such as data labels.

Although the clustering analysis is in fact an unsupervised learning technique, it can be used as the basis for a classification model, if the dataset contains a classification variable. It can be defined as "*the organization of a collection of patterns (usually represented as a vector of measurements, or a point in a multidimensional space) into clusters based on similarity*" [3].

MacQueen's *K-Means* [4] is the simplest and most well-known clustering algorithm, and it is commonly used as a baseline when comparing the effectiveness of clustering algorithms. It groups the data into *K* clusters, creating a *k*-partition of the dataset. It starts with a random initial partition (initial *seed centroids*) and iteratively refines the clustering by assigning each instance to the nearest centroid; after that, the algorithm will compute each centroid again, as the mean of the instances of each cluster until a convergence criterion is met (e.g. there is no reassignment of any pattern from one cluster to another, or the squared error doesn't decrease significantly after some number of iterations).

K-Means try to minimize the total mean squared error of each point to its assigned cluster by minimizing the objective function. Typically, the initial centroids are chosen randomly, although the initial seed influences the final clusters determined by K-Means.

Traditional clustering is done in an unsupervised way, which means that, they only have access to a set of features describing each instance. In this manner, they do not use any other information about the domain or the dataset, ignoring for example constraints about how the instances should be placed within a partition. Because traditional algorithms have no way of taking advantage of this information, some clustering algorithms have been modified to incorporate background information, originating *semi-supervised clustering* algorithms. In particular, they can use some labelled instances to define constraints to be used during the partition process.

By making use of such information, semi-supervised clustering occupies the middle ground between supervised classification and unsupervised clustering.

The most common semi-supervised algorithms studied are modifications of the *K-Means* algorithm to incorporate domain knowledge. Typically, this knowledge can be incorporated when the initial centroids are chosen (*by seeding*) or in the form of constraints that have to be satisfied when grouping similar objects (*constrained algorithms*).

2.1 Semi-Supervised Algorithms by Seeding

The idea behind this kind of algorithms is to initialize the centroids of each cluster in a non-arbitrary way, and instead of it, to initialize each centroid with some labelled instance, usually called *seed*. Naturally, this procedure is only possible whenever there is at least one seed for each cluster. In such case, two of the main issues of clustering are solved. In one hand, the number of clusters is known in advance: it just corresponds to the number of labels, like for classification. On the other hand, each final partition is supposed to represent the set of instances that corresponds to the given label, which means that the meaning of each cluster is known.

The *Seeded-KMeans* algorithm [5] is an extension of *K-Means*, where the difference resides in the initialization procedure. Instead of initializing the centroids with random instances, it initializes the centroid of the i^{th} cluster with one instance from the set of instances labelled with the i^{th} label (or *class*). Seeds are only used for initialization, and are not used in the following steps of the algorithm.

As for *Seeded-KMeans*, in *Constrained-KMeans* [5] seeds are used for initializing the centroids, but the rest of labelled data is also assigned to corresponding clusters, remaining unchanged during all the discovery process. Only the labels of non-seeded data are re-estimated. *Constrained-KMeans* is appropriate when the initial seed labelling is noise-free, or if the user does not want the labels of the seed data to change.

2.2 Constrained Algorithms

For constrained clustering, grouping of similar objects into several clusters has to satisfy some additional conditions. Algorithms for this incorporate a set of *must-link* constraints (that impose that two specified instances should be in the same cluster), *cannot-link* constraints (that impose that two specified instances cannot be in the same cluster) or both. These constraints provide *apriori* knowledge about which instances should be grouped or not. In clustering with hard constraints, the goal is to minimize the objective function subject to satisfying the constraints. In this case, the objective function is the vector quantization error, or variance, of the partition.

MPCK-Means (*Metric Pair-wise Constrained K-Means*) [6] is a semi-supervised algorithm derived from *K-Means* that incorporates both metric learning and the use of pair wise constraints. These constraints are used both to seed the initial cluster centroids and to guide the clustering via the objective function. It is also able to learn individual metrics for each cluster, which allows for clusters with different shapes. Additionally, it allows for the existence of constraints violations, if it leads to a better cohesive clustering.

In order to achieve its goal, *MPCK-Means* generates a *K*-partitioning of the dataset that locally minimizes the objective function by using the *EM* algorithm. In this algorithm, the objective function combines the objective function in *K-Means* with the costs of violating constraints. In concrete, the *E-step* consists on assigning each point to the cluster that minimizes the objective function given each data point and the previous assignments to clusters. The *M-step* consists of two parts: re-estimating the cluster centroids given the *E-step* cluster assignments and updating the metric matrices to decrease the objective function.

MPCK-Means can be simplified to the *PCK-Means* algorithm, which works in the same manner but without the metric learning component.

3. SOCIAL NETWORK ANALYSIS

In the last century social networks were recognized to be a key factor on the behaviour of humans, and in the last decade, computer scientists have developed fundamental research on the area of social network analysis.

From the point of view of computer science, a social network corresponds to a directed graph, whose nodes correspond to the agents in the society and the edges to the relationships among agents. Several have been the topics under research, but the determination of the centrality of an agent in the network has deserved a significant attention. This property reveals the relative importance of a node in the entire network, and can be seen as the impact of a node in the other nodes in the network.

The basic algorithms for this task are *Hits* [7] and *PageRank* [8], both developed to determine the centrality of web pages. Several others have been proposed after them [9].

Among these is *BadRank*, an algorithm used by Google to help detect spam web pages [10]. This algorithm is based on the principle that "*a page will get high BadRank value if it points to some pages with high BadRank values*". *BadRank* could be considered as a reversion of *PageRank*, because it gathers information on the forward links of a web page instead of using the backward links, as in *PageRank*.

The formula of *BadRank* is given by the following equation:

$$b(i) = e(i)(1-q) + q \sum_{j:i\to j} \frac{b(j)}{d_{in}(j)}, \qquad j = 1,2,\dots N$$

Where $b(i)$ is the *BadRank* value of page i; j is a page pointed by page i, with $b(j)$ as *BadRank* value for page j; $d_{in}(j)$ is the total number of the backward links of page j; q is a damping factor; $e(i)$ is the original *BadRank* value for page i, which is determined by the spam filter.

While *PageRank* measures the relative importance of a node in the network, based on the number of relations that it presents; *BadRank* measures the impact of the other nodes in each node, based on the "importance" of their relations.

Note that on both cases, each node in the network, which corresponds to each agent in the society, gains a new attribute – its rank, which can be used to help in its characterization.

4. *S2C+SNA* METHODOLOGY

Usually, the data for training classifiers, either supervised or unsupervised methods are based on static features. This is due to the nature of the data usually stored in information systems that mainly capture nominal and numeric attributes for instances, ignoring others with temporal or social nature.

Moreover, whenever the instances to mine are social agents, stored data is not enough to describe the instances, since it completely ignores the relationships among agents. For example, in the area of churn detection in telecommunications, experts argue that once someone (an influencer) at the centre of a network decides to change provider, the others in the network are likely to follow. Since, general information systems do not store the social network for each instance, traditional classifiers cannot explore that knowledge.

A form to overcome this lack of data is to enrich each instance with additional information, collected from the analysis of its social network. Indeed, whenever social networks involving those instances are known, or can be inferred, the description of instances can be extended by adding attributes resulting from the analysis of their social networks, like the rank determined by *PageRank* and *BadRank*.

Combined with semi-supervised clustering, those ranks contribute to incorporate existing domain information into the clustering process.

The *S2C+SNA* methodology is suitable for cases where there is not enough labelled data to train a supervised model and where the analysis of social networks can add value in the classification process. As Fig. 1 suggests, this methodology follows the different stages of the knowledge discovery process.

The proposed methodology assumes the existence of a database containing *n* data instances with labelled (*l*) and unlabelled instances (*u*), identified in the first step of the process (*Data Selection*). After this, it is required to proceed to *data cleaning* stage, as usual [11].

The enrichment stage considers the existence of information in the input database that allows the representation of instances in a graph *G*=(*V*, *E*) with the set of nodes *V* corresponding to the *n* instances, and the set of edges *E* corresponding to the links between those instances. With these two sets, is then possible to infer the graph representing the social networks for labelled instances *l*. With this graph we proceed to link analysis, using *BadRank*. The computation of the *BadRank* index for each instance is done through the analysis of its corresponding social network. This value is then added as a new feature to the instance.

Instances with a high *BadRank* rate have more chances of being in the same class.

At the coding stage, the dataset can be transformed or simplified in order to prepare it for semi-supervised clustering. For example, it is sometimes useful to code binary attributes into one bit, as this facilitates an efficient execution of clustering algorithms.

The application of semi-supervised clustering algorithms corresponds to the discovery stage of the proposed methodology. The main goal here is to assign labels to unlabelled instances *u,* using the domain information of labelled instances *l.*

Seeded-KMeans, *Constrained-KMeans* and *PCK-Means* are some of the most simple and popular algorithms that can be applied in this stage of the process. These algorithms are usually applied in cases where there is lack of labelled data, necessary to train supervised algorithms.

Evaluation of the results is not trivial because in most of the cases there isn't enough labelled data to compare results.

In semi-supervised clustering, the reporting stage differs with the application where it is used. For example in churn detection, these algorithms should indicate to marketers which individuals are

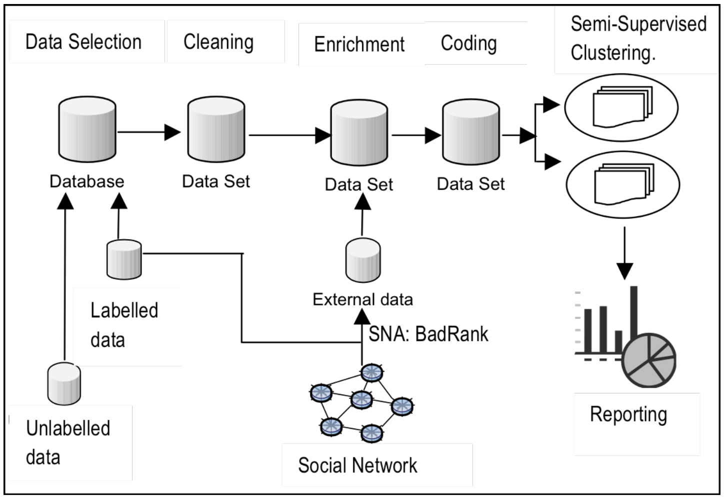

Fig. 1 ***S2C+SNA*** **Methodology**

more likely to churn, so that major efforts could be focused on them.

4.1 Example on Fraud Detection

The structure of social networks, in general, is very similar to the web-link structure, since social relations form a huge directed graph. Consider for example the case of taxes payment, where the taxes collector maintains individual records about each agent in the society, including his social relations, like familiar and work relations.

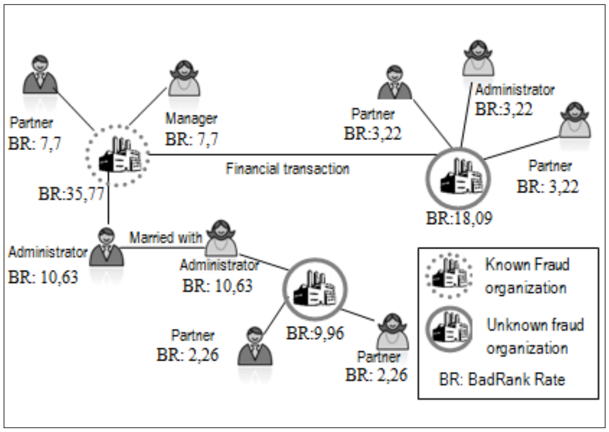

Fig. 2 Application of BadRank in a criminal network for taxes payment

It is clear, from Fig. 2, that we can use a directed graph to denote the nodes and relationships between the agents in this domain. The directed graph is marked as G=(V, E), where V can represent the set of all operators (companies and singulars) and E represents the set of all relationships (administrator of, married with, etc.). Applying social network analysis algorithms in fraud domain graphs allow us, not only to determine the centrality of all operators in the network (for example by using *PageRank*) but also the risk that is associated to an operator by analysing their links to other (fraudulent) operators, using the *BadRank* algorithm.

Fig. 2 illustrates the application of *BadRank* in a criminal network between three organizations, where only one of them is known to be fraudulent. By using this recognized case of fraud in the initialization of *BadRank*, is then possible to expose the other two organizations, since they will have a high *BadRank* rate for having some kind of relations with the fraudulent organization. Indeed, *BadRank* can be used to spread a fraud risk from a set of known fraudulent organizations to all their neighbours: the entities that are directly or indirectly connected to them.

5. CASE STUDY

The present case study is based on a dataset extracted from the information system of a financial organization. All information about the data is confidential because it could give to criminals the information that they required to evade detection. Because of that, the dataset will not be described in great detail.

It is important to refer that, like in other fraud domains, the original dataset was unbalanced, with only 10% of fraud instances in the entire set. This is a common problem in fraud detection (known as "skewed class distribution") and can be addressed in by forms of sampling and other techniques that transform the dataset into a more balanced one [2].

Since the experiments presented in this paper will focus only in the problem of detecting fraud with small fractions of labelled data, it was extracted a balanced dataset with three thousand instances, perfectly balanced, using sub sampling.

In coding stage, all nominal attributes were converted to numerical ones, missing values replenished by a default value (average) and all instances normalized to a range between 0 and 1.

In addition, we used a huge social network, with millions of nodes, containing different types of relationships between the entities stored in the information system, from working relations (like CEO, partner, manager) to family relations (like married to, child of, parent of). Despite its dimension, the social network is fragmented in disjoint sub-networks. Since the analysis of the entire network is largely time consuming and expects a great processing power, a sample of the network was extracted using Snowball Sampling [12]. The sample contemplates the sub-networks for each instance in the dataset.

With Snowball Sampling it is possible to locate one or more key individuals and ask them to name others who would be likely candidates for the research.

In studies of social networks, this sampling is used where the goal is to find out people's connections, and how they know each other. In this particular case, the goal is to find all the individuals that interact directly and indirectly with the individuals that will be analysed (contained in the dataset). In other words, the application of snowball sampling, having the individuals in the dataset as key nodes in the algorithm, allows for the identification of all the sub-components on the social network that are necessary to study interactions of these individuals.

To gain information about interactions among individuals with "fraud neighbours", the *BadRank* algorithm is applied to the social networks obtained by *Snowball Sampling* computation, using a fraction of pre-labelled fraud nodes. This originates a new attribute for each individual in the network, to be used in the classification process.

The study compares the accuracy obtained by applying three semi-supervised clustering algorithms: *K-Means*, *Constrained-KMeans*, *MPCK-Means* and *PCK-Means*, with and without the *BadRank* attribute.

Since the results on semi-supervised clustering and *BadRank* can diverge with the size of pre-labelled instances selected, all the experiments were conducted using different fractions of pre-labelled instances. These fractions go from 1% to 50%.

Using x% of pre-labelled instances in the experiment means that x% of the dataset was incorporated in the respective semi-supervised clustering algorithm and that x% of all fraud instances were used in the *BadRank* computation.

All the experiments were conducted selecting randomly 10 different sets of pre-labelled instances for each algorithm and for different fractions of labelled instances.

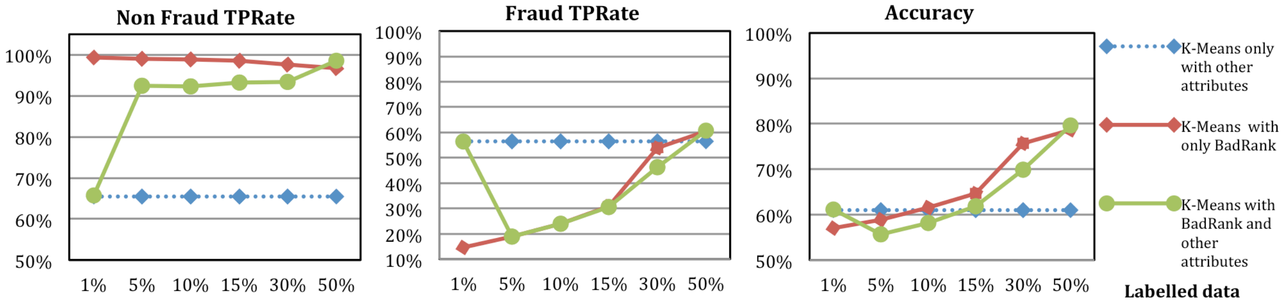

Fig. 3 K-Means results

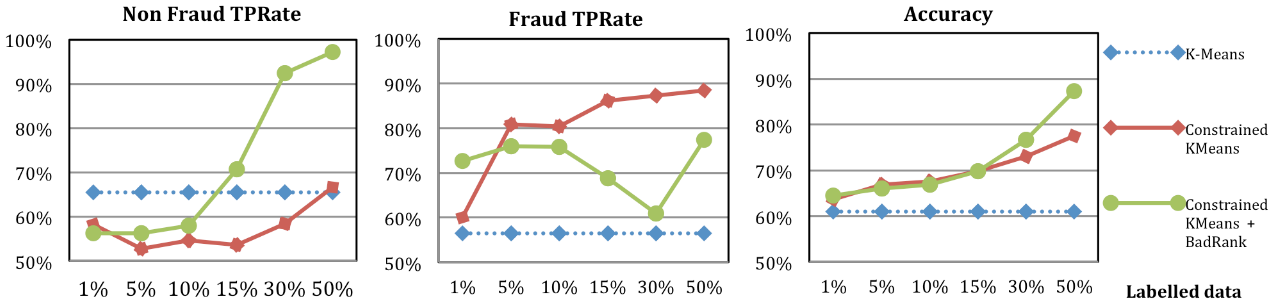

Fig. 4 Constrained-KMeans results

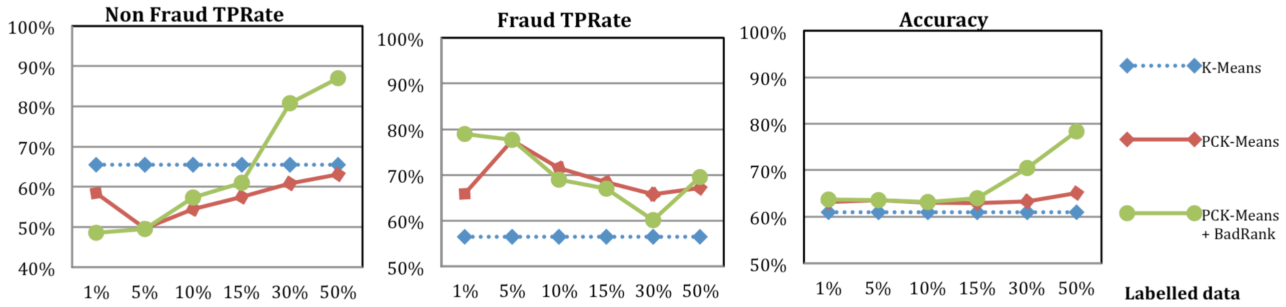

Fig. 5 PCK-Means results

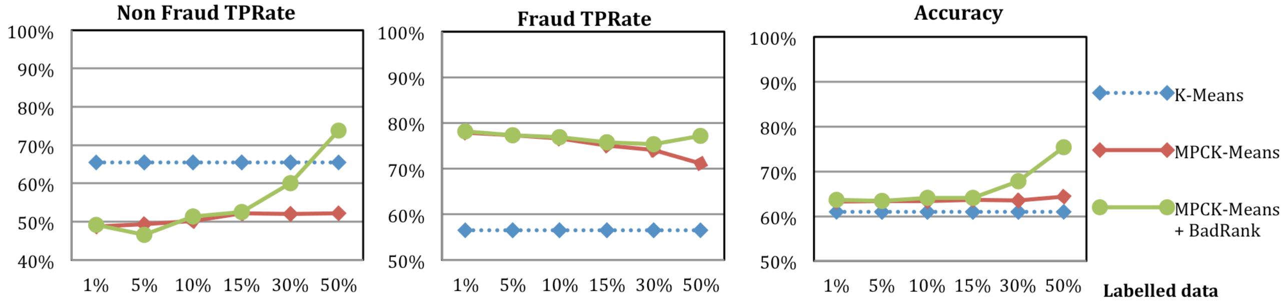

Fig. 6 MPCK-Means results

The results presented above report the best, worst and the average results obtained on those datasets. The algorithms used are implementations available at WEKAUT machine learning toolkit.[1]

The accuracy of the obtained clusters was computed using the *Rand Index* metric [13], which compares the resulting clusters with labelled data. This index has a value between 0 and 1, with 0 indicating that two clusters do not agree on any pair of clusters and 1 indicating that the clusters are exactly the same comparing to the "true" labels.

The charts on Fig. 3, Fig. 4, Fig. 5 and Fig. 6 show a summary of the results obtained in this case study. These charts show how the different algorithms behave in terms of total accuracy (fraction of instances correctly classified), *Non Fraud True Positive Rate* (the

[1] WEKAUT machine learning toolkit is a Data Mining open source tool, available online at http://www.cs.utexas.edu/users/ml/risc/code/.

fraction of non fraud instances correctly classified) and *Fraud True Positive Rate* (the fraction of fraud instances correctly classified), when the fraction of labelled instances varies.

Since *K-Means* algorithm makes no use of existing labelled data, the results produced remains constant with the variation of labelled data used, producing a straight line that is used as a baseline in all results.

Fig. 3 compares the results of applying *K-Means* before and after the enrichment stage, with the incorporation of the *BadRank* attribute. This figure also shows the performance of applying *K-Means* using the *BadRank* attribute by itself.

With *BadRank* we can notice that it produces a raised increase on *Non Fraud TPRate* and a reduction on *Fraud TPRate*. The accuracy only starts to increase, relatively to the baseline, after 10% to 15% of labelled instances. By using just the *BadRank* attribute, it produces little better results than with its combination with other attributes.

It is also important to notice that for small fractions of labelled data (less than 10%) K-Means outperforms the more informed alternatives, resulting from the incorrectly estimation of fraudulent cases from a small portion of labelled data.

Fig. 4 shows that the combination of *BadRank* risk factor and *Constrained-KMeans*, also achieved good results, by improving the total accuracy in 10% with the presence of *BadRank*. Note that, in this case, *Fraud TPRate* decreases and *Non Fraud TPRate* increases, meaning that less false positives will be prompted. Again the differences become notorious with more than 15% of labelled instances.

Also notice that the accuracy for *Constrained K-Means* is always better than the baseline.

Fig. 5 shows that *PCK-Means* always tends to decrease *Fraud TPRate* and to increase *Non Fraud TP Rate*. Significant improvements are observed after 15% of labelled instances used with the incorporation of *BadRank*.

The results of *MPCK-Means* presented on Fig. 6 shows that Fraud TPRate remains almost the same with and without *BadRank*. The most significant differences are detected on *Non Fraud TPRate*, which is higher with *BadRank* using more than 15% of labelled instances.

Fig. 7 presents a summary of the accuracies obtained with the experiments. It also compares accuracy results with and without *BadRank*.

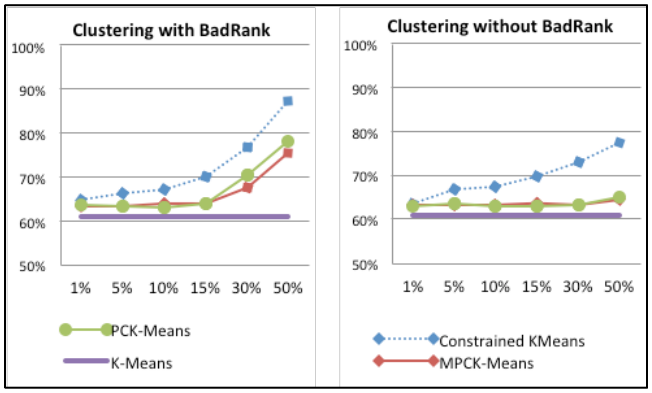

Fig. 7 Average results with and without BadRank attribute

From these results it is clear that with a small fraction of labelled instances (about 15%) all semi-supervised algorithms obtain a significant improvement when comparing to the unsupervised clustering (*K-Means*). When the fraction of labelled instances grows these algorithms reacts in different ways. *Constrained K-Means* have the best performance when comparing to other semi-supervised algorithms. *PCK-Means* and *MPCK-Means* don't reveal significant differences on accuracy without *BadRank*. With the incorporation of *BadRank*, the results show significant improvements in all experiments, after 15% of labelled instances used.

The best and worst results obtained with experiments without *BadRank*, presented on Fig. 8, shows that, although all algorithms increase their best results as more labelled instances are made available, constrained algorithms (*MPCK-Means* and *PCK-Means*) tend to decline as the number of labelled instances (used as constraints) grows.

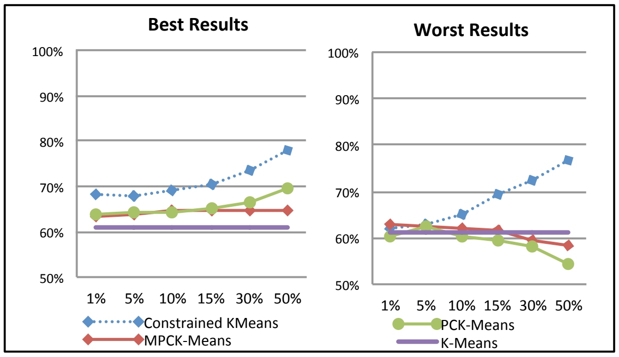

Fig. 8 Best and worst results without BadRank

The best results with *BadRank* (see Fig. 9) show a considerable increase for all semi-supervised algorithms, after 15% of labelled instances are made available. In the worst results, the trend of decrease on constrained algorithms, observed in Fig. 9, seems to be attenuated as more labelled instances are made available.

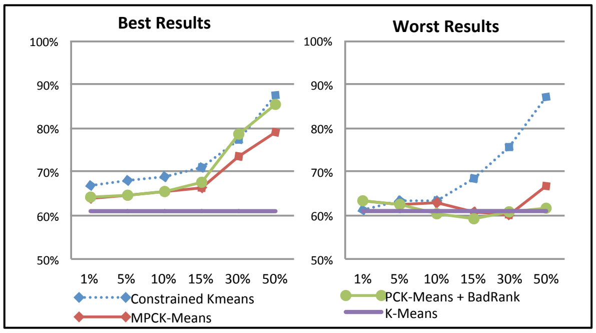

Fig. 9 Best and worst results with BadRank

In general, semi-supervised clustering gives significant improvement over unsupervised clustering. Definitely, in general situations, semi-supervised algorithms tend to increase their performance with the increasing number of constraints.

It is clear in our experiments, that seeding helped the algorithm on finding a good clustering, since the best results on accuracy were obtained with *Constrained-KMeans*, a semi-supervised clustering algorithm by seeding.

Similarly to *Constrained-KMeans*, the results for *MPCK-Means* and *PCK-Means* performance tend to increase as the number of labelled instances grows. However, for these two constrained algorithms, the results show that with the increase of constraints the accuracies obtained not always follows this raise. In fact, there were some experiments where the use of constraints produced worst results than without using any constraints (see Fig. 8 and Fig. 9).

Like has been stated in specific studies for semi-supervised clustering, the difference of performances verified for a fixed number of constraints is explained by two properties for the set of constraints: *informativeness* and *coherence*. While the first one refers to “the amount of information in the constraint set that the algorithm cannot determine on its own”, the second one is related to the amount of agreement between the constraints in the set, given a distance metric” [14]. Positively, the dataset used in this study is a hard problem to solve by clustering methods, due to the existence of overlapping clusters. But this is a feature for any fraud detection dataset.

From experimental results it is possible to notice that *BadRank* produce significant improvements on accuracy results after incorporating 15% of labelled instances. These results seem to show a decrease of the overlapping clusters, since the fraudulent instances get more differentiated from others.

On this dataset we can also notice that *MPCK-Means* does not provide significant accuracy improvement when comparing to *PCK-Means*. Actually, using *BadRank* on both algorithms, *PCK-Means* exceeds the accuracy of *MPCK-Means*, demonstrating that not always the metric learner take advantage on accuracy performance.

6. CONCLUSIONS

While semi-supervised clustering algorithms have been shown to be useful whenever there are a small amount of labelled samples, social networks have been accepted as an important tool to represent the existing information about entities’ social relations.

The fraud detection is a privileged case that combines both situations. On one hand, it is difficult to have a representative labelled dataset, since validated frauds are just a small percentage of cases, and the probability of having misclassified instances is high. On the other hand, fraud is perpetrated by humans, which live in a society and have multiple social relations. Indeed, organizations that collect taxes have data that can be used to determine the social network for each entity, and then use it to better classify it.

From our experiments, it is clear that semi-supervised clustering performs better when data is enriched with social network analysis. This improvement is higher when the percentage of labelled instances increases. Among the tested algorithms, *Constrained K-Means* presents the best average results, and also the worst.

These results evidence that social network analysis plays a significant role on incorporating domain knowledge into the mining process, and opens new opportunities to improve semi-supervised methods.

7. ACKNOWLEDGEMENTS

This work was partially supported by Fundação para a Ciência e Tecnologia under research project D2PM (PTDC/EIA-EIA/110074/2009).

8. REFERENCES

1. Bolton, R. J., Hand, D. J.: Statistical fraud detection: A review. Statistical Science 17(3), 235-255 (2002)

2. Scholz, M. .: Sampling-Based Sequential Subgroup Mining. In Grossman, R., Bayardo, R., Bennett, K., Vaidya, J., eds. : ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp.265 - 274 (2005)

3. Jain, A.: Data Clustering: A Review. ACM Computing Surveys 31(3) (1999)

4. MacQueen, J.: Some methods for classification and analysis of multivariate observations. In : Berkeley Symposium on Mathematical Statistics and Probability, p.281–297 (1967)

5. S., B., Banerjee, A., Mooney, R.: Semi-supervised clustering by seeding. In : International Conference on Machine Learning, Sydney, Australia, pp.27-34 (2002)

6. Basu, S., Bilenko, M., Mooney, R.: Integrating Constraints and Metric Learning in Semi-Supervised Clustering. In : International Conference on Machine Learning, pp.81-88 (2004)

7. Kleinberg, J. M.: Authoritative Sources in a Hyperlink Environment. In : ACM SIAM Symposium on Discrete Algorithms, New York, pp.668- 677 (1998)

8. Haveliwala, T.: Efficient computation of pagerank. Technical report (1999)

9. Wasserman, S. .: Social Network Analysis: Methods and Applications. Cambridge University Press (1994)

10. Gyongyi, Z., Garcia-Molina, H., Pedersen, J.: Combating web spam with trustrank. In : International Conference on Very Large Data Bases, Toronto, Canada , pp.576-587 (2004)

11. Adriaans, P. .: Data Mining. Addison-Wesley, Harlo England (1996)

12. Neuman, L.: Social Research Methods: Qualitative and Quantitative Approaches 5th edn. Allyn & Bacon (2002)

13. Rand, W. M.: Objective criteria for the evaluation of clustering methods. Journal of the American Statistical Association 66, 846–850 (1971)

14. Wagstaff, K. ., Basu, S., Davidson, I.: When is Constrained Clustering Beneficial, and Why? In : National Conference on Artificial Intelligence (2006)

15. McCarthy, J., Shelton, S.: Customer intelligence: It's all in the network. White paper (2008)