A density-based backward approach to isolate rare events in large-scale applications

Enikö Székely[1], Pascal Poncelet[1], Florent Masseglia[2], Maguelonne Teisseire[3], and Renaud Cezar[4]

[1] Computer Science Department, University of Montpellier, Montpellier, France
[2] INRIA, Montpellier, France
[3] Maison de la Télédetection, Montpellier, France Institute for Research in Biotherapy, Montpellier, France

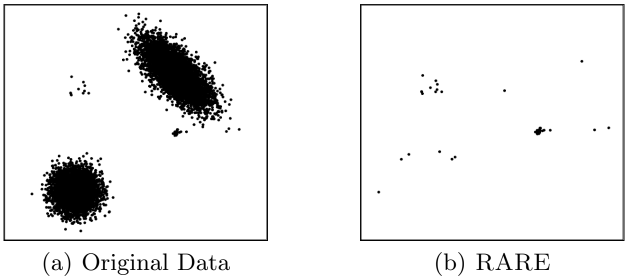
Abstract. While significant work in data mining has been dedicated to the detection of single outliers in the data, less research has approached the problem of isolating a group of outliers, i.e. rare events representing micro-clusters of less – or significantly less – than 1% of the whole dataset. This research issue is critical for example in medical applications. The problem is difficult to handle as it lies at the frontier between outlier detection and clustering and distinguishes by a clear challenge to avoid missing true positives. We address this challenge and propose a novel two-stage framework, based on a backward approach, to isolate abnormal groups of events in large datasets. The key of our backward approach is to first identify the core of the dense regions and then gradually augments them based on a density-driven condition. The framework outputs a small subset of the dataset containing both rare events and outliers. We tested our framework on a biomedical application to find micro-clusters of pathological cells. The comparison against two common clustering (DBSCAN) and outlier detection (LOF) algorithms show that our approach is a very efficient alternative to the detection of rare events while also providing a $\mathcal{O}(N)$ solution to the existing algorithms dominated by a $\mathcal{O}(N^2)$ complexity.

Keywords: rare events, outlier/anomaly detection, large scale, *k*-means

1 Introduction

”An outlier is an observation which deviates so much from the other observations as to arouse suspicions that it was generated by a different mechanism” (Hawkins 1980). Similarly, a *rare event* – cluster of outliers (Rocke and Woodruff 1996), clustered anomaly (Liu et al 2010, 2012), anomaly collection (Dai et al 2012), micro-cluster (Bae et al 2012) – is a group of observations which deviates so much from the other groups of observations as to arouse suspicions that it was generated by a different mechanism.

The detection of rare events with a high recall, i.e. no false negatives, is intrinsic to those domains where the cost of missing rare events is significantly


(a) Original Data

(b) RARE

Fig. 1. Detection of rare events with RARE on artificially generated data. The dataset contains two normal populations and two rare events: one sparse and global and one dense and local.

high. The most representative example is the medical domain where, for example, the cost of missing a pathological group of cells in a blood sample is significantly higher than the cost of classifying a healthy group of cells as pathological, i.e. favouring false positives over false negatives. Disease outbreaks in biosurveillance (Shmueli and Burkom 2010), bursts of clustered attacks (Liu et al 2010) or groups of spammers/fraudulent reviewers in social media (Dai et al 2012) are other examples of scenarios where the detection of rare events is prevailing over the cost of detecting them.

An anomaly – single or clustered – is an event considered as not normal with respect to a normal behaviour (Chandola et al 2009). With any type of anomaly, the open issue is to define normality. For *single outliers*, normality is defined in terms of distance, distribution or neighbourhood similarity with other data instances. For *spatial anomalies*, it is their occurence in a specific region of the space that makes them abnormal. For *collective anomalies*, individual instances are normal but it is their co-occurence that makes them anomalies. For *rare events*, it is their small relative size with respect to other data subpopulations that makes them anomalies. Contrary to collective anomalies, every instance contained in a rare event is an anomaly. We consider an example of rare events detection in Figure 1. The data distribution contains two normal populations of 10,000 points and two rare events: a sparser one of 10 points far from the normal populations, i.e. a *global* anomaly, and a denser one of 20 points close to one of the normal populations, i.e. a *local* anomaly. Figure 1(b) shows the output of our approach, RARE, isolating the rare events from the rest of the data.

Sharing common characteristics with both outliers and clusters, the detection of rare events lies at the frontier between *outlier detection* and *strongly imbalanced/unbalanced clustering*. Both clustering and outlier detection algorithms, by their construction, are generally prone at misclassifying positive examples, i.e. rare events, as negative. Algorithms for unbalanced data have been mainly proposed in supervised scenarios (Tang et al 2009) for classification problems in the presence of unbalanced training data where the problem is generally handled using resampling, cost-sensitive or one-class learning methods (Chawla et al

2004). In unsupervised scenarios the lack of ground truth information makes the problem even more difficult to handle. One of the main causes is the size balancing effect, as for example in k-means, which tends to reduce the variation in cluster sizes as a trade-off for a better accuracy (Xiong et al 2006). In spectral clustering, both RatioCut and Ncut (von Luxburg 2007) put more emphasis on balancing clusters than on minimizing cut values. Both algorithms propose through the balancing constraints introduced to handle the outlier sensitivity of the initial MinCut solution. On the other hand, outlier/anomaly detection algorithms (Aggarwal 2013) are very effective at discovering single anomalies. Different approaches (density-based, distance-based, distribution-based) have been proposed in the literature. The most common outlier detection algorithm, LOF (Breunig et al 2000), Local Outlier Factor, outputs a list of top-k outliers according to an outlierness score obtained by comparing the local density of each point against the local density of the points in its neighbourhood. The performance of LOF depends mainly on the construction of the local neighbourhood (parameter $MinPts$).

In this paper we address this gap between outlier detection and clustering methods. Given our main challenge to avoid false negatives, i.e. avoid missing true positives, we propose a density-based *backward* or bottom-up approach, i.e. going from the most dense regions to the least dense ones. Common outlier detection methods use a forward or top-down approach, i.e. they take the top-k outliers according to an outlierness threshold score. The paper is organized as follows. Section 2 is dedicated to a literature review for finding rare events in large datasets. Section 3 introduces our RARE framework. We first perform a clustering using DenseKMeans, a modified variant of k-means, designed to find and cluster only points that lie in dense regions of the space. In the second step, we gradually augment the dense regions found by DenseKMeans using a density-based sliding region. As soon as the density inside the sliding region fails to fullfill a density condition, we consider to have reached the border of the dense regions. Rare events lie outside these borders. In section 4 experiments on a biomedical data benchmark show that RARE is capable of isolating the rare events with a higher precision than both DBSCAN and LOF. We discuss the advantages and limitations of RARE in Section 5.

2 Related work

Different approaches (Chandola et al 2009; Ertoz et al 2003; Ester et al 1996; He et al 2003; Liu et al 2010, 2012; Papadimitriou et al 2003; Zhu et al 2010) in the literature have been proposed for the detection of rare events in large datasets. A few techniques approach it as cluster-based anomaly detection (Chandola et al 2009): normal instances belong to large and dense clusters, while anomalies either belong to small or sparse clusters. Such methods rely on the output of a clustering algorithm. CBLOF (He et al 2003) first performs a clustering, using any clustering method, and subsequently separates small from large clusters based on a predefined threshold. Using this threshold, it defines a Cluster-Based

Local Outlier Factor (CBLOF) outlierness score by taking into account both the size of the cluster and the distance to the closest cluster center. Overall, the performance of such techniques relies strongly on the choice and quality of the initial clustering.

Employing explicit cluster size constraints is another solution (Zhu et al 2010) that can be used to handle the detection of rare events in datasets. While the tendency in the literature is to concentrate on balancing clusters, this approach allows to generate a partitioning with different cluster sizes. It can be very helpful when an a priori knowledge on the size of each cluster in the data is known in advance. Still, only a few applications benefit from such a faithful information.

A third approach is to use or adapt single outlier detection algorithms and make them suitable for detecting micro-clusters of outliers. In LOF (Breunig et al 2000) the detection of outlying clusters depends on the choice of the number of nearest neighbours *MinPts* that define the local neighbourhood. The detection of very small clusters requires a *MinPts* large enough to contain all the points in a cluster, i.e. larger than the size of the cluster. LOCI (Papadimitriou et al 2003) defines a multi-granularity deviation factor (MDEF) and identifies outliers as those points whose neighbourhood size is significantly different than the neighbourhood size of their neighbours. Similarly to LOF, LOCI relies on an appropriate choice of the neighbourhood size, except that, contrary to LOF, it requires the maximum radius of the neighbourhood as input parameter.

Another different direction is to consider that normal instances belong to a cluster in the data, while outliers do not belong to any cluster (Chandola et al 2009). This approach requires the use of methods (DBSCAN (Ester et al 1996), SNN-based clustering (Ertoz et al 2003)) that do not force every point to belong to one of the clusters. DBSCAN (Ester et al 1996) is the most common density-based clustering algorithm. Its novel notion of *density reachability* allows the detection of clusters of arbitrary sizes ans shapes, but it cannot handle clusters of different densities. Both the run time complexity and memory requirements of the original alorithm are high $\mathcal{O}(N^2)$. Using efficient indexing structures like k-d trees to find the nearest neighbours, the run time complexity can be reduced to $\mathcal{O}(N \log N)$. However such indexing structures are not suitable for high-dimensional data.

A relatively recent concept – *isolation* – was proposed (Liu et al 2008, 2010, 2012) as an alternative to the concepts of distance and density used in most outlier detection methods. The notion of isolation relies on the property of anomalies of being 'few and different'. The two methods, iForest (Liu et al 2008, 2012) and SCiForest (Liu et al 2010), that rely on this concept build in the training phase forests of t binary trees using sub-samplings of the data and compute in an evaluation step an anomaly score based on the path length of each point, defined as the path from the root of the tree to the node. While both methods are effective at discovering global clustered anomalies, i.e. clusters far apart from normal populations, only SCiForest is able to detect local clustered anomalies (Liu et al 2012), i.e. clusters close to normal populations (we presented both types of clustered anomalies in our example in Figure 1). However the high complexity of SCiForest

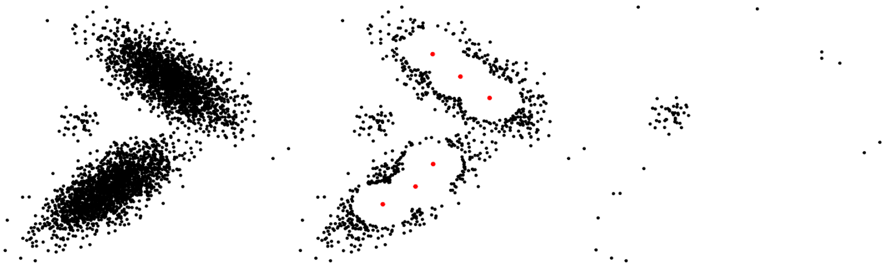


Fig. 2. Illustrative example: a) Original data: the rare event contains 1% of the entire data collection. b) The data subset after eliminating the core of the dense regions with DenseKMeans. c) Rare events after DenseSlide.

in both training and evaluation stages, respectively $\mathcal{O}(t\tau\psi(q\psi + log\psi + \psi))$ and $\mathcal{O}(qNt\psi)$, where ψ is the sampling size for building the iTrees and t the number of trees to build in the training phase, makes it suitable only in the presence of local clustered anomalies.

The RARE framework that we propose in this paper proposes: 1) a *backward* approach to the detection of rare events by first identifying the normal/dense regions; 2) an approach designed to avoid false negatives and therefore accepting false positives, favouring recall over precision; 3) a low complexity due to the use of a variant of k-means (linear, scalable); 4) a lower bound density-driven approach in both steps of the framework that allow the detection of rare events.

3 The RARE framework

We describe in this section our two-stage framework for the detection of rare events in large datasets. Given a dataset $\mathbf{X}$ with N data points, we consider a rare event as a *micro-cluster* of size N_R, where N_R is significantly smaller than the total size of the dataset ($N_R \ll N$).

When expressed in terms of the ratio $\varepsilon = \frac{N_R}{N}$ between the number of points in the rare event and the total number of points in the dataset, the above rare event condition becomes $\varepsilon \ll 1$. Very small values of ε, i.e. $\varepsilon < 10^{-2}$, place the problem of abnormal events detection at the frontier between *outlier detection* and *strongly imbalanced clustering*.

3.1 The backward approach: an illustrative example

We illustrate the backward approach of RARE by means of an example in Figure 2. We consider a dataset $\mathbf{X}$ with two normal subpopulations and a rare event representing 1% of the whole dataset.

First, we want to identify the core of the dense regions while handling two major issues at this stage: the *scalability* and the *density*. We have no a priori knowledge on the number of subpopulations in the data. To handle the *scalability* issue we choose to cluster the dataset using k-means (MacQueen 1967) due to both its linear complexity and parallelization power. The *density* problem is then handled by modifying k-means so that only points that lie in dense regions are clustered. We do this by changing in the re-assignement phase of k-means the way cluster centers are estimated, i.e. only points that lie within a maximum radius around cluster centers contribute to the reestimation of the centers. This radius-limited approach does not force all points to belong to one of the clusters, i.e. some points will be left unclustered. As the actual number of clusters in the dataset is unknown, we use a large initial number of clusters K_I and let each population be modelled using multiple clusters. Figure 2(b) illustrates this first step of the analysis after the convergence of the centers to the core of the dense regions. We use $K_I = 6$ cluster centers in this example and plot the output of DenseKMeans, i.e. the points left unclustered after the first step, $\mathbf{X}_{KEEP}$.

In the second stage (Figure 2(c)) the clusters that belong to the same population, i.e. they are adjacent as will be defined in Section 3.3, are merged to form connected components. In our example each group of 3 clusters forms a connected component. The two components are then gradually augmented, by means of a density-based Gaussian sliding region (DenseSlide), to reach the border of the dense regions. Everything that is outside these borders, $\mathbf{X}_{RARE}$, is considered a rare event. The framework retrieves both true positives, i.e. the rare event, and false positives, i.e. points that lie close to the border of the dense regions or outliers.

3.2 Dense regions clustering

The principle behind k-means relies on the minimization of a distance-based objective function that clusters the dataset $\mathbf{X}$ around K cluster centers. But this distance-based approach leaves k-means sensitive to density-related issues and to the presence of outliers and noise. To adress this density problem and cluster only points that lie in dense regions we propose a variant of k-means – DenseKMeans – by bringing two modifications to the original algorithm:

$$
\begin{aligned}
&\min \sum_{k=1}^{K} \sum_{\mathbf{x}_i \in \mathcal{C}_k} \parallel \mathbf{x}_i - \mu_k \parallel^2 \qquad (1)\\
&\text{s.t. } \mid \mathcal{C}_k \mid > N_I \\
&dist(\mathbf{x}_i, CC_k) < D_{MAX}, \forall \mathbf{x}_i \in \mathcal{C}_k
\end{aligned}
$$

1. ***initialization***: choose cluster centers iteratively so that each new center is positioned at a minimum of D_{MAX} distance from all the other centers and that each cluster center is assigned at least N_I data points.
2. ***re-assignement***: reestimate cluster centers using only points that are at a maximum of D_{MAX} distance from one of the cluster centers and remove cluster centers that fall below the initial N_I threshold during the re-assignement phase.

```
Algorithm 1: DenseKMeans
Input: $\mathbf{X} = \{\mathbf{x}_i\}, i = 1..N$, $\mathbf{x}_i \in \mathbb{R}^D$
        $K_I$ - initial number of clusters
        $N_I$ - minimum number of points (density)
        $D_{MAX}$ - radius
Output: $\mathbf{CC} = \{CC_k\}$, $k = 1..K_F$ - final cluster centers
        $\mathbf{X}_{KEEP}$ - the subset of points left unclustered
        $\mathbf{X}_{RMV}$ - the subset of points clustered
Initialization:
1': Choose cluster centers CC iteratively so that they are further than $D_{MAX}$ one from each other:
        $\|CC_k, CC_l\|_2 > D_{MAX}, \forall k, l = 1..K_I$
2': Check the density condition: $card\{\mathcal{C}_k\} > N_I$
3': Repeat steps 1' and 2' until convergence: all $K_I$ centers are assigned at least $N_I$ points.
DenseKMeans:
1": Select all points $\mathbf{X}_{KEEP}$ that are further than $D_{MAX}$ from all centers:
        $min(\mathbf{x}_i, CC_k) > D_{MAX}$
2": Reestimate cluster centers using $\mathbf{X}_{RMV} = \mathbf{X} \setminus \mathbf{X}_{KEEP}$
3": If a cluster center falls under the initial density threshold ($card\{\mathcal{C}_k\} < N_I$) remove it.
4": Repeat steps 1"-3" until convergence: a maximum number of iterations is reached or centers do not change significantly.
```

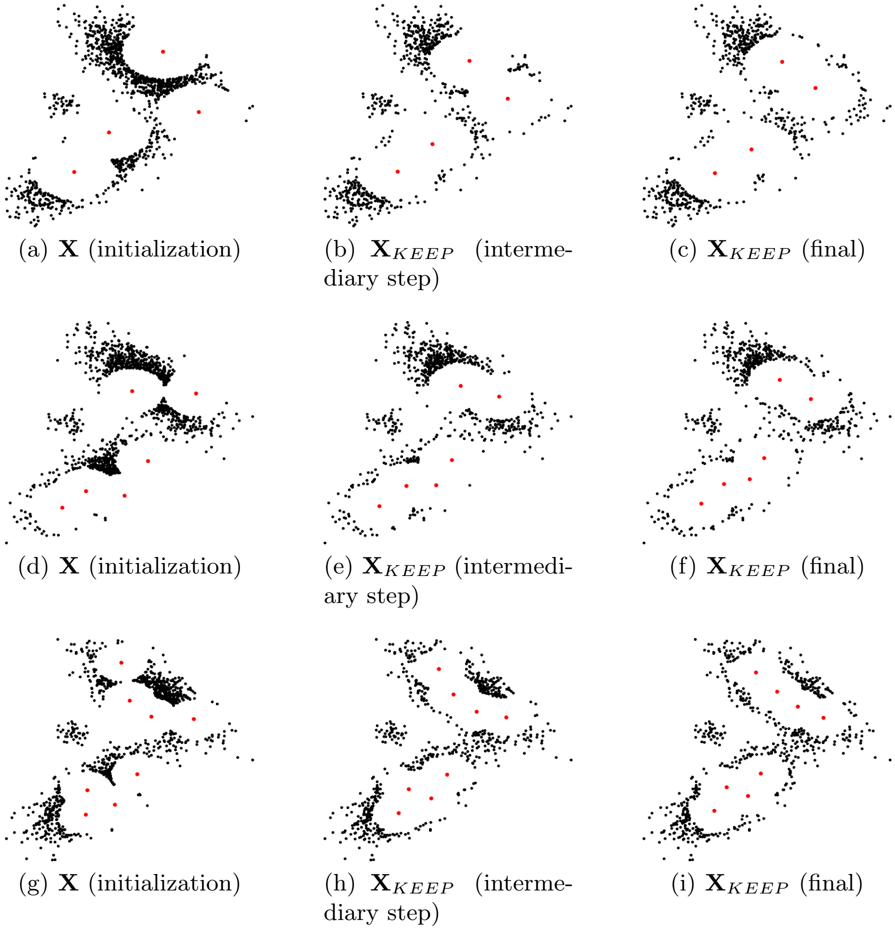
Table 1. DenseKMeans.

DenseKMeans is summarized in Table 1. The reestimation of cluster centers using only points that are at a maximum of D_{MAX} distance from one of the cluster centers eliminates k-means' sensitivity to outliers – in our case to rare events – as long as the radius D_{MAX} is smaller than the distance to outliers. Moreover clusters $\mathcal{C}_k$ that are not dense enough, $card\{\mathcal{C}_k\} < N_I$, are discarded in the re-assignement phase.

These two modifications allow to restrict the region of the space considered by k-means to only dense regions and iteratively move cluster centers towards the core of the dense regions. Figure 3 illustrates a few examples with different parameter combinations D_{MAX} vs. K: 1) $D_{MAX} = 1.4$, $K_I = 4$ (Figure 3(a, b, c)); 2) $D_{MAX} = 1.2$, $K_I = 6$ (Figure 3(d, e, f);) 3) $D_{MAX} = 1$, $K_I = 8$ (Figure 3(g, h, i)). The output of this first stage of the algorithm divides the original dataset into two disjoint subsets $\mathbf{X} = \mathbf{X}_{RMV} \cup \mathbf{X}_{KEEP}$: 1) $\mathbf{X}_{RMV}$ = points falling within a maximum of D_{MAX} distance from the final cluster centers, 2) $\mathbf{X}_{KEEP}$ = points falling outside the region defined by the maximum D_{MAX} distance from the final cluster centers. Using this approach, only points that are in dense regions are clustered.

3.3 Dense regions augmentation

DenseKMeans identifies the core of the dense regions using an initial number of clusters K_I larger than the actual number of clusters/data subpopulations. The radius-limited approach of DenseKMeans allows to define the *cluster adjacency* property as in the following:



(a) $\mathbf{X}$ (initialization)

(b) $\mathbf{X}_{KEEP}$ (intermediary step)

(c) $\mathbf{X}_{KEEP}$ (final)

(d) $\mathbf{X}$ (initialization)

(e) $\mathbf{X}_{KEEP}$ (intermediary step)

(f) $\mathbf{X}_{KEEP}$ (final)

(g) $\mathbf{X}$ (initialization)

(h) $\mathbf{X}_{KEEP}$ (intermediary step)

(i) $\mathbf{X}_{KEEP}$ (final)

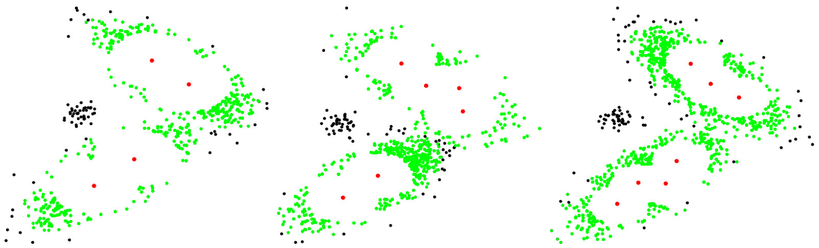
Fig. 3. Varying D_{MAX} and K_I in DenseKMeans considering the original data from Figure 2: (a,b,c) $D_{MAX} = 1.4$, $K_I = 4$; (d,e,f) $D_{MAX} = 1.2$, $K_I = 6$; (g,h,i) $D_{MAX} = 1$, $K_I = 8$. Red points represent cluster centers. The initial, intermediary and final step for each case illustrate the convergence of cluster centers towards the core of the dense regions, eliminating the sensitivity of the original k-means to outliers.

Definition 1. *Two clusters defined by centers CC_k and CC_l and maximum radius D_{MAX} are adjacent if they are overlapping, i.e. the Euclidean distance between the centers CC_k and CC_l is less than $2 \times D_{MAX}$:*

$$\|CC_k, CC_l\|_2 < 2 \times D_{MAX}$$

Among the final K_F dense clusters found by DenseKMeans, adjacent clusters are merged to build connected components and provide a more faithful representation of the real data subpopulations.

A spherical model like the one used by k-means and DENSEKMEANS considers that the intrinsic dimensionality of the data is equal to the original dimensionality. However in real scenarios the intrinsic dimensionality of the data - especially locally, i.e. one data subpopulation/cluster - is rarely equal to the original dimensionality (Levina and Bickel 2005). To address this challenge, we treat the output of the spherical model by means of a model that is better adapted to handle the intrinsic dimensionality of the data. The most common is the Gaussian model. In the first step of the analysis, the spherical approach was preferred due to the scalability advantage of k-means. The use of the Gaussian mixture model in the first step would have required the estimation of $K(D^2 + D + 1)$ parameters for every value of K – as K is not known in advance. Even if parsimonius models, e.g. diagonal, can replace the full Gaussian model, the challenge to detect rare events is too sensitive and requires the use of a full model.



(a) $D_{MAX} = 1.4$, $K_I = 4$ (b) $D_{MAX} = 1.2$, $K_I = 6$ (c) $D_{MAX} = 1$, $K_I = 8$

Fig. 4. Points in green are eliminated through DENSESLIDE. The same combinations of D_{MAX} and K_I as in Figure 3 are used. c) Only 7 out of 8 clusters are left, one was eliminated because it did not fullfill the density condition (N_I) in DENSEKMEANS.

The subset $\mathbf{X}_{RMV}$ allows to quickly estimate both the means μ_j and covariance matrices Σ_j of the core dense regions defined by the connected components. These dense regions are augmented using a sliding region S_R defined based on the Mahalanobis distance D_M and an increase parameter ϵ_S. The sliding regions approach the border of the dense regions gradually and the process is repeated as long as a density condition is fullfilled, $nbPoints(S_R) > N_S$, i.e. the number of points inside the sliding region is larger than a predefined threshold N_S. When the density inside the sliding region drops below this threshold, we consider to have reached the border of the dense regions. The algorithm for dense regions augmentation, DENSESLIDE, is summarized in Table 2 and a few examples for various combinations of parameters D_{MAX} and K_I are shown in Figure 4. The parameters for DENSESLIDE were $\epsilon_S = 0.1$ and $N_S = 10$. The output of the algorithm returns the subset $\mathbf{X}_{RARE}$ of positive examples.

```
Algorithm 2: DenseSlide
Input: $\mathbf{X}_{KEEP}$, $\mathbf{X}_{RMV}$, $\mathbf{CC}$ - output of DenseKMeans
       $\epsilon_S$ - increase parameter for the sliding region
       $N_S$ - number of points in the sliding region
Output: $\mathbf{X}_{RARE}$ - output of RARE
Connected components:
1': Build the graph $G = (\mathbf{CC}, E)$ using the cluster adjacency property.
2': Find connected components $G_j$ in $G$.
3': Use $\mathbf{X}_{RMV}$ to model $G_j$ as $\mathcal{N}(\mu_j, \Sigma_j)$.
Sliding Region:
1": Initialize $\mathbf{X}_{RARE}$=$\mathbf{X}_{KEEP}$.
2": For each $G_j$ compute the Mahalanobis distance:
        $D_M^j = \sqrt{(\mathbf{X}_{RARE} - \mu_j)^T \Sigma_j^{-1} (\mathbf{X}_{RARE} - \mu_j)}$
3": Eliminate points from $\mathbf{X}_{RARE}$ that are closer to one of the component centers than the farthest point from $\mathbf{X}_{RMV}$: $D_M^j(x_i) > D_{max}^j$.
4": Create a moving sliding region $S_R(D_{max}^j, \epsilon_S)$ around each component $\mathcal{N}(\mu_j, \Sigma_j)$.
5": Eliminate points from $\mathbf{X}_{RARE}$ inside $S_R$.
6": Repeat steps 4" and 5" as long as the density condition is respected: $nbPoints(S_R) > N_S$.
```

Table 2. DenseSlide.

4 Experimental results

In this section we test RARE on a large-scale biomedical application in a diagnosis purpose, to isolate pathological group of cells in flow cytometry. We perform experiments on multiple data sets with varying sizes of the rare event. A practical analysis of the influence of parameter values is also performed on the benchmark data. Finally we compare RARE against both clustering – DBSCAN – and outlier detection – LOF – algorithms[4]. We experiment with various parameter values to illustrate the behaviour of each of the above methods.

We use Precision and Recall to evaluate the performance of the algorithms. Given our main challenge to avoid missing true positives, it is Recall that becomes the most important evaluation measure in this scenario.

$$\mathrm{P} = \frac{TP}{TP + FP} = \frac{TP}{|\mathbf{X}_{RARE}|}, \qquad \mathrm{R} = \frac{TP}{TP + FN} = \frac{TP}{N_{RS}} \tag{2}$$

where $|\mathbf{X}_{RARE}|$ = the number of data points retrieved by RARE and N_{RS} = the number of positives in the data, i.e. the size of the rare event.

4.1 A real case: flow cytometry

In flow cytometry each cell is characterised by fluorescence levels in response to cell markers, i.e. attributes. Nowadays flow cytometers can count up to tens of millions of cells representing normal cell populations found in any healthy patient, such as lymphocytes or monocytes. In patients presenting a blood pathology, the blood samples also contain micro-clusters of cells with abnormal signatures, i.e. abnormal combinations of cell marker fluorescence levels. The human detection of these rare events is performed visually by sequentially inspecting two-dimensional spaces, i.e. combinations of two markers.

[4] We used the ELKI implementation available at: http://elki.dbs.ifi.lmu.de/.

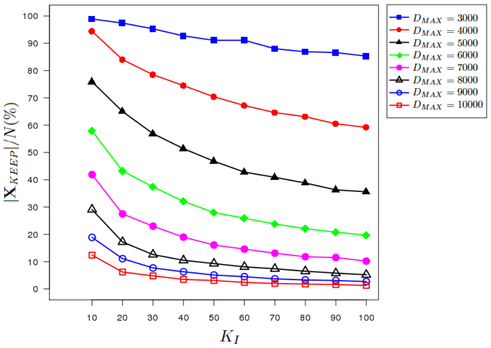


Fig. 5. Initialization of D_{MAX} and K_I for the flow cytometry dataset.

Experiment 1. D_{MAX} and K_I. We first estimate the percentage of data covered by DenseKMeans in the first step of the algorithm for various values of the parameters D_{MAX} and K_I (Figure 5). We fixed $N_I = \frac{N}{100 \times K_I}$ because we know that the rare event is signficantly smaller that the total size of the dataset. Those combinations of values for D_{MAX} and K_I – closely related – covering approximately $80 - 90\%$ of the dataset in DenseKMeans ($\mathbf{X}_{RMV}$) generally led to very good final results in the experiments. This is due to the fact that the rare events represent significantly less than the rest of $10 - 20\%$ of the whole dataset, allowing in the meantime the detection of the core dense regions by DenseKMeans.

Throughout our evaluation, we experimented with different values of the parameters and observed that the choice of the parameter values was consistent across different datasets for a given application.

Experiment 2: Varying N_R. We now wish to test the performance of RARE for varying levels of unbalancedness. In this purpose we will keep the total size of the dataset fixed and vary the size of the rare event - which is an indicator of the phase of the pathology. On the biological side, this experiment was performed by injecting grown cells from a blood pathology into a cell sampling of a healthy patient. The size of the rare population injected was of $\{5, 10, 20, 50, 100, 500\}$. Due to machine error, a difference appears between the number of injected cells and the actual size N_{RS} of the rare cell population found in the blood samples, i.e. positive examples (corresponding to a pathology signature in flow cytometry). The whole dataset contained $N = N_H + N_{RS}$ cells, where $N_H \approx 700.000$ cells. In this experiment the free parameters D_{MAX} and K_I in DenseKMeans were chosen to guarantee the ratio $\frac{|\mathbf{X}_{KEEP}|}{N} \approx 10 - 20\%$ across the different blood samples (as discussed in Experiment 1). Here we choose $D_{MAX} = 8000$ and $K_I = 40$, but other value combinations that respect the above ratio are also valid (as will be seen in Experiment 3). The parameters for DenseSlide were chosen: $\epsilon_S = 0.1$ and $N_S = 10$.

N_R	N	$\frac{\lvert\mathbf{X}_{KEEP}\rvert}{N}$(%)	$\lvert\mathbf{X}_{RARE}\rvert$	TP	FP	P	R	N_{RS}
0	151,388	7.7%	64	**5**	59	7.8%	100%	**5**
5	646,149	8.1%	42	**4**	38	9.5%	100%	**4**
10	780,988	7.6%	54	**13**	39	24%	92.8%	**14**
20	757,234	7.5%	70	**17**	53	24.2%	100%	**17**
50	752,987	7.4%	65	**30**	35	46.1%	96.7%	**31**
100	760,842	7.2%	132	**80**	52	60.6%	97.5%	**82**
500	718,743	7.7%	415	**358**	57	86.2%	99.7%	**359**
0	696,465	10.9%	102	**14**	88	13.7%	100%	**14**
5	731,576	11.0%	98	**9**	89	9.1%	75%	**12**
10	720,945	9.9%	114	**14**	100	12.2%	100%	**14**
20	484,285	10.5%	129	**25**	104	19.3%	96.1%	**26**
50	630,341	10.4%	40	**35**	5	87.5%	97.2%	**36**
100	676,745	10.2%	142	**69**	77	48.5%	98.5%	**70**
500	516,981	11.2%	541	**366**	175	67.6%	98.6%	**371**
0	671,582	10.1%	94	**8**	86	8.5%	100%	**8**
5	707,535	10.8%	100	**7**	93	7%	100%	**7**
10	714,081	10.2%	135	**13**	122	9.6%	100%	**13**
20	621,155	11.8%	155	**11**	144	7%	100%	**11**
50	599,851	10.2%	144	**26**	118	18%	100%	**26**
100	711,801	10.5%	204	**84**	120	41.1%	100%	**84**
500	993,671	10.7%	552	**312**	240	56.5%	100%	**312**

Table 3. RARE on three samples for each of the varying $N_R = \{5, 10, 20, 50, 100, 500\}$

.

The results in Table 3 show an excellent performance for RARE which finds almost all positive examples, i.e. true positives TP (column 3), among the positive examples N_{RS} found with the signature provided by domain experts (column 5). The size of the false positives FP returned by RARE (column 4) depends mainly on the size and structure of the original dataset, i.e. FP remains relatively constant with increasing TP. We also observe that the recall is relatively high and the precision increases with the size of the rare event.

Experiment 3. Comparison with DBSCAN and LOF. A comparison of the parameters required by the three methods is presented in Table 4. While LOF requires only one parameter – $MinPts$ – in the construction phase, DBSCAN and RARE both require two parameters, thus adding more flexibility but also more complexity to the model. Both RARE and LOF require a stopping criteria while DBSCAN considers all points left unclustered as noise. Rare events will often fall in the noise category with DBSCAN (as shown in the next experiment). RARE uses two parameters – ϵ_S and N_S, the growing rate of the sliding region and the minimal density (ϵ_S is generally fixed to either 10^{-1} or 10^{-2}) – to define the stopping criteria. Their influence is equivalent to the cutting threshold in LOF, but it is the approach that is different: LOF has a top-down approach while RARE has a bottom-up approach. The bottom-up approach is preferred in scenarios where avoiding false negatives is the priority.

Method	Model parameters	Stopping criteria	Approach
RARE	(D_{MAX}, K_I)	(ϵ_S, N_S)	Bottom-up (backward)
DBSCAN	$(\epsilon, MinPts)$	–	Bottom-up
LOF	$MinPts$	*Threshold* or top-k	Top-down (forward)

Table 4. Parameters in RARE, DBSCAN and LOF.

In Table 5 we analysed a data sample chosen at random from the second experiment with a medium rare event (752987 samples and 31 positive examples) using various parameter values for the three methods. We compute the number of true positives (TP) and false positives (FP) retrieved by the algorithms. Both RARE and DBSCAN have a high recall (generally 100%) while RARE has a significantly higher precision than DBSCAN. In DBSCAN for most parameter values the rare event is left unclustered and belongs to the subset classified as noise[5] – except in the two cases where a fraction of the rare event clusters separately in a small cluster (14 and 25 points). While DBSCAN requires the $MinPts$ parameter to be lower than the size of the rare event for a relatively good performance, LOF on the contrary requires the $MinPts$ parameter higher than the size of the rare event, i.e. this is necessary for the detection of micro-clusters in LOF. While DBSCAN requires no stopping criteria, in LOF we need to choose either the cutting threshold value or the number of outliers. We use here two cutting threshold values for each value of $MinPts$ in LOF and indicate the number of false positives in each case. The two values were chosen so that the vast majority of the rare event has an LOF outlierness score in the range bounded by the two values.

5 Discussion and conclusion

We proposed in this paper a two-stage framework to isolate rare events in large datasets. The size of these events makes their detection difficult by both clustering and outlier detection algorithms as both tend to missclasify true positives as false negatives. We have shown that RARE has a good performance and also the advantage of the linear complexity, largely dominated by the complexity of k-means and low memory requirements $\mathcal{O}(NK_I)$. The new variant of k-means was proposed to handle the scalability and density issues in this type of problems and the sliding region was designed in a backward/bottom-up approach to avoid false negatives. Overall, the RARE framework targets applications where recall prevails over precision. We did not approach here complexity improvements. Both DBSCAN and LOF have a $\mathcal{O}(N^2)$ memory requirement and runtime complexity – that can be improved to $\mathcal{O}(N \log N)$ using indexing structures such as k-d trees for low-dimensional data. In its current stage RARE has a $\mathcal{O}(N)$ complexity and DenseKMeans is easily parallelizable – it is the most time consuming

[5] Here $TP + FP$ equals the size of the noise subset in DBSCAN.

Method	Parameters	TP	FP
RARE($D_{MAX}, K_I, \epsilon_S, N_S$)	(6000, 80, 0.1, 10)	31	193
	(6000, 100, 0.1, 10)	31	48
	(7000, 40, 0.1, 10)	31	43
	(7000, 60, 0.1, 10)	31	60
	(7000, 80, 0.1, 10)	31	57
	(7000, 100, 0.1, 10)	30	40
	(8000, 20, 0.1, 10)	31	184
	(8000, 40, 0.1, 10)	31	60
	(8000, 60, 0.1, 10)	31	22
	(9000, 10, 0.1, 10)	31	284
	(9000, 30, 0.1, 10)	31	48
	(9000, 50, 0.1, 10)	31	35
	(10000, 10, 0.1, 10)	31	51
	(10000, 30, 0.1, 10)	31	35
DBSCAN($\epsilon, MinPts$)	(5000, 10)	31	1286
	(5000, 20)	31	1998
	(5000, 30)	31	2703
	(6000, 10)	31	457(14)
	(6000, 20)	31	699
	(6000, 30)	31	934
	(7000, 10)	31	197(25)
	(7000, 20)	31	331
	(7000, 30)	31	396
LOF($MinPts, Threshold$)	(30, 1)	31	589039
	(30, 1.1)	3	132890
	(50, 1.5)	31	2133
	(50, 1.6)	8	945
	(100, 2)	31	230
	(100, 2.5)	3	54
	(150, 2.1)	31	206
	(150, 2.7)	3	43

Table 5. Comparison between RARE, DBSCAN and LOF. The parameter values in the second column correspond to the respective parameters of each method from the first column.

in RARE. We consider these complexity improvements as a next step for future work.

Acknowledgements

This work has been mainly funded by the Labex NUMEV. The authors would like to thank Jean-Pierre Vendrell from the Institute for Research in Biotherapy in Montpellier, France for enabling the access to the data for the experimental part and Dino Ienco, Andre Mas, Pierre Pudlo and Sandra Bringay for valuable discussions.

Bibliography

Aggarwal C (2013) Outlier analysis. Springer

Bae DH, Jeong S, Kim SW, Lee M (2012) Outlier detection using centrality and center-proximity. In: Proceedings of CIKM

Breunig M, Kriegel HP, Ng RT, Sander J (2000) Lof: identifying density-based local outliers. In: Proceedings of ACM SIGMOD

Chandola V, Banerjee A, Kumar V (2009) Anomaly detection: a survey. ACM Computing Surveys 41

Chawla NV, Japkowich N, Kolcz A (2004) Editorial: special issue on learning from imbalanced data. SIGKDD Explorations 6

Dai H, Zhu F, Lim EP, Pang H (2012) Mining coherent anomaly collections on web data. In: Proceedings of CIKM

Ertoz L, Steinbach M, Kumar V (2003) Finding clusters of different sizes, shapes and densities in noisy, high-dimensional data. In: Proceedings of SDM

Ester M, Kriegel HP, Sander J, Xu X (1996) A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceedings of ACM SIGKDD

Hawkins D (1980) Identification of outliers. Chapman and Hall

He Z, Xu X, Deng S (2003) Discovering cluster-based local outliers. Pattern Recognition Letters 24

Levina E, Bickel PJ (2005) Maximum likelihood estimation of intrinsic dimension. Advances in Neural Information Processing Systems 17

Liu FT, Ting KM, Zhou ZH (2008) Isolation forest. In: Proceedings of ICDM

Liu FT, Ting KM, Zhou ZH (2010) On detecting clustered anomalies using sciforest. In: Proceedings of ECML/PKDD

Liu FT, Ting KM, Zhou ZH (2012) Isolation-based anomaly detection. ACM Transactions on Knowledge Discovery from Data 6

von Luxburg U (2007) A tutorial on spectral clustering. Statistics and Computing 17

MacQueen JB (1967) Some methods for classification and analysis of multivariate observations. In: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability

Papadimitriou S, Kitagawa H, Gribbons PB, Faloutsos C (2003) Loci: Fast outlier detection using the local correlation integral. In: Proceedings of ICDE

Rocke DM, Woodruff DL (1996) Identification of outliers in multivariate data. Journal of the American Statistical Association

Shmueli G, Burkom H (2010) Statistical challenges facing early outbreak detection in biosurveillance. Technometrics, Special Issue on Anomaly Detection

Tang Y, Zhang YQ, Chawla NV, Krasser S (2009) Svms for highly imbalanced classification. IEEE Transactions on Systems, Man and Cybernetics 39

Xiong H, Wu J, Chen J (2006) K-means clustering versus validation measures: a data distribution perspective. In: Proceedings of SIGKDD

Zhu S, Wang D, Li T (2010) Data clustering with size constraints. Knowledge-Based Systems, Elsevier 23