LERI: Local Exploration for Rare-Category Identification

Hao Huang, Qian Yan, Wei Lu, Huaizhong Lin, Yunjun Gao, *Member, IEEE*, and Lei Chen, *Member, IEEE*

Abstract—To identify the data examples of rare categories that form small compact clusters in large data sets, existing approaches mostly require enough labeled data examples as a training set to learn a classifier, assuming that the rare-category clusters are spherical or nearly spherical. Nonetheless, a large enough training set is usually difficult to obtain in practice, and rare categories in many real-world applications often form small compact clusters with arbitrary shapes. In this paper, we investigate how to identify all data examples of a rare category with an arbitrary shape based on only one seed (i.e., a labeled rare-category data example). Instead of finding a compact and spherical local region around the seed, we locally explore the data set from the seed by continuously searching and visiting the k-nearest neighbors of each newly visited data example. The local exploration connects the data examples in the objective rare category by the relationship of k-nearest neighbors, and meanwhile, suspected external data examples are filtered out if they are not close enough to any visited data example. Experimental results on both synthetic and real-world data sets are conducted, and the results verify the effectiveness and efficiency of our approach.

Index Terms—Rare category, classification, local exploration, arbitrary shape, k-nearest neighbors

1 Introduction

Compared with majority categories that occupy the vast majority of a data set, each rare category has only a few data examples, forming a small compact cluster [8], [9], [10], [11], [13], [14], [25], [30], [34]. An accurate identification of these rare categories is often of a high practical significance.

Example 1. In financial trading systems, most of financial transactions are legitimate, and fraudulent transactions can be deemed as rare categories. The fraudulent transactions from a same rare category share similar operating characteristics. For instance, the fraudsters may exploit the same loophole (e.g., the time zone difference) and adopt similar strategies to implement the fraudulent transactions. Discerning these fraudulent transactions from the massive legitimate ones helps uncover the vulnerabilities of the financial trading systems.

Example 2. In the electric bicycle sharing business, to decide where to deploy charging piles is essential for the service provider. It is easy to identify major clusters at hotspots where a lot of electric bicycles are parked. Nonetheless, at certain places, there may be a rare cluster with a much smaller (but non-negligible) number of bicycles, whose charging demand also needs to be addressed.

Apart from the above two examples, rare categories are also common in medicine, astronomy and geography, where their discovery may lead to new scientific advancements [34]. Thus, over the last decade, the problem of rare-category identification has received considerable attention from the research communities of database and data mining.

Existing approaches to rare-category identification mostly require enough labeled data examples, especially those from rare categories, to learn an accurate classifier [10], [11]. Nonetheless, this requirement on the quantity of labeled data examples may limit the applicability of the existing approaches. This is because, in practice, it is often the case that a data example from a previously unknown rare category is accidentally discovered by users, or may be detected by approaches like rare-category detection [9], [14] which aims to find out at least one data example for each rare category in an unlabeled data set with the help of a labeling oracle (e.g., human experts with domain specific knowledge). In other words, discovering and labeling adequate data examples from rare categories are often difficult and expensive [25].

Moreover, the existing approaches to rare-category identification either explicitly or implicitly assume that each rare category is confined within a small spherical local region, such as a hyper-ball [10]. Nevertheless, this assumption on the shapes of rare categories may be untenable in many real-world applications like image segmentation [31], spatial data mining [16], and biomedicine [39], where the data examples (e.g., pixels, locations, and protein molecules) of rare categories usually form clusters with arbitrary shapes.

- *H. Huang and Q. Yan are with the School of Computer Science, Wuhan University, Wuhan, Hubei 430072, China. E-mail: {haohuang, qy}@whu.edu.cn.*
- *W. Lu is with the School of Information and DEKE, MOE, Renmin University of China, Beijing 100872, China. E-mail: lu-wei@ruc.edu.cn.*
- *H. Lin and Y. Gao are with the College of Computer Science and Technology, Zhejiang University, Hangzhou, Zhejiang 310027, China. E-mail: {linhz, gaoyj}@zju.edu.cn.*
- *L. Chen is with the Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Kowloon, Hong Kong, China. E-mail: leichen@cse.ust.hk.*

Manuscript received 24 Apr. 2018; revised 29 Mar. 2019; accepted 9 Apr. 2019. Date of publication 17 Apr. 2019; date of current version 5 Aug. 2020. (Corresponding authors: Huaizhong Lin and Yunjun Gao.) Recommended for acceptance by J. M. Phillips. Digital Object Identifier no. 10.1109/TKDE.2019.2911941

In this paper, we relax the above requirement on the quantity of labeled data examples and the assumption on the shapes of rare categories, and investigate how to identify all data examples of an arbitrary shaped rare category based on only one seed, i.e., a labeled data example from the rare category. Towards this, we propose LERI (short for **L**ocal **E**xploration for **R**are-category **I**dentification) algorithm, which starts from visiting the seed and carries out an iterative local exploration. In each iteration of local exploration, LERI searches the k-nearest neighbors (abbreviated as kNN) of each newly visited data example as the candidate rare-category data examples that will be visited in the next iteration of local exploration. As the local exploration moves outward, data examples distributed in different parts of the arbitrary shaped rare category are connected by the kNN relationships and visited by LERI. When the kNN search fails to involve any unvisited data example, the local exploration terminates and returns all visited data examples as the result of rare-category identification.

To achieve high recall and precision for rare-category identification, there are two two vital issues should be focused on in the local exploration. (1) First, sometime the kNN search may fail to connect the visited part and adjacent unvisited part of the objective rare category, and thus only find partial data examples of the objective rare category. To address this kinds of issues, LERI enhances the connection effect of kNN search by shifting the position of each newly visited data examples towards their homogeneous neighbors during the local exploration to create more new kNN relationships across the visited and adjacent unvisited part of the objective rare category, in case these two parts can not be connected by original kNN relationships. (2) Second, the kNN search may involve external data examples that are from the outside of the objective rare category, and thus degrades the accuracy performance. To prevent this kinds of situations, among the kNN of each newly visited data example, LERI prudently reserves the data examples which are most close to the newly visited data example, and considers the other ones as the suspected external data examples. Each suspected external data example will be prevented from being visited for the next round of local exploration, but will not be prevented forever. As the local exploration continues, if a former suspected external data example is close enough to any subsequent newly visited data example, it still has the opportunity to be visited.

Our key contributions is two-fold. (1) Based on the kNN-based local exploration framework, we present a position shift method for data examples visited in the local exploration, which enhances the kNN relationship between different parts of the objective rare category and strengthens the search capability of the local exploration framework. (2) We combine prudent selection and multiple inspection to filter the candidate rare-category data examples, which help reduce the chances of false positives (i.e., having wrong candidates) and false negatives (i.e., missing right candidates) for rare-category identification. Besides, we also discuss how to select appropriate parameters (e.g., k for kNN search) for our proposed algorithm, and verify the effectiveness and efficiency of our approach on both synthetic and real-world data sets.

The remaining sections are organized as follows. We review the related work in Section 2, and present our LERI algorithm by elaborating how to connect rare-category data examples and filter out external data examples in Section 3. Experimental results and our findings are reported in Section 4 before concluding the paper in Section 5.

2 Related Work

In this section, we first review the existing paradigms for rare-category identification, followed by reviewing a related work known as rare-category detection.

2.1 Rare-Category Identification

According to the type of input data, the existing approaches to rare-category identification can be classified into three groups, namely (1) the supervised learning, (2) the semi-supervised learning, and (3) the seed-based approaches.

Supervised learning tries to construct a classifier based on the labeled data examples from both the rare and majority categories. The main challenge comes from the imbalanced category membership. To solve this problem, five types of methods have been proposed, i.e., (1) finding a small hyper-ball or hyper-plane that encloses all data examples of a rare category [10], [32], [37], (2) sampling or generating appropriate data examples for classifier construction [1], [7], [11], [21], (3) weighting the data to compensate the proportion bias between data examples from rare categories and majority categories [22], [26], [27], (4) boosting the ensemble performance of multiple classifiers [6], [18], [20], [28], and (5) dynamically modifying classifiers' decision thresholds [15]. Nonetheless, compared with our proposed LERI algorithm, these supervised learning approaches may be more computationally expensive since their analyses are mostly carried out on the entire given data set. In contrast, LERI tends to be relatively more efficient since it executes a local exploration on a small part of the data set, rather than a holistic analysis.

Semi-supervised learning tries to identify rare categories based on both labeled rare-category data examples and unlabeled data examples. The basic idea is to make good use of the unlabeled data examples that are close to the labeled rare-category data examples, under the assumption that the data examples of a rare category are close to each other and form a small compact cluster. So far, various approaches have been proposed for semi-supervised learning, such as co-training with both labeled and most confidently predicted unlabeled data examples [36], spreading label information to unlabeled data examples through manifold [2], [12], [41], and learning from positive and unlabeled data examples [4], [23]. Nevertheless, to obtain accurate identification results, both of the supervised and semi-supervised learning require enough labeled rare-category data examples, which are often difficult and expensive to obtain in practice.

To relax the requirement on the quantity of labeled rare-category data examples, several *seed-based approaches* have been proposed to help identify a rare category based on only one seed. FRANK (short for **F**ast **R**are c**A**tegory explorati**N** using a **K**-nearest neighbors graph) algorithm [13] assumes that each rare category forms a small compact cluster, and transforms the problem of rare-category identification from a seed to that of local community detection from a node [38] by constructing a kNN graph on the given data set. RCEWA (short for **R**are **C**ategory **E**xploration via **W**avelet **A**nalysis) algorithm [25] has the same shape assumption on rare categories, and adopts a two-phase strategy, in which the first phase aims for high recall by collecting all candidate data examples of the objective rare category into a small bin, and the second phase refines the collection to remove false positives. Similar two-phase strategies are also used in a few classical approaches to imbalance classification [17], [19], although they usually require more labeled data examples for classifier construction. However, for rare categories that have arbitrary shapes, the assumption on the shapes of rare categories in the

TABLE 1
Notation

Symbol	Description
D	The set of unlabeled data examples.
n	The number of all data examples in D.
d	The dimensionality of D.
x_i	The ith data examples in D ($i \in \{1, \ldots, n\}$).
s	A known rare-category data example in D.
k	The number of nearest neighbors.
x'	The new position of data example x after position shift.
α	The step size coefficient for position shift ($0 < \alpha \leqslant 1$).
$k\text{NN}_x$	The k-nearest neighbors of data example x.
$k\text{NN}_x(j)$	The jth-nearest neighbor of data example x ($j \in \{1, \ldots, k\}$).
μ_x	The mean of the k-nearest neighbors of data example x.
t	The upper bound for the times of position shift of each data example in D.
$d_x(j)$	The distance between data example x and its jth nearest neighbor ($j \in \{1, \ldots, k\}$). Without loss of generality, $d_x(1)$ denotes the distance between x and its nearest neighbor.
$\overline{d_x}$	The harmonic mean of the distances $\{d_x(1), \ldots, d_x(k)\}$.
$A_{j\ell}$	The affinity between distances $d_x(j)$ and $d_x(\ell)$ ($j, \ell \in \{1, \ldots, k\}$).
$A \in \mathbb{R}^{k \times k}$	The affinity matrix of the distances $\{d_x(1), \ldots, d_x(k)\}$.
L	The set of indices that indicate which data examples in $k\text{NN}_x$ are regarded to be close enough to data example x.
$\mathbf{v} \in \mathbb{R}^k$	The probability vector of which each element v_j indicates the probability that distance $d_x(j)$ is in the cluster of distances found by Eq. (4) ($\sum_j v_j = 1, v_j \geqslant 0, j \in \{1, \ldots, k\}$).
$w_L(j)$	The advantage of the average affinity between $d_x(j)$ and distances $\{d_x(\ell) \mid \ell \in L \setminus \{j\}\}$ over the average affinity among the distances $\{d_x(\ell) \mid \ell \in L \setminus \{j\}\}$.

above two seed-based approaches may reduce their utility, resulting in a performance degradation. Besides the above two seed-based approaches, some traditional clustering algorithms like DBSCAN algorithm [5] can also be used to identify the cluster of a rare category based on a seed. Nonetheless, all these clustering algorithms as well as the existing seed-based approaches determine whether or not a data example is from the objective rare category by once estimating. As a rule, if the estimation is over-prudent, it will miss a few real rare-category data examples, otherwise it may have more wrong candidates in the result of rare-category identification. To avoid the above flaws, our LERI algorithm adopts a kNN-based local exploration framework to identify the data examples of arbitrary shaped rare categories, and jointly utilizes prudent selection and multiple inspection to achieve a complete and accurate identification result.

2.2 Rare-Category Detection

Rare-category detection aims at finding out at least one representative data example for each rare category in an unlabeled data set with the help of a labeling oracle. According to the selection strategy for representative rare-category data examples, the existing rare-category detection approaches can be categorized into the following four groups: (1) classification-based methods construct a classifier based on the data examples that have been labeled by labeling oracle, and select data examples that cannot be well explained by this classifier for the next round of labeling [11], [30]; (2) clustering-based methods decompose the data set into a few cohesive clusters, and select data examples that are isolated from the clusters for labeling [33], [34]; (3) density-based methods assume that large variations of local density indicate the presence of rare categories [8], [40], [42]; and (4) nearest-neighborhood-based methods investigate the changes in local data distribution around each data example to discover the candidate data examples of rare categories [14], [35].

Although rare-category detection algorithms do not identify all data examples of a rare category, they can help us to find a seed for rare-category identification. The results of rare-category identification in turn can help the rare-category detection algorithms to update their classifiers or to extract the characteristics of rare categories more exactly.

3 The LERI Algorithm

In this section, we first define some notations used in this paper, and then elaborate how to connect rare-category data examples and filter out external data examples, followed by presenting the detailed steps of our LERI algorithm together with a complexity analysis, before concluding this section with a discussion on parameter estimation for LERI.

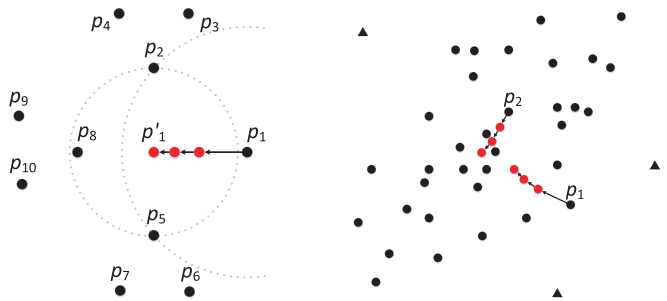
3.1 Notation

In rare-category identification, we are given a set of unlabeled data examples $D = \{x_1, \ldots, x_n\}$, where $x_i \in \mathbb{R}^d$ and $1 \leqslant i \leqslant n$, and a seed $s \in D$ that is a known data example from the objective rare category. Our goal is to identify all the data examples from the objective rare category, minimizing the false positive and false negative errors of the identification result. Tabel 1 lists the notation that will be used henceforth.

3.2 Connecting Rare-Category Data Examples

For rare categories, we make the following assumption, i.e., the distribution of the rare categories is locally smooth. This assumption is much more general than the compact and spherical-shape assumption adopted by existing approaches. Based on our assumption, whatever the overall shape and distribution scope of a rare category are, the adjacent small local parts in this rare category are smoothly connected. Hence, we propose to exploit this kind of local connectivity to connect the seed and the other data examples in the objective rare category. To this end, our proposed LERI algorithm utilizes the kNN relationship with a relatively small k value to perform this connection work (as the kNN relationship with a large k value is not appropriate to reflect local connectivity). To be more efficient, LERI does not try to find out the kNN of all data examples. Instead, it carries out a local exploration on the given data set, which starts from visiting seed and keeps on searching and visiting the kNN of each newly visited data example until the latest kNN search can not find any unvisited data example.

Note that with a relatively small k value, the connection scope of each data example's kNN is limited, and data examples in two adjacent small local parts of the objective rare category may not be connected by the kNN relationship. For



(a) $k = 2, \alpha = 0.5$ (b) $k = 4, \alpha = 0.5$

Fig. 1. Examples of position shift (black triangles are external data examples, black dots are rare-category data examples, and red dots are the new positions of corresponding black dots after position shifts).

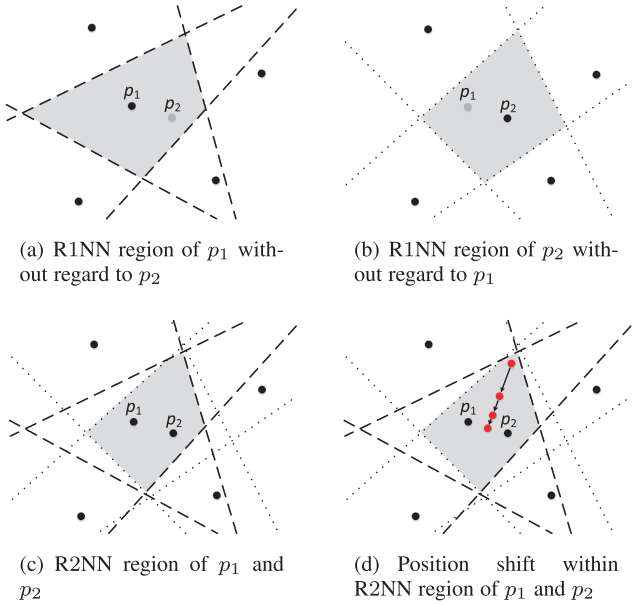
example, in Fig. 1a, let data example p_1 be the seed and $k = 2$. The k NN relationship can only connect p_2, p_3, p_4 and p_5, p_6, p_7 with p_1, while missing the local part consisting of p_8, p_9 and p_{10}. To avoid this kind of situation and extend the connection scope, we can shift the position of each newly visited data example during the local exploration to help create more new k NN relationships across the newly visited local parts and adjacent unvisited local parts. Nonetheless, a random position shift may move the newly visited data examples out of the rare-category cluster, increasing the risk of involving extra external data examples into the local exploration. A good position shift method should help us eliminate the above risk. To achieve this, we shift the position of each newly visited data example x towards its nearest neighbors, and updates the kNN relationships of shifted data examples in the next round of exploration. Formally, the position shift can be formulated as

$$x' = \frac{\alpha}{k}\sum_{j=1}^{k} k\mathrm{NN}_x(j) + (1-\alpha)x, \tag{1}$$

where x' is the new position of data example x, $k\mathrm{NN}_x(j)$ is the jth nearest neighbor of x in the given data set D, and α is a step size coefficient for position shift ($0 < \alpha \leqslant 1$).

According to Eq. (1), a data example x is shifted towards the mean $\mu_x = \frac{1}{k}\sum_{j=1}^{k} k\mathrm{NN}_x(j)$ of its current kNN with a step size $\triangle x$ ($\triangle x = |x' - x| = \alpha|\frac{1}{k}\sum_{j=1}^{k} k\mathrm{NN}_x(j) - x|$). For example, in Fig. 1a, the seed p_1 will be gradually shifted to the mean of p_2 and p_5, i.e., p'_1, and then find p_8 as its kNN, which will involve p_9 and p_{10} into the next round of local exploration. Then, all data examples in Fig. 1a will be connected by kNN relationship due to the position shift.

As a rule, the kNN of shifted data examples will not involve extra external data examples due to the following observations, i.e., the position shift tends to shift boundary rare-category data examples towards the core parts of the rare category, and helps prevent inner rare-category data examples from drifting apart. For example, Fig. 1b illustrates these two observations by depicting the trajectories (in red) of boundary data example p_1 and inner data example p_2 when we continuously shift their positions by Eq. (1). The reason behind is that each data example is more likely to find its kNN from a nearby local part which has a higher local density and more data examples. Compared with the boundary parts of a rare category, the core parts of the rare category often have relatively higher local densities and more data examples for kNN search, and thus have greater attraction for nearby data examples, since the position shift moves data examples closer to their kNN.



(a) R1NN region of p_1 without regard to p_2 (b) R1NN region of p_2 without regard to p_1

(c) R2NN region of p_1 and p_2 (d) Position shift within R2NN region of p_1 and p_2

Fig. 2. Examples of RkNN region (in gray).

Besides, when the position shift continues, we can also have the following observation, i.e., the shifted data examples converge towards certain positions after a few rounds of position shift, and each of these certain positions is a mean of k data examples in D. To explain the reason behind this observation, based on the definition of reverse k-nearest neighbors (abbreviated as RkNN henceforth), i.e., data example p_j is one of the RkNN of data example p_i iff p_i is one of the kNN of p_j, we introduce a new concept called RkNN region defined as follows.

Definition 1. *For any k data examples in a given data set D, the RkNN region of the k data examples refers to an area in which each position point will find these k data examples, rather than the other data examples in D, as its kNN.*

Based on the concept of RkNN region, the convergence property of our position shift method can be intuitively explained by the examples illustrated in Fig. 2. The gray parts in Figs. 2a and 2b represent the R1NN region of data example p_1 (without regarding the existence of p_2), and the R1NN region of data example p_2 (without regarding the existence of p_1), respectively. The intersection of these two R1NN regions consists of the R2NN region of p_1 and p_2, which is depicted in Fig. 2c. As shown in Fig. 2d, when a data example moves within this R2NN region, it will always find p_1 and p_2 as its 2NN, and will be gradually shifted to the mean of p_1 and p_2 by Eq. (1). This observation can be also theoretically explained by the following Theorem 1 and Corollary 1, which reveals the convex property of RkNN region and presents the convergence condition of the position shift, respectively.

Theorem 1. *A RkNN region is a convex set.*

Proof. For any two data examples x_1, x_2 in a data set D, there is a half-space $H^+_{x_1}(x_2)$, in which any position point is closer to x_1 than to x_2. For any two position points $p_1, p_2 \in H^+_{x_1}(x_2)$, and any real number $\alpha \in (0, 1]$, the relationship

$$\alpha p_1 + (1-\alpha)p_2 \in H^+_{x_1}(x_2), \tag{2}$$

always holds, indicating that any position point on the line segment between p_1 and p_2 is still in the half-space $H^+_{x_1}(x_2)$. Thus, according to the definition of convex set, $H^+_{x_1}(x_2)$ is a convex set. In a given data set D, the R1NN region of any data example $x_1 \in D$ is an intersection of a set of half-spaces $\{H^+_{x_1}(x_2) \mid x_2 \in D \setminus \{x_1\}\}$, and the R$k$NN region of any k data examples is an intersection of a set of R1NN regions. As the intersection of convex sets is still a convex set, a RkNN region is still a convex set. The proof completes. □

Corollary 1. *Given a data set D and any k data examples in D, let p denote the mean of the k data examples, and $x \in D$ be a data example of which the position is continuously shifted by Eq. (1). If p is in the RkNN region of the k data examples, then once the position of x is shifted into this RkNN region, its newly shifted position will converge to p.*

Proof. If the mean of the given k data examples, say p, is in the RkNN region of the k data examples, and data example x is shifted to a new position, say x', which is also in the RkNN region, then according to Eq. (2), any data point on the directed line segment from x' to p is still within the RkNN region of the given k data examples. This is because according to Theorem 1, each RkNN region is a convex set. According to Eq. (1), the subsequent positions of x are still on the the directed line segment from x' to p, and thus are still within the RkNN region of the given k data examples. In other words, for the subsequent positions of x, the kNN are still the given k data examples, and the distance from each newly shifted position of x to p will gradually decrease with a ratio of α. The proof completes. □

Based on the above Corollary 1, we also have the following corollary, which provides a stopping criterion for the position shift, and can help us avoid unnecessary position shifts during the local exploration.

Corollary 2. *Given a data set D and a data example $x \in D$, let $k\text{NN}_x$ denote the kNN of x, μ_x denote the mean of $k\text{NN}_x$, and $k\text{NN}_{\mu_x}$ denote the kNN of μ_x. When the position of x is shifted to x' by Eq. (1), if $k\text{NN}_x = k\text{NN}_{\mu_x}$, then the kNN of x', i.e., $k\text{NN}_{x'}$, are always equal to $k\text{NN}_x$.*

Proof. Relationship $k\text{NN}_x = k\text{NN}_{\mu_x}$ indicates that both x and μ_x are in the RkNN region of $k\text{NN}_x$. According to Corollary 1, the shifted position x' of x will be closer to μ_x, but still in the RkNN region of $k\text{NN}_x$. Therefore, x' will still find $k\text{NN}_x$ as its kNN. The proof completes. □

According the corollary above, for any data example x, if relationship $k\text{NN}_x = k\text{NN}_{\mu_x}$ holds, it is unnecessary to shift the position of x by Eq. (1) any more, since no more new kNN relationship will be created by the position shift.

3.3 Filtering Out External Data Examples

With the local exploration, the seed and the other data examples in objective rare category will be gradually connected by the kNN relationship, and be visited by our LERI algorithm. To prevent external data examples, especially those near the boundary of the objective rare category, from being connected and visited as rare-category data examples, for each newly visited example $x \in D$, we need to effectively filter out external data examples in the kNN of x.

In the existing work of rare-category identification, the data examples of a rare category are assumed to be isolated from the other data examples [34] or have relatively higher local densities than the data examples nearby [9]. In either of these two cases, the distances between adjacent rare-category data examples are relatively smaller than the distances between rare-category and external data examples. Therefore, our LERI algorithm utilizes this distance differentials to identify the suspected external data examples.

Moreover, instead of filtering out a suspected external data example by once estimating, our LERI algorithm carries out this work via multiple inspection. In other words, a suspected external data example in current round of local exploration will not be abandoned permanently. It still has the opportunity to be visited, if it is close enough to any subsequent newly visited data example. This is because although a suspected data example is relatively far away from current newly visited rare-category data examples, it may be close enough to the other unvisited rare-category data examples of the objective rare category. Therefore, in our LERI algorithm, a suspected external data example will be prevented from being visited for only the next round of local exploration, but not forever.

Given the above multiple inspection-based filtering strategy, our LERI algorithm executes the identification of suspected external data examples by prudently reserving the data examples in the kNN of a newly visited data example x which are close enough to x, and regarding the rest of data examples in the kNN as the suspected external data examples. Specifically, we conduct an affinity analysis on the distances $\{d_x(1), \ldots, d_x(k)\}$ between x and its kNN to find out which distances have the highest affinities to the distance $d_x(1)$ between x and its nearest neighbor.[1] The affinity $A_{j\ell}$ between each two distances $d_x(j)$ and $d_x(\ell)$ ($j, \ell \in \{1, \ldots, k\}$) is defined as

$$A_{j\ell} = \begin{cases} \exp(-\frac{|d_x(j)-d_x(\ell)|}{\overline{d_x}}), & j \neq \ell, \\ 0, & j = \ell, \end{cases} \tag{3}$$

where $\overline{d_x}$ is the harmonic mean of $\{d_x(1), \ldots, d_x(k)\}$, and $\exp(-\frac{|d_x(j)-d_x(\ell)|}{\overline{d_x}})$ reflects the relative affinity between $d_x(j)$ and $d_x(\ell)$ w.r.t. the harmonic mean of $\{d_x(1), \ldots, d_x(k)\}$. The problem of affinity analysis can be carried out by pairwise clustering in a form of standard quadratic program.

$$\max \mathbf{v}^{\mathrm{T}} A \mathbf{v} \quad \text{s.t.} \quad \mathbf{v} \in \Delta, \tag{4}$$

where $A \in \mathbb{R}^{k \times k}$ is the affinity matrix of $\{d_x(1), \ldots, d_x(k)\}$, $\mathbf{v}$ is a probability vector of which each element v_j indicates the probability that distance $d_x(j)$ is in the objective cluster found by solving the standard quadratic program in Eq. (4), and $\Delta = \{\mathbf{v} \in \mathbb{R}^k \mid \sum_j v_j = 1, v_j \geq 0, j = 1, \ldots, k\}$.

To solve the above pairwise clustering problem, traditional approaches like evolutionary computation [3] and dominant-set [29] require user-specified stopping criteria for the convergence, such as a change threshold for $\mathbf{v}$, which often significantly affect their final clustering results. More importantly, their final clustering results may not include distance $d_x(1)$. This is because directly solving the pairwise clustering problem in Eq. (4) with traditional approaches will approximate an optimal cluster which has the highest internal affinity, rather than a cluster of distances that have the highest affinities to distance $d_x(1)$. To avoid the above flaws,

1. Without loss of generality, let $d_x(1)$ denote the distance between x and its nearest neighbor.

we propose the following three-step approach, denoted as **find_externals_in_*k*NN**($k\text{NN}_x, x$), to solve the pairwise clustering problem and identify the suspected external data examples without any parameters to determine the convergence of the final results.

Step 1. Initialize $L = \{1\}$, and $\mathbf{v} = (1, 0, \ldots, 0)^{\mathrm{T}}$, where set L is used to record the indices of distances that have the highest affinities to distance $d_x(1)$.

Step 2. Find out the $d_x(j^*) \in \{d_x(1), \ldots, d_x(k)\}$ with the highest affinity to distances $\{d_x(\ell) \,|\, \ell \in L\}$ by calculating j^* as follows.

$$j^* = \arg\max_j (\mathbf{v}^{\mathrm{T}} A \mathbf{e}^j - \mathbf{v}^{\mathrm{T}} A \mathbf{v}), \tag{5}$$

where $\mathbf{e}^j \in \mathbb{R}^k$ refers to the jth column of the identity matrix, $\mathbf{v}^{\mathrm{T}} A \mathbf{e}^j$ is the average affinity between distance $d_x(j)$ and distances $\{d_x(\ell) \,|\, \ell \in L\}$, and $\mathbf{v}^{\mathrm{T}} A \mathbf{v}$ reflects the average affinity of the distances $\{d_x(\ell) \,|\, \ell \in L\}$.

Step 3. If $j^* \in L$, then return the corresponding data examples in $k\text{NN}_x$ of which the indices are not recorded in L as the elements of set E_x; otherwise, go back to Step 2 after updating $L = L \bigcup \{j^*\}$ and updating each element v_j of $\mathbf{v}$ by

$$v_j = \begin{cases} \frac{w_L(j)}{\sum_{\ell=1}^{|L|} w_L(\ell)}, & j \in L, \\ 0, & j \notin L, \end{cases} \tag{6}$$

where $w_L(j)$ reflects the advantage of the average affinity between distance $d_x(j)$ and distances $\{d_x(\ell) \,|\, \ell \in L \setminus \{j\}\}$ over the average affinity among distances $\{d_x(\ell) \,|\, \ell \in L \setminus \{j\}\}$, and can be calculated as follows.

$$w_L(j) = \begin{cases} 1, & \text{if } |L| = 1, \\ \sum_{\ell \in L \setminus \{j\}} \beta w_{L\setminus\{j\}}(\ell), & \text{otherwise.} \end{cases}$$
$$\beta = A_{\ell j} - \frac{1}{|L \setminus \{j\}|} \sum\nolimits_{i \in L \setminus \{j\}} A_{\ell i}. \tag{7}$$

Intuitively, the above three steps keep on finding distances that can increase the intra-set average affinity of the distances in $\{d_x(\ell) \,|\, \ell \in L\}$, and adding their indices into current L. When there is no more such distances, the execution of the three steps terminates, and returns a set E_x of suspected external data examples that will be prevented from being visited in the next round of local exploration.

3.4 Algorithm

To find out the data examples of a arbitrary shaped rare category based on a seed, we introduce a kNN-based local exploration framework, which utilizes kNN relationship to gradually connect the seed and the other rare-category data examples, and present a position shift method to enhance the connection effect. To prevent external data example from being connected with the data examples in the objective rare category, we only involve data examples that are close enough to visited data examples into the local exploration. Based on these preparing work, we propose an algorithm called LERI for the problem of rare-category identification.

The proposed LERI algorithm, outlined in Algorithm 1, takes as inputs a given unlabeled data set $D = \{x_1, \ldots, x_n\}$ of n data examples, a seed $s \in D$, and two parameters k and α. It performs an iterative local exploration on D, beginning from seed s (line 2). Each round of local exploration (lines 4–13) consists of two main phases, i.e., (1) the phase of connecting rare-category data examples (lines 7–9), which finds the kNN of each newly visited data example x among the data examples in data set D (line 6), and shifts x closer to its kNN (i.e., $k\text{NN}_x$) by using Eq. (1) if the position shift of x is not satisfied in stopping criterion (i.e., $kNN_x \neq kNN_{\mu_x}$), and (2) the phase of identifying suspected external data examples (line 10), which utilizes the three-step approach proposed in Section 3.3 to find out data examples that are not close enough to x among the kNN of x as the suspected external data examples. A suspected external data example will be temporarily locked and can not be visited in next round of local exploration. At the end of each round of local exploration, LERI updates the newly visited data examples (line 13). Specifically, unvisited and unlocked data examples involved in kNN search and newly shifted data examples are updated as newly visited data examples for the next round of local exploration. When the kNN search fails to involve any unvisited and unlocked data example in D, i.e., $\big(\bigcup_{x \in N}(k\text{NN}_x \setminus E_x)\big) \bigcap \big(D \setminus C\big) = \emptyset$, and there is no newly shifted data example, i.e., $S = \emptyset$, the local exploration terminates and all visited data examples in C are returned as the data examples of the objective rare category.

Algorithm 1. The LERI Algorithm

Input: $D = \{x_1, \ldots, x_n\}$, a seed $s \in D$, k, α.
Output: The set C of rare-category data examples.
1 $C \leftarrow \emptyset$; // the set of visited data examples in D
2 $V \leftarrow \{s\}$; // the set of newly visited data examples
3 **while** $V \neq \emptyset$ **do**
4 $S \leftarrow \emptyset$; // the set of data examples newly shifted
5 **for** each $x \in V$ **do**
6 $k\text{NN}_x \leftarrow$ ***k*NN_search**(D, x, k);
7 **if** $k\text{NN}_x \neq k\text{NN}_{\mu_x}$ **then**
8 $x' \leftarrow$ **position_shift**$\big(k\text{NN}_x, x, \alpha\big)$;
9 $S \leftarrow S \bigcup \{x'\}$; // $S \bigcap D = \emptyset$
10 $E_x \leftarrow$ **find_externals_in_*k*NN**$\big(k\text{NN}_x, x\big)$;
11 **if** $x \in D$ **then**
12 $C \leftarrow C \bigcup \{x\}$;
13 $V \leftarrow \Big(\big(\bigcup_{x \in N}(k\text{NN}_x \setminus E_x)\big) \bigcap \big(D \setminus C\big)\Big) \bigcup S$;

In each round of local exploration, the most computationally expensive process is the kNN search, which requires $O(n)$ time for each newly visited data example x. Each position shift requires $O(k)$ time to move x closer to its kNN, and identifying the suspected external data examples in $k\text{NN}_x$ takes $O(k^2)$ time as a pair-wise affinity analysis on the distances from x to its kNN is carried out to help find out relatively greater ones. Let τ denote the total number of visited data examples at the end of local exploration, and t denote the upper bound for the times of position shift of each data example, there will be at most τt shifted data examples, and the overall time complexity of our proposed LERI algorithm is about $O\big(\tau(t+1)(n+k^2+k)\big)$, where $\tau \ll n$, $t \ll n$, and $k \ll n$.

3.5 Parameter Estimation

Besides the given data set D and seed $s \in D$, our LERI algorithm has two extra parameters, namely (1) a value k for kNN relationship, and (2) a step size coefficient α for position shift. In what follows, we discuss how to estimate or select these two parameters.

Estimation of k. As pointed out in Section 3.2, we should select a relatively small k value for our LERI algorithm.

Nonetheless, a fixed small k value may not be appropriate for different data sets. Hence, we propose to adjust the k value for different data sets via the following two steps. (1) Executing LERI on the given data set with $k = 2$ (when $k = 1$, our proposed approach for suspected external data example identification will not work). (2) Based on the data examples visited in the first execution, increasing the k value progressively until the kNN of the visited data examples include any data example that is unvisited in the first execution. This estimation approach for k helps select a k value as small as possible, while bringing a relatively higher connection effect for the kNN relationship.

Selection of $\boldsymbol{\alpha}$. Our position shift method moves a data example x towards the mean μ_x of its kNN, with a step size $\triangle x = \alpha|\frac{1}{k}\sum_{j=1}^{k} k\mathrm{NN}_x(j) - x|$. A larger value for the step size coefficient α will lead to a greater step size $\triangle x$. If data example x can find some new kNN when it moves from current position to μ_x, the position shift will create new kNN relationship to enhance the connection effect, no matter it adopts a large or small step size. From the perspective of efficiency, a large step size (e.g., using $\alpha = 1$) will help x to quickly find new kNN over a longer distance, while it may miss a few new kNN that are closer to the original position of x. In contrast, a relatively small step size (e.g., using $\alpha = 0.1$) will help x to avoid missing any new kNN nearby at the cost of more useless steps of position shift. In fact, as long as the step size is not extremely small, it will not significantly affect the efficiency performance of LERI, since the runtime for position shift is not the dominant item in the time complexity of LERI. In this paper, we suggest to use a moderate step size, i.e., $\alpha = 0.5$.

4 Experimental Evaluation

In this section, we verify the effectiveness and efficiency of our LERI algorithm on both synthetic and real-world data sets, followed by discussing the effects of position shift, external data example identification, estimated k, and parameter α. All algorithms in the experiments are implemented in MATLAB 8.3, running on a desktop PC with Intel Core i3-6100 CPU at 3.70GHz and 8GB RAM.

4.1 Effectiveness Study

In this experiment, we evaluate the accuracy performance of our LERI algorithm in terms of F-score, which is the harmonic mean of precision and recall, and is calculated as follows.

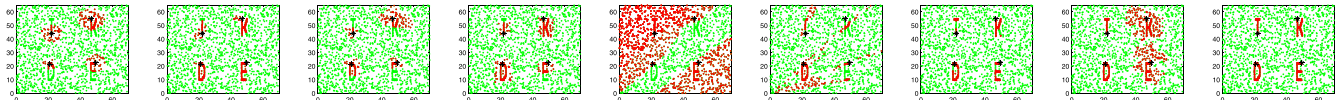
$$F{-}score = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall},$$
$$Precision = \frac{N_{TP}}{N_{TP} + N_{FP}}, \quad Recall = \frac{N_{TP}}{N_{TP} + N_{FN}},$$

where N_{TP}, N_{FP} and N_{FN} refer to the numbers of true positives, false positives, and false negatives, respectively.

To study the accuracy performance of our LERI algorithm with different seeds, we separately execute the representative algorithms of the existing four types of rare-category detection approaches, i.e., the classification-based approach Interleave [30], the clustering-based approach HMS [34], the density-based approach NNDM [8], and the nearest-neighborhood-based approach CLOVER [14], on the given data set, and select the first rare-category data example detected by each of these approaches as one of the seeds. Intuitively, these seeds are regarded as the most characteristic rare-category data examples from different aspects.

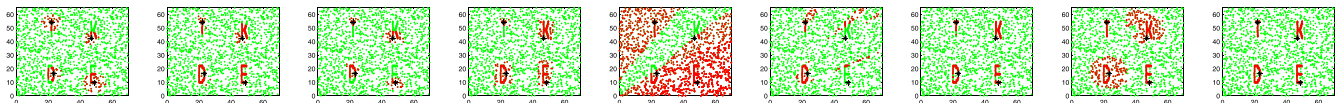
We compare the accuracy performance of LERI with that of a state-of-the-art supervised learning approach for rare-category identification called RACH (short for **RA**re category **CH**aracterization) [10], a high-performance semi-supervised learning approach based on manifold learning [41] (abbreviated as Manifold henceforth), two seed-based approaches to rare-category identification known as FRANK [13] and RCEWA [25], as well as a few classical classification and clustering approaches, including Roc-SVM (short for **Roc**chio based **S**upport **V**ector **M**achine) [23] which can train a classifier based on unlabeled data examples and one or more labeled seeds from the objective class, an artificial neural network (abbreviated as ANN henceforth) with 10 layers and 100 neurons on each layer, kernel logistic regression (abbreviated as Kernel LR henceforth) using Gaussian Radial Basis Function (RBF) kernel [27], and DBSCAN which can find a cluster starting from a seed. As only one labeled rare-category data example is not enough for RACH, Manifold, ANN and Kernel LR to construct their classifiers, we use each seed and 10 percent data examples randomly selected from the given data set to construct a training set. Each data example in the training set is labeled. Furthermore, we give RACH, Manifold, ANN and Kernel LR a privilege, i.e., for each seed, we construct 10 training sets with different randomly selected data examples, and use the best accuracy performance of RACH, Manifold, ANN and Kernel LR on these training sets as their final performance with the seed. Moreover, as the performance of DBSCAN is often affected by its two parameters (i.e., the radius *Eps* and the required minimum number *MinPts* of data examples in the *Eps*-neighborhood of a data example), we give DBSCAN a privilege, i.e., for each seed and its corresponding rare category, we vary the parameter *Eps* from 0.1*MaxDist* to *MaxDist* (with interval 0.01*MaxDist*, where *MaxDist* is the maximal distance from the seed to the data examples in the rare category), vary the parameter *MinPts* from $0.1r$ to r (with interval $0.01r$, where r is the total number of data examples in the rare category), and then, for each data set with four seeds selected by four different rare-category detection approaches, we find the optimal *Eps* and *MaxDist* that bring DBSCAN the best average accuracy performance, and report this best average accuracy performance as the final performance of DBSCAN on this data set. In what follows, we report the accuracy performance comparison results on both synthetic data set and UCI data sets.

Synthetic Data Set. The synthetic data set contains 1700 data examples, including a majority category, which has 1467 data examples, and 4 rare categories, which have 35, 70, 66, 62 data examples, respectively. The shape of each rare category forms one of characters 'T', 'K', 'D', and 'E', respectively. Figs. 3, 4, 5, and 6 illustrate the identification results of LERI and the other tested algorithms, in which black asterisks are the seeds selected by four rare-category detection approaches, red points and green points are respectively the rare-category data examples and majority-category data examples returned by each algorithm. From the figures, we can have the following observations. (1) LERI is able to accurately identify the data examples of each rare category based on only one seed. (2) RACH prefers to find clusters with spherical shapes, and involves many nearby external data examples into its final results. This is because RACH identifies the cluster of the objective rare category by finding



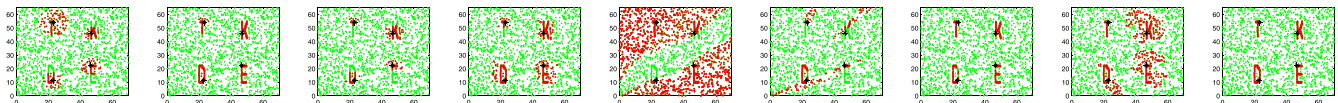
(a) RACH (b) Manifold (c) FRANK (d) RCEWA (e) Roc-SVM (f) ANN (g) Kernel LR (h) DBSCAN (i) LERI

Fig. 3. Rare-category identification result on synthetic data set using seeds selected by Interleave.



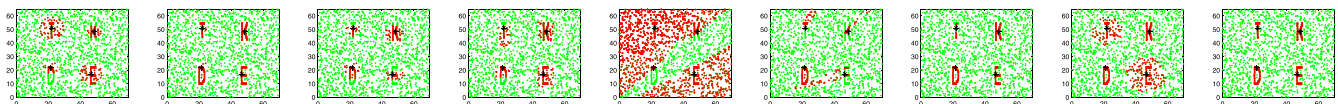
(a) RACH (b) Manifold (c) FRANK (d) RCEWA (e) Roc-SVM (f) ANN (g) Kernel LR (h) DBSCAN (i) LERI

Fig. 4. Rare-category identification result on synthetic data set using seeds selected by HMS.



(a) RACH (b) Manifold (c) FRANK (d) RCEWA (e) Roc-SVM (f) ANN (g) Kernel LR (h) DBSCAN (i) LERI

Fig. 5. Rare-category identification result on synthetic data set using seeds selected by NNDM.



(a) RACH (b) Manifold (c) FRANK (d) RCEWA (e) Roc-SVM (f) ANN (g) Kernel LR (h) DBSCAN (i) LERI

Fig. 6. Rare-category identification result on synthetic data set using seeds selected by CLOVER.

a compact hyper-ball to enclose the labeled rare-category data examples in its training set. (3) Manifold has a relatively better performance on handling the arbitrary shaped rare categories, but fails to screen a few adjacent external data examples from the data examples of rare categories. (4) The results of FRANK often have a low recall. The reason behind is that FRANK assumes that the data examples of a rare category are restricted in a small convex local region, and are close to each other. However, in an arbitrary shaped rare category, data examples of different parts may not be close to each other due to the arbitrary shape. (5) RCEWA prefers to return data example groups with rectangle shapes. The sizes of these groups change a lot. This is because RCEWA decomposes the given data set into small rectangle bins. The bin size is estimated based on the size of cluster returned by performing density-based clustering process which starts from the seed. If the returned cluster is over-small or over-large, the result of RCEWA tends to have a similar mistake. (6) The classification hyperplanes constructed by Roc-SVM and ANN are less competent to accurately extract the rare categories which are hidden in the majority category, since these two methods are originally proposed to handle the linearly separable classification problems. In contrast, the classification results of Kernel LR are much better, although they often miss partial data examples in the objective rare categories. The reason behind is that kernel tricks can make the linearly non-separable categories to be more separable in a new high dimensional feature space. (7) Despite DBSCAN with the optimal parameters has the best average accuracy performance on each data set, it often fails to identify adjacent external data examples from the data examples of partial rare categories.

UCI Data Set. We also evaluate each tested algorithm on six real data sets from UCI machine learning repository [24], i.e., data sets DS1 to DS6 described in Table 2, where D denotes the data set, n is the number of data examples in D, d is the data set dimension, m is the number of categories. These six data sets have gradually increasing number n of data examples, and multiple rare categories in each of them. For each data set, we select four seeds, each of which is the first rare-category data example detected by one of the four rare-category detection approaches, namely Interleave, HMS, NNDM, and CLOVER. Table 3 reports the average F-score and the standard error of F-scores of each tested algorithm with different seeds on each UCI data set. Since the runtime of Manifold on data set DS6 is prohibitively long and beyond acceptable, we omit the performance result of Manifold on DS6. From the table, we can observe that LERI outperforms the other tested algorithms in terms of F-score, and has a reasonably better accuracy performance for rare-category identification.

4.2 Efficiency Study

In this experiment, we first compare the runtime of LERI with that of the other tested algorithms on DS1–DS6, and then conduct a scalability study for these algorithms with varying number of data examples in the given data set, different dimensionality of the given data set, and varying number of data examples in the objective rare category. To ensure a fair comparison, no index is used to accelerate the kNN search in LERI or any other algorithm.

Runtime Comparison. Table 4 reports the average runtime of each tested algorithm with different seeds on each UCI data

TABLE 2
Description of UCI Data Sets

D	Name	n	d	m
DS1	Ecoli	336	7	8
DS2	Pen Digits	1040	16	10
DS3	Page Blocks	5473	20	5
DS4	Letter	18688	16	26
DS5	Shuttle	58000	9	7
DS6	KDD99	494021	39	23

set, from which we can have the following observations. (1) When data sets grow larger, ANN and Kernel LR usually have the better efficiency performance than most of the other tested algorithms. This is because the most computationally expensive process of these two approaches is constructing the classifier based on the training set, which is only a small part (i.e., about 10 percent) of each given data set in this experiment. (2) Compared with our LERI algorithm, DBSCAN has a similar but slightly better efficiency performance. This is because both of the two approaches adopt an iterative local exploration strategy rather than a global analysis on the whole data set, while LERI executes more analysis to prevent external data examples from the next round of local exploration. (3) LERI is significantly faster than RACH, Manifold, FRANK, RCEWA and Roc-SVM. The reason behind is that LERI has a low time complexity, which is nearly linear to the number n of data examples in the given data set. When n is fixed, the runtime of LERI is mainly determined by the total number τ of data examples visited in local exploration. As a rule, this number will not be too great when LERI has an accurate performance, since a rare category only occupies the minority of a data set. In contrast, Manifold and FRANK carry out pairwise similarity (or distance) analysis on the entire data set. Although RACH, RCEWA and Roc-SVM adopt pruning methods to reduce the number of data examples to be analysed, RACH and Roc-SVM repeatedly use the remaining data examples to improve its accuracy performance, and RCEWA carries out pairwise similarity analysis on the remaining data examples, both resulting in a relatively long runtime.

Scalability Study. To study the the scalability of our LERI algorithm to the number n of data examples in the given data set, based on the synthetic data set ($n = 1700, d = 2$) and four seeds used in the experiment of effectiveness study, we generate a series of five data sets with increasing n by creating new data example near the original ones from number 1700 to 8500 (with interval 1700), and execute the tested algorithms on each of these five data sets with the same four seeds. Fig. 7a illustrates the average runtime of each tested algorithm on each of these five data sets (the runtime in Fig. 7 is in log scale). From the figure, we can observe that as the number n of data examples in the given data set increases (with fixed data set dimensionality d and fixed number r of data examples in objective rare category), (1) DBSCAN and ANN show a greater advantage over the other tested algorithms in terms of runtime; (2) the runtime of LERI is close to that of Kernel LR; (3) compared with RACH, Manifold, FRANK, RCEWA and Roc-SVM, LERI shows a better scalability to larger data sets.

To investigate the scalability of our LERI algorithm to the data set dimensionality d, based on the 2-dimensional synthetic data set ($n = 1700, d = 2$), we generate another series of five data sets with increasing d by duplicating the original 2 dimensions to 10 dimensions (with interval 2). Fig. 7b illustrates the average runtime of each tested algorithm on each of these five data sets with the same four seeds used in the experiment of effectiveness study. From the figure, we can observe that compared with the other tested algorithms, our LERI algorithm, RCEWA, Roc-SVM and Kernel LR are less affected by the increasing dimensionality in terms of runtime, indicating that these four approaches have better scalability to the dimensionality of the given data set.

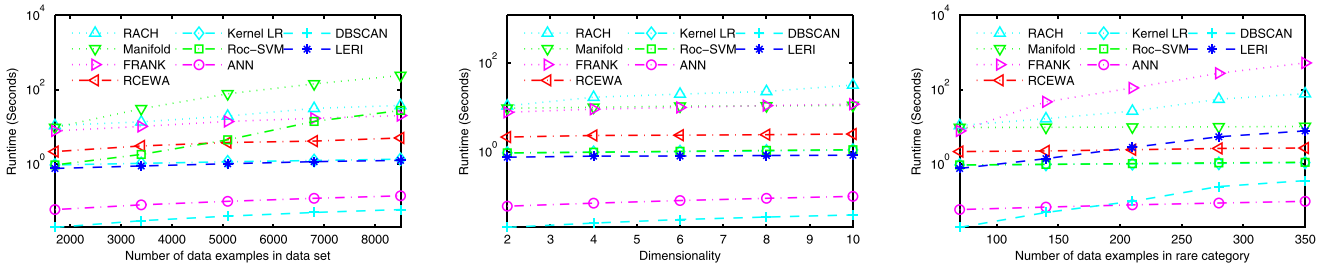
To evaluate the scalability of our LERI algorithm to the number r of data examples in the objective rare category, we modify the synthetic data set ($n = 1700, d = 2$) used in the experiment of effectiveness study, and adjust the number of data examples in the rare category ($r = 70$) that forms the character 'K' from number 70 to 350 (with interval 70). We execute the tested algorithms on each modified data set with the same four seeds used in the experiment of effectiveness study. Fig. 7c illustrates the average runtime of each tested algorithm on each modified data set. From the figure, we can

TABLE 3
Algorithm's Accuracy Performance on DS1–DS6 (Average F-Score $\pm$ Standard Error)

D	RACH	Manifold	FRANK	RCEWA	Roc-SVM	ANN	Kernel LR	DBSCAN	LERI
DS1	0.68 ± 0.09	0.57 ± 0.13	0.36 ± 0.15	0.64 ± 0.15	0.60 ± 0.07	0.53 ± 0.06	0.29 ± 0.13	0.68 ± 0.12	0.72 ± 0.10
DS2	0.65 ± 0.05	0.46 ± 0.03	0.90 ± 0.02	0.71 ± 0.15	0.61 ± 0.09	0.85 ± 0.07	0.80 ± 0.06	0.58 ± 0.14	0.91 ± 0.03
DS3	0.26 ± 0.02	0.30 ± 0.02	0.32 ± 0.02	0.33 ± 0.01	0.32 ± 0.02	0.37 ± 0.01	0.26 ± 0.01	0.12 ± 0.01	0.41 ± 0.02
DS4	0.25 ± 0.06	0.28 ± 0.05	0.35 ± 0.08	0.43 ± 0.07	0.33 ± 0.05	0.54 ± 0.09	0.44 ± 0.08	0.32 ± 0.06	$\mathbf{0.70} \pm 0.09$
DS5	0.35 ± 0.04	0.42 ± 0.02	0.48 ± 0.04	0.53 ± 0.08	0.58 ± 0.01	0.70 ± 0.03	0.51 ± 0.05	0.47 ± 0.13	$\mathbf{0.78} \pm 0.03$
DS6	0.77 ± 0.07	—	0.72 ± 0.06	0.80 ± 0.02	0.82 ± 0.01	0.85 ± 0.00	0.68 ± 0.12	0.80 ± 0.00	0.86 ± 0.01

TABLE 4
Algorithm's Average Runtime on DS1–DS6 (seconds)

D	RACH	Manifold	FRANK	RCEWA	Roc-SVM	ANN	Kernel LR	DBSCAN	LERI
DS1	33.16	10.82	27.06	5.06	0.95	0.05	0.68	0.04	0.55
DS2	214.31	809.59	8.82	2.11	1.25	0.05	0.91	0.02	0.15
DS3	1106.53	2108.72	44.25	12.35	4.02	0.06	1.08	0.02	0.10
DS4	1537.61	12051.07	812.24	45.12	11.63	0.10	1.22	0.04	0.47
DS5	3617.19	66931.41	3104.73	61.09	59.94	0.12	1.50	0.04	0.51
DS6	38639.23	—	154908.53	225.61	6151.87	1.20	1.91	14.60	20.62



(a) Scalability to number n of data examples ($r = 70,\ d = 2$)

(b) Scalability to dimensionality d of data set ($n = 1700,\ r = 70$)

(c) Scalability to number r of rare-category data examples ($n = 1700,\ d = 2$)

Fig. 7. Algorithm's scalability study.

observe that the runtime curve's gradient of LERI is similar to that of DBSCAN and RACH, greater than that of Manifold, RCEWA, Roc-SVM, ANN and Kernel LR, and less than that of FRANK. The reasons behind are as follows. (1) To identify a larger objective rare category, LERI often needs to visit a greater total number τ of data examples in its local exploration, and find the kNN of them among the data examples in the given data set, resulting in more computation. (2) DBSCAN adopts a similar local exploration strategy, and thus also requires more computation when there are more data examples in the objective rare category. (3) RACH repeatedly uses the data examples around the objective rare category. Hence, the size of objective rare category has an evident impact on the runtime of RACH. (4) In Manifold, the most computationally expensive steps are learning a manifold and spreading label information to unlabeled data example through the manifold. Their runtime depends on the data set size rather than the size of objective rare category. (5) RCEWA decomposes the given data set, and find a bin (or bins) that can cover the objective rare category. Since we have not extend the distribution scope of the objective rare category (although we have adjusted the number r), RCEWA can find the bin with approximately the same time in each modified data set, and takes a little more time to analyse a bin containing more rare-category data examples. (6) Regardless of how many data example the objective rare category has, Roc-SVM uses only one seed to construct its classifier. (7) The runtime of ANN and Kernel LR depend on the sizes of their training sets, which change a little in this experiment. (8) FRANK finds the local community of the objective rare category, and conducts a pairwise similarity analysis between the community members, of which the time complexity is quadratic.

In summary, our LERI algorithm can achieve a nearly linear time complexity w.r.t. the data set size and data set dimensionality, which makes it more efficient to handle large and high-dimensional data sets. Given a data set, the runtime of LERI mainly depends on the number of data examples in the objective rare category. Nonetheless, since a rare category often has only a few data examples, as a rule LERI will be competently efficient for rare-category identification.

4.3 Effect of Position Shift

In this experiment, we discuss the effect of position shift on the performance of LERI. To this end, we set the upper bound t for the times of position shift of each data example as to 0, 5, 10, 15 and 20, respectively. When $t = 0$, it means LERI is executed without position shift. We report the corresponding accuracy and efficiency performance of LERI on data sets DS1–DS6 in Figs. 8a and 8b. From the figures, we can observe that (1) in half of the tested data sets, position shift helps improve the accuracy performance of LERI. (2) When $t \geqslant 5$, although the runtime of LERI slightly increases with the growth of t, a higher t has no effect on the accuracy performance of LERI. This is because the shifted data examples converge to certain positions, and will not create new kNN relationships to involve new unvisited data examples for the subsequent rounds of local exploration.

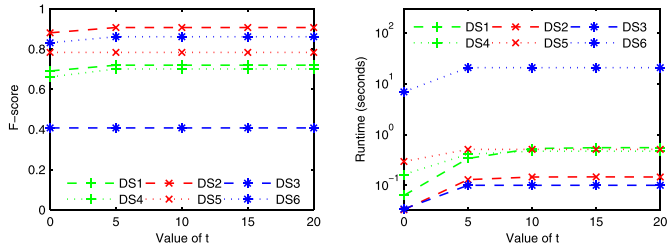
4.4 Effect of External Data Example Identification

In this experiment, we discuss the effect of our proposed external data example identification approach on the performance of LERI. Towards this, we run LERI without external data example identification (denoted as LERI-NoEx) on data set DS1–DS6, and set the stopping condition as either having r data examples visited during the local exploration or no more newly visited data example being found. In addition, we also run LERI with a naive strategy for external data example identification (denoted as LERI-Naive), i.e., for each newly visited data examples, we simply keep the top k' ($k' \in \{1, \ldots, k\}$) closest data examples in its kNN, and regard the rest of data examples in its kNN as the external data examples. Furthermore, we give LERI-Naive a privilege, i.e., we vary the value of k' from 1 to k, and use the best accuracy performance of LERI-Naive as its final performance. Figs. 9a and 9b compare the average F-score and average runtime of LERI, LERI-NoEx and LERI-Naive on each UCI data set (with the same seeds used in the experiment of effectiveness study). From the figures, we can observe that compared with LERI-NoEx and LERI-Naive, (1) LERI achieves a reasonably better accuracy performance due to our proposed external data example identification approach, and (2) the compensation is a relatively higher computation cost.

4.5 Effect of Estimated k

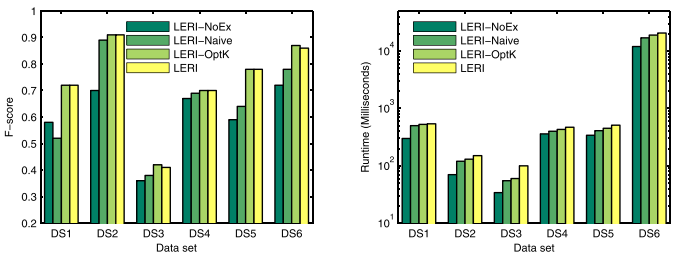
In our LERI algorithm, a parameter k is used to determine the kNN relationship among data examples. As presented in Section 3.5, we propose an estimation method for k to help find a relatively smaller k value, which can also bring a relatively higher connection effect for the kNN relationship.

In this experiment, we discuss the effect of the k value estimated by our proposed approach on the performance of LERI. To this end, based on pilot experiments, we find out the optimal k that makes LERI achieve the highest F-score. Then, we compare LERI using our estimated k and LERI using the optimal k (denoted as LERI-OptK) by reporting their average F-score and average runtime on DS1–DS6 (with the same seeds used in the experiment of effectiveness study). Figs. 9a and 9b show the comparison results, from



(a) Accuracy performance (b) Efficiency performance

Fig. 8. Performance of LERI with different t.



(a) Accuracy performance (b) Efficiency performance

Fig. 9. Performance of LERI without external data example identification (LERI-NoEx), LERI with naive strategy for external data examples identification (LERI-Naive), and LERI with optimal k (LERI-OptK).

which we can observe that the accuracy and efficient performance of LERI using our estimated k is approximately the same as that of LERI using the optimal k, indicating that our proposed estimation method for parameter k is effective.

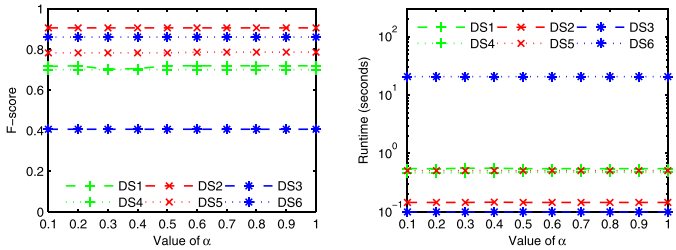
4.6 Effect of Parameter α

In our LERI algorithm, a parameter α ($0 < \alpha \leqslant 1$) is used as the step size coefficient for position shift. Intuitively, a greater value for α will lead to a relatively larger step size for each position shift in the local exploration. Nonetheless, as discussed in Section 3.5, as long as the value of α is not extremely small (e.g., very close to 0), it will enable the position shift to play its role without a significant effect on the efficient performance of LERI.

In this experiment, we experimentally evaluate the effect of parameter α on the performance of LERI. Towards this, we change the value of parameter α from 0.1 to 1 (with interval 0.1), and report the corresponding accuracy and efficiency performance of LERI on data sets DS1–DS6 in Figs. 10a and 10b. From the figures, we can observe that with different values for parameter α, the changes on the F-score and runtime of LERI are very little, indicating that our LERI algorithm is reasonably insensitive to this parameter α.

5 Conclusion

In this paper, we have investigated the problem of how to identify the data examples of an arbitrary shaped rare category based on a seed. Towards this, we have proposed the LERI algorithm, which executes a local exploration on the given data set from the seed. During each round of local exploration, LERI utilizes the kNN relationship among data examples and position shift to gradually connect the seed and the other data examples in the objective rare category, and identifies the suspected external data examples from the rare-category data examples based on the differentials between their distances to kNN. Extensive experimental results have verified the effectiveness and efficiency of LERI.



(a) Accuracy performance (b) Efficiency performance

Fig. 10. Performance of LERI with different α.

Acknowledgments

This work was supported in part by the National Key R&D Program of China (2018YFB1004003), NSFC Grants (61522208 and 61502347), the Technological Innovation Major Projects of Hubei Province (2017AAA125), the Science and Technology Program of Wuhan City (2018010401011288), and Xiaomi-WHU AI Lab.

References

[1] S. Barua, M. M. Islam, X. Yao, and K. Murase, "MWMOTE–majority weighted minority oversampling technique for imbalanced data set learning," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 26, pp. 405–425, Feb. 2014.

[2] M. Belkin, P. Niyogi, and V. Sindhwani, "Manifold regularization: A geometric framework for learning from labeled and unlabeled examples," *J. Mach. Learn. Res.*, vol. 7, no. 1, pp. 2399–2434, 2006.

[3] S. R. Bulò, M. Pelillo, and I. M. Bomze, "Graph-based quadratic optimization: A fast evolutionary approach," *Comput. Vis. Image Underst.*, vol. 115, no. 7, pp. 984–995, 2011.

[4] C. Elkan and K. Noto, "Learning to classify texts using positive and unlabeled data," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, 2008, pp. 213–220.

[5] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, 1996, pp. 226–231.

[6] A. Fernández-Baldera and L. Baumela, "Multi-class boosting with asymmetric binary weak-learners," *Pattern Recognit.*, vol. 47, no. 5, pp. 2080–2090, 2014.

[7] T. S. Haines and T. Xiang, "Active rare class discovery and classification using dirichlet processes," *Int. J. Comput. Vis.*, vol. 106, no. 3, pp. 315–331, 2014.

[8] J. He and J. Carbonell, "Nearest-neighbor-based active learning for rare category detection," in *Proc. Advances Neural Inf. Process. Syst.*, 2007, pp. 633–640.

[9] J. He, Y. Liu, and R. Lawrence, "Graph-based rare category detection," in *Proc. IEEE Int. Conf. Data Min.*, 2008, pp. 833–838.

[10] J. He, H. Tong, and J. Carbonell, "Rare category characterization," in *Proc. IEEE Int. Conf. Data Min.*, 2010, pp. 226–235.

[11] T. M. Hospedales, S. Gong, and T. Xiang, "Finding rare classes: Active learning with generative and discriminative models," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 2, pp. 374–386, Feb. 2013.

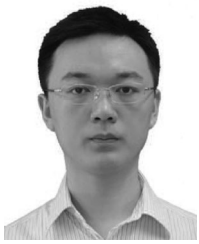
[12] G. Huang, S. Song, J. N. D. Gupta, and C. Wu, "Semi-supervised and unsupervised extreme learning machines," *IEEE Trans. Cybern.*, vol. 44, no. 12, pp. 2405–2417, Dec. 2014.

[13] H. Huang, K. Chiew, Y. Gao, Q. He, and Q. Li, "Rare category exploration," *Expert Syst. Appl.*, vol. 41, no. 9, pp. 4197–4210, 2014.

[14] H. Huang, Q. He, K. Chiew, F. Qian, and L. Ma, "Clover: A faster prior-free approach to rare-category detection," *Knowl. Inf. Syst.*, vol. 35, no. 3, pp. 713–736, 2013.

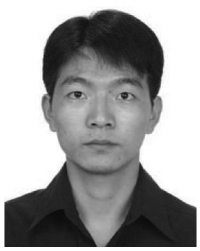
[15] K. Huang, H. Yang, I. King, and M. R. Lyu, "Learning classifier from imbalanced data based on biased minimax probability machine," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2004, pp. 558–563.

[16] R. Jiamthapthaksin, C. F. Eick, and S. Lee, "GAC-GEO: A generic agglomerative clustering framework for geo-referenced datasets," *Knowl. Inf. Syst.*, vol. 29, no. 3, pp. 597–628, 2011.
[17] M. V. Joshi, R. C. Agarwal, and V. Kumar, "Mining needles in a haystack: Classifying rare classes via two-phase rule induction," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2001, pp. 91–102.
[18] M. V. Joshi, R. C. Agarwal, and V. Kumar, "Predicting rare classes: Can boosting make any weak learner strong?" in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, 2002, pp. 297–306.
[19] M. V. Joshi, R. C. Agarwal, and V. Kumar, "Predicting rare classes: Comparing two-phase rule induction to cost-sensitive boosting," in *Proc. Eur. Conf. Princ. Data Min. Knowl. Discov.*, 2002, pp. 237–249.
[20] M. V. Joshi, V. Kumar, and R. C. Agarwal, "Evaluating boosting algorithms to classify rare classes: Comparison and improvements," in *Proc. IEEE Int. Conf. Data Min.*, 2001, pp. 257–264.
[21] G. King and L. Zeng, "Explaining rare events in international relations," *Int. Organ.*, vol. 55, no. 3, pp. 693–715, 2001.
[22] G. King and L. Zeng, "Logistic regression in rare events data," *Political Anal.*, vol. 9, pp. 137–163, 2001.
[23] X. Li and B. Liu, "Learning to classify texts using positive and unlabeled data," in *Proc. Int. Joint Conf. Artif. Intell.*, 2003, pp. 587–594.
[24] M. Lichman, "UCI machine learning repository," Irvine, CA: University of California, School of Information and Computer Science, 2013. [Online]. Available: http://archive.ics.uci.edu/ml
[25] Z. Liu, K. Chiew, L. Zhang, B. Zhang, Q. He, and R. Zimmermann, "Rare category exploration via wavelet analysis: Theory and applications," *Expert Syst. Appl.*, vol. 63, pp. 173–186, 2016.
[26] M. Maalouf and M. Siddiqi, "Weighted logistic regression for large-scale imbalanced and rare events data," *Knowl. Based Syst.*, vol. 59, pp. 142–148, 2014.
[27] M. Maalouf and T. B. Trafalis, "Robust weighted kernel logistic regression in imbalanced and rare events data," *Comput. Stat. Data Anal.*, vol. 55, no. 1, pp. 168–183, 2011.
[28] Y. Park and J. Ghost, "Ensembles of α-trees for imbalanced classification problems," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 1, pp. 131–143, Jan. 2014.
[29] M. Pavan and M. Pelillo, "Dominant sets and pairwise clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 1, pp. 167–172, Jan. 2007.
[30] D. Pelleg and A. W. Moore, "Active learning for anomaly and rare-category detection," in *Proc. Advances Neural Inf. Process. Syst.*, 2004, pp. 1073–1080.
[31] J. M. Siskind, J. J. Sherman, I. Pollak, M. P. Harper, and C. A. Bouman, "Spatial random tree grammars for modeling hierarchical structure in images with regions of arbitrary shape," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 9, pp. 1504–1519, Sep. 2007.
[32] A. Tayal, T. F. Coleman, and Y. Li, "RankRC: Large-scale nonlinear rare class ranking," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 12, pp. 3347–3359, Dec. 2015.
[33] D. Tu, L. Chen, X. Yu, and G. Chen, "Semisupervised prior free rare category detection with mixed criteria," *IEEE Trans. Cybern.*, vol. 48, no. 1, pp. 115–126, Jan. 2018.
[34] P. Vatturi and W.-K. Wong, "Category dection using hierarchical mean shift," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, 2009, pp. 847–856.
[35] S. Wang, H. Huang, Y. Gao, T. Qian, L. Hong, and Z. Peng, "Fast rare category dection using nearest centroid neighborhood," in *Proc. Asia Pac. Web Conf.*, 2016, pp. 383–394.
[36] W. Wang and Z.-H. Zhou, "A new analysis of co-training," in *Proc. Int. Conf. Mach. Learn.*, 2010, pp. 1135–1142.
[37] J. Wu, H. Xiong, and J. Chen, "COG: Local decomposition for rare class analysis," *Data Min. Knowl. Discov.*, vol. 20, no. 2, pp. 191–220, 2010.
[38] Y. Wu, R. Jin, J. Li, and X. Zhang, "Robust local community detection: On free rider effect and its elimination," *Proc. VLDB Endowment*, vol. 8, no. 7, pp. 798–809, 2015.
[39] R. Xu and D. C. Wunsch, "Clustering algorithms in biomedical research: A review," *IEEE Rev. Biomed. Eng.*, vol. 3, pp. 120–154, 2010.
[40] P. Yang, J. He, and J.-Y. Pan, "Learning complex rare categories with dual heterogeneity," in *Proc. SIAM Int. Conf. Data Min.*, 2015, pp. 523–531.
[41] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf, "Learning with local and global consistency," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2003, pp. 321–328.
[42] D. Zhou, J. He, K. S. Candan, and H. Davulcu, "Learning complex rare categories with dual heterogeneity," in *Proc. Int. Joint Conf. Artif. Intell.*, 2015, pp. 4098–4104.

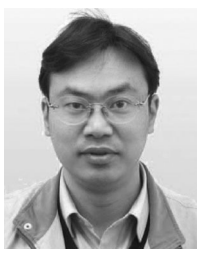


Hao Huang received the PhD degree in computer science from Zhejiang University, China, in 2012. He is currently an associate professor with the School of Computer Science, Wuhan University, China. His research interests include big data management and analytics, statistical learning, and optimization problems.

Qian Yan received the BS degree in computer science from Wuhan University, China, in 2016. He is currently working toward the MS degree in the School of Computer Science, Wuhan University, China. His research interests include data mining and statistical learning.

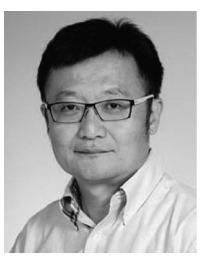


Wei Lu received the PhD degree in computer science from the Renmin University of China, in 2011. He is currently an associate professor with the Renmin University of China. His research interests include query processing in the context of spatiotemporal, cloud database systems, and applications.



Huaizhong Lin received the PhD degree in computer science from Zhejiang University, China, in 2002. He is currently an associate professor with the College of Computer Science, Zhejiang University, China. His research interests include spatial database, data mining, and information retrieval.

Yunjun Gao received the PhD degree in computer science from Zhejiang University, China, in 2008. He is currently a professor with the College of Computer Science, Zhejiang University, China. His research interests include spatial and spatio-temporal databases, metric and incomplete/uncertain data management, and spatio-textual data processing. He is a member of the ACM and the IEEE, and a senior member of the CCF.



Lei Chen received the BS degree in computer science and engineering from Tianjin University, China, in 1994, the MA degree from the Asian Institute of Technology, Thailand, in 1997, and the PhD degree in computer science from the University of Waterloo, Canada, in 2005. He is currently a professor of computing science with the Hong Kong University of Science and Technology, China. His research interests include crowdsourcing, uncertain and probabilistic databases, web data management, multimedia and time series databases, and privacy. He is a member of the IEEE.