Interactive Rare-Category-of-Interest Mining from Large Datasets*

Zhenguang Liu,*[1] Sihao Hu,*[2,4] Yifang Yin,[3] Jianhai Chen,[2]
Kevin Chiew, Luming Zhang,[2] Zetian Wu[2]
[1]Zhejiang Gongshang University
[2]Zhejiang University
[3]National University of Singapore
[4]Alibaba Group
liuzhenguang2008@gmail.com, {husihao26, chenjh919}@zju.edu.cn, idsyin@nus.edu.sg

Abstract

In the era of big data, rare category data examples are often of key importance despite their scarcity, *e.g.,* rare bird audio is usually more valuable than common bird audio. However, existing efforts on rare category mining consider only the statistical characteristics of rare category data examples, while ignoring their 'true' interestingness to the user. Moreover, current approaches are unable to support real-time user interactions due to their prohibitive computational costs for answering a single user query.

In this paper, we contribute a new model named *IRim*, which can interactively mine rare category data examples of interest over large datasets. The mining process is carried out by two steps, namely *rare category detection (RCD)* followed by *rare category exploration (RCE)*. In *RCD*, by introducing an offline phase and high-level knowledge abstractions, IRim reduces the time complexity of answering a user query from quadratic to logarithmic. In *RCE*, by proposing a collaborative-reconstruction based approach, we are able to explicitly encode both user preference and rare category characteristics. Extensive experiments on five diverse real-world datasets show that our method achieves the response time in seconds for user interactions, and outperforms state-of-the-art competitors significantly in accuracy and number of queries. As a side contribution, we construct and release two benchmark datasets which to our knowledge are the first public datasets tailored for rare category mining task.

Introduction

The big data era provides tremendous opportunities for extracting valuable knowledge from large datasets (Cao et al. 2015). In many real-world applications, it is often the case that the dataset is mixed with a great number of data examples belonging to a major category together with a small number of data examples belonging to a few rare categories, whereas the rare categories are more valuable than the major category (Pelleg and Moore 2004; He and Carbonell 2007). For example, a network access dataset may contain a big portion of normal network connections forming the major category and a small portion of intrusions forming a few rare categories, which however are usually more significant. The motivation and aims of rare category mining have enabled it to have a wide variety of applications such as fraudulent transaction detection (Zhou et al. 2018), network security bug discovering, and forest fire identification in satellite images (Mithal et al. 2017), and etc (He, Tong, and Carbonell 2010; Svenstrup, Jørgensen, and Winther 2015; Liu et al. 2017; Cheng et al. 2019).

Following the existing work (Huang et al. 2014; He and Carbonell 2007; He, Tong, and Carbonell 2010), rare category mining can be decomposed into two sequential subtasks, namely RCD (*rare category detection*) and RCE (*rare category exploration*). (1) RCD targets to detect a few data examples for an undiscovered rare category to prove its existence in the unlabeled dataset, *e.g.*, detecting an instance of a network attack. (2) If the user finds the detected rare category data examples valuable or interesting, RCE further tries to identify other similar and interesting data examples in the same rare category, *e.g.,* identifying interesting instances of the same attack type as the detected one.

Facing a few challenges such as (1) skewed category distribution, (2) the non-separability nature of interesting data examples from uninteresting data examples, and (3) the extremely limited number of labeling budget of the user, most of the existing methods (*e.g.*, (He and Carbonell 2007; Huang et al. 2013; Yu and Lam 2019; Pérez-Ortiz et al. 2019; Feuz and Cook 2017)) have been focused on the discovery of statistically significant data examples of a rare category. However, not all rare category data examples are necessarily of equal importance (Vatturi and Wong 2009). For examples, a user might be interested in only a few fighting scenes in a rare game instead of all game images; a doctor in digestive diseases may not be interested in a rare psychopath instance. This motivates us to identify rare category data examples that are of interest to a user *subjectively* besides of their statistical significance (which is *objective*).

More importantly, scrutinizing the implementations of existing methods, *e.g.,* (Vatturi and Wong 2009; Tu et al. 2018), we empirically found that current RCD approaches often have quadratic time complexities w.r.t. dataset size in an-

*The first two authors are of equal contributions. Corresponding to: Jianhai Chen(chenjh919@zju.edu.cn), Yifang Yin (idsyin@nus.edu.sg).

swering a single user query. To demonstrate their high computational costs, we plot the required time of different methods for answering an RCD query in Fig. 1. The x-axis are five real-world datasets sorted by their numbers of data examples, while the y-axis represents the required time in seconds. The methods presented are NNDM (He and Carbonell 2007), HMS (Vatturi and Wong 2009), Clover(Huang et al. 2013), and our method. All experiments are conducted on a server equipped with 40 Intel Xeon E5-2640V4 vCPUs and 96 GB RAM. We can clearly observe that as the dataset size increases, the required time increases, which confirms the challenges in dealing with large datasets. Moreover, when the dataset size exceeds 494,021 (*e.g.*, on the KDD-CUP dataset), the existing methods take hours or days to answer a single query. This is far from achieving the second-level response time required by interactive systems.

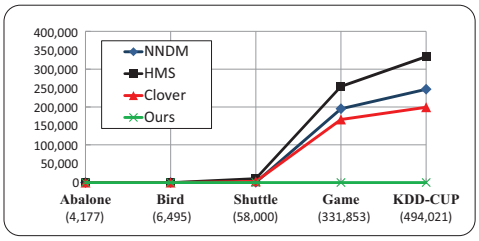


Figure 1: Required time for answering a single RCD query. Empirical results for four distinct RCD methods are presented. The x-axis are five datasets sorted by their numbers of data examples. The numbers in the parentheses show the numbers of data examples in the corresponding datasets. The y-axis denotes the required time in seconds.

To address these challenges, we propose a novel online rare category mining platform, which is able to fast interact with users and effectively identify rare category data examples of interest over large datasets. The platform is designed to jointly address the two subtasks, namely RCD and RCE. (1) *For RCD*, a logarithmic time complexity algorithm is proposed to enable real-time user interactions, which to our knowledge, is two orders of magnitude faster than state-of-the-art competitors. (2) *For RCE*, a novel collaborative-reconstruction approach is presented, which explicitly encode user interest using positive and negative contexts.

We also notice that there is a lack of benchmark datasets for rare category mining. We construct two benchmark datasets, which are obtained from practical problems, and make them publicly available. To our knowledge, they are the first public datasets tailored for rare category mining task. The datasets will be released at https://github.com/Bayi-Hu/Interactive-Rare-Category-of-Interest-Mining.

To summarize, the key contributions are:

- **Problem formulation.** We propose to discover data examples that not only fulfill rare category compactness assumption but also are interesting.
- **Methods.** We present a novel collaborative reconstruction approach for RCE, and propose to model raw big data into compact knowledge-rich abstractions for RCD.
- **Datasets and codes**. Our method sets the new state-of-the-art performance for interactive rare category mining and overall provides insights into the challenges and opportunities in this task. The implementation codes and datasets will be released upon acceptance.

Related Work

In this section, we briefly review the related work regarding the two subtasks of rare category mining, namely *rare category detection* (RCD) and *rare category mining* (RCE).

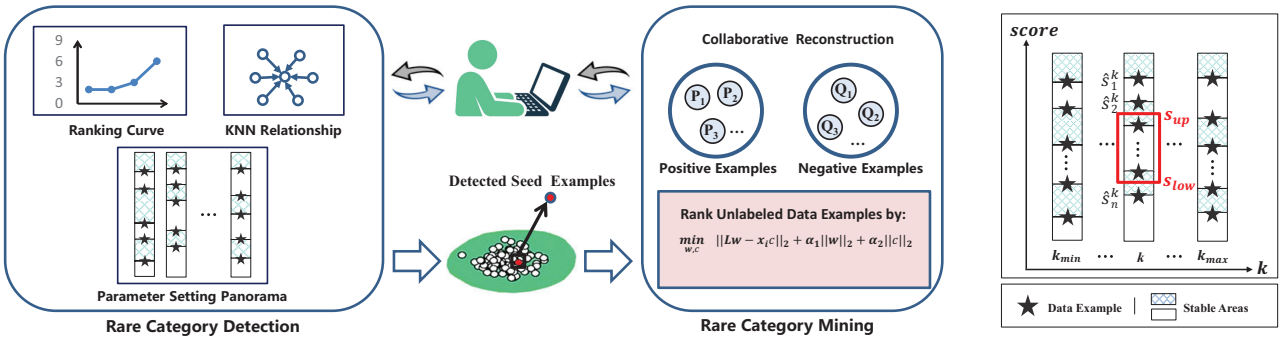
Rare Category Detection

RCD is an emerging topic in security and data mining, which targets to find a few data examples for a rare category from an unlabeled dataset. It is firstly formulated by (Pelleg and Moore 2004), where a rare category is characterized as a tiny and compact cluster of similar data examples. Following this compactness assumption, (He and Carbonell 2007) and (Huang et al. 2013) propose to rank all data examples according to user specified parameters, and return top data examples to a user. The user provides labels that indicate whether a data example belongs to an undiscovered rare category. (He and Carbonell 2009) and (Liu et al. 2014) instead resort to semi-parametric density estimation and wavelet transform respectively to rank all data examples. (Vatturi and Wong 2009) employs hierarchical mean shift clustering to detect rare categories of different scales. Recently, (Zhou et al. 2015) explores utilizing multi-view features, while (Tu et al. 2018) introduces a prior-free RCD method composed of active learning and semi-supervised hierarchical clustering. (Lin et al. 2018) further presents a user-guided RCD approach via visualization.

Rare Category Exploration RCE is a natural follow-up action of RCD, *i.e.,* after detecting a few interesting rare-category data examples called *seeds*, we may want to find more interesting data examples in the same rare category. (He, Tong, and Carbonell 2010) transforms RCE into a convex optimization problem and proposes to characterize the entire rare category as the set of data examples within a hyperball. (Huang et al. 2014) converts RCE to a local community detection problem, which keeps absorbing external data examples until there is no improvement in local community quality. (Wu, Xiong, and Chen 2010) advocates to address the imbalanced category distribution by performing clustering within each large category, producing subcategories with relatively balanced sizes. (Zhou et al. 2018) develops a self-paced framework that gradually learns the rare category oriented representation and the rare category exploration model. It is worth noting that there are research efforts that jointly address the RCD and RCE tasks, e.g., (Hospedales, Gong, and Xiang 2013) tries to solve RCD and RCE simultaneously with a generative and discriminative model.

Our Approach

In this section, we present the details of our proposed *IRim* (Interactive Rare-category-of-interest mining) system. Be-



(a) (b)

Figure 2: (a) An overview of the proposed system, which consists of RCD and RCE. RCD queries are supported by multiple high-level knowledge spaces, while RCE is conducted by collaborative reconstruction using both positive and negative data examples. (b) A toy example to illustrate the stable areas.

fore that, we first introduce the basic concepts and assumptions in rare category mining.

Concepts and Assumptions

Presented with an imbalanced dataset $X = \{\mathbf{x}_i\}_{i=1}^n$, where the category label for each data example $\mathbf{x}_i \in \mathbb{R}^d$ is lacked, we are interested in interacting with the user to effectively identify rare category data examples of her/his interest.

Assumptions. A commonly adopted assumption in rare category mining is the *compactness assumption* (Zhou et al. 2018; Vatturi and Wong 2009; Wang et al. 2016). That is, data examples of a rare category are assumed to exhibit intra-category similarity and inter-category dissimilarity.

Candidate score. Following the *compactness assumption*, we rank all the unlabeled data examples by their potentials to be from a rare category. Inspired by (He and Carbonell 2007; Lin et al. 2018), the rare category candidate score s_i^k for data example $\mathbf{x}_i$ is defined as:

$$s_i^k = \frac{d_k}{avg\{d_1, \cdots, d_{k-1}\}}, \tag{1}$$

where d_j denotes the distance between $\mathbf{x}_i$ and its j^{th} nearest neighbor, and avg represents average. The *numerator* and *denominator* measure *inter-category* and *intra-category* distances, respectively. Score s_i^k measures how likely $\mathbf{x}_i$ is from a compact rare category that consists of k similar data examples, higher s_i^k corresponds to higher likelihood. It is worth pointing out that parameter k specifies the scale (number of data examples) of the rare category to be detected, a different value of k leads to different scores for all data examples.

System Overview Given the above background knowledge, now we are ready to present the *IRim* system. Fig. 2(a) depicts the overall architecture of *IRim*, which consists of two key components: 1) RCD basing on high-level knowledge abstractions, and 2) RCE basing on collaborative reconstruction. Next, we elaborate the two key components.

Rare Category Detection (RCD) Model

RCD seeks to detect a few data examples of an undiscovered rare category hidden in an unlabeled dataset.

Conventional RCD approaches generally operate in a trial and error manner (He and Carbonell 2007; Tu et al. 2018):

1) A user selects particular values for the three parameters in the query triple $< k, s_{low}, s_{up} >$, where k represents the expected number of data examples in the rare category to be detected, s_{low} and s_{up} denote the lower and upper bounds of the rare category candidate score (Eq. 1).
2) This instantiated triple is then submitted as an RCD query to the model, which computes and returns all the data examples x_i that have candidate scores s_i^k satisfying $s_{low} \leqslant s_i^k \leqslant s_{up}$. The returned data examples are ranked by their candidate scores s_i^k.
3) The user investigates the returned data examples and provides category labels for them. If the user finds the returned data examples insignificant, she will adjust the parameters in the triple and executes another RCD query to obtain a different set of returned data examples.

This conventional framework suffers from severe limitations: (1) A good parameter setting for the triple is the key to gain insight into the data (Cao et al. 2015). However, in this framework, the user has to consistently re-submit isolated queries with different parameter settings in a trial-and-error manner. This is extremely inefficient because of the infinite number of possible parameter settings. (2) To answer a query, candidate scores $\{s_i^k\}_{i=1}^n$ of all data examples are computed from scratch using Eq. 1, which requires time consuming kNN (k nearest neighbor) calculations and thus has an $O(n^2)$ time complexity. Consequently, the user has to wait hours or days to get the result for a single RCD query.

To address these issues, we propose to divide the RCD process into *offline* and *online* phases. In the *offline* phase, we construct high-level knowledge base to avoid queries being executed from scratch, and guide the user to explore different parameter settings in a systematic way. The knowledge base mainly includes parameter setting panorama, ranking curve, and kNN relationship.

Parameter Setting Panorama Our insight is that the three parameters in the query triple $< k, s_{low}, s_{up} >$ actu-

ally fall in a 2D space since s_{low} and s_{up} are in one dimension. Fig. 2(b) visualizes the 2D space, which is spanned by parameter k in the x-axis and candidate score s_i^k in the y-axis. In the figure, k_{min} and k_{max} stand for the user-specified lower and upper bounds for k value, respectively. For a fixed $k \in [k_{min}, k_{max}]$, its parameter space corresponds to a column in Fig. 2(b). Let $< \hat{s}_1^k, \hat{s}_2^k, \cdots, \hat{s}_n^k >$ denote the descending ordered scores of all data examples under a fixed k, and $< \hat{\boldsymbol{x}}_1, \hat{\boldsymbol{x}}_2, \cdots, \hat{\boldsymbol{x}}_n >$ be their corresponding data examples. As depicted in Fig. 2(b), the parameter space of k are partitioned into $n+1$ areas (rectangles in the figure) by the sorted scores $\{\hat{s}_i^k\}_{i=1}^n$, with the $n+1$ areas being $(-\infty, \hat{s}_n^k)$, $(\hat{s}_n^k,\ \hat{s}_{n-1}^k)$, $(\hat{s}_{n-1}^k,\ \hat{s}_{n-2}^k), \cdots, (\hat{s}_2^k,\ \hat{s}_1^k)$, and $(\hat{s}_1, +\infty)$.

Interestingly, once we fix k and s_{up} in the query triple, no matter how we adjust the value of s_{low} within any one of the $n+1$ areas, the returned result, *i.e.,* all data examples that have scores s_i^k within range $(s_{low},\ s_{up})$, remains the same. This conclusion can be easily proved as below, while each of the $n+1$ areas is called a *stable area*.

Proof. Given the descending ordered scores $< \hat{s}_1^k, \hat{s}_2^k, \cdots, \hat{s}_n^k >$ for all data examples, without loss of generality, s_{up} can be assumed to fall in $(\hat{s}_{m+1}^k, \hat{s}_m^k)$, *i.e.,* $\hat{s}_{m+1}^k < s_{up} < s_m^k$. We fix k and s_{up}, and adjust s_{low} within an area $(\hat{s}_{j+1}^k,\ \hat{s}_j^k)$. For $\forall s_{low} \in (\hat{s}_{j+1}^k,\ \hat{s}_j^k)$, conditions $\hat{s}_{j+1}^k < s_{low}$ and $s_{low} < \hat{s}_j^k$ hold, so the scores that fall in range $[s_{low},\ s_{up}]$ are $\{\hat{s}_j, \hat{s}_{j-1}, \hat{s}_{j-2}, \cdots, \hat{s}_{m+1}\}$ for $\forall s_{low} \in (\hat{s}_{j+1}^k,\ \hat{s}_j^k)$. As a result, the returned data examples for the query triple $< k, s_{low}, s_{up} >$ remain the same as $\{\hat{\boldsymbol{x}}_j, \hat{\boldsymbol{x}}_{j-1}, \cdots, \hat{\boldsymbol{x}}_{m+1}\}$ for $\forall s_{low} \in (\hat{s}_{j+1}^k,\ \hat{s}_j^k)$. In other words, once we fix k and s_{up} in the query triple, no matter how we adjust the value of s_{low} within an area $(\hat{s}_{j+1}^k,\ \hat{s}_j^k)$, the returned result remains the same. It is worth pointing out that variants of this proof apply if we assume s_{up} fall on the area boundaries, *i.e.,* $s_{up} = \hat{s}_m^k, m \in [1, n]$. ■

Similarly, once we fix k and s_{low} in the query triple, no matter how we adjust the value of s_{up} within any one of the $n+1$ areas, the returned data examples remain the same.

Therefore, given a specific k, its parameter space is partitioned into $n+1$ stable areas. Within a stable area, no matter how we adjust the value of s_{low} or s_{up}, the returned data examples do not change. Traverse all possible $k \in [k_{min}, k_{max}]$, the entire parameter space is partitioned into $(n+1)*(k_{max}-k_{min}+1)$ stable areas, which are shown by the rectangles in Fig. 2(b). These stable areas form the entire *parameter setting panorama*. It is worth pointing out that: (1) despite the infinite number of possible parameter settings for the query triple, the number of possible returned results is limited to at most $(n+1)*(k_{max}-k_{min}+1)$, and (2) the limited number of stable areas offers the user an opportunity to avoid blindly trying all possible parameter settings since different s_{up} (or s_{low}) values within a stable area yield the same result. To store the *parameter setting panorama*, for each specific k, we only need to store n sorted candidate scores and their corresponding data example indices. In total, $n*(k_{max}-k_{min}+1)$ scores and indices are stored.

Ranking Curve and kNN Relationship Besides the parameter setting panorama, we further construct (1) kNN abstraction, which stores the k_{max} nearest neighbors of each data example, and (2) ranking abstraction, which maintains the score ranking of each data example under each feasible k value. These multi-view semantic spaces facilitate user's insights into the data and provide supporting evidence when the user is labeling an ambiguous data example.

Online query. For an RCD query with parameter setting $< k, s_{low}, s_{up} >$, we can easily answer it by consulting the parameter setting panorama. More specifically, as shown in Fig. 2(b), we adopt binary search to lookup the positions of s_{low} and s_{up} in the ranked score list $\{\hat{s}_i^k\}_{i=1}^n$, and then intercept the scores in the ranked list that fall in $[s_{low},\ s_{up}]$. Since for each score $\hat{s}_i^k$, its corresponding data example index is already stored in the offline phase, we can easily return all data examples that have scores within $[s_{low},\ s_{up}]$. Overall, the time complexity for answering an RCD query in our model is $O(\log(n))$, while it is $O(n^2)$ for existing methods.

Rare Category Exploration (RCE) Model

RCE is a natural follow-up action of RCD, *i.e.,* after detecting a few interesting or valuable rare-category data examples, RCE further seeks to identify other similar and interesting data examples in the same rare category. Formally,

RCE problem formulation. *Given (i) few-shot positive data examples $\{\boldsymbol{x}_p\}_{p=1}^{|P|}$ that are labeled as* interesting *data examples of a rare category C, and (ii) optionally few-shot negative data examples $\{\boldsymbol{x}_g\}_{g=1}^{|G|}$ that are labeled as* uninteresting*, RCE aims to identify other interesting data examples of C by interacting with the user.*

Positive data examples $\{\mathbf{x}_p\}_{p=1}^{|P|}$ are often referred to as *seeds*. The unique challenges of RCE (Zhou et al. 2018) come from (1) the extremely limited number of labeled data examples, (2) the fact that the support region of interesting data examples may be non-separable from that of uninteresting ones in the feature space, and 3) the subjective nature of user interest. Note that the positive and negative data examples are obtained in the RCD process, where the user provides labels for the returned data examples.

Conventional approaches (He, Tong, and Carbonell 2010; Wu, Xiong, and Chen 2010) adopt supervised methods such as imbalanced classification or convex optimization for RCE, which heavily rely on a number of labeled data examples for training. When presented with only one or a few labeled data examples, their performance degenerates greatly. To address this issue, (Huang et al. 2014) and (Liu et al. 2015) propose semi-supervised methods using local community detection and wavelet transform, respectively. These methods, however, consider only the rare category compactness characteristics and ignore the true interest of the user.

We propose to explicitly encode both *compactness assumption* and *user interest* in RCE using collaborative reconstruction. In particular, for an unlabeled data example $\boldsymbol{x}_i$, we collaboratively reconstruct $\boldsymbol{x}_i$ by using either positive or negative data examples. Then, the residuals r_i^+ and r_i^- for reconstructing $\boldsymbol{x}_i$ using either positive or negative data ex-

ample set can be computed, respectively. r_i^+ measures the distance between $\boldsymbol{x}_i$ and the positive data example set, while r_i^- measures the distance between $\boldsymbol{x}_i$ and the negative data example set. We rank all the unlabeled data examples $\boldsymbol{x}_i$ by $r_i = r_i^- - r_i^+$, and ask the user to label the top data example $\boldsymbol{x}$, *i.e.,* indicating $\boldsymbol{x}$ is *positive* (interesting) or *negative* (uninteresting). Afterwards, we update the positive or negative data example set accordingly and re-rank the unlabeled data examples for another round of interesting data example mining.

We first use all positive data examples to reconstruct $\boldsymbol{x}_i$. Formally, let $\boldsymbol{P} = \{\boldsymbol{x}_p\}_{p=1}^{|P|}$ denote the positive data example sets. We model all the positive data examples as a hull by $hull(\boldsymbol{P}) = \boldsymbol{P}\boldsymbol{w}$, where $\boldsymbol{w} = [w_1, w_2, ..., w_{|P|}]^T$ is the weight vector for all positive data examples and $\sum w_i = 1$. We reconstruct an unlabeled data example $\boldsymbol{x}_i$ using all positive data examples with the objective to minimize the reconstruction residual as follows:

$$\min_{\boldsymbol{w},c} \|\boldsymbol{P}\boldsymbol{w} - \boldsymbol{x}_i c\|_2^2 + \alpha_1\|\boldsymbol{w}\|_2 + \alpha_2\|c\|_2 \quad s.t. \sum \boldsymbol{w}_j = 1, \tag{2}$$

where c is the coefficient (scalar) for $\boldsymbol{x}_i$, while $\alpha_1\|\boldsymbol{w}\|_2$, and $\alpha_2\|c\|_2$ are the regularization terms. Here l_2-norm regularization is used to achieve a closed-form solution. Constraint $\boldsymbol{w}_j = 1$ is required by the *hull* definition (Zhu et al. 2014) and can avoid the trivial solution $\boldsymbol{w}_j = c = 0$ (Liu et al. 2016). Element $\boldsymbol{w}_j$ corresponds to the weight of the j^{th} positive data example. By minimizing the distance between $\boldsymbol{P}\boldsymbol{w}$ and $\boldsymbol{x}_i c$, different $\boldsymbol{w}_j$ will possess distinct values, thus each positive data examples makes its individual contribution in the final representation of $\boldsymbol{x}_i$.

Solution derivation for minimizing Eq. 2. To solve Eq. 2, we transform Eq. 2 into its Lagrangian form:

$$f(\boldsymbol{w}, c, \lambda) = \|\boldsymbol{P}\boldsymbol{w} - \boldsymbol{x}_j c\|_2^2 + \alpha_1\|\boldsymbol{w}\|_2 + \alpha_2\|c\|_2 + \lambda(\boldsymbol{e}\boldsymbol{w} - \boldsymbol{1}) \tag{3}$$

where $\boldsymbol{e}$ is a row vector with all elements equal to 1. Let $\boldsymbol{M} = [\boldsymbol{P}, \quad -\boldsymbol{x}_j]$, $\boldsymbol{z} = \begin{bmatrix}\boldsymbol{w}\\ c\end{bmatrix}$, $\boldsymbol{U} = \begin{bmatrix}\boldsymbol{\alpha_1 I} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{\alpha_2 I}\end{bmatrix}$ and $\boldsymbol{v} = [e, \ \boldsymbol{0}]^T$, then Eq. (3) can be deformed into:

$$f(\boldsymbol{z}, \lambda) = \boldsymbol{z}^T(\boldsymbol{M}^T\boldsymbol{M})\boldsymbol{z} + \boldsymbol{z}^T\boldsymbol{U}\boldsymbol{z} + \lambda(\boldsymbol{v}^T\boldsymbol{z} - 1) \tag{4}$$

By setting the gradients w.r.t. $\boldsymbol{z}$ and λ to zero:

$$\begin{cases} \dfrac{\partial f}{\partial \boldsymbol{z}} = (\boldsymbol{M}^T\boldsymbol{M})\boldsymbol{z} + \boldsymbol{U}\boldsymbol{z} + \dfrac{1}{2}\lambda\boldsymbol{v} = 0 \\ \dfrac{\partial f}{\partial \lambda} = \boldsymbol{v}^T\boldsymbol{z} - 1 = 0 \end{cases} \tag{5}$$

we finally arrive at the closed form solution to Eq. 2:

$$\boldsymbol{z} = \frac{(\boldsymbol{M}^T\boldsymbol{M} + \boldsymbol{U})^{-1}\boldsymbol{v}}{\boldsymbol{v}^T(\boldsymbol{M}^T\boldsymbol{M} + \boldsymbol{U})^{-1}\boldsymbol{v}}, \lambda = -\frac{2}{\boldsymbol{v}^T(\boldsymbol{M}^T\boldsymbol{M} + \boldsymbol{U})^{-1}\boldsymbol{v}} \tag{6}$$

■

After solving Eq. 2, we can obtain the optimal weight vector $\boldsymbol{w}^*$ and optimal coefficient c^*. We define the distance between $\boldsymbol{x}_i$ and positive data example set $\boldsymbol{P}$ as $r_i^+ = \|\boldsymbol{P}\boldsymbol{w}^* - \boldsymbol{x}_i c^*\|_2^2$.

Similarly, we then use all negative data examples to reconstruct $\boldsymbol{x}_i$. Let r_i^- be the distance between $\boldsymbol{x}_i$ and negative set $\boldsymbol{G}$, namely $r_i^- = \|\boldsymbol{G}\boldsymbol{\omega}^* - \boldsymbol{x}_i\varsigma^*\|_2^2$. Note $\boldsymbol{\omega}^*$ and ς^* are the optimal weight vectors obtained when reconstructing $\boldsymbol{x}_i$ using $\boldsymbol{G}$. We rank unlabeled data examples $\{\boldsymbol{x}_i\}_{i=1}^n$ according to $\{r_i^- - r_i^+\}_{i=1}^n$ and request the user to label the top data example.

Discussions. We would like to point out that we do not actually reconstruct all the unlabeled data examples. Instead, we only reconstruct the unlabeled data examples that have similar candidate scores as the *seeds* under $k = k_0$, where k_0 is the parameter value for k under which the *seeds* are detected. This is because data examples in the same rare category are within a same compact cluster, and thus should have similar candidate scores.

Experiments

In this section, we evaluate the proposed methods on five diverse and complex datasets. The datasets were obtained from real-world applications in different fields and involve different data types including images, audio, and numerical data. We seek to answer the following research questions.

- **Q1**: How is the efficiency of the proposed RCD and RCE models comparing to state-of-the-art methods? Can they achieve second-level response time on large datasets to support real-time user interactions?
- **Q2**: Can the proposed RCE model effectively capture user interest? How is its accuracy performance comparing to existing RCE approaches?
- **Q3**: Are the high-level knowledge spaces useful in reducing the number of user queries in RCD? Can they facilitate systematic and deeper insights into the data?

Next, we first present the experimental settings, followed by answering the above research questions one by one.

Experimental Settings

Datasets. Since there is a lack of benchmark datasets that are specially tailored for rare category mining task, we construct two datasets, Game and Bird, which come from two practical problems and contain images and audio data, respectively. Game consists of $331,853$ images from electronic games. The images are sampled from videos on the web[1] and have no category labels. Conventionally, in order to discover interesting rare game images, the user has to carefully sift through all the images, which is tedious and time consuming (Changpinyo, Chao, and Sha 2017; Ma and Zhang 2019; Huang, Long, and Wang 2019). Rare category mining methods enable the user to fast identify interesting rare game images out from the massive dataset. We employed the ResNet-50 model (He et al. 2015) pretrained on ImageNet to extract a 2,048 dimensional feature for each image. Bird dataset consists of 6,495 audio

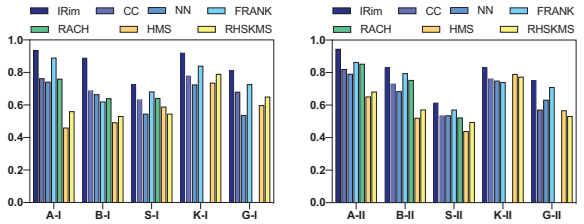
[1]Mainly from https://www.twitch.tv/directory

recordings of various birds. We extracted acoustic features from the audio utilizing the CNN network proposed in (Kahl et al. 2017). Besides Game and Bird datasets, three public datasets are also engaged in the experiments, namely Kddcup (on network intrusion), Abalone (on physical measurements of abalones), and Shuttle (on space shuttle), which are widely used in existing works (He and Carbonell 2007; Vatturi and Wong 2009; Zhou et al. 2018; Huang et al. 2013). The properties of the 5 datasets are summarized in Table 1.

Table 1: Properties of different datasets.

Dataset	Dimensions	Number of Data Exmaples
Abalone	7	4,177
Bird	512	6,495
Shuttle	9	58,000
Game	2,048	331,853
Kddcup	41	494,021

Parameter Settings. For RCD, the lower bound k_{min} of the k values is constantly set to 2 across different datasets, while the upper bound k_{max} is set to 200, 500, 200, 1,000, and 1,000 respectivley for Abalone, Bird, Shuttle, Kddcup, and Game datasets. All experiments were conducted on a server equipped with 40 Intel Xeon E5-2640V4 vCPUs and 96 GB RAM.



(a) Accuracy-I (b) Accuracy-II

Figure 3: Accuracy comparison between seven distinct RCE methods on ten different rare categories. I and II denote the first and second rare categories in the dataset.

Study of Response Time (Q1)

First, we benchmark our approach against state-of-the-art methods on all the five datasets, with a concentration on the response time required for a user interaction.

Response time in RCD. In RCD, the user interacts with the model by specifying a query triple and requesting for ranked rare category candidates that match this specification. Table 2 illustrates the average response time of five state-of-the-art methods for a single RCD query, where '-' denotes longer than 24 hours (86,400 seconds). The five methods presented are Interleave (Pelleg and Moore 2004), NNDM (He and Carbonell 2007), HMS (Vatturi and Wong 2009), Clover (Huang et al. 2013), and our IRim. (1) From the table, our first observation is that as the dataset size n increases, the response time of each method increases. (2) The second observation is that our method consistently and significantly outperform existing methods. In particular, for the Kddcup dataset with $494,021$ data examples, our method is shown to be 100 times faster than other methods. We conjecture the reasons behind these observations are: the time complexity for answering a user query in our RCD model is $O(\log(n))$ while it is usually $O(n^2)$ for existing methods. We would like to highlight that 1s response time is achieved by our method even for large dataset with more than 0.49 million data examples.

Table 2: Response time (in seconds) for an RCD query.

Methods	Abalone	Bird	Shuttle	Game	Kddcup
Interleave (Pelleg and Moore 2004)	0.324	4.341	2.420	410.663	76.960
NNDM (He and Carbonell 2007)	0.175	0.536	393.894	-	-
HMS (Vatturi and Wong 2009)	4.689	5.447	5406.318	-	-
Clover (Huang et al. 2013)	0.498	2.696	1684.139	-	-
our IRim	**0.047**	**0.081**	**0.133**	**0.568**	**0.762**

Response time in RCE. We further evaluate the efficiency of different RCE methods. The time (in seconds) for one round data example *interestingness* ranking in RCE is shown in Table 3. A total of seven methods are studied, including FRANK (Huang et al. 2013), RACH (He, Tong, and Carbonell 2010), HMS (Vatturi and Wong 2009), R-HSKMS (Tu et al. 2018), NN (nearest neighbor model), CC (cluster centroid model), and our IRim approach. The NN model ranks the unlabeled data example by their smallest distance to a positive data example, while CC model ranks the unlabeled data example by their smallest distance to the centroid of all seeds (i.e., positive data examples). For fair comparison, each time the seven methods were presented with the same set of seeds. For each RCE method, we tried different seed sets and report the averaged response time. From Table 3, we can observe that IRim significantly outperforms existing methods across different datasets. We can also see that besides our model NN performs the best.

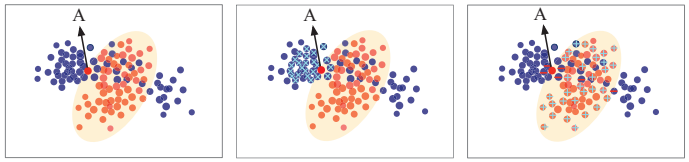
Table 3: Efficiency comparison of different RCE methods. The time unit is seconds. '-' denotes longer than 24 hours.

	Our IRim	CC	NN	FRANK	RACH	HMS	R-HSKMS
Abalone	**0.271**	0.475	0.398	0.284	1.848	0.553	0.597
Bird	**0.265**	0.571	0.545	0.294	2.32	0.629	0.753
Shuttle	**0.284**	1.088	1.073	6.427	91.549	1.078	1.104
Game	**0.577**	7.452	6.182	165.9	-	6.503	8.107
Kddcup	**0.498**	3.912	3.769	148.47	-	2.376	2.689

Accuracy Comparison with Existing RCE Approaches (Q2)

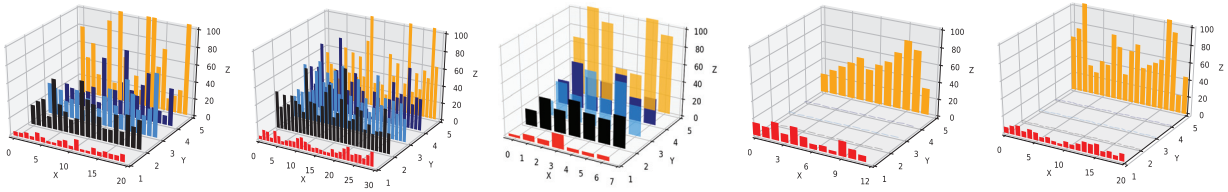
In this subsection, we compare our RCE approach against state-of-the-art methods employing the *accuracy* metric (Zhou et al. 2018; He and Carbonell 2007; Huang et al. 2014), namely the ratio of true interesting data examples to all sampled data examples.

In Fig. 3, the performance of seven different methods are presented in terms of accuracy. In the figure, datasets Abalone, Bird, Shuttle, Kddcup, and Game are abbreviated as A, B, S, K, and G, respectively. For each of the five datasets, we selected two rare categories for RCE. Thus, in total ten different rare categories are engaged in the evaluation. The two rare categories of each dataset are respectively denoted as I and II. Empirical evidences in Fig. 3 show that IRim consistently and significantly outperforms existing



(a) (b) (c)

Figure 4: Analysis on the labeling strategies of our RCE method and existing methods.



(a) Abalone (b) Bird (c) Shuttle (d) Kddcup (e) Game

Figure 5: Query number comparison on five different datasets. The first row plotted in red represents the number of queries of IRim. The second to fifth rows report the results for Clover, NNDM, HMS, and Interleave respectively.

methods on different datasets. More specifically, comparing to the state-of-the-art method, IRim achieves a 11.75% improvement in accuracy. For RACH, since it exceeds the time threshold (24 hours) on Kddcup and Game, its results on the two datasets are omitted.

We attribute the good performance of our method to its ability to actively learn from both positive and negative neighboring contexts. Fig. 4 visualizes an example scenario where the interesting data examples is non-separable from uninteresting ones in the feature space. In Fig. 4 (a), the data examples plotted in orange are interesting, while data examples plotted in blue are uninteresting. Red data example A is a given seed data example. As shown by the data examples with crosses in Fig. 4 (b), starting from A existing methods, such as CC, NN, and FRANK, blindly and consistently sample data examples in the left since they only consider the compactness assumption and ignore negative feedbacks. This greatly reduces their accuracy. In contrast, as shown in Fig. 4 (c), where the plus and minus signs represent positive and negative labels respectively, our method will adjust to sample data examples in the right side of A after few negative feedbacks in the left side. For HMS and R-HSKMS, their poor performance mainly dues to the fact that they are easily affected by noisy data examples and thus often drift to invalid centroids. For RACH, a number of labeled data examples are required, when the labeled data examples are extremely limited, its performance degenerates.

Number of Queries Comparison with Existing RCD Approaches (Q3)

In this subsection, we compare our RCD model against existing methods with respect to number of queries.

Labeling is tedious and time-consuming, a better RCD model should require less queries to detect an interesting rare category data example. Therefore, we evaluate our RCD model on the number of queries required to detect a seed. Fig. 5 demonstrate the results for detecting 20 rare categories in Abalone dataset, 30 in Bird, 7 in Shuttle, 12 in KDDcup, and 20 in Game, respectively. For all the methods, the average query number over 17 participants are reported. For NNDM, HMS, and CLOVER, since they exceed the time threshold (24 hours) for a single query on Game and Kddcup, their results are omitted for the two datasets. In Fig. 5(a), the first row plotted in red represents the number of queries of IRim for detecting the first seed of each rare category in Abalone dataset. The second to fifth rows report the results for Clover, NNDM, HMS, and Interleave respectively. Query numbers larger than 100 are truncated to 100. Figs. 5(b)–(e) follow the same convention. Empirical evidences show that our method consistently requires much less queries than other methods. We attribute this mainly to the stable areas constructed in the parameter setting panorama. Different from existing methods, for the infinite parameter settings of a stable area, we only need to try one of them. Other semantic knowledge spaces such as kNN relationship further facilitate user's insights and reduce number of queries.

Conclusions

We have proposed a novel rare-category-of-interest mining system termed *IRim*, which is able to interact with the user in real time and actively learn true interest of the user. For RCE, a collaborative-reconstruction based approach has been proposed to explicitly incorporate positive and negative contexts for user interest modeling. For RCD, a logarithmic time complexity method has been introduced. Extensive experiments demonstrate that IRim addresses user interactions within 1 second, and significantly outperforms state-of-the-

art competitors. For future work, we will investigate incorporating expert knowledge graph in rare category mining.

Acknowledgements

This paper is supported by the National Key R&D Program of China (2017YFB1401300, 2017YFB1401304), the Natural Science Foundation of Zhejiang Province, China (No. LQ19F020001), by the National Natural Science Foundation of China (No. 61902348), by the Fundamental Research Funds for the CentralUniversities, and by Singapore Ministry of Education Academic Research Fund Tier 2 under MOE's grant number MOE2018-T2-1-103.

References

Cao, L.; Wei, M.; Yang, D.; and Rundensteiner, E. A. 2015. Online outlier exploration over large datasets. In *KDD*, 89–98.

Changpinyo, S.; Chao, W.; and Sha, F. 2017. Predicting visual exemplars of unseen classes for zero-shot learning. In *ICCV 2017*, 3496–3505.

Cheng, Z.; Chang, X.; Zhu, L.; Catherine Kanjirathinkal, R.; and Kankanhalli, M. S. 2019. MMALFM: explainable recommendation by leveraging reviews and images. *ACM Trans. Inf. Syst.* 37(2):16:1–16:28.

Feuz, K. D., and Cook, D. J. 2017. Modeling skewed class distributions by reshaping the concept space. In *AAAI*, 1891–1897.

He, J., and Carbonell, J. G. 2007. Nearest-neighbor-based active learning for rare category detection. In *Advances in Neural Information Processing Systems 20, Proceedings of the Twenty-First Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 3-6, 2007*, 633–640.

He, J., and Carbonell, J. G. 2009. Prior-free rare category detection. In *SDM*, 155–163.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2015. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*.

He, J.; Tong, H.; and Carbonell, J. G. 2010. Rare category characterization. In *ICDM 2010, The 10th IEEE International Conference on Data Mining, Sydney, Australia, 14-17 December 2010*, 226–235.

Hospedales, T. M.; Gong, S.; and Xiang, T. 2013. Finding rare classes: Active learning with generative and discriminative models. *IEEE Trans. Knowl. Data Eng.* 25(2):374–386.

Huang, H.; He, Q.; Chiew, K.; Qian, F.; and Ma, L. 2013. CLOVER: a faster prior-free approach to rare-category detection. *Knowl. Inf. Syst.* 35(3):713–736.

Huang, H.; Chiew, K.; Gao, Y.; He, Q.; and Li, Q. 2014. Rare category exploration. *Expert Syst. Appl.* 41(9):4197–4210.

Huang, Y.; Long, Y.; and Wang, L. 2019. Few-shot image and sentence matching via gated visual-semantic embedding. In *AAAI 2019*, 8489–8496.

Kahl, S.; Wilhelm-Stein, T.; Hussein, H.; Klinck, H.; Kowerko, D.; Ritter, M.; and Eibl, M. 2017. Large-scale bird sound classification using convolutional neural networks. *Working notes of CLEF*.

Lin, H.; Gao, S.; Gotz, D.; Du, F.; He, J.; and Cao, N. 2018. Rclens: Interactive rare category exploration and identification. *IEEE Trans. Vis. Comput. Graph.* 24(7):2223–2237.

Liu, Z.; Chiew, K.; He, Q.; Huang, H.; and Huang, B. 2014. Prior-free rare category detection: More effective and efficient solutions. *Expert Syst. Appl.* 41(17):7691–7706.

Liu, Z.; Huang, H.; He, Q.; Chiew, K.; and Gao, Y. 2015. Rare category exploration on linear time complexity. In *DASFAA*, 37–54.

Liu, A.; Nie, W.; Gao, Y.; and Su, Y. 2016. Multi-modal clique-graph matching for view-based 3d model retrieval. *IEEE Trans. Image Processing* 25(5):2103–2116.

Liu, A.; Su, Y.; Nie, W.; and Kankanhalli, M. S. 2017. Hierarchical clustering multi-task learning for joint human action grouping and recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* 39(1):102–114.

Ma, T., and Zhang, A. 2019. Affinitynet: Semi-supervised few-shot learning for disease type prediction. In *AAAI 2019*, 1069–1076.

Mithal, V.; Nayak, G.; Khandelwal, A.; Kumar, V.; Oza, N. C.; and Nemani, R. R. 2017. RAPT: rare class prediction in absence of true labels. *IEEE Trans. Knowl. Data Eng.* 29(11):2484–2497.

Pelleg, D., and Moore, A. W. 2004. Active learning for anomaly and rare-category detection. In *NIPS*, 1073–1080.

Pérez-Ortiz, M.; Tiño, P.; Mantiuk, R.; and Hervás-Martínez, C. 2019. Exploiting synthetically generated data with semi-supervised learning for small and imbalanced datasets. In *AAAI 2019*, 4715–4722.

Svenstrup, D.; Jørgensen, H. L.; and Winther, O. 2015. Rare disease diagnosis: a review of web search, social media and large-scale data-mining approaches. *Rare Diseases* 3(1):e1083145.

Tu, D.; Chen, L.; Yu, X.; and Chen, G. 2018. Semisupervised prior free rare category detection with mixed criteria. *IEEE Trans. Cybernetics* 48(1):115–126.

Vatturi, P., and Wong, W. 2009. Category detection using hierarchical mean shift. In *SIGKDD 2009*, 847–856.

Wang, S.; Huang, H.; Gao, Y.; Qian, T.; Hong, L.; and Peng, Z. 2016. Fast rare category detection using nearest centroid neighborhood. In *APWeb*, 383–394.

Wu, J.; Xiong, H.; and Chen, J. 2010. COG: local decomposition for rare class analysis. *Data Min. Knowl. Discov.* 20(2):191–220.

Yu, Q., and Lam, W. 2019. Data augmentation based on adversarial autoencoder handling imbalance for learning to rank. In *AAAI 2019*, 411–418.

Zhou, D.; He, J.; Candan, K. S.; and Davulcu, H. 2015. MUVIR: multi-view rare category detection. In *IJCAI*, 4098–4104.

Zhou, D.; He, J.; Yang, H.; and Fan, W. 2018. SPARC: self-paced network representation for few-shot rare category characterization. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018*, 2807–2816.

Zhu, P.; Zuo, W.; Zhang, L.; Shiu, S. C.; and Zhang, D. 2014. Image set-based collaborative representation for face recognition. *IEEE Trans. Information Forensics and Security* 9(7):1120–1132.