Interactive Analytics System for Exploring Outliers

Mingrui Wei
Worcester Polytechnic Institute
Worcester, MA 01609
netwmr01@wpi.edu

Lei Cao
Massachusetts Institute of Technology
Cambridge, MA 02139
lcao@csail.mit.edu

Chris Cormier
Worcester Polytechnic Institute
Worcester, MA 01609
ccormier@wpi.edu

Hui Zheng
Worcester Polytechnic Institute
Worcester, MA 01609
hzheng@wpi.edu

Elke A. Rundensteiner
Worcester Polytechnic Institute
Worcester, MA 01609
rundenst@wpi.edu

ABSTRACT

ONION is the first system with rich interactive support for efficiently analyzing outliers. ONION features an innovative exploration model that offers an "outlier-centric panorama" into big datasets. The ONION system is composed of an offline preprocessing phase followed by an online exploration phase that supports rich classes of novel exploration operations. As our demonstration illustrates, this enables analysts to interactively explore outliers at near real-time speed even over large datasets. We demonstrate ONION's capabilities with urban planning applications use cases on the Open Street Maps dataset.

1 INTRODUCTION

This big data era provides tremendous opportunities for extracting insights from datasets via advanced analytics. Among these analytical tasks, understanding "abnormalities" in the data is one of the fundamental services for applications ranging from credit fraud prevention, financial strategy to urban planning. They all rely on effective outlier detection techniques to discover suspicious card usage and potential identity theft, identify development prospects, and to predict market changes and trade opportunities [2, 6, 8].

In this context, we focus on a well-established anomaly definition [9], called distance-based outliers, that effectively captures 'outliers' [6, 12] – data points behaving significantly differently from others in a dataset.

Limitations of Traditional One-At-A-Time Query Approach. Traditional distance-based outlier detection systems require the analyst to select a fixed set of parameter values, most notably a distance threshold r and a count threshold k [9]. Then they submit this instantiated request to the system in an attempt to detect outliers.

Similar to many other data analytical tasks, a good input parameter setting (in this case a pair of appropriate values for k and r parameters) is the key to gain insights about the data and to identify "true" outliers meaningful to the domain. To achieve this using current systems [1, 3, 4, 7], the analyst thus has to continuously re-submit individual requests with different parameter settings in a trial-and-error fashion. This repetitive process is slow, tedious, and error-prone.

Although some optimization strategies for executing such requests have been proposed [1, 3, 4, 7, 10], mining outliers according to a particular parameter setting from scratch still tends to take hours on large data [1, 5]. Clearly, this is not matching the stringent response time of seconds or less required by interactive systems.

Worst yet, each individual query would independently generate an outlier set as separate answer. Without establishing an explicit connection among these 'isolated outlier views', it is challenging to compare and contrast outlier sets produced by different queries over time. This is especially troublesome when working with a big dataset and in turn a large outlier base.

Furthermore, important insights, such as how stable the outlier status of each point is across a range of parameters; how the detected outlier set migrates across different parameter settings; or what relationships hold among different outlier points might be missed during this tedious yet expensive exploration process. For example, if some points are "more robust" outliers than others during parameter settings changes, they may warrant closer inspection. One may also use such points comparatively, to weed out inliers. This information can be critical for the analysts to interpret the characteristics of the outliers hidden in the dataset. In short, this one-at-a-time approach is neither effective nor efficient for interactive analytics.

ONION System We have extended the novel online outlier exploration platform[5], called ONION, with advance visual exploration system to addresses the above challenges.

ONION system* offers users an innovative "outlier-centric panorama" into the outliers present within the original raw dataset by establishing an interactive outlier exploration model. The ONION framework, composed of a comprehensive knowledge base called ONION model, serves as the foundation of outlier exploration operations. On top of that, the visual explorer provides easy-to-use interactions and visual result displays enabling the exploration.

The ONION knowledge base explicitly models the distribution of the outliers with respect to their associated parameter settings. This is achieved by abstracting the points of a dataset D relative

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
CIKM'17 , November 6–10, 2017, Singapore, Singapore
© 2017 Association for Computing Machinery.
ACM ISBN 978-1-4503-4918-5/17/11…$15.00
https://doi.org/10.1145/3132847.3133189

*An preview of the ONION system can be found at https://goo.gl/x62i60

to their characteristics with respect to parameter settings into a multi-dimensional space. These *abstractions* provide insights into the stability of the parameter settings. The hierarchical domination relationships in anomaly among the outlier candidates independent of particular parameter setting are also featured.

A rich class of outlier exploration operations beyond the traditional outlier detection operation now allows the analysts to understand how parameter changes would impact the captured outliers, while offering analysts a "parameter-free" approach to identifying outliers. Our system supports these rich outlier exploration operations with a milliseconds response time. Therefore it offers true interactive outlier analytics.

Furthermore, the ONION system offers users rich visual exploration interfaces that not only highlight the underlying interaction between parameters and outlierness degree, but also facilitate the analysts to visually verify the detected outliers and in turn proactively guide the outlier exploration process.

2 ONION INFRASTRUCTURE

To support instantaneous responsivenss for interactive outlier exploration, our ONION system adopts the "process once, query many" paradigm instead of the traditional one-at-a-time approach. The ONION model is constructed during the offline phase and stored in memory using compact data structures. Online interaction such as parameter exploration and comparative outlier analytics is supported by our compact in-memory ONION model. Fig. 1 illustrates the ONION model, while Fig. 2 depicts the overall architecture.

ONION Preprocessor reads the input dataset from the data source and produces a much smaller set of outlier candidates.

ONION Execution Engine consumes these outlier candidates for on-line processing. The engine features three subcomponents that together constitute the spaces of the ONION model.

ONION Space maintains the outlier candidate set sufficient to support any outlier query.

Parameter Space partitions the possibly infinite number of parameter settings into a finite number of stable regions.

Data Space maintains the domination relationships among outlier candidates. It also establishes explicit linkages between the above two spaces, called the *PD-Linkage*.

ONION Request Dispatcher accepts requests submitted by the analysts through the visual interfaces. Such requests are executed, leveraging the appropriate space. Execution results are then displayed through the ONION visualizer.

ONION Visualizer directly interacts with users and visualizes the "outlier-centric panorama". It allows the users to utilize our novel outlier analytic operations in a visual manner.

3 KEY INNOVATIONS OF ONION SYSTEM

The ONION system features several innovations that form the foundation for effective management and exploration of outliers. Our key contributions includes (1) ONION abstraction model; (2) ONION outlier knowledge management; (3) ONION interactive analytic operators; and (4) ONION query processing strategies.

3.1 ONION Model

ONION Space (O-Space): Conceptually, the O-Space encodes the outlier status of all points in the original dataset with respect to

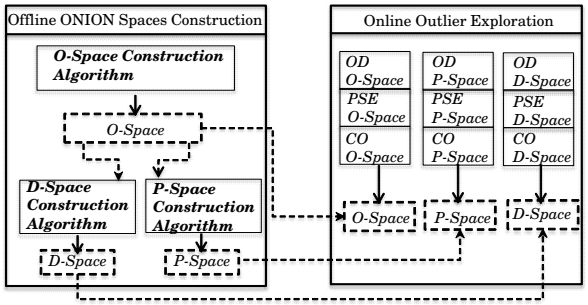


Figure 1: ONION Knowledge Base

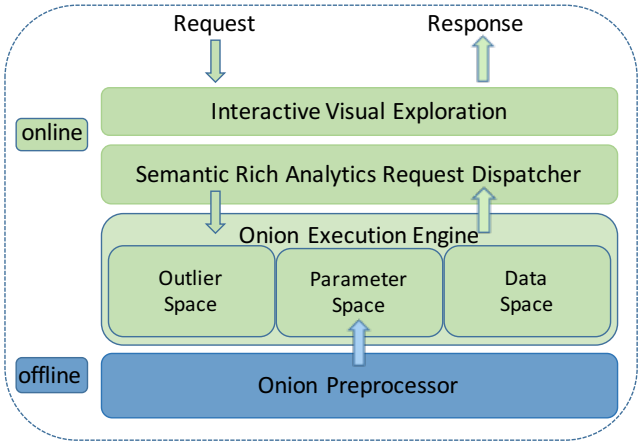


Figure 2: ONION System Architecture

all possible parameter settings. Yet, physically we designed a very compact solution as described in Section 3.2.

Parameter Space (P-Space): P-Space is based on the observation that despite the infinite number of possible parameter settings, a large range of continuous parameter settings generate the same set of outliers. P-Space offers analysts the opportunity to determine the appropriate parameter settings using a systematic methodology by revealing the influence of possible parameter adjustments.

Data Space (D-Space): D-Space leverages the key anomaly properties, namely outlier candidacy status and domination relationships. The outlier candidacy property allows us to significantly reduce the number of data points to be maintained in our system. D-Space also enables analysts to better understand key characteristics of the detected outliers ranging from their sensitivity to their stability by revealing domination relationships among outlier candidates.

3.2 ONION Knowledge Management

Data structures for the management of the three spaces in the ONION model, namely how to manage the outlierness measures and domination relationships effectively, are sketched next.

First, ONION extracts the distances to a data point's k nearest neighbors. These distances are utilized to construct the O-Space. The O-Space data structure is composed of a set of arrays. Each of the arrays contains $(k_{max} - k_{min} + 1)$ distance values, where k_{max}, k_{min} are provided to the system as parameter ranges of general interest to analysts. Therefore the space complexity of O-Space is linear in the number of outlier candidates.

Second, P-Space is constructed by further abstracting the O-Space model into a parameter space centric data structure. In P-Space, each outlier candidate *oc* will be mapped to k nodes, where k denotes the neighbor count of each *oc*. Then we organize the parameter nodes into k lists and sort the parameter nodes in each of the lists by the distance values in the nodes.

Third, the D-Space abstracts the domination relationships among data points. Some outlier candidates may demonstrate a much *stronger abnormality* than others independent of any particular parameter setting within the parameter space $\mathbb{P}$. In other words, some data points *dominate* others in abnormality. This domination relationship represents the significance of abnormality. Based on these domination relationships, outlier candidates are partitioned into several groups. Within each group, an outlier candidate either dominates or is dominated by all other outlier candidate. Each group formed a domination tree. The D-space is thus effectively represented by a domination forest composed of multiple domination trees.

3.3 ONION Analytics Interaction Operations

Our ONION system supports rich classes of analytical operators.

Outlier Detection (OD): OD corresponds to classical outlier detection requests that specify the k and r parameters.

Comparative Outlier (CO) Analytics: Leveraging the domination relationships in D-Space, the CO operation offers users a "parameter-free" approach to identifying outliers using their domain knowledge. Specifically, the CO operation helps analysts to retrieve the most extreme outliers which dominate other known outliers in the original dataset.

Outlier-Centric Parameter Space Exploration (PSE): The PSE operations enables analysts to determine the stability of a outlier set by leveraging the stable region property of the ONION model.

3.4 ONION Execution Strategies

The ONION system analyzes the original dataset and then abstracts outlier candidates into the ONION model, thereby supporting interactive outlier analytics with real-time responsiveness which is essential for interactive analytics.

Execution Strategy For Outlier Detection (OD): O-Space supports OD operation in linear time by scanning each outlier candidate. P-Space divides the two-dimensional parameter space formed by the two axes corresponding to k and r parameters into a set of disjoint stable regions. After ordering the stable regions based on the number of outliers each stable region recognizes, a binary search can then be applied to discover the outliers instead of scanning through all outlier candidates like the baseline case. Thus, any OD query can be satisfied in logarithmic in the size of outlier candidates.

Execution Strategy For Comparative Outlier (CO): D-Space organizes outlier candidates into several domination trees base on their domination relationship. CO then can be supported by conducting binary search on each domination tree. The time complexity is logarithmic to the size of each domination tree.

Execution Strategy For Parameter Space Exploration (PSE): By using the domination trees in the D-Space and inspecting the parameters at the boundary of current stable regions of each tree, PSE operation can be supported in the time complexity linear to the number of trees.

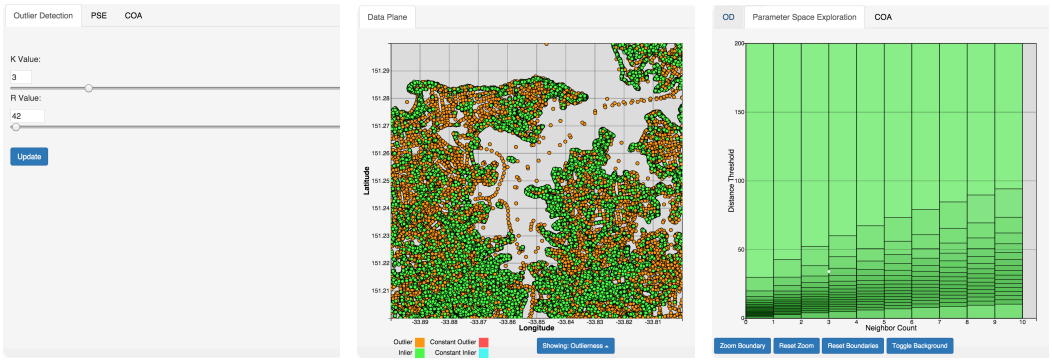
4 DEMONSTRATION DETAILS

In this section we demonstrate how an analyst explore a dataset using our ONION system. In the demonstration, we utilize the real open street map dataset [11]. The audiences can explore the dataset for urban development opportunities, environmental protection issues, etc. For example, an analyst may want to find real estate development opportunity near Sydney Opera House. Intuitively, ONION system provides comprehensive insights at three different levels of abstraction: 1) the application parameter settings, 2) the significance of the outlierness, and 3) the relationships among outliers. These may mean size of the neighborhood, quality of the development opportunity, and how the opportunity compare to other opportunities respectively.

Interactive Outlier Detection. In this demonstration, the analyst will experience the interactive outlier detection power of ONION. The analyst can test out any parameter setting as shown in Figure 3(a) and get an instantaneous visual result display as shown in Figure 3(b). The result display consists of a plot of the dataset depicting the location and distance between points in the original dataset. As users change the parameters, the dataset visualization plot display instantaneously highlights their outlier status change due to the variation of the parameter settings.

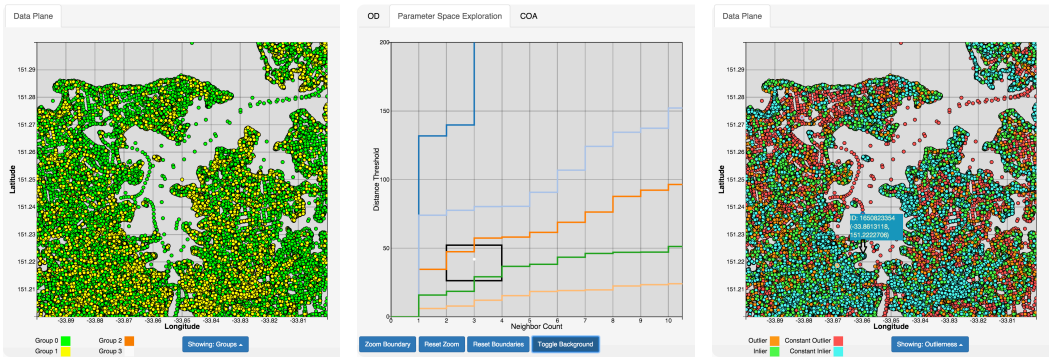
Cross Space Exploration. After some initial exploratory data analysis, users may want to gain a deeper understanding of the linkage between parameter settings and outliers. At this time, the user may explore the parameter space using the interactive capabilities of the *stable region* view. Without any guidance, the analyst will not know what is the appropriate parameter to use. Using our visual exploration tools, the analyst intuitively understands, the majority of the data points will be inlier with radius larger than 50 meters. Therefore, it is not necessary to go any larger than that as shown in Figure 3(c). The stable region view consists of a plot of the two dimensional parameter space composed of k (x-axis) and radius/distance r (y-axis) dimensions. Stable regions are colored in different shades of green to denote the density of the outliers in the region. Regions with the same outlier density share the same shade of color. The analyst may want to explore the dense area which are sensitive to parameter changes by shift-clicking the mouse on a stable region and instantaneously see the outliers in the original data set. In addition, the analyst may zoom into and zoom out of a specific region for a closer inspection of the outliers.

Comparative Outlierness Exploration. The user may wish to find out points which show different patterns comparing to other outlier candidates when the parameters change. Such points can be considered as the "outliers of outlier". Comparative outlier exploration plot visualizes these "outliers". Points in the same domination group are depicted using the same color as shown in Figure 4(a). Points in different domination groups have different outlierness pattern corresponding to the variation of the parameters as shown in Figure 4(b). Furthermore, data points in original data set are colored based on their types: 1) constant outliers, 2) constant inliers, 3)



(a) ONION Detection Analytic (b) ONION Data Plane (c) ONION Parameter Exploration

Figure 3: ONION Front End GUI



(a) ONION Outlierness Grouping (b) ONION Boundary Comparison (c) ONION Outlierness Catagories

Figure 4: ONION Front End GUI

outlier candidate which are outliers according to the current parameter setting, and 4) outlier candidate which are inliers according to the current parameter setting as shown in Figure 4(c).

5 CONCLUSION

This demonstration presents the key innovations of the ONION system, a novel outlier analytics tool that not only supports real time outlier detection but also rich analytics semantics for gaining deep insights into the outlier space. Its features of rich analytics semantics, parameter recommendation, and instantaneous responsiveness significantly reduce the effort of the data analysts to detect true outliers. We demonstrated ONION's capabilities using the real life open street map dataset to explore potential urban development opportunities.

REFERENCES

[1] F. Angiulli and F. Fassetti. Dolphin: An efficient algorithm for mining distance-based outliers in very large datasets. *TKDD*, 3(1), 2009.
[2] V. Barnet and T. Lewis. Outliers in statistical data. *International Journal of Forecasting*, 12(1):175–176, 1996.
[3] S. Bay and M. Schwabacher. Mining distance-based outliers in near linear time with randomization and a simple pruning rule. In *KDD*, pages 29–38, 2003.
[4] K. Bhaduri, B. L. Matthews, and C. Giannella. Algorithms for speeding up distance-based outlier detection. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, CA, USA, August 21-24, 2011*, pages 859–867, 2011.
[5] L. Cao, M. Wei, D. Yang, and E. A. Rundensteiner. Online outlier exploration over large datasets. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, NSW, Australia, August 10-13, 2015*, pages 89–98, 2015.
[6] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection. *ACM Computing Surveys*, 41(3):1–58, 2009.
[7] A. Ghoting, S. Parthasarathy, and M. E. Otey. Fast mining of distance-based outliers in high-dimensional datasets. *Data Min. Knowl. Discov.*, 16(3):349–364, 2008.
[8] D. M. Hawkins. *Identification of Outliers*. Springer, 1980.
[9] E. M. Knorr and R. T. Ng. Algorithms for mining distance-based outliers in large datasets. In *VLDB*, pages 392–403, 1998.
[10] H.-P. Kriegel, P. Kröger, and A. Zimek. Outlier detection techniques. In *Tutorial at the 16th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD), Washington, DC*, 2010.
[11] OpenStreetMap contributors. Planet dump retrieved from https://planet.osm.org . https://www.openstreetmap.org, 2017.
[12] G. H. Orair, C. H. C. Teixeira, Y. Wang, W. M. Jr., and S. Parthasarathy. Distance-based outlier detection: Consolidation and renewed bearing. *PVLDB*, 3(2):1469–1480, 2010.