

Distance-based Outlier Detection: Consolidation and Renewed Bearing

Gustavo. H. Orair, Carlos H. C. Teixeira, Wagner Meira Jr.,
Ye Wang, Srinivasan Parthasarathy

September 15, 2010



Table of contents

Introduction

- Outlier Detection

- Distance-based Algorithms

Optimizations and Taxonomy

- Pruning Strategies

- Ranking Strategies

The DIODE framework

Experiments

- Consolidation and Renewed Bearing

Renewed Bearing and Conclusions

References

Why is Outlier Detection important?

- ▶ Stephen Hawking defines an outlier as “an observation which deviates so much from the other observations as to arouse suspicions that it was generated by a different mechanism”.
- ▶ In other words, one can describe it as a pattern in the data that does not conform to the expected behavior.
- ▶ Applications in data cleaning, financial fraud detection, network intrusion detection, medical diagnostic, etc.

The Advantages of Distance-based Algorithms

- ▶ There are many methods to detect outliers – statistical methods, geometric methods and density/distance based approaches.
- ▶ Distance-based techniques are popular due to their scalability and non-parametric nature (they do not need any knowledge about data distribution).
- ▶ The main requirement for employing them is to define a distance (similarity) function among objects.

Motivation for this Study

Over the last decade several algorithms/optimizations have been proposed in distance-based outlier detection. Often they have been evaluated in isolation. In this study we try to ask and answer the following questions:

- ▶ What are the best/most frequently used optimizations? Can we characterize them (say in a taxonomy)?
- ▶ Can these optimizations be combined (i.e. do they interact with each other limiting effectiveness)?
- ▶ What are the current bottlenecks of these approaches?
- ▶ What lessons can we learn moving forward?

Definition and Background

The kNN definition of an outlier

“Outliers are the n objects presenting the highest distance values to their respective k^{th} nearest neighbor”

- ▶ In other words, the outlier score of a object pt is given by the distance between pt and its k^{th} nearest neighbor.
- ▶ The above is one definition, used in this study, it can be generalized to use average or density-based.
- ▶ Several distance-based algorithms were proposed based on this definition, such as Ramaswamy's techniques[1], ORCA[2], RBRP[3], among others.

Popular Optimizations in Existing Literature

- ▶ Approximate Nearest Neighbor Searching (ANNS) – Used almost universally now.
- ▶ Partition-based Pruning Strategies
 1. Partition Pruning while Searching for Neighbors (PPSN)
 2. Partition Pruning while Searching for Outliers (PPSO)
- ▶ Ranking-based Strategies
 1. Rank Object Candidates while searching for Neighbors (ROCN)
 2. Rank Object Candidates while searching for Outliers (ROCO)

Notation

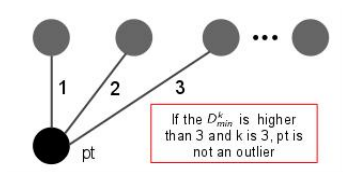
- ▶ Let D_{min}^k be the the shortest distance between an object in the current top outlier set and its k^{th} nearest neighbor (the lowest outlier score in the current outliers set);
- ▶ Let $D^k(pt)$ be the distance of an object pt to its current k^{th} nearest neighbor;
- ▶ Let $MINDIST(pt, P)$ be the lower bound of the shortest distance between pt and an object in partition P ;
- ▶ Let $MINDIST(P, H)$ be the lower bound of the distance between any two objects, pt and ht such as pt and ht belong to P and H , respectively;
- ▶ Let $MAXDIST(P)$ be the upper bound of the distance between any two objects in partition P

Approximate Nearest Neighbor Searching (ANNS)

ANNS definition

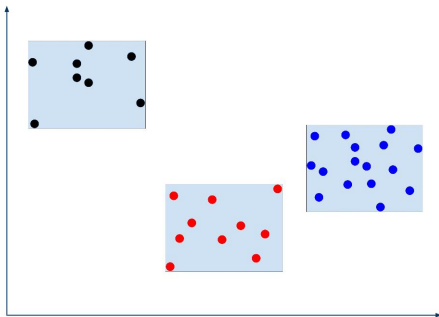
“An object pt can be disregarded as a candidate outlier if $D^k(pt) < D_{min}^k$ while computing its $D^k(pt)$ ”

- ▶ Intuition : If the current distance of pt to its k^{th} nearest neighbor is lower than the lowest outlier score of a point in the current outlier set, pt is not an outlier.
- ▶ Example: suppose that the number of neighbors considered (k) is 3 and the D_{min}^k is 4.



The Case for Partitioning

- ▶ We partition the database and get summary statistics for each partition.
- ▶ For example, MBR (Minimum Bound Rectangle) structures are useful to estimate distances among objects and then used to prune and rank.

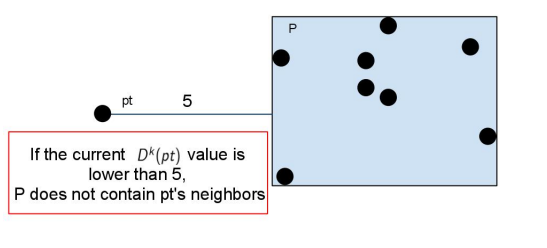


Partition Pruning while Searching for Neighbors(PPSN)

PPSN definition

“The objects in P can be disregarded as pt 's neighbors if, while computing its $D^k(pt)$, $D^k(pt) < MINDIST(pt, P)$ ”

- ▶ Intuition : As the current distance of pt to its k^{th} nearest neighbor is lower than the minimum distance among pt and P 's objects, then P does not contain pt 's neighbors.
- ▶ Example : suppose that the current $D^k(pt)$ value is 3.



Partition Pruning while Searching for Outliers (PPSO)

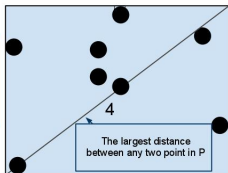
PPSO definition

“The objects in P can be disregarded as outliers if

$$D_{min}^k > MAXDIST(P) \text{ and } |P| \geq k$$

- ▶ An entire partition can be pruned if it is dense enough – the upper bound of the outlier score of any point in the partition is less than the minimum outlier score of the current outlier set.
- ▶ Example : suppose that the current D_{min}^k value is 5 and the number of neighbors considered (k) is 3.

If the D_{min}^k is 5 and k is 3, this partition does not contain outliers



Rank Objects Candidates while searching for Neighbors (ROCN)

ROCN heuristic

While searching for the neighbors of an object, search nearby partitions first, as they are more likely to contain neighbors.

- ▶ The objective here is to decrease the $D^k(pt)$ value quickly and, thereby improve the effectiveness of the ANNS pruning rule.
- ▶ Let pt an object in partition P . We begin the search of pt 's neighbors in P and, subsequently, examine the closest partition H according to the function $\text{MINDIST}(P, H)$.

Rank Objects Candidates while searching for Outliers (ROCO)

ROCO heuristic

Process objects with higher score (i.e, increase quickly the D_{min}^k value) and, as consequence, improve the efficiency of ANNS pruning rule.

- ▶ ROCO based on Partitions: low-density regions tend to contain higher-score objects.
- ▶ Ranking objects within a partition are done based on their distances from the mean value of the objects in the partition.
- ▶ This is a new idea to our work.

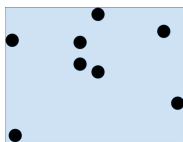


Figure: Low density partition

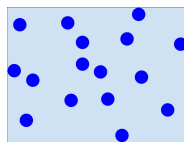


Figure: High density partition

Placing Existing Algorithms within our Taxonomy

Table: Summary of distance-based outlier detection strategies

Method	Clustering	ANNS	PPSN	PPSO	ROCN	ROCO	Algorithms
1	No	No	No	No	No	No	Nested Loop [1]
2	No	Yes	No	No	No	No	Enhanced NL with ANNS
3	No	Yes	No	No	Yes	No	ORCA [2]
4	No	Yes	No	No	No	Yes	Wu and Jermaine [4]
5	Yes	Yes	No	No	Yes	No	RBRP [5]
6	Yes	Yes	Yes	No	Yes	No	Index-based join algorithm [1]
7	Yes	Yes	Yes	Yes	Yes	No	Partition-based algorithm [1]
8	Yes	Yes	Yes	Yes	No	No	MIRO [6]
9	Yes	Yes	No	No	No	Yes	Our approach presented in this paper
10	Yes	Yes	No	No	Yes	Yes	Our approach + ROCN

It is important to note that ANNS is used by all recent proposals to solve this problem.

The DIODE framework¹

- ▶ The Distance-based Outlier DEtection framework (DIODE) provides a unified framework to implement and analyze distance based outlier detection algorithms.
- ▶ DIODE supports the evaluation of each dimension in isolation and in combination to study interaction effects.
- ▶ DIODE supports ANNS, PPSO, PPSN, ROCN and ROCO.
- ▶ DIODE applies a scalable recursive partitioning algorithm for computing partitions and various summary statistics.

¹www.speed.dcc.ufmg.br/Speed/DIODE/

Experimental Environment and Datasets

- ▶ The experiments were performed on an AMD Athlon 64 with 2 GB RAM.
- ▶ In experiments reported in this talk we looked for the top 30 outliers ($n = 30$), using $k = 5$ as the number of nearest neighbors. See paper for additional results.
- ▶ The size of any partition is limited to 16000 points.
- ▶ All timing results were averaged over 10 runs.
- ▶ Dataset details (more datasets in paper):

Table: Database descriptions

Database	# Objects	# Attributes	
		# Cont	# Cat
Government Auctions	268,170	6	7
KddCup1999	4,898,430	34	8
Forest Covertype	581,012	10	45

Factorial Design

- ▶ The factorial design model tries to explain the impact for each factor and possible interactions of factors.
- ▶ The Q_i values shows the value of the factors and interactions on execution time by the factor i w.r.t. the average time (Q_0).
- ▶ The $SS_i(\%)$ values shows the percentage of variation explained by the factor i .

Table: Factors employed on Experimental Design

Factor	Description
A	ROCO
B	ROCN
C	PPSO
D	PPSN

KDDCup Dataset

Table: Running time

ROCO (A)	ROCN (B)	PPSO (C)	PPSN (D)	Time (s)
-1	-1	-1	-1	1317.15
-1	-1	+1	-1	1249.95
-1	+1	-1	-1	1249.49
+1	-1	-1	-1	748.24
+1	-1	+1	+1	179.5
+1	+1	+1	+1	178.4

- ▶ ROCN, PPSO does not provide significant gains alone.
- ▶ PPSN provides significant gains in isolation. It reduces the number of comparisons by a factor of 20 (from 2 billion to 100 million), explaining more than 70% of the total variation.
- ▶ These results can be explained by the fact that KDDCup has dense clusters with many duplicates.
- ▶ Best configuration is when all optimizations are applied (speedup 7-8).

Forest Covertypes

Table: Running time

ROCO (A)	ROCN (B)	PPSO (C)	PPSN (D)	Time (s)
-1	-1	-1	-1	327.03
-1	-1	-1	+1	232.7
-1	-1	+1	-1	329.47
-1	+1	-1	-1	262.31
+1	-1	-1	-1	280.83
+1	+1	-1	+1	<u>169.76</u>

- ▶ The most effective strategy for reducing the execution time was PPSN (in isolation, the execution time is 232.7s). This factor explains 52% of the total variation.
- ▶ ROCO offers less benefits in isolation, it does not have a significant interaction with PPSN.
- ▶ PPSO was found to be ineffective in the context of this dataset.
- ▶ The best configuration was the combination of ROCN, ROCO and PPSN (speedup around 2).

Government Auctions

Table: Running time

ROCO (A)	ROCN (B)	PPSO (C)	PPSN (D)	Time (s)
-1	-1	-1	-1	57.88
-1	-1	-1	+1	51.23
-1	-1	+1	-1	55.91
-1	+1	-1	-1	44.01
+1	-1	-1	-1	11.89
+1	+1	-1	+1	<u>9.6</u>

- ▶ The ROCO is the most effective optimization in isolation decreasing running time from 57 to 12 seconds. This factor explains 68% of the total variation.
- ▶ ROCN provides gains in isolation, but, when applied together with ROCO, it is not as effective.
- ▶ PPSN was not effective in this dataset because one rarely had to go outside a partition when searching for k neighbors.
- ▶ The best average time (9.6 seconds) is associated with the PPSN, ROCO and ROCN (speedup of 6).

Consolidation of Results

- ▶ ANNS is very effective and should always be used as a basic component for distance-based outlier detection
- ▶ PPSN is a very important optimization strategy among the ones we evaluated.
- ▶ ROCO, a new idea in this work helps improve the effectiveness of the ANNS rule and interacts least with PPSN. ROCN is also useful although it interacts with PPSN.
- ▶ PPSO is not very effective. There are few opportunities for applying it and the interactions with other (more recent) optimizations make it ineffective.
- ▶ No single optimization used in isolation is always the best. However a combination of PPSN, ROCO and ROCN is generally effective.

Renewed Bearing

- ▶ The application of a different ranking strategy than the simple one we used for ROCO (e.g., based on Locality Sensitive Hashing) could improve the performance.
- ▶ Study of the relation among the clustering preprocessing phase and the optimizations/factors.
- ▶ PPSO is not very useful but improved mechanisms to support PPSN and ROCN may be investigated.
- ▶ Parallel implementations may be of interest.

Conclusion

- ▶ We identified a region in the design space that has been relatively unexplored and propose a strategy for ranking outliers candidates.
- ▶ We presented a unified framework for distance-based outlier detection algorithm (DIODE) with several state-of-art optimizations.
- ▶ We conducted a factorial design experiment implemented on the DIODE framework and found that no single optimization always performs the best.

References



S. Ramaswamy, R. Rastogi, and K. Shim, "Efficient algorithms for mining outliers from large data sets," in *SIGMOD '00: Proc. ACM SIGMOD Int. Conf. on Management of data*. New York, NY, USA: ACM Press, 2000, pp. 427–438.



S. D. Bay and M. Schwabacher, "Mining distance-based outliers in near linear time with randomization and a simple pruning rule," in *9th ACM SIGKDD Int. Conf. on Knowledge Discovery on Data Mining*, 2003.



A. Ghoting, S. Parthasarathy, and M. E. Otey, "Fast mining of distance-based outliers in high-dimensional datasets," *6th SIAM Int. Conf. on Data Mining*, April 2005.



M. Wu and C. Jermaine, "A bayesian method for guessing the extreme values in a data set?" in *VLDB '07: Proceedings of the 33rd international conference on Very large data bases*. VLDB Endowment, 2007, pp. 471–482.



A. Ghoting, S. Parthasarathy, and M. E. Otey, "Fast mining of distance-based outliers in high-dimensional datasets," *Data Min. Knowl. Discov.*, vol. 16, no. 3, pp. 349–364, 2008.



N. Vu and V. Gopalkrishnan, "Efficient Pruning Schemes for Distance-Based Outlier Detection," in *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases: Part II*. Springer, 2009, p. 175.

Thank you!



carlos@dcc.ufmg.br