Introduction

Motivation: Discovering abnormal phenomena in data streams has become more critical. There are many applications such as:

- Credit card fraud detection
  Based on abnormal transactions
- Network intrusion prevention
  Based on abnormal log activities
- Stock investment tactical planning
  Based on abnormal changes in price
- Event detection in sensor network
  Based on abnormal measuring values

In this poster, we study the problem of distance-based outlier detection for data streams and some of the most recent algorithms proposed by the scientific community.

Distance Based Outlier

- Data point \(o\) is a distance-based outlier if less than \(k\) neighbors in a window lie within distance \(R\) from \(o\).

\[K=3, W=16\] (window size)

- Current window: 1\(8\) (dash lines)
  \(o9\) has 4 neighbors \((o5, o10, o14, o15)\) \(\rightarrow\) inlier
  \(o11\) has 4 neighbors \(\rightarrow\) inlier
- Consider window \(W2\)
  \(o9\) is an inlier, neighbors set = \((o10, o14, o15)\)
  \(o11\) is an outlier neighbors set = \((o13)\):
  \(o3, o4, o6\) expired

- Succeeding neighbor of point \(o\) is a neighbor that comes after \(o\).
- Preceding neighbor of point \(o\) is a neighbor that comes before \(o\).
- Safe inlier is a point that never becomes an outlier: has at least \(k\) succeeding neighbors
- Naive approach: For each data point, store a list of neighbors.

Algorithms

- Exact Storm [1]
  For each data point, store number of succeeding neighbors and list of preceding neighbors

\[
\begin{array}{c|c|c|c}
\text{Preceding neighbor 1} & \text{Preceding neighbor 2} & \text{Sum of succeeding neighbors} \\
\hline
o1 & o2 & o3 \\
\hline
\end{array}
\]

- Abstract-C [2]
  For each point, store total number of neighbors in every window that it participates in. For red point: \(<W1: 3, W2: 3>\)

- Event Based Approach (LUE) [3]
  Use a priority queue to store un-safe inliers. Poll from event queue when a data object expires.

Algorithms (cont.)

- Micro-Cluster Based Algorithm [3]
  A data point in each cluster is certainly an inlier

- Lifespan - Aware Minimal Probing Algorithm (MESI) [4]
  Only find enough \(k\) neighbors, instead of all neighbors, for each data point, to prove it is an inlier or not.

Results

- Windows PC with Intel core i5, 1.8GHz, 4GB RAM
- Datasets:
  - TAO (Tropical Atmosphere Ocean), real-world, outlier rate = 0.01%
  - Gaussian, synthetic, outlier rate = 0.1%
- CPU time (s)
  - Exact Storm - Micro: large memory consumption
  - Abstract-C - Micro: large memory consumption
  - Event Based Approach: Micro: large memory consumption

Conclusion and Future Work

- Observations:
  - Micro-cluster based algorithm is the best at CPU time and memory consumption.
  - MESI, Abstract-C are comparable when window size is small.
- Future work:
  - Complete evaluation with more real-world datasets and larger parameter ranges
  - Identify the characteristics of real-world outliers and propose efficient methods

References