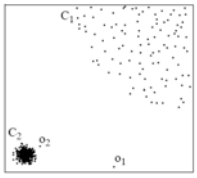
Unifying Global and Local Outlier Detection Using Commute Time Distance

Author: Nguyen Lu Dang Khoa, khoa@it.usyd.edu.au
Supervisor: Associate Professor Sanjay Chawla
School of Information Technologies

Aim:

- To find outliers using 'commute time distance' which is computed from a random walk of a weighted graph derived derived from the data similarity matrix.
- To avoid the O(n^3) direct computation of computation of commute time distance, a distance, a subspace approximation is combined with a pruning technique.

Introduction:



- Global and local outliers
- Statistical and distance based approaches can only find global outliers. outliers.
- Density based method (LOF) can find local outliers. Disadvantages of LOF are its are its computational time O(n^3) and global global outliers may be ranked lower than than local ones.
- We present a new method to find outliers outliers using a measure called 'commute 'commute time distance' (CTD). Unlike Euclidian distance, CTD between two nodes captures both the distance between between them and their neighborhood neighborhood densities. Using CTD, we can capture both global and local outliers outliers using a distance based method. method. So we can unify both global and and local outlier detection using the CTD CTD measure.

Background:

- Similarity graph:
  - ε-neighborhood graph
  - Fully connected graph
  - *k*-nearest neighbor graph

- Random walk: a sequence of nodes on a on a graph visited by a random walker, and described by a Markov chain:
  - $P(s(t+1)=j|s(t)=i) = a_{ij}/a_i = p_{ij}$ where $a_i = \sum_j a_{ij}$
  - The rule of walk: $\pi(t+1) = P^T\pi(t)$
    $\pi(t) = (P^T)^t\pi(0)$
  $\pi(t) = [\pi_1(t), \pi_2(t), \ldots, \pi_n(t)]^T$: the state probability distribution at time *t*

- Commute time distance:
  - Access time *m(i, j)*: the expected number of steps a random walker starting at *i* will take to reach *j* for the the first time:

$$m(i,j) = \begin{cases} 0 \text{ if } i = j \\ 1 + \sum_{k \in N(i)} p_{ik} \times m(k,j) \text{ otherwise} \end{cases}$$

- Commute time: the expected number of steps that a random walker walker starting at *i* will take to reach *j* reach *j* once and go back to *i* for the the first time:

$$n(i,j) = m(i,j) + m(j,i)$$

- Compute direct from the pseudo-pseudo-inverse of the graph Laplacian matrix L^+:

$$n(i,j) = V_G(l_{ii}^+ + l_{jj}^+ - 2l_{ij}^+)$$

$$n(i,j) = V_G(e_i - e_j)^T L^+ (e_i - e_j)$$

$V_G = \sum_i d_{ii}$: graph volume
e_i: *i*-th column of identity matrix I

- Complexity: space: O(n^2)
  time: O(n^3)

Subspace Approximation:

$$\tilde{n}(i,j) = V_G(\tilde{l}_{ii}^+ + \tilde{l}_{jj}^+ - 2\tilde{l}_{ij}^+)$$

$$\text{where } \tilde{l}_{ij}^+ = \sum_{k=1}^{m} \lambda_k^+ v_{ik} v_{jk}$$

λ_k^+ : *m* largest eigenvalues of L^+ ($m << n$)
v_{ij} : entries of matrix $\tilde{V}$
$\tilde{V}$: matrix containing *m* largest eigenvectors of L^+

- Compute $\tilde{n}(i,j)$'on demand': pruning pruning technique reduces the computation significantly
- Bound of the approximation:

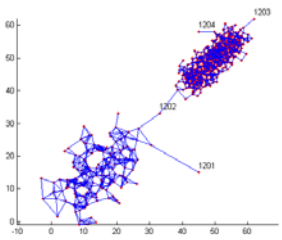
$$\left\| n(i,j) - \tilde{n}(i,j) \right\| \le V_G \sum_{i=1}^{m} \lambda_i^+$$

Algorithm:

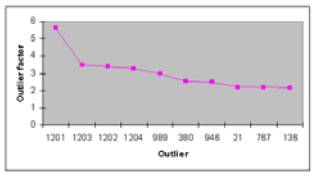
- Construct the mutual *k*-nearest neighbor graph from the dataset
- Compute the graph Laplacian matrix *L* matrix *L*
- Compute the matrix $\tilde{V}$ (the *m* smallest smallest eigenvectors with nonzero eigenvalues of *L*)
- Find top *N* outliers using the approximate commute time distance based technique with pruning rule

Preliminary Results:

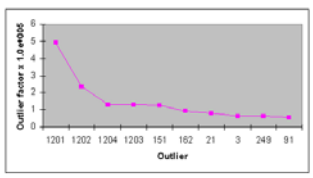
- Accuracy and ranking:



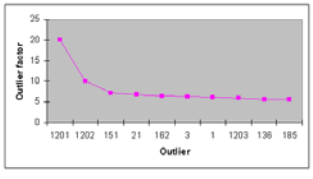
k nearest neighbor graph



LOF

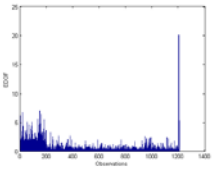


CDOF



EDOF

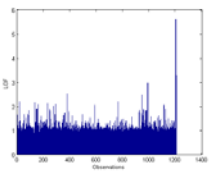
- Distance and density capture:



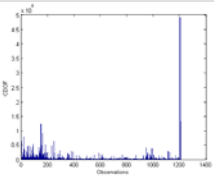
EDOF

	EDOF	LOF	CDOF
EDOF	1	0.4634	0.9805
LOF	0.4634	1	0.5116
CDOF	0.9805	0.5116	1

Spearman rank tests for LOF, EDOF, and CDOF



LOF



CDOF

Conclusions:

- Unify the detection of both global and and local outliers using CTD
  - CTD captures both distances between observations and their local neighborhood densities
- The ability to rank outliers globally
- Approximate CTD and the use of pruning rule
  - Accelerate the algorithm significantly
  - Still maintain the accuracy of the the detection

References:

[1] S. D. Bay and M. Schwabacher. Mining distance-distance-based outliers in near linear time with randomization and a simple pruning rule. In SIGKDD SIGKDD 2003.
[2] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander. LOF: Identifying density-based local outliers. outliers. In SIGMOD 2000.
[3] F. Fouss and J.-M. Renders. Random-walk computation of similarities between nodes of a graph graph with application to collaborative recommendation. In IEEE TKDE 2007.
[4] E. M. Knorr and R. T. Ng. Algorithms for mining distance based outliers in large datasets. In VLDB 1998.
[5] M. Saerens, A. Pirotte, and F. Fouss. Computing Computing dissimilarities between nodes of a graph: graph: Application to collaborative filtering. Technical Technical report, IAG, Universite Catholique de Louvain, 2004.

The University of Sydney