ADCN: An Anisotropic Density-Based Clustering Algorithm for Discovering Spatial Point Patterns with Noise

Gengchen Mai[1], Krzysztof Janowicz[1], Yingjie Hu[2], Song Gao[1]

[1]STKO Lab, Department of Geography, University of California, Santa Barbara
[2]Department of Geography, University of Tennessee, Knoxville

OUTLINE

- Introduction & motivation
- Research Question
- Clustering Algorithm
- Evaluation of Clustering Quality & Efficiency
- Conclusion

CLUSTERING ANALYSIS

- Cluster analysis is a key component of modern knowledge discovery. It is a technique used for reducing dimensionality, identifying prototypes, cleansing noise, determining core regions, or segmentation.
- A wide range of clustering algorithms, such as *DBSCAN*, *OPTICS*, *K-means*, and *Mean Shift*, have been proposed and implemented over the last decades.
- Among many clustering algorithms, DBSCAN or Density based clustering is a popular one, because:

K-means	DBSCAN
Clustering with circular shape	Clustering with arbitrary shape
Can not detect noise	Robust to noise
Require # of clusters beforehand	Do not require prior knowledge

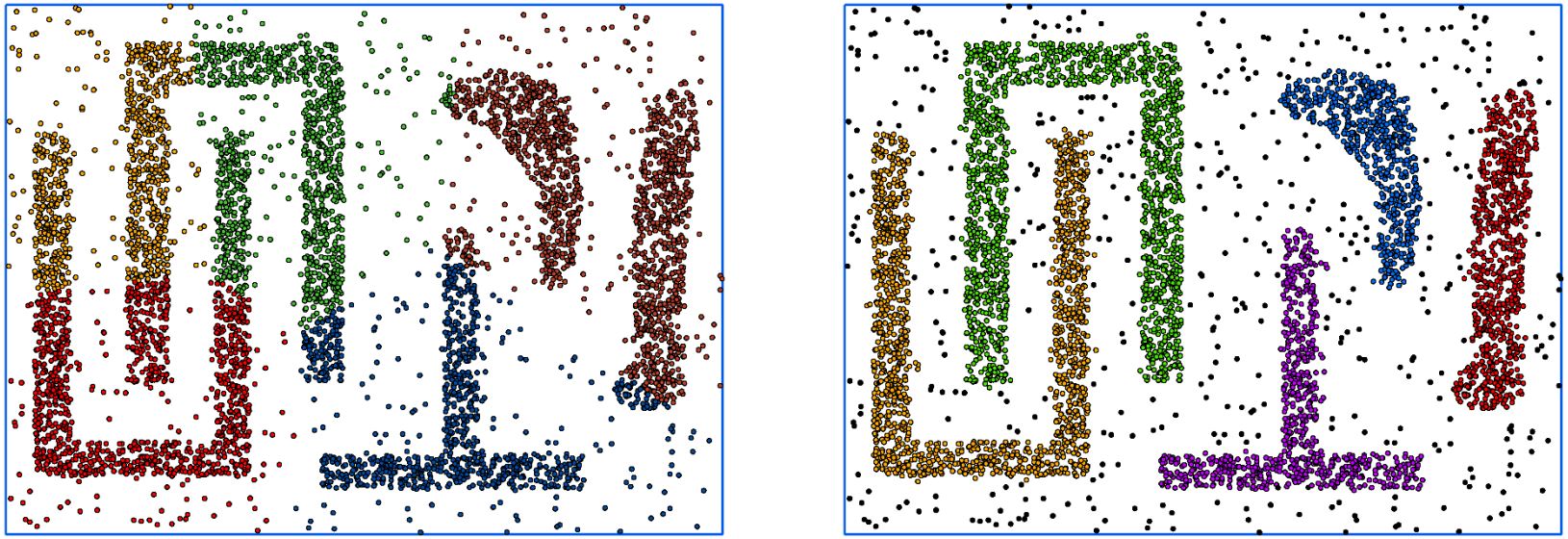
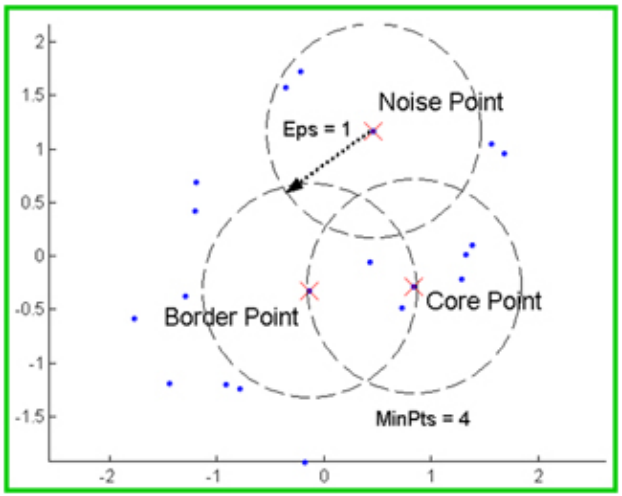


Fig. 1. Clustering result comparison between K-means and DBSCAN.

DBSCAN ALGORITHM

- Parameter:
  - **Eps**: the radius of circle neighbourhood of Pi
  - **MinPts**: minimum number of points in circle neighborhood of Pi



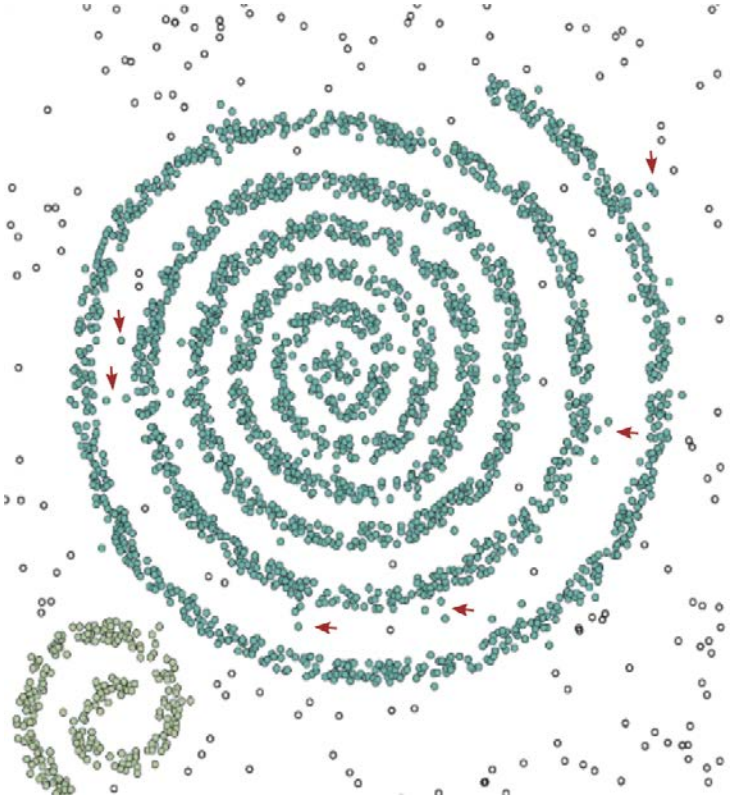


Fig. 2. Spiral cases: DBSCAN fails and misclassify the inter-spiral points as parts of the spiral

Disadvantage

- Can not deal with spatial point patterns with **varied density**
- Assume **isotropic** second-order effects among spatial objects which implies that the magnitude of similarity and interaction between two objects mostly depends on their distance.

A lot of works has been done to fix the 1st problem. In this work, we will focus on the 2nd one.

ANISOTROPIC SPATIAL POINT PROCESS

- The genesis of many geographic phenomena demonstrates clear *anisotropic* spatial processes which means the spatial interaction also depends on direction.
- Geo-tagged social media data reflects **human dynamic mobility** in/across urban area which are highly restricted by the **urban spatial structure** (road network)

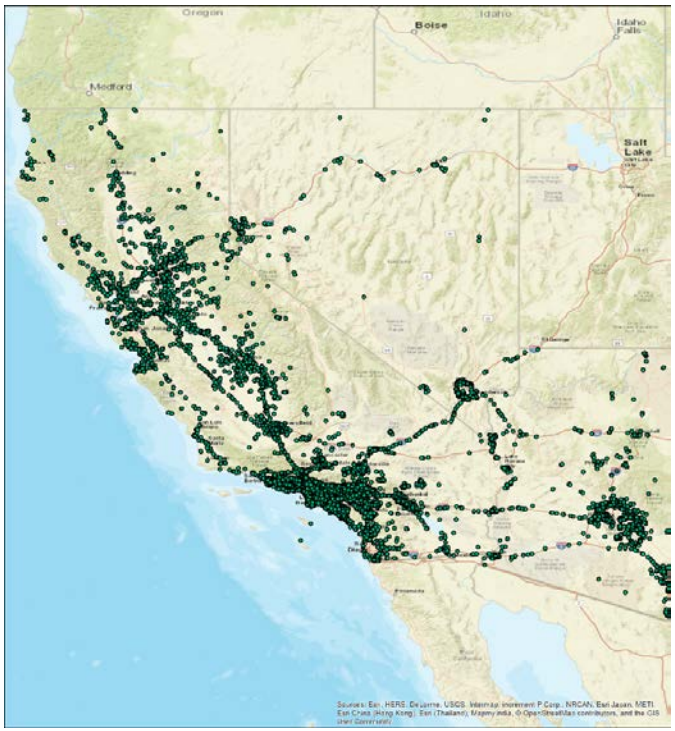


Fig. 3. Geo-tagged Twitter in California during Sep 2015.

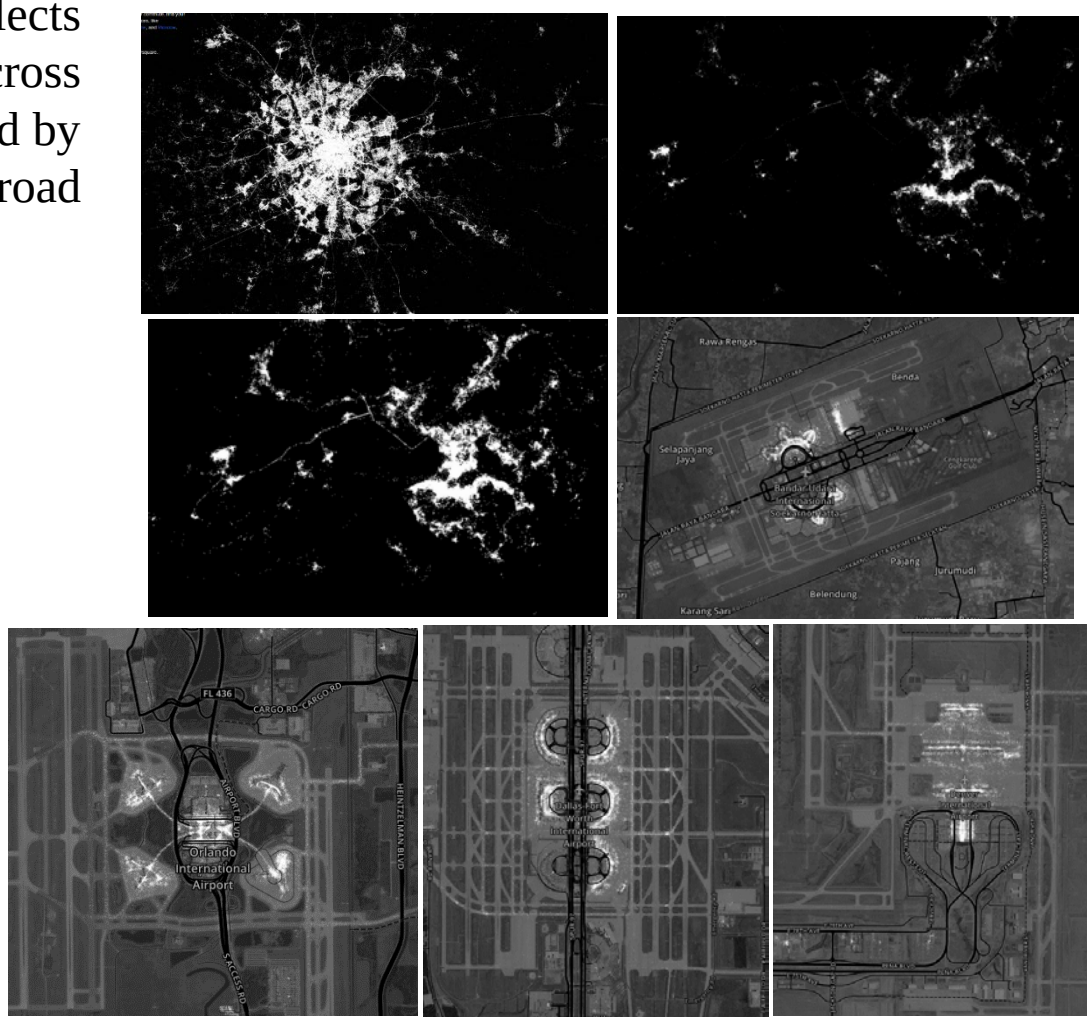
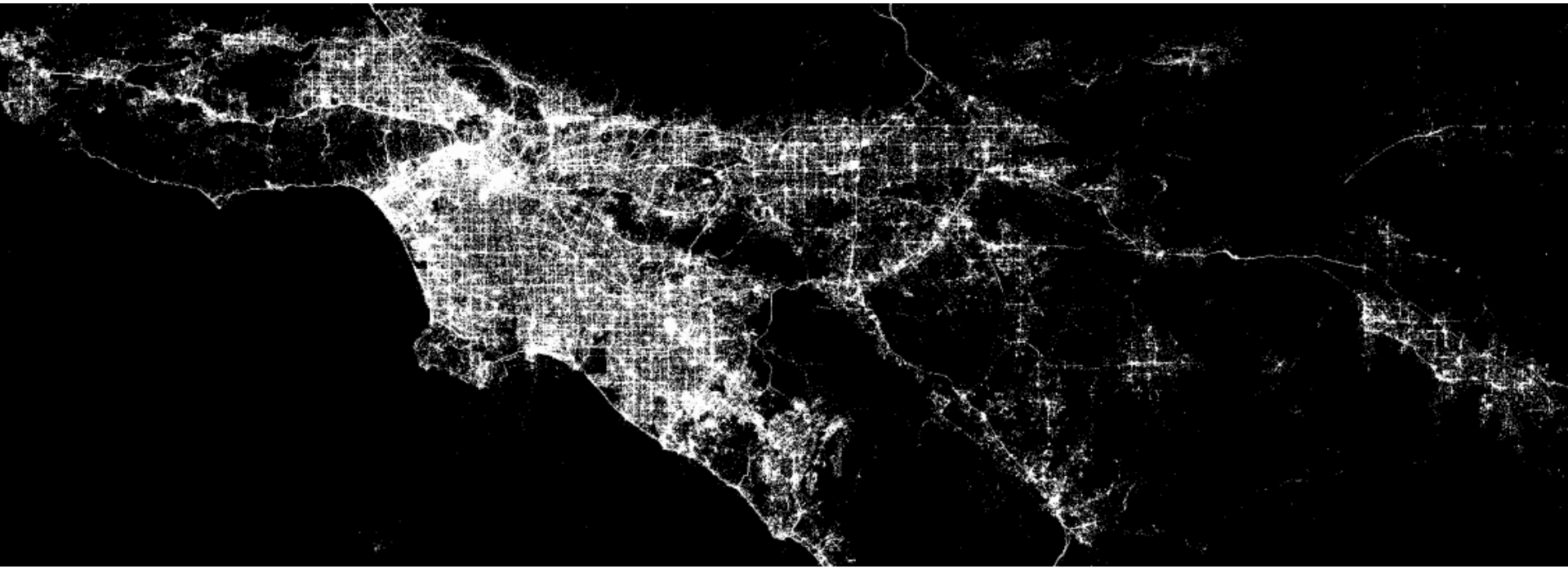


Fig. 4. Snapshots of Geo-tagged Foursquare data

RESEARCH QUESTION

- Develop an algorithm based on DBSCAN family that can capture anisotropic spatial point patterns



ADCN ALGORITHM

- *Using a ellipse instead of a circle as the neighborhood of each point*
- Using the nearby points of *pi* to calculate **Standard Deviation Ellipse** (SDE) to get the direction and shape of local scan ellipse
- Rescale the ellipse (ERi):

$$Area(ER_i) = PI \times Eps^2$$

How to define the nearby points for an arbitrary point Pi in order to calculate SDE?

Algorithm 1: ADCN(D, $MinPts$, Eps)

Input : A set of n points $D(X,Y)$; $MinPts$; Eps;
Output: Clusters with different labels $C_i[]$; A set of noise points $Noi[]$

```
1  foreach point p_i(x_i, y_i) in the set of points D(X,Y) do
2      Mark p_i as Visited;
3      //Get Eps-ellipse-neighborhood EN_Eps(p_i) of p_i
4      ellipseRegionQuery(p_i, D, MinPts, Eps);
5      if |EN_Eps(p_i)| < MinPts then
6          Add p_i to the noise set Noi[];
7      else
8          Create a new Cluster C_i[];
9          Add p_i to C_i[];
10         foreach point p_j(x_j, y_j) in EN_Eps(p_i) do
11             if p_j is not visited then
12                 Mark p_j as visited;
13                 //Get Eps-ellipse-neighborhood
                    EN_Eps(p_j) of Point p_j
14                 ellipseRegionQuery(p_j, D, MinPts,
                    Eps);
15                 if |EN_Eps(p_j)| ⩾ MinPts then
16                     Let EN_Eps(p_i) as the merged set of
                        EN_Eps(p_i) and EN_Eps(p_j);
17                     if p_j hasn't been assigned a label
                        then
18                         Add p_j to current cluster C_i[];
19                     end
20                 else
21                     Add p_j to the noise set Noi[];
22                 end
23             end
24         end
25     end
26 end
```

Fig. 8. ADCN: An Anisotropic Density-Based Clustering Algorithm for Discovering Spatial Point Patterns with Noise.

TWO WAYS TO DEFINE THE NEARBY POINTS

- Using scan circle to get nearby points
  ADCN-Eps
- Using k Nearest Neighbor Points
  ADCN-KNN

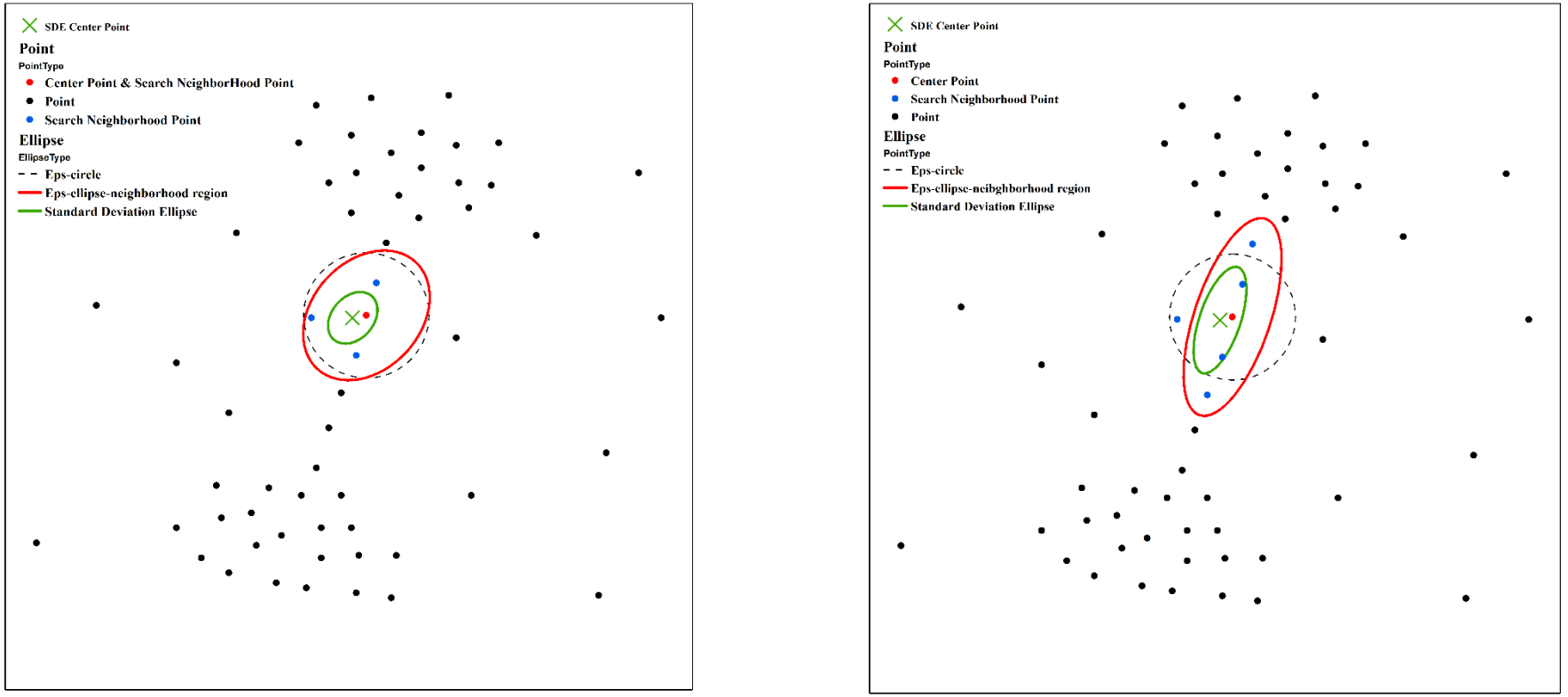


Fig. 9. Illustration for ADCN-Eps and ADCN-KNN.

- **A toy example simulates the geo-tagged photos around Golden Gate Bridge**
- **How to extract the bridge while filtering out the noises on the both sides of the "bridge"?**

EVALUATION OF CLUSTERING QUALITY

- Many statistic indices have been proposed to evaluate the result of clustering which can be classified into three categories:

Tab. 1. Clustering Comparison Index Categories.

Clustering Comparison Index Categories	Example
Pair-counting based indices	Rand Index Jaccard index Fowlkes–Mallows index
Set-matching based indices	Clustering Error
Information theoretic indices	Normalized Mutual Information (NMI) Variation of Information (VI)

- All of these indices are called extrinsic clustering evaluation methods. They compare the clustering results of one algorithms with the “ground truth”.

NORMALIZED MUTUAL INFORMATION (NMI)

Let X, Y are two random variables describe by two different the cluster labeling, NMI is defined as the mutual information between X and Y normalized by the entropy of X and Y:

$$NMI(X,Y) = \frac{I(X,Y)}{\sqrt{H(X)H(Y)}}$$

Where *I(X, Y)* denote the mutual information between X and Y, and H(X) denote the entropy of X. So does H(Y).

Let n be the number of points in a point datasets D. $X = (X_1, X_2, \ldots, X_r)$ and $Y = (Y_1; Y_2, \ldots, Y_s)$ are two clustering results from the same or different clustering algorithms, NMI is defined as:

$$\Phi^{(NMI)}(X,Y) = \frac{\sum_{h=1}^{r}\sum_{l=1}^{s} n_{h,l}^{(x,y)} \log \frac{n \cdot n_{h,l}^{(x,y)}}{n_h^{(x)} \cdot n_l^{(y)}}}{\sqrt{\left(\sum_{h=1}^{r} n_h^{(x)} \log \frac{n_h^{(x)}}{n}\right)\left(\sum_{l=1}^{s} n_l^{(y)} \log \frac{n_l^{(y)}}{n}\right)}}$$

Where $n_h^{(x)}$ be the number of points in cluster X_h and $n_l^{(y)}$ the number of points in cluster Y_l. Let $n_{h,l}^{(x,y)}$ be the number of points in the intersect of cluster X_h and Y_l.

RAND INDEX

Let n be the number of points in a point datasets D. $X = (X_1, X_2, \ldots, X_r)$ and $Y = (Y_1; Y_2, \ldots, Y_s)$ are two clustering results from the same or different clustering algorithms, Rand index is defined as the agreement between X and Y divided by the total pairs.

$$\Phi^{(Rand)}(X, Y) = \frac{a + b}{a + b + c + d}$$

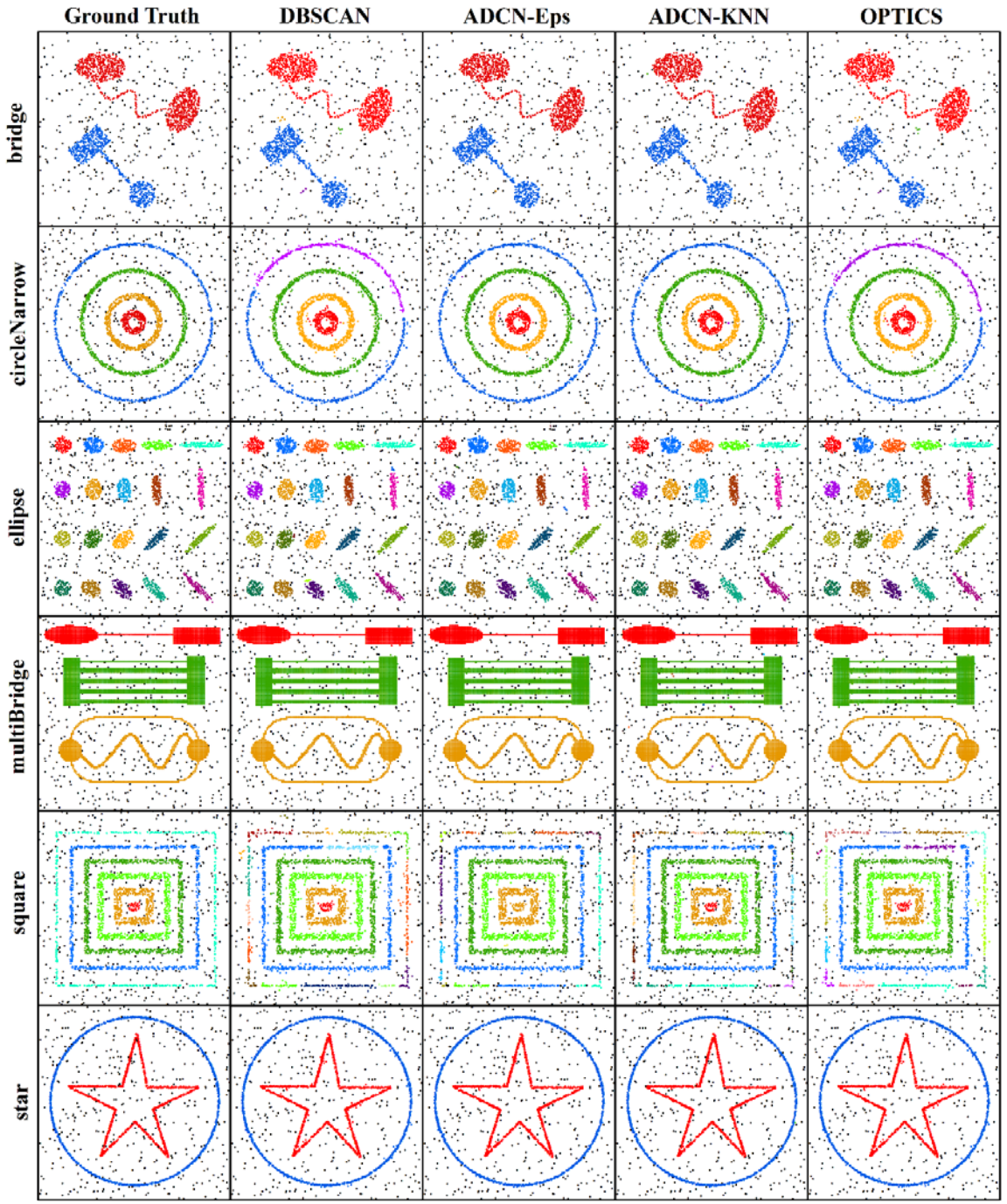
- a: the number of pairs of points in D that are in the same clusters in X and Y .
- b: the number of pairs of points in D that are in different clusters in X and Y .
- c: the number of pairs of points in D that are in the same clusters in X and in different cluster in Y .
- d: the number of pairs of points in D that are in different clusters in X and in the same cluster in Y .
- a+b is the agreement between X and Y, c+d is the disagreement between X and Y.

Both NMI and Rand index measure the similarity between two clustering results (between clustering result and "ground truth"). Higher NMI and Rand index means higher similarity (higher clustering accuracy).

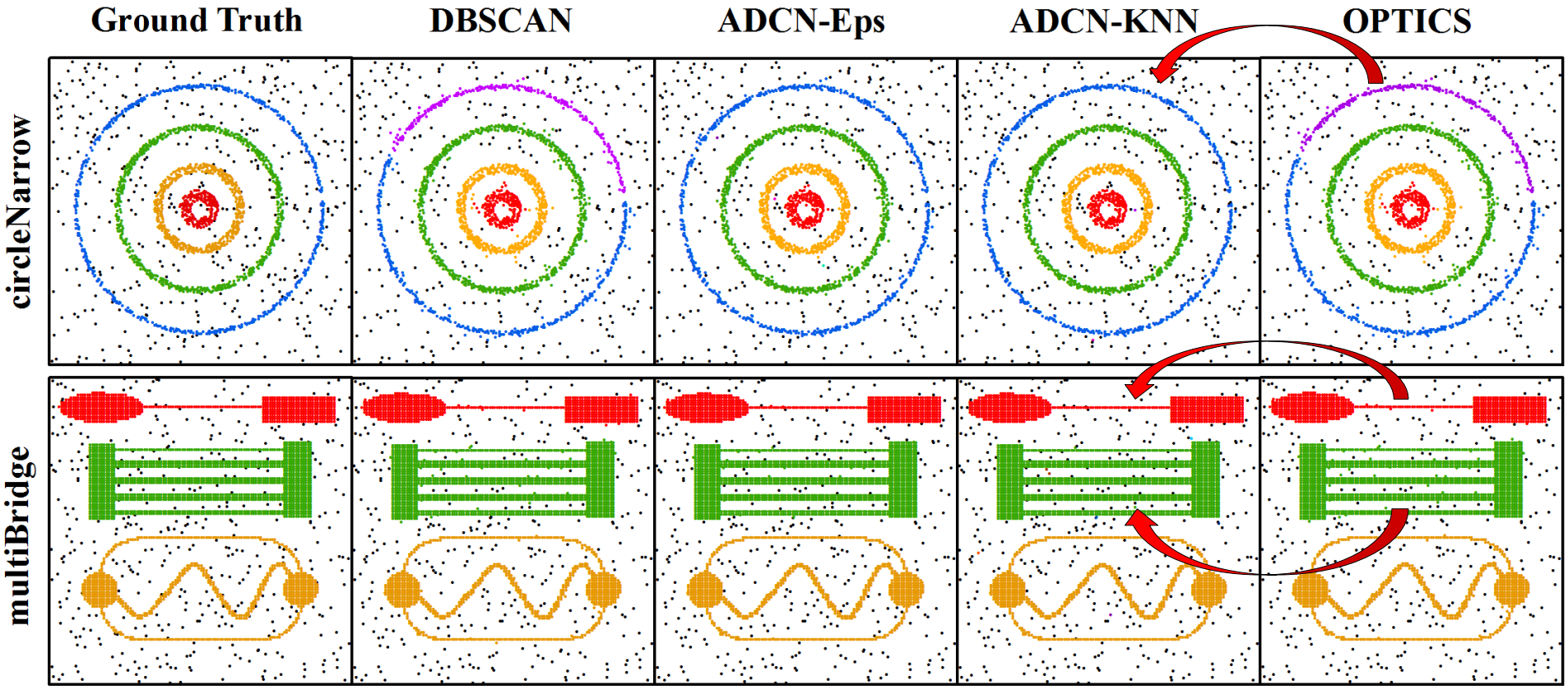
SYNTHETIC CASES STUDY

- Create random points within polygons (cluster points)
- Create random points outside these polygons (noise points).
- Run DBSCAN, ADCN-Eps, ADCN-KNN, OPTICS.
- Try every possible parameter combination of Eps and MinPts to get the “best” clustering result with highest NMI or Rand Index.

We also run OPTICS because OPTICS is the algorithm aiming at fixing some problems of DBSCAN

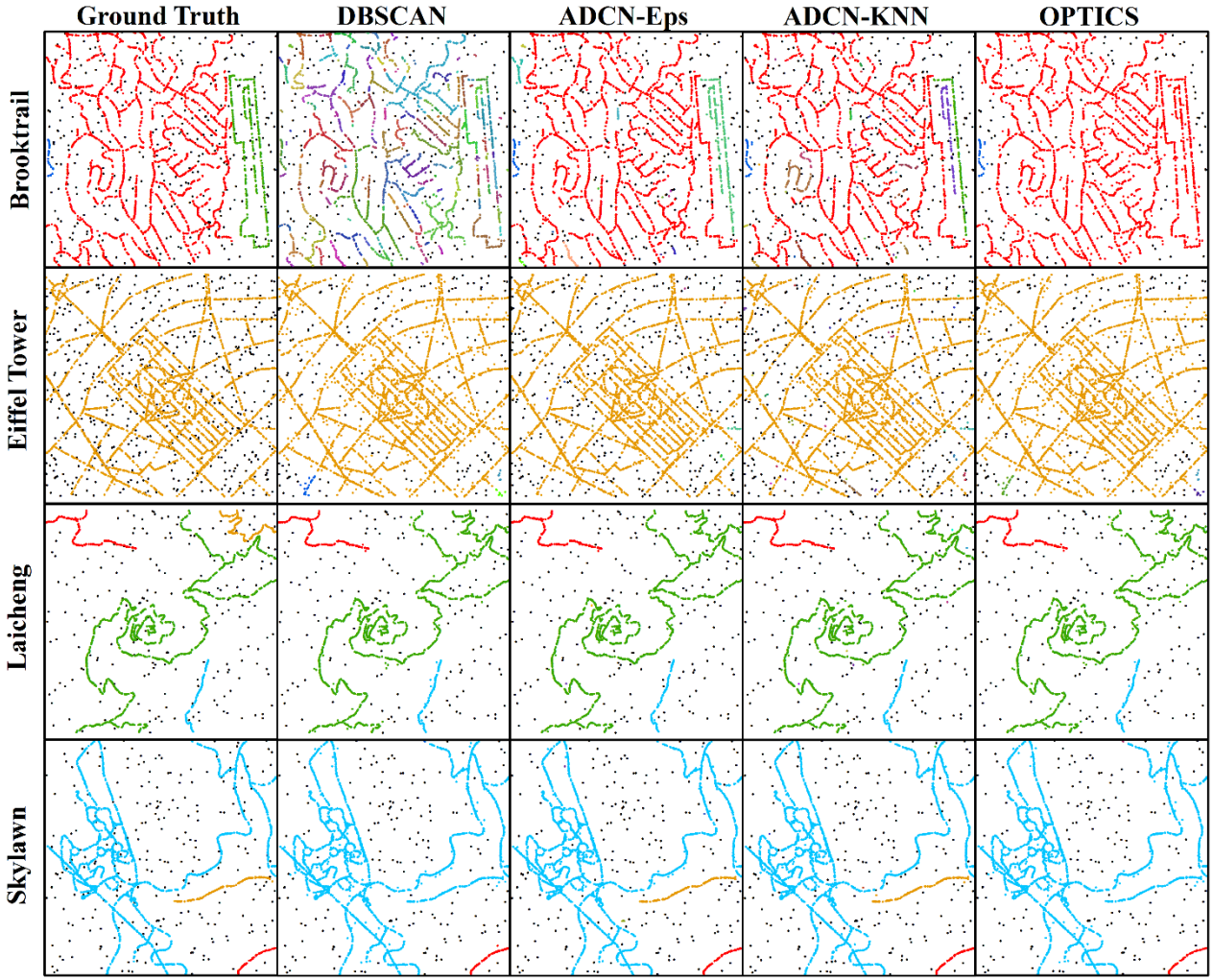


Synthetic cases Study

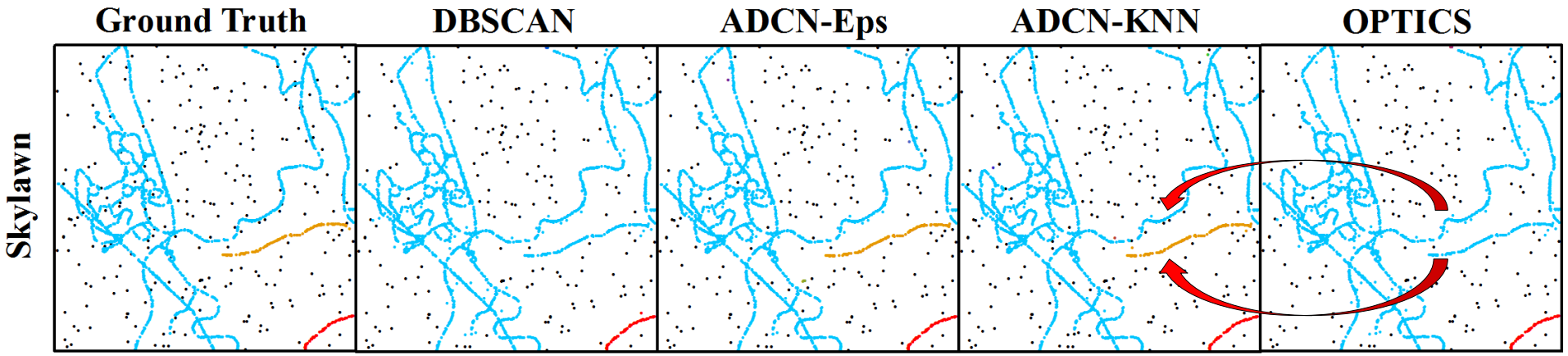


REAL WORLD CASES STUDY

- Same as synthetic cases except that the polygon is the buffer zone (3m) created from road network.

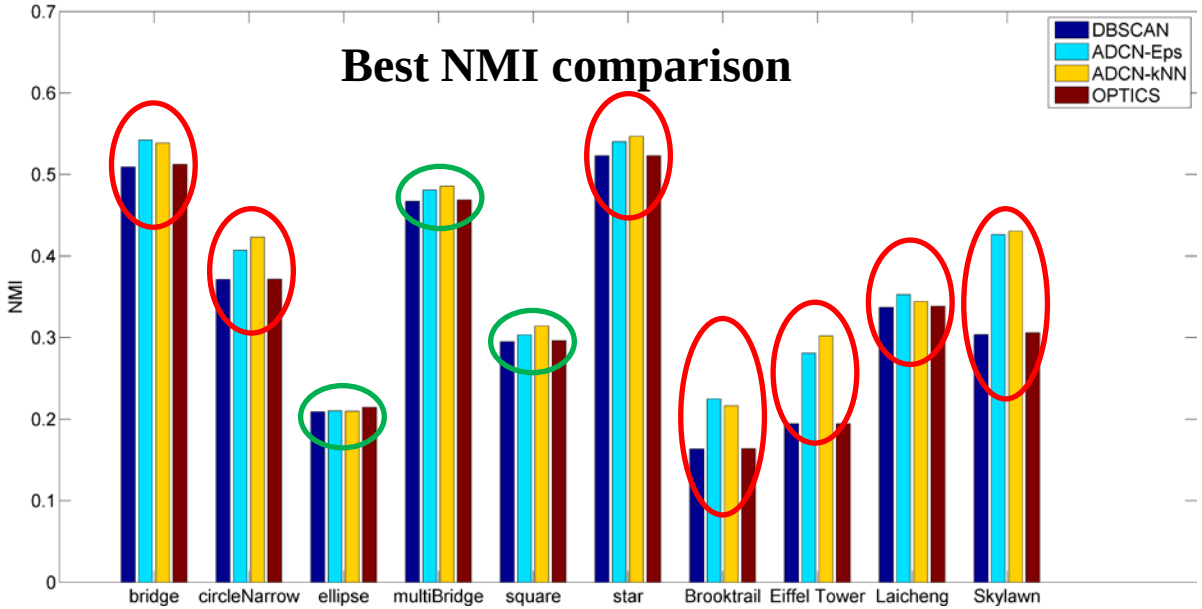


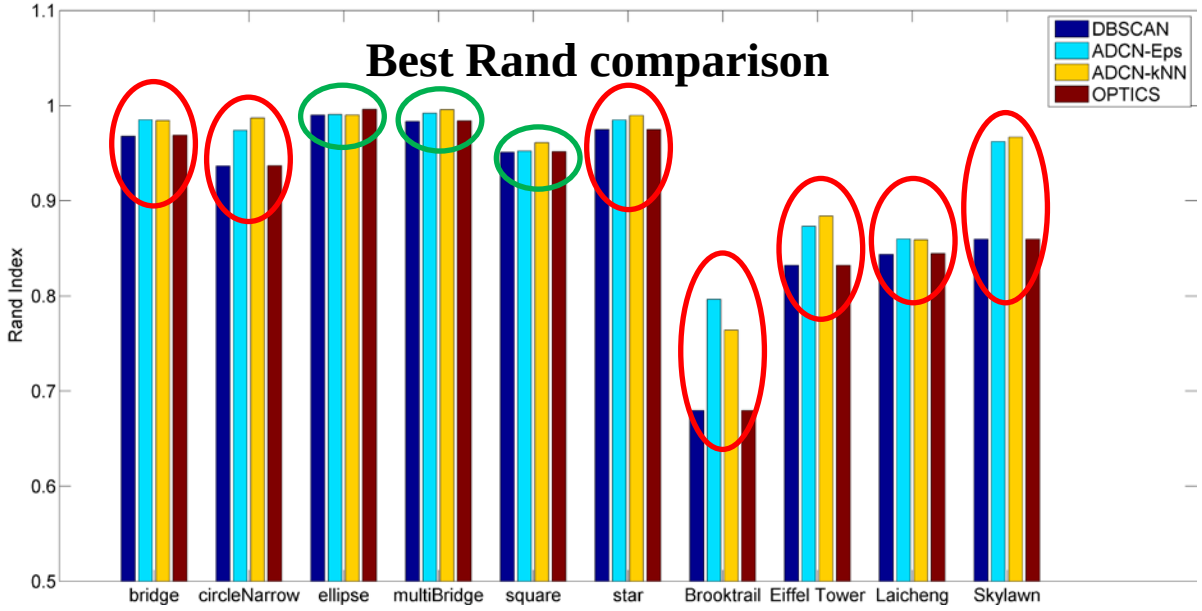
Real world cases Study



CLUSTERING RESULT COMPARISON BY STATISTIC INDICES

- ADCN outperforms DBSCAN and OPTICS when the datasets have obvious anisotropic spatial point patterns. (red circle)
- ADCN performs equally well in cases that do not explicitly benefit from an anisotropic perspective. (green circle)

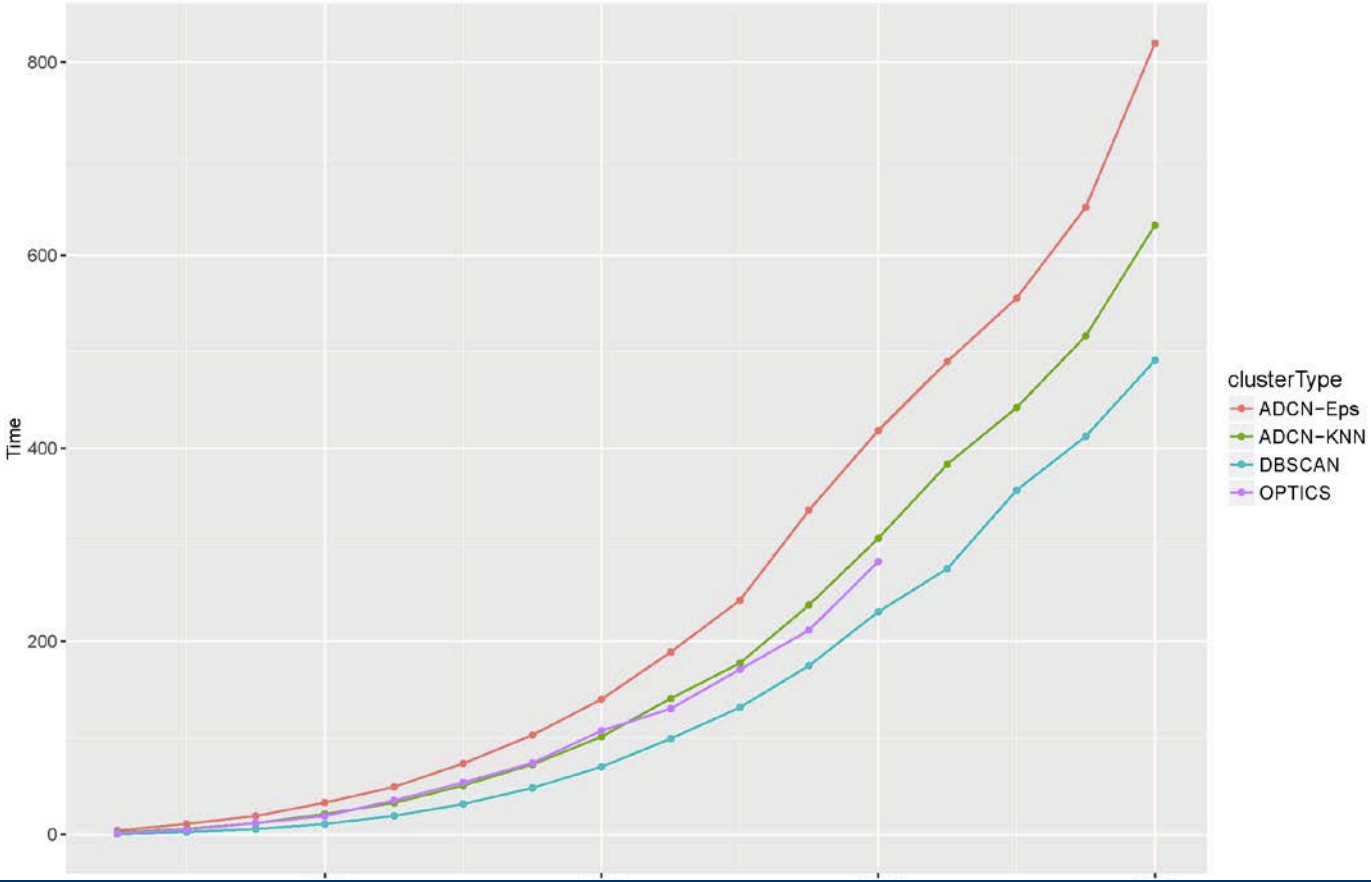




EVALUATION OF CLUSTERING EFFICIENCY

Comparison of clustering efficiency among DBSCAN, OPTICS, ADCN-Eps, ADCN-KNN:

- Theoretical time complexity: All these four algorithms are $O(n^2)$ without spatial index; O(nlogn) with R-tree
- Run time comparison: As the size of the point dataset increases, the ratio of the runtimes of ADCN-KNN to DBSCAN decrease from 2.80 to 1.29. The original OPTICS paper states a 1.6 runtime factor compared to DBSCAN. For ADCN, we test point-in-circle for the radius of the major axis before computing point-in-ellipse to reduce the runtime.



CONCLUSION

- ADCN-KNN outperforms DBSCAN and OPTICS for the detection of anisotropic spatial point patterns and performs equally well in cases that do not explicitly benefit from an anisotropic perspective.
- ADCN has the same time complexity and similar run time as DBSCAN and OPTICS.
- Our algorithm is particularly suited for linear features such as typically encountered in urban structures.

[1] Gengchen Mai, Krzysztof Janowicz, Yingjie Hu, Song Gao. ADCN: An Anisotropic Density-Based Clustering Algorithm for Discovering Spatial Point Patterns with Noise, *Computers, Environment and Urban Systems* (Under Review)

Thanks & Question?