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

Gengchen Mai, Krzysztof Janowicz, Yingjie Hu, Song Gao

STKO Lab, Department of Geography, UC Santa Barbara

{gengchen_mai, jano, yingjiehu, sgao}@geog.ucsb.edu

Abstract

In this work we introduce an anisotropic density-based clustering algorithm. It 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 as DBSCAN and OPTICS, namely O(n log n) when using a spatial index, O(n^2) otherwise.

Introduction

Cluster analysis is a key component of modern knowledge discovery. A wide range of clustering algorithms, such as DBSCAN, OPTICS, K-means, and Mean Shift, have been proposed and implemented over the last decades. Many clustering algorithms assume isotropic second-order effects among spatial objects thereby implying that the magnitude of similarity and interaction between two objects mostly depends on their distance. However, the genesis of many geographic phenomena demonstrates clear anisotropic spatial processes. Isotropic clustering algorithms such as DBSCAN have difficulties dealing with the resulting point patterns and either fail to eliminate noise or do so at the expense of introducing many small clusters. To address this problem, we propose an anisotropic density-based clustering algorithm.

ADCN

Some important definitions of ADCN:

Definition 2. (Search-neighborhood of a point) A set of points $S(p_i)$ around Point p_i is called search-neighborhood of Point p_i and can be defined in two ways:

1. The Eps-neighborhood $N_{Eps}(p_i)$ of Point p_i.
2. The k-th nearest neighbor $KNN(p_i)$ of Point p_i. Here $k = MinPts$ and $KNN(p_i)$ does not include p_i itself.

Definition 3. (Eps-ellipse-neighborhood region of a point) An ellipse ER_i is called Eps-ellipse-neighborhood region of a point p_i if:

1. Ellipse ER_i is centered at Point p_i.
2. Ellipse ER_i is scaled from the standard deviation ellipse SDE_i computed from the Search-neighborhood $S(p_i)$ of Point p_i.
3. $\frac{\sigma_{max}'}{\sigma_{min}'} = \frac{\sigma_{max}}{\sigma_{min}}$;
   where σ_{max}',σ_{min}' , σ_{max},σ_{min} are the length of semi-long , semi-short axis of ellipse ER_i and SDE_i.
4. $Area(ER_i) = \pi ab = \pi Eps^2$

Definition 4. (Eps-ellipse-neighborhood of a point) An Eps-ellipse-neighborhood $EN_{Eps}(p_i)$ of point p_i is defined as all the point inside the eillpse ER_i, which can be expressed as $EN_{Eps}(p_i) = \{p_j(x_j, y_j) \in D | \frac{((y_j-y_i)\sin\theta_{max}+(x_j-x_i)\cos\theta_{max})^2}{a^2} + \frac{((y_j-y_i)\cos\theta_{max}-(x_j-x_i)\sin\theta_{max})^2}{b^2} \leqslant 1\}$.

Definition 5. (Directly anisotropic-density-reachable) A point p_j is *directly anisotropic density reachable* from point p_i wrt. Eps and $MinPts$ iff:

1. $p_j \in EN_{Eps}(p_i)$.
2. $|EN_{Eps}(p_i)| \geqslant MinPts$. (Core point condition)

The pseudo code of ADCN-KNN algorithm

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

```
Input  : A set of n points D(X,Y) ; MinPts; Eps;
Output: Clusters with different labels C_i[]; Set of noise points Noi[]
foreach point p_i(x_i, y_i) in the set of points D(X,Y) do
    Mark p_i as Visited;
    //Get Eps-ellipse-neighborhood EN_Eps(p_i) of p_i
    ellipseRegionQuery(p_i, D, MinPts, Eps);
    if |EN_Eps(p_i)| < MinPts then
        Add p_i to the noise set Noi[];
    else
        Create a new Cluster C_i[];
        Add p_i to C_i[];
        foreach point p_j(x_j, y_j) in EN_Eps(p_i) do
            if p_j is not visited then
                Mark p_j as visited;
                //Get Eps-ellipse-neighborhood EN_Eps(p_j) of
                  Point p_j
                ellipseRegionQuery(p_j, D, MinPts, Eps);
                if |EN_Eps(p_j)| >= MinPts then
                    Let EN_Eps(p_i) as the merged set of
                      EN_Eps(p_i) and EN_Eps(p_j);
                    if p_j hasn't been assigned a label then
                        Add p_j to current cluster C_i[];
                    end
                else
                    Add p_j to the noise set Noi[];
                end
            end
        end
    end
end
```

Algorithm 2: ellipseRegionQuery(p_i, D, $MinPts$, Eps)

```
Input  : p_i, D, MinPts, Eps
Output: Eps-ellipse-neighborhood EN_Eps(p_i) of Point p_i
//Get the Search-neighborhood S(p_i) of Point p_i. ADCN-Eps and
  ADCN-KNN use different functions.
ADCN-Eps: searchNeighborhoodEps(p_i, D, Eps); ADCN-KNN:
  searchNeighborhoodKNN(p_i, D, MinPts);
Compute the standard deviation ellipse SDE_i base on the
  Search-neighborhood S(p_i) of Point p_i;
Scale Ellipse SDE_i to get the Eps-ellipse-neighborhood region ER_i
  of Point p_i to make sure Area(ER_i) = PI × Eps^2;
if The length of short axis of ER_i == 0 then
    // the Eps-ellipse-neighborhood region ER_i of Point p_i is
      diminished to a straight line Get Eps-ellipse-neighborhood
      EN_Eps(p_i) of Point p_i by finding all points on this straight line
      ER_i;
else
    // the Eps-ellipse-neighborhood region ER_i of Point p_i is an
      ellipse Get Eps-ellipse-neighborhood EN_Eps(p_i) of Point p_i by
      finding all the points inside Ellipse ER_i;
end
return EN_Eps(p_i);
```

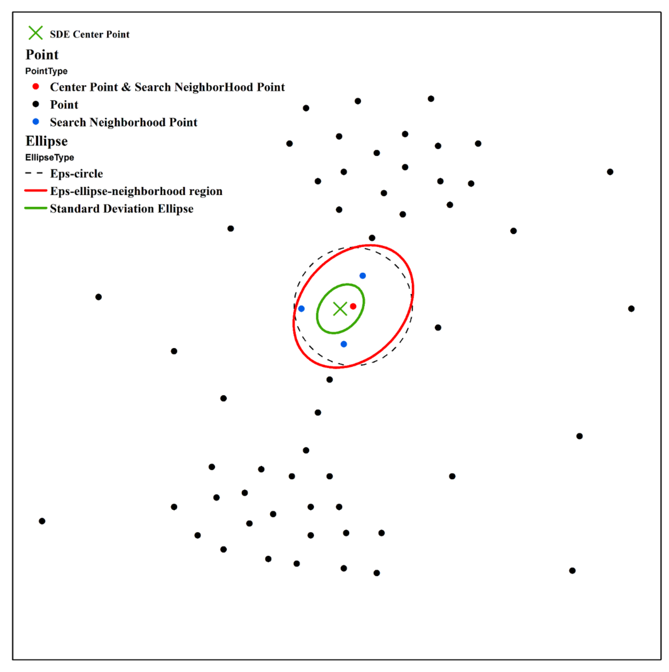

Fig.1 Illustration for ADCN-Eps.

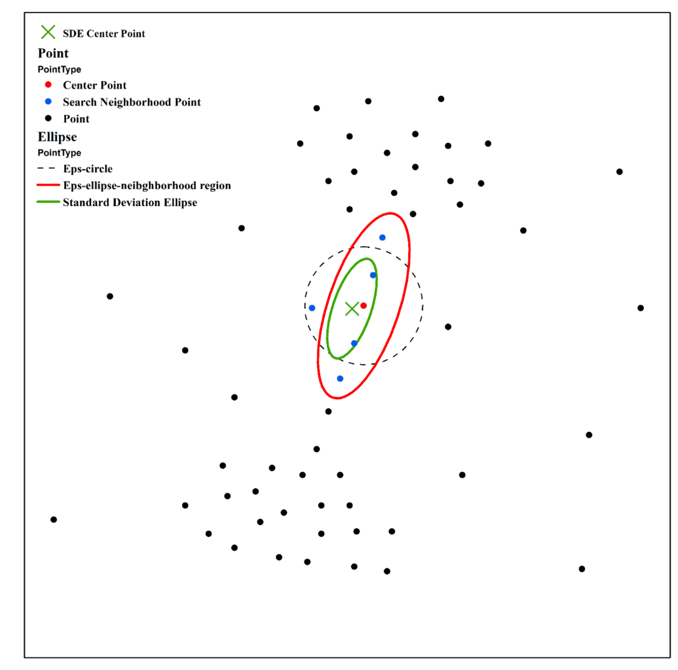

Fig.2 Illustration for ADCN-KNN.

Evaluation of Clustering Quality

We use two clustering indices - normalized mutual information (NMI), Rand Index - to measure the quality of clustering results of all algorithms.

$$\Phi^{(NMI)}(X,Y) = \frac{\sum_{h=1}^{r}\sum_{l=1}^{s} n_{h,l}^{(x,y)} \log \frac{n \cdot n_{h,l}^{(x,y)}}{n_h^{(x)} \cdot n_l^{(y)}}}{\sqrt{(\sum_{h=1}^{r} n_h^{(x)} \log \frac{n_h^{(x)}}{n})(\sum_{l=1}^{s} n_l^{(y)} \log \frac{n_l^{(y)}}{n})}} \qquad \Phi^{(Rand)}(X,Y) = \frac{a+b}{a+b+c+d}$$

We generated 6 synthetic and 4 real-world use cases with 3 different noise settings. In order to simulate a “ground truth” for the synthetic cases, we created polygons to indicate different clusters and randomly generated points within these polygons and outside of them. We took a similar approach for the four real-world cases. The only difference is that the polygons for real world cases have been generated from buffer zones with a 3m radius of the real-world features.

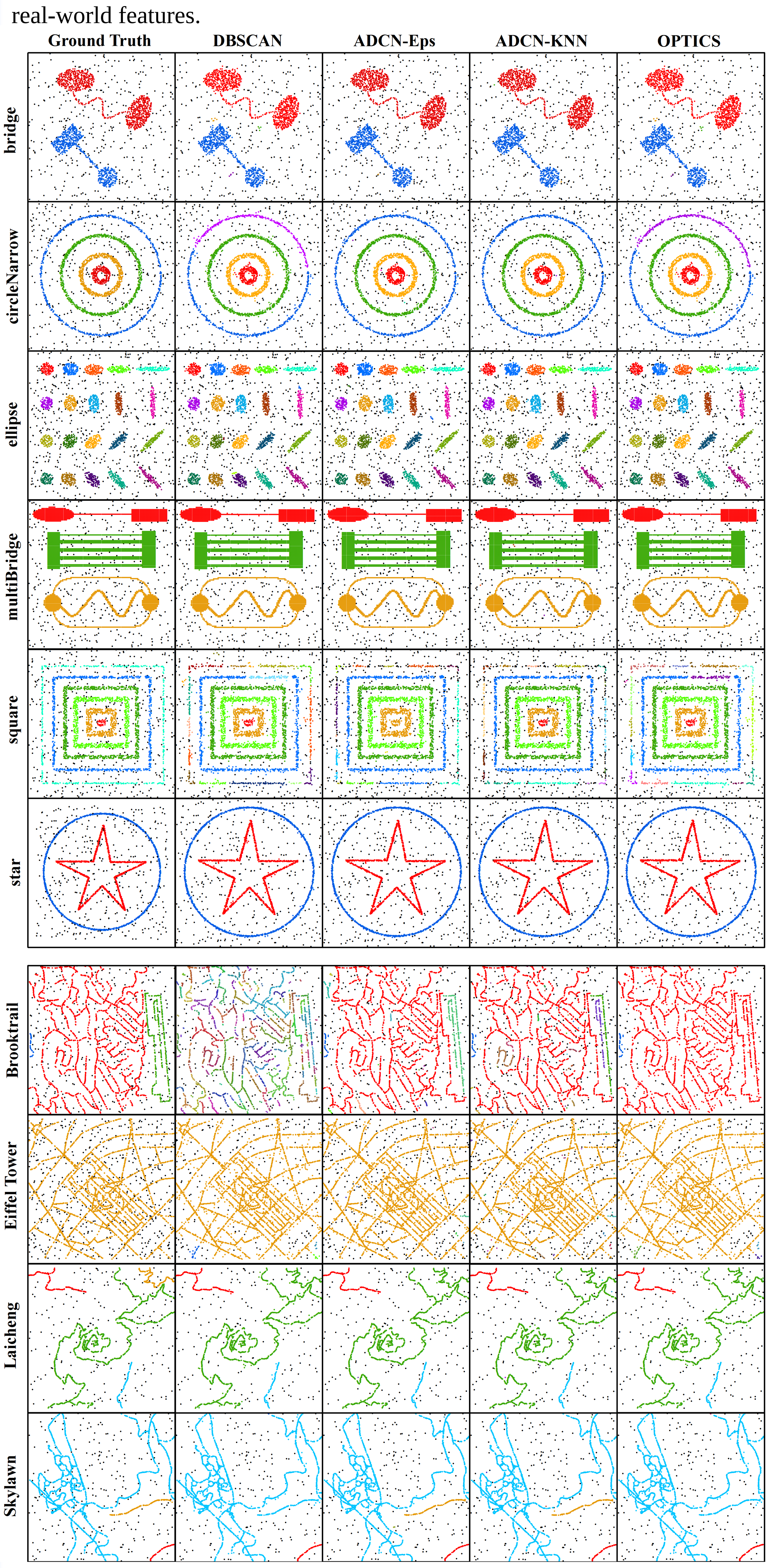

Fig.3 Ground truth and best clustering result comparison for 6 synthesis cases and 4 real world cases.

As for the 30 test cases, ADCN-KNN has a higher maximum NMI/Rand Index than DBSCAN in 27 cases and has a higher maximum NMI/Rand Index than OPTICS in 26 cases

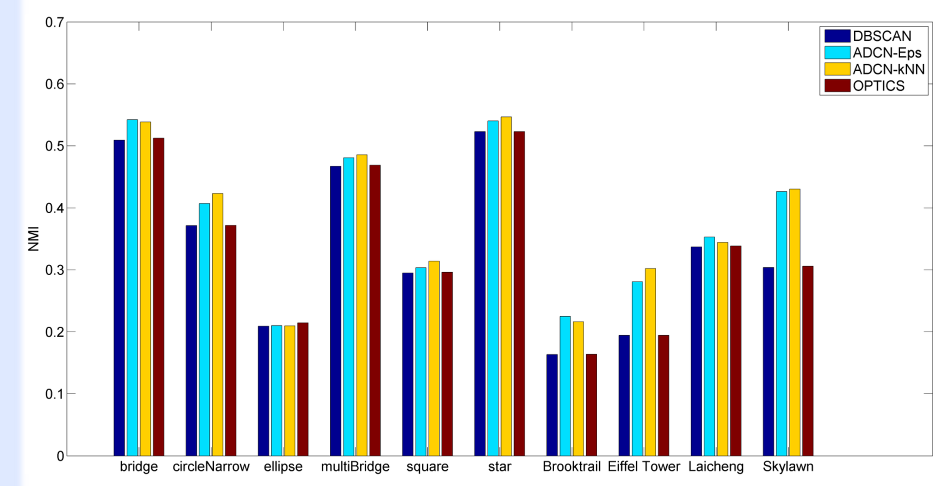

Fig.4 NMI clustering quality comparisons.

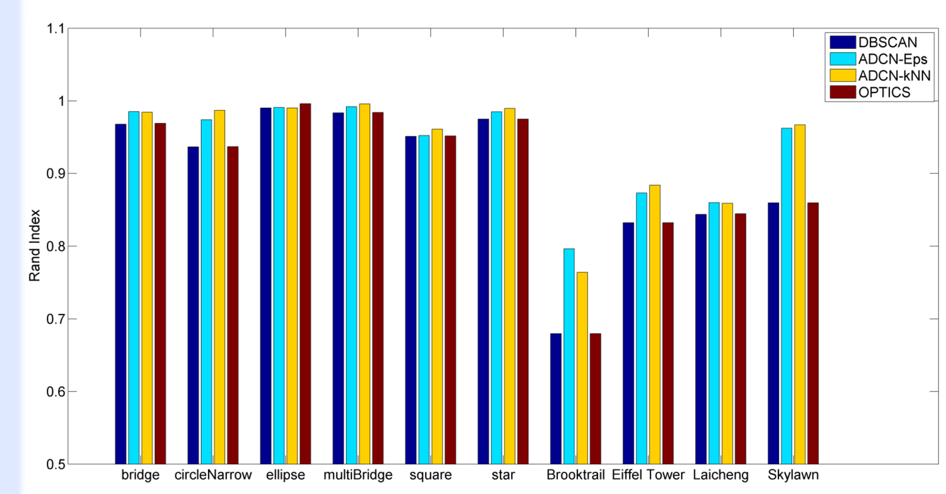

Fig.5 Rand index clustering quality comparisons

Evaluation of Clustering Efficiency

In order to enable a comprehensible comparison of the run times of all algorithms on different sizes of point datasets, we performed a batch of performance tests. The polygons from the 10 cases shown above have been used to generated point datasets of different sizes ranging from 500 to 8000 in 500 step intervals. The ratio of noise points to cluster points is set to 0.25. *Eps*, *MinPts* are set to 15, 5 for all of these experiments. The median of the run times for the same size of point datasets is depicted in Fig. 6. We implemented an R-tree to accelerate the neighborhood queries for all algorithms

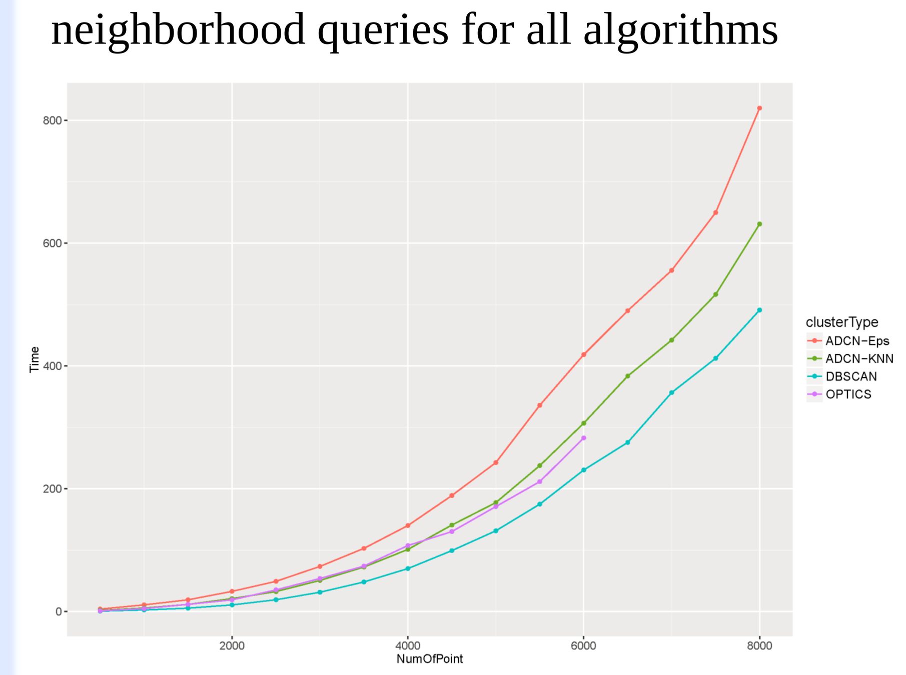

Fig.6 Comparison of clustering efficiency with different dataset sizes; runtimes are given in ms.

Conclusion

We proposed an anisotropic density-based clustering algorithm (ADCN). Synthetic & real-world cases have been used to verify its quality and efficiency compared to DBSCAN & OPTICS. ADCN outperforms DBSCAN & OPTICS for the detection of anisotropic spatial point patterns and performs equally well in cases that do not show anisotropicity. ADCN has the same time complexity as DBSCAN & OPTICS, namely O(n log n) using a spatial index, O(n2) otherwise. Its average runtime is comparable to OPTICS. ADCN is particularly suited for linear features such as encountered in urban structures. Application areas include social media data, trajectories from car sensors, wildlife tracking, and so forth.

References

M. Ankerst, M.M. Breunig, H.-P. Kriegel, and J. Sander. OPTICS: ordering points to identify the clustering structure. In ACM SIGMOD Record, volume 28, pages 49-60. ACM, 1999.

M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In KDD, volume 96, pages 226-231, 1996.