FLORIDA STATE UNIVERSITY

RNN-DBSCAN: A Density-Based Clustering Algorithm Using Reverse Nearest Neighbor Density Estimates

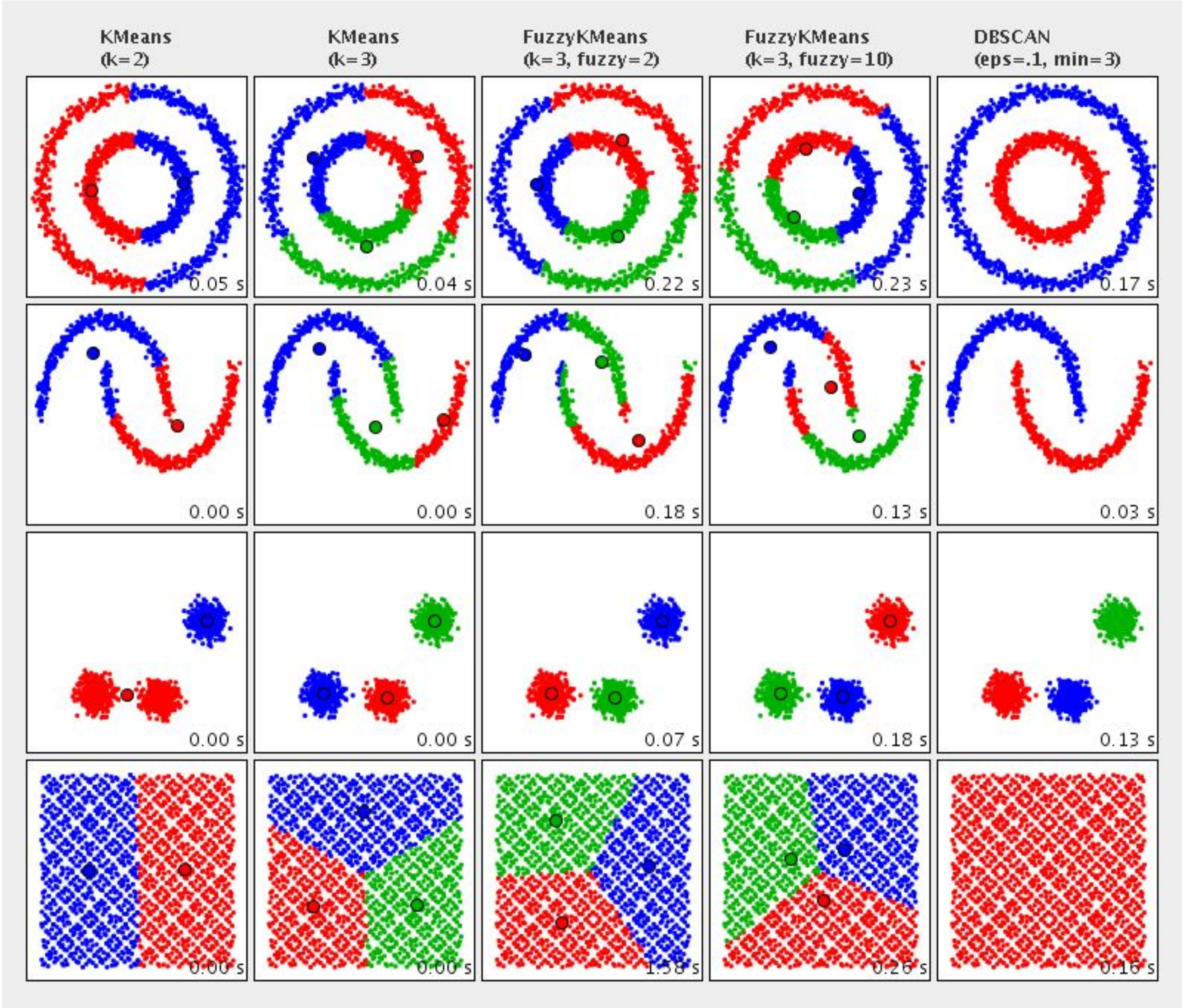
Avory Bryant and Krzysztof Cios
Presented by Thai Flowers

What is clustering?

- "an unsupervised pattern recognition problem"
- "grouping data such that observations within a group are similar to each other while being dissimilar to observations within other groups"

- Partitioning, hierarchical, model, density, and grid based algorithms are common

Clustering Examples



Source:
https://hipparchus.org/hipparchus-clustering/index.html

DBSCAN

- Density Based Spatial Clustering for Applications with Noise
- min_{pts} and eps parameters
- Pros
  - Determines number of clusters automatically
  - Handles irregularly shaped clusters
  - Simple algorithm
- Cons
  - Difficult to determine parameters
  - Distance measure must be symmetric
  - Errors on nearby clusters of differing density

DBSCAN--Algorithm

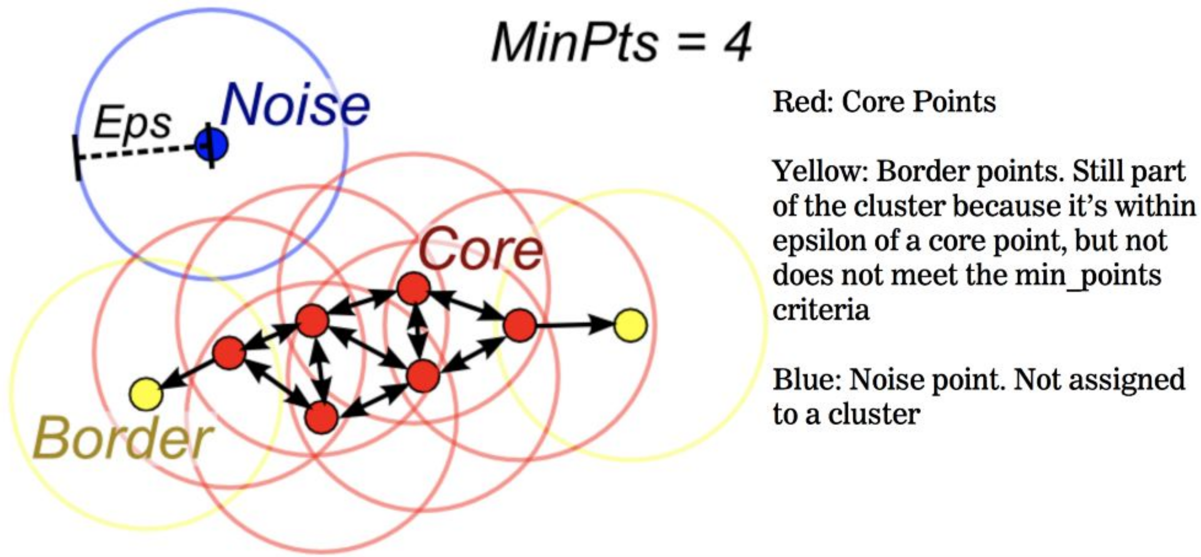
```
DBSCAN (SetOfPoints, Eps, MinPts)

// SetOfPoints is UNCLASSIFIED
  ClusterId := nextId(NOISE);
  FOR i FROM 1 TO SetOfPoints.size DO
    Point := SetOfPoints.get(i);
    IF Point.ClId = UNCLASSIFIED THEN
      IF ExpandCluster(SetOfPoints, Point
             ClusterId, Eps, MinPts) THEN
        ClusterId := nextId(ClusterId)
      END IF
    END IF
  END FOR
END; // DBSCAN
```

```
ExpandCluster(SetOfPoints, Point, ClId, Eps,
              MinPts) : Boolean;
  seeds:=SetOfPoints.regionQuery(Point,Eps);
  IF seeds.size<MinPts THEN // no core point
    SetOfPoint.changeClId(Point,NOISE);
    RETURN False;
  ELSE   // all points in seeds are density-
         // reachable from Point
    SetOfPoints.changeClIds(seeds,ClId);
    seeds.delete(Point);
    WHILE seeds <> Empty DO
      currentP := seeds.first();
      result := SetOfPoints.regionQuery(currentP,
                                        Eps);
      IF result.size >= MinPts THEN
        FOR i FROM 1 TO result.size DO
          resultP := result.get(i);
          IF resultP.ClId
              IN {UNCLASSIFIED, NOISE} THEN
            IF resultP.ClId = UNCLASSIFIED THEN
              seeds.append(resultP);
            END IF;
            SetOfPoints.changeClId(resultP,ClId);
          END IF; // UNCLASSIFIED or NOISE
        END FOR;
      END IF; // result.size >= MinPts
      seeds.delete(currentP);
    END WHILE; // seeds <> Empty
    RETURN True;
  END IF
END; // ExpandCluster
```

Source: A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise
Martin Ester, Hans-Peter Kriegel, Jiirg Sander, Xiaowei Xu

DBSCAN Concept Diagram

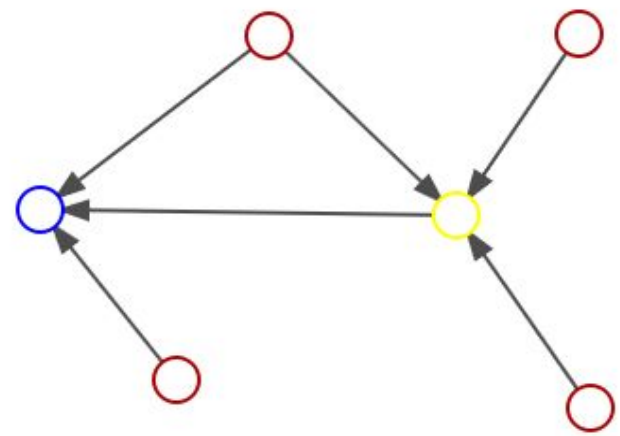


Source: https://medium.com/@elutins/dbscan-what-is-it-when-to-use-it-how-to-use-it-8bd506293818

- Start with a core observation, visit neighbors recursively until you hit a non-core (aka border) observation
- Points not in a cluster after all visited are noise
- Difficult to choose correct parameters when clusters have varying densities.
  - Sparse cluster → large eps → merge with dense cluster
  - Lower eps → sparse cluster labeled as noise

Reverse Nearest Neighbor

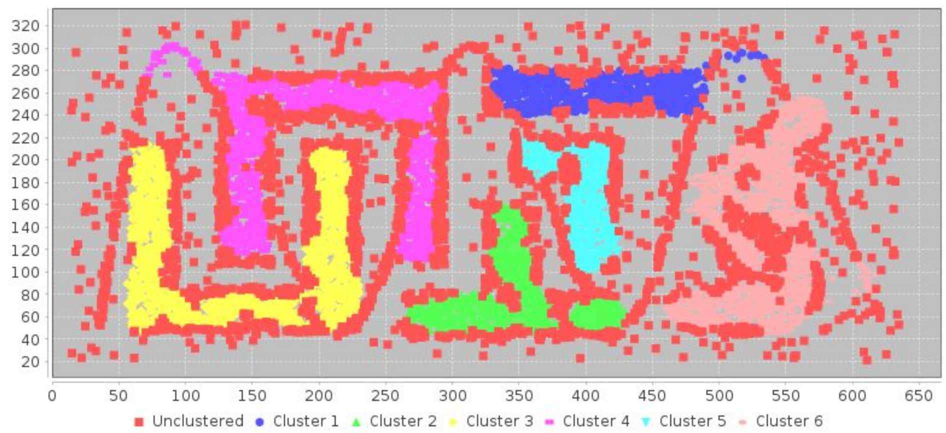
- RNN approaches attempt to address parameter issue
- Single parameter, k, number of reverse nearest neighbors
- Blue and Yellow nodes shown with their RNN_3
- Transpose of nearest neighbor graph
- Conceptually simple, but difficult to efficiently compute.



RNN Clustering Algorithms

RECORD

- $G_{RkNN}(V,E)$ is the reverse nearest neighbor graph
- Core = {v in V | outdegree(v) >= k}
- Bad a detecting border points, many false positives for noise



RNN Clustering Algorithms

- IS-DBSCAN and ISB-DBSCAN
- Define “influence space”, intersection of kNN and RkNN graphs
  - This is symmetric, thus distance measure need not be.
- Core = {v in V | |IS(v)| >= 2k/3}
  - 2k/3 viewed as hidden (fixed) parameter
- IS has STRATIFY pass which often has false positive noise detection
  - Incorrect results when no noise, Complicated algorithm
- ISB has post clustering border point classifying pass and has no lower bound on cluster size

RNN-DBSCAN Benefits

- Doesn’t require a symmetric distance measure
- No hidden parameters
- Automatically handles differential densities between clusters correctly
  - As will be seen with the grid dataset

RNN-DBSCAN Definitions

- dist(x,y) is euclidean distance in original paper
  - Though others may be used
- X is set of observations

Definition 1 (k-nearest neighborhood of observation $\mathbf{x}$). *The k-nearest neighborhood of observation $\mathbf{x}$ is defined by the function $N_k(\mathbf{x}) = N$ where N satisfies the following conditions:*

1) $N \subseteq X/\{\mathbf{x}\}$
2) $|N| = k$
3) $\forall \mathbf{y} \in N, \mathbf{z} \in X/(N + \{\mathbf{x}\}) : dist(\mathbf{x}, \mathbf{y}) \leq dist(\mathbf{x}, \mathbf{z})$

Definition 2 (reverse nearest neighborhood of observation $\mathbf{x}$). *The reverse nearest neighborhood of observation $\mathbf{x}$ is defined by the function $R_k(\mathbf{x}) = R$ where R satisfies the following conditions:*

1) $R \subseteq X/\{\mathbf{x}\}$
2) $\forall \mathbf{y} \in R : \mathbf{x} \in N_k(\mathbf{y})$

Definition 3 (directly density-reachable). *A observation $\mathbf{x}$ is directly density reachable from a observation $\mathbf{y}$ if*

1) $\mathbf{x} \in N_k(\mathbf{y})$
2) $|R_k(\mathbf{y})| \geq k$ *(core observation condition)*

Definition 4 (density-reachable). *A observation $\mathbf{x}$ is density-reachable from a observation $\mathbf{y}$ if there is a chain of observations $\mathbf{x}_1, ..., \mathbf{x}_m$, $\mathbf{x}_1 = \mathbf{y}$, $\mathbf{x}_m = \mathbf{x}$ such that where $|R_k(\mathbf{x})| \geq k$*

1) *$\mathbf{x}_m$ is directly density-reachable from $\mathbf{x}_{m-1}$*
2) *$\forall 1 \leq i \leq m-2 : \mathbf{x}_{i+1}$ is directly density-reachable from $\mathbf{x}_i$ or $\mathbf{x}_i$ is directly density-reachable from $\mathbf{x}_{i+1}$*

Definition 5 (density-connected). *A observation $\mathbf{x}$ is density-connected to a observation $\mathbf{y}$ if there is a observation $\mathbf{z}$ such that both, $\mathbf{x}$ and $\mathbf{y}$ are density reachable from $\mathbf{z}$.*

Density connected is a symmetric relationship over all observation types. Now using the above reachability definitions a cluster of observations is defined as follows.

Definition 6 (cluster). *A cluster C is a non-empty subset of X, $\emptyset \neq C \subseteq X$, satisfying the following conditions:*

1) *$\forall \mathbf{x}, \mathbf{y} \in X$: if $\mathbf{x} \in C$ and $\mathbf{y}$ is density-reachable from $\mathbf{x}$ then $\mathbf{y} \in C$ (maximality)*
2) *$\forall \mathbf{x}, \mathbf{y} \in C : \mathbf{x}$ is density-connected to $\mathbf{y}$ (connectivity)*

Definitions Continued

Definition 7 (density of cluster C). *The density of cluster C is defined by the function $den(C)$:*

$$den(C) = \max_{(\mathbf{x},\mathbf{y})} dist(\mathbf{x}, \mathbf{y}),$$
$$\forall \mathbf{x}, \mathbf{y} \in C : |R_k(\mathbf{x})| \geq k,\ |R_k(\mathbf{y})| \geq k, \text{ and}$$
$$\mathbf{y} \text{ is directly density-reachable from } \mathbf{x}$$

Once all clusters have been identified, given cluster C and its density $den(C)$, let the extended cluster of C, C^{ex}, be a superset of C, $C^{ex} \supseteq C$, that is extended by inserting unclustered observations into C. An unclustered observation is inserted into C^{ex} if its closest core k nearest neighbor, within distance $den(C)$, is in C.

RNN-DBSCAN Algorithm

Algorithm 1. $RNN - DBSCAN(X, k)$

```
assign[∀x ∈ X] = UNCLASSIFIED
cluster = 1
for all x ∈ X do
  if assign[x] = UNCLASSIFIED then
    if ExpandCluster(x, cluster, assign, k) then
      cluster = cluster + 1
    end if
  end if
end for
ExpandClusters(X, k, assign)
return assign
```

Algorithm 4. $ExpandClusters(\mathbf{x}, k, assign)$

```
for all x ∈ X do
  if assign[x] = NOISE then
    neighbors = N_k(x), mincluster = NOISE, mindist = ∞
    for all n ∈ N do
      cluster = assign[n], d = dist(x, n)
      if |R_k(n)| ≥ k & d ≤ den(cluster) & d < mindist then
        mincluster = cluster, mindist = d
      end if
    end for
    assign[x] = mincluster
  end if
end for
```

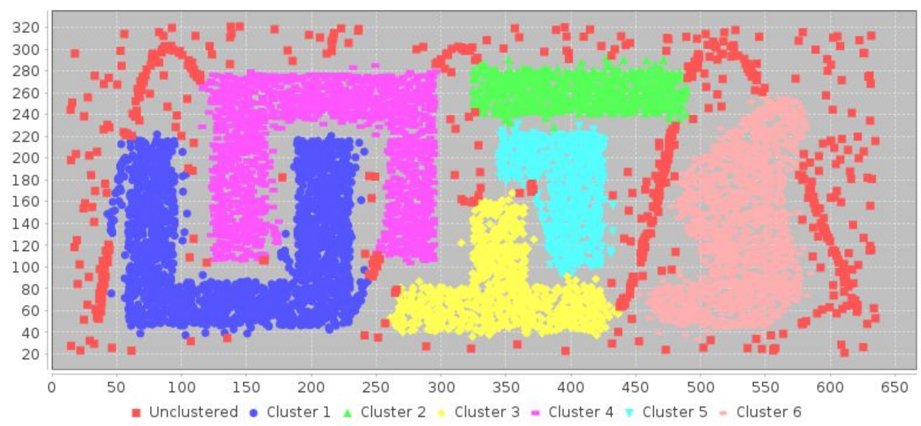
Algorithm 2. $ExpandCluster(\mathbf{x}, cluster, assign, k)$

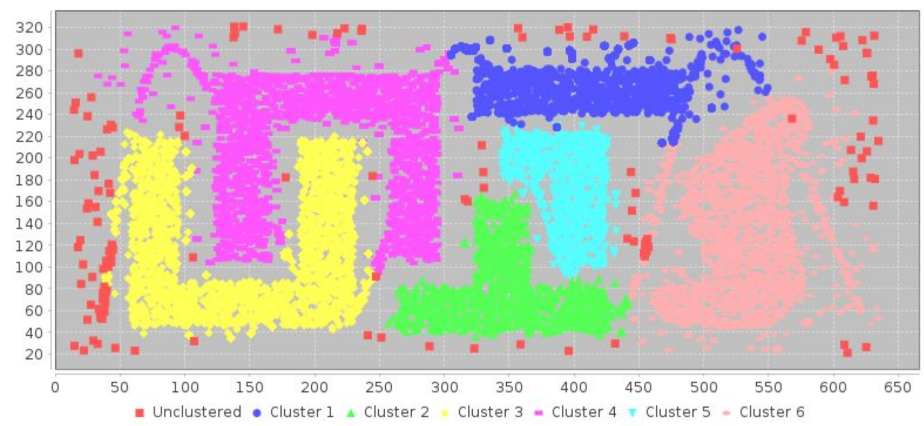
```
if |R_k(x)| < k then
  assign[x] = NOISE
  return FALSE
else
  initialize empty queue seeds
  seeds.enqueue(Neighborhood(x, k))
  assign[x + seeds] = cluster
  while seeds ≠ ∅ do
    y = seeds.dequeue()
    if R_k(y) ≥ k then
      neighbors = Neighborhood(y, k)
      for all z ∈ neighbors do
        if assign[z] = UNCLASSIFIED then
          seeds.enqueue(z)
          assign[z] = cluster
        else if assign[z] = NOISE then
          assign[z] = cluster
        end if
      end for
    end if
  end while
  return TRUE
end if
```

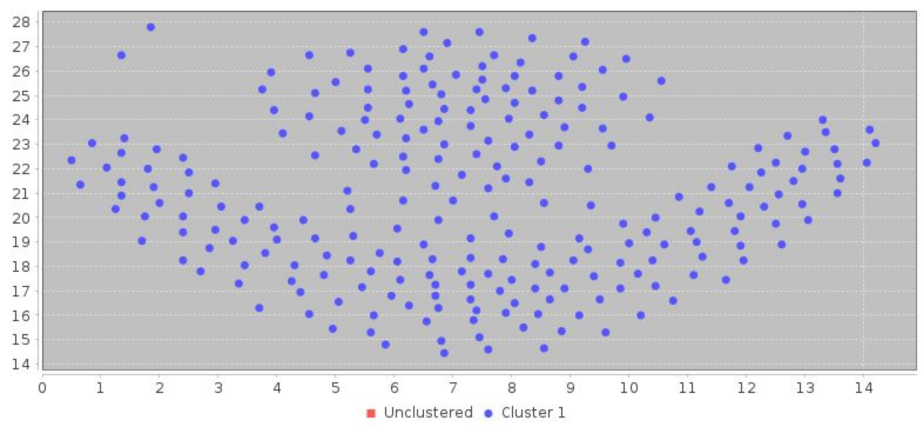
Algorithm 3. $Neighborhood(\mathbf{x}, k)$

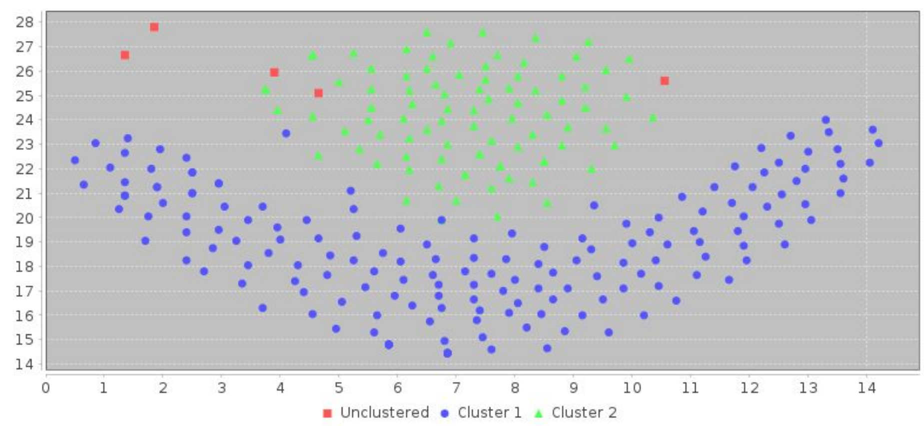
```
neighbors = N_k(x) + {y ∈ R_k(x) : |R_k(y)| ≥ k}
return neighbors
```

DBSCAN vs RNN-DBSCAN









DBSCAN vs RNN-DBSCAN Differential Density

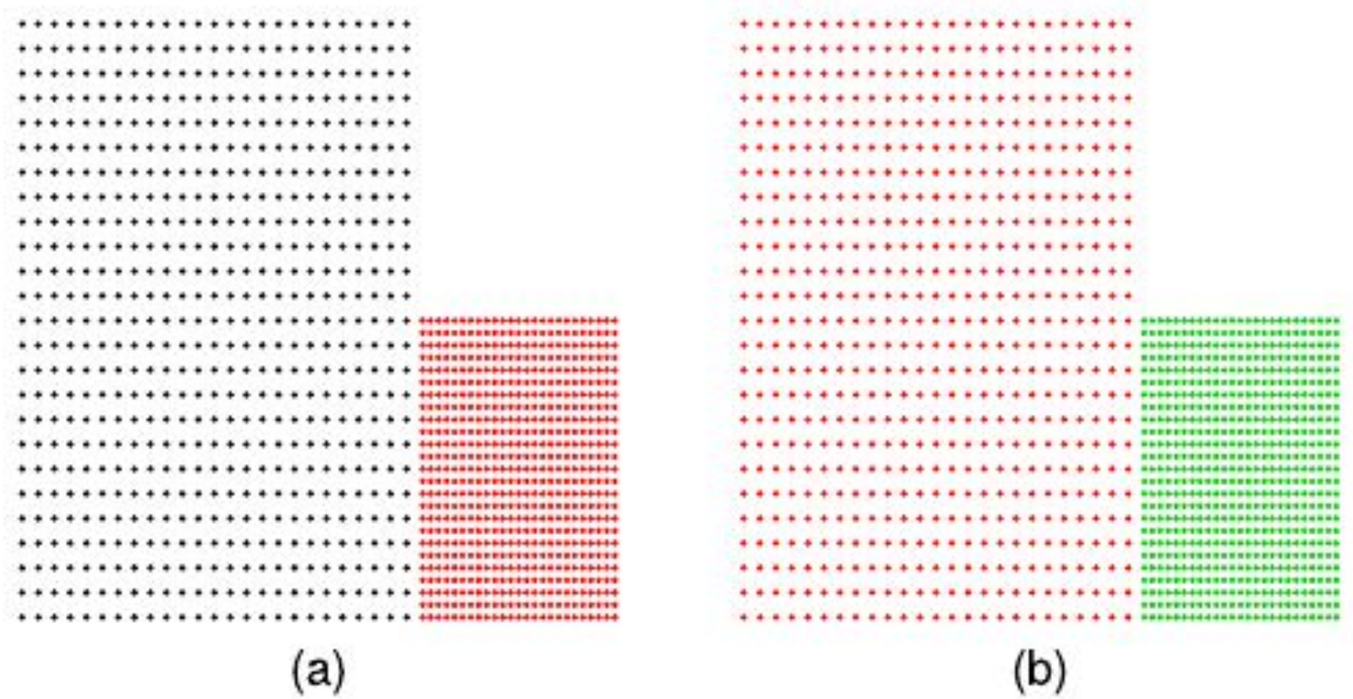


Fig. 7. *DBSCAN* (a) and *RNN-DBSCAN* (b) clustering results (maximum ARI solution) for the *grid* dataset. Note that observations colored black were identified as noise by the clustering.

DBSCAN vs RNN-DBSCAN Complexity

- DBSCAN $O(n^2)$ naive
  - O(nlogn) with fast index (R* tree, K-d tree)
- RNN methods dependent on all k nearest neighbor problem
  - $O(kn^2)$ naive
  - $O(\eta \log n)$ with Cover tree (where η is dependent on dimensionality)
  - Approximate methods faster but less reliable
  - This is an area of active research
- DBSCAN dependent on fixed radius nearest neighbor problem
  - $O(n^2)$ naive, O(logn) with index

NN-Descent

- Iterative Nearest Neighbor approximation algorithm
- Takes p and d parameters
  - p : Neighborhood sampling rate
  - d :minimum changes per iteration
- $O(n^{1.3})$ empirical performance
- $O(pnk^2)$ per iteration theoretical
- Paper uses p=0.1, d=0.001, and 12 iterations max
  - My results with 12 are wildly inaccurate
  - Library I used has high overhead compared to more direct (index based) approaches.

A Few More Informal Definitions

- Adjusted Rand Index (ARI) -- “the similarity measure between two clusterings that is adjusted for chance and is related to accuracy”
- Normalized Mutual Information (NMI) -- “amount of information obtained about one clustering though the other”
  - Their mutual dependence
- Purity -- “Weighted average of percentage of observations belonging to the dominant class in each cluster”

Methodology

- Parameter search space for RNN methods: 1<=k<=100
- Parameter search space for DBSCAN:
  - min_{pts} from {1,5,10,20}
  - eps selected over the set of eps values equal to the min_{pts} nearest neighbor distance of each observation
- Two sets of parameters were selected which maximize ARI and NMI respectively
- Only euclidean distance was used

Datasets

TABLE 1
Artificial Datasets

Data	Observations	Classes	Dimensions
aggregation [15]	788	7	2
d31 [15]	3100	31	2
flame [15]	240	2	2
jain [15]	373	2	2
pathbased [15]	300	3	2
r15 [15]	600	15	2
spiral [15]	312	3	2
grid [15]	1250	2	2
blobs [16]	1K,10K,100K,1M	5	3
circle [16]	1K,10K,100K,1M	2	2
moons [16]	1K,10K,100K,1M	2	2
swissroll [16]	1K,10K,100K,1M	2	3

TABLE 2
Real-World Datasets

Data	Observations	Classes	Dimensions
banknote [17]	1372	2	4
ctg [17]	2126	10	19
digits [17]	1,797	10	64
ecoli [17]	336	8	7
htru2 [17], [19]	17,898	2	8
iris [17]	150	3	4
seeds [17]	210	3	7
farm [18], [20]	3,627,086	-	4
house [17], [18]	2,049,280	-	6

Results (Artificial)

TABLE 3
ARI Performance on Artificial Datasets

Data		RNN	REC	IS	ISB	DBS	OPT
aggr	ari	**0.998**	0.752	0.872	0.914	0.994	0.979
	clu	7	7	6	6	7	8
	pur	0.999	1.0	0.956	0.956	0.999	0.987
	noi	0	163	34	0	2	0
d31	ari	**0.896**	0.539	0.71	0.739	0.868	0.874
	clu	31	38	34	43	31	60
	pur	0.975	0.928	0.901	0.861	0.982	0.95
	noi	167	1051	492	244	286	0
flam	ari	**0.971**	0.631	0.682	0.215	0.944	0.928
	clu	2	2	2	23	2	3
	pur	0.996	0.995	0.981	1.0	0.992	0.983
	noi	2	43	31	33	4	0
jain	ari	0.983	0.417	0.819	**1.0**	0.941	**1.0**
	clu	2	2	2	2	4	2
	pur	1.0	1.0	1.0	1.0	1.0	1.0
	noi	2	115	34	0	1	0
path	ari	**0.917**	0.763	0.759	0.789	0.655	0.684
	clu	3	3	5	5	10	7
	pur	0.99	1.0	0.986	0.989	0.986	0.957
	noi	11	50	21	16	11	0
r15	ari	0.984	0.751	0.807	**0.993**	0.979	0.956
	clu	15	14	15	15	15	16
	pur	0.995	0.932	0.986	0.997	0.995	0.977
	noi	3	103	91	0	6	0
spir	ari	**1.0**	**1.0**	0.947	**1.0**	**1.0**	0.653
	clu	3	3	3	3	3	6
	pur	1.0	1.0	1.0	1.0	1.0	0.888
	noi	0	0	11	0	0	0
grid	ari	**1.0**	0.922	0.994	0.997	0.5	0.997
	clu	2	2	2	2	1	3
	pur	1.0	1.0	0.999	0.999	1.0	0.999
	noi	0	50	2	0	625	0

TABLE 4
NMI Performance on Artificial Datasets

Data	RNN	REC	IS	ISB	DBS	OPT
aggr	**0.996**	0.742	0.888	0.954	0.991	0.969
d31	**0.934**	0.772	0.844	0.879	0.911	0.921
flam	**0.931**	0.54	0.567	0.362	0.869	0.875
jain	0.97	0.376	0.709	**1.0**	0.862	**1.0**
path	**0.872**	0.706	0.735	0.772	0.704	0.686
r15	0.988	0.881	0.871	**0.994**	0.984	0.964
spir	**1.0**	**1.0**	0.917	**1.0**	**1.0**	0.685
grid	**1.0**	0.824	0.983	0.991	0.301	0.991

Results (Real)

TABLE 5
ARI Performance on Real-World Datasets

Data		RNN	REC	IS	ISB	DBS	OPT
bank	ari	**0.771**	0.086	0.596	0.594	0.558	0.225
	clu	3	4	2	3	8	35
	pur	0.985	0.828	0.894	0.896	0.896	0.98
	noi	34	580	25	10	1	0
ctg	ari	0.951	0.057	0.883	0.902	**0.992**	0.892
	clu	10	14	6	9	13	17
	pur	1.0	0.372	0.999	0.999	1.0	0.995
	noi	91	796	409	179	5	0
digi	ari	**0.739**	0.011	0.462	0.695	0.684	0.315
	clu	34	3	25	18	21	29
	pur	0.936	0.245	0.957	0.977	0.983	0.733
	noi	104	1524	564	298	355	0
ecol	ari	0.526	0.14	0.474	0.46	**0.639**	0.591
	clu	8	2	4	3	3	5
	pur	0.736	0.538	0.714	0.711	0.582	0.708
	noi	10	89	63	55	100	0
htru	ari	0.334	0.146	0.147	0.166	**0.552**	0.146
	clu	204	56	310	204	4	26
	pur	0.976	0.915	0.981	0.949	0.977	0.976
	noi	236	1909	4206	2270	2289	0
iris	ari	0.644	0.289	0.566	0.568	**0.703**	0.643
	clu	4	2	2	2	7	4
	pur	0.963	0.674	0.671	0.667	0.978	0.847
	noi	16	55	1	0	16	0
seed	ari	**0.617**	0.416	0.383	0.361	0.491	0.498
	clu	4	3	2	9	4	6
	pur	0.898	0.903	0.653	0.888	0.95	0.857
	noi	4	65	34	22	51	0

TABLE 6
NMI Performance on Real-World Datasets

Data	RNN	REC	IS	ISB	DBS	OPT
bank	**0.68**	0.213	0.59	0.585	0.579	0.363
ctg	0.934	0.399	0.803	0.886	**0.99**	0.902
digi	**0.824**	0.47	0.648	0.775	0.77	0.67
ecol	0.569	0.538	0.551	0.571	**0.6**	0.55
htru	0.195	0.116	0.117	0.109	**0.25**	0.109
iris	0.683	0.445	0.723	**0.734**	**0.734**	**0.734**
seed	**0.618**	0.48	0.487	0.495	0.533	0.525

Approximation Accuracy

TABLE 7
Approximate k Nearest Neighbors Results

Data	Scan Rate	Recall	$\mathbf{ARI}_{k=10}$	$\mathbf{ARI}_{k=100}$
blobs	0.0038	0.996	0.998	0.999
circle	0.0042	0.997	0.973	0.998
moons	0.0042	0.997	0.982	0.998
swissroll	0.0038	0.997	0.982	0.997
farm	0.0013	0.999	0.942	0.982
house	0.0022	0.996	0.978	0.988

Summary

- RNN-DBSCAN is a simple and effective RNN clustering algorithm
- It addresses parameter complexity and multiple density performance issues of existing methods
- Provided you have an efficient nearest neighbor search algorithm and/or indexing scheme then the algorithm is also moderately efficient

References

1. A. Bryant and K. Cios, "RNN-DBSCAN: A Density-Based Clustering Algorithm Using Reverse Nearest Neighbor Density Estimates," in IEEE Transactions on Knowledge and Data Engineering, vol. 30, no. 6, pp. 1109-1121, 1 June 2018. doi: 10.1109/TKDE.2017.2787640
2. Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. 1996. A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining* (KDD'96), Evangelos Simoudis, Jiawei Han, and Usama Fayyad (Eds.). AAAI Press 226-231.
3. https://hipparchus.org/hipparchus-clustering/index.html
4. Evan Lutins, “DBSCAN: What is it? When to Use it? How to use it.” Medium.com, Sep 5 2017 https://medium.com/@elutins/dbscan-what-is-it-when-to-use-it-how-to-use-it-8bd506293818