# OPTICS: Ordering Points To Identify the Clustering Structure

Mihael Ankerst, Markus M. Breunig, Hans-Peter Kriegel, Jörg Sander

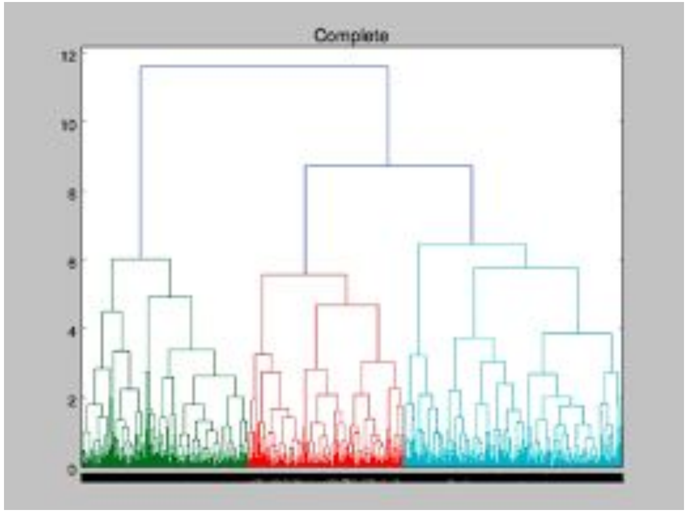
# В чем мотивация?

Еще до статьи существовало множество алгоритмов кластеризации, но их объединяли 3 связанные друг с другом проблемы:

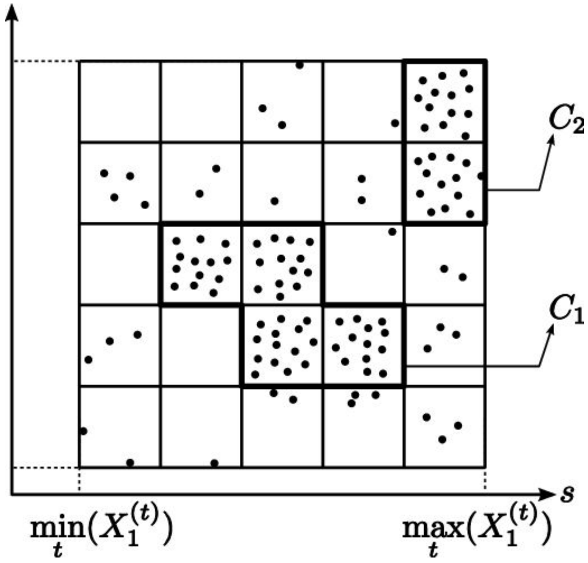
- Они требуют гиперпараметры (например количество кластеров), которые довольно сложно подобрать, особенно для высокоразмерных данных
- При этом модель еще очень чувствительна к подбору гиперпараметров
- И высокоразмерные данные довольно часто имеют какое-то смещенное распределение, которое не описывается одним набором гиперпараметров.

# Обзор предыдущих решений

Иерархические(Single-link)

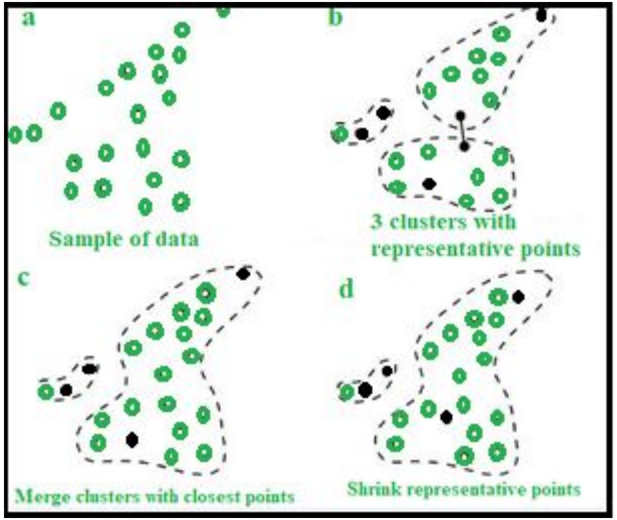



Grid-based clustering

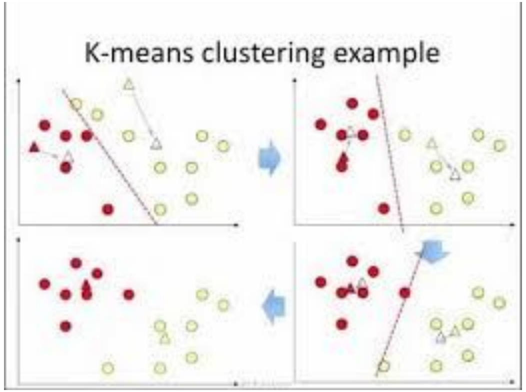


# Обзор предыдущих решений

CURE

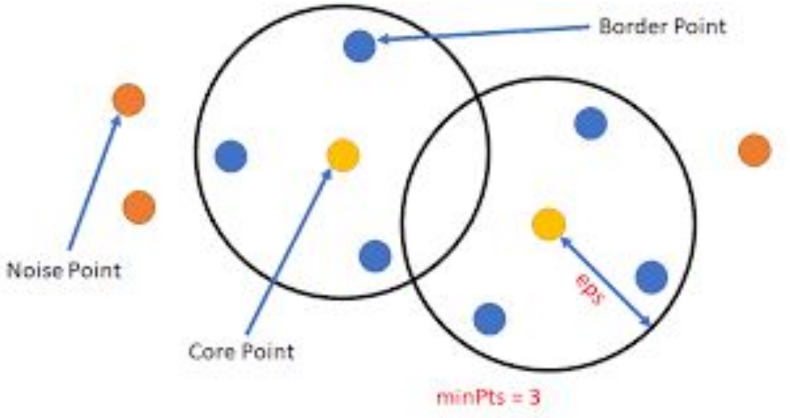



K-means (K-modes, K-medoids)

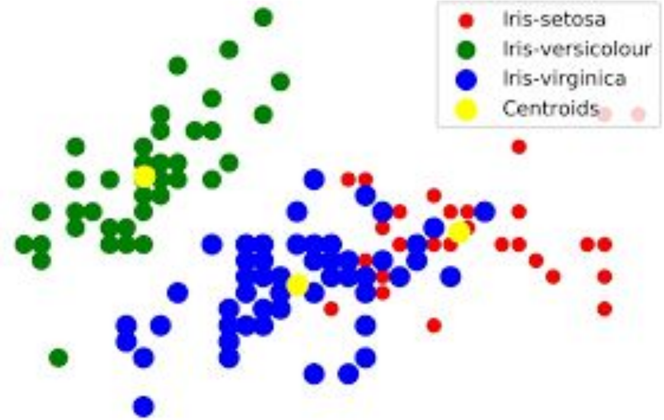


# Обзор существующих решений

DBSCAN

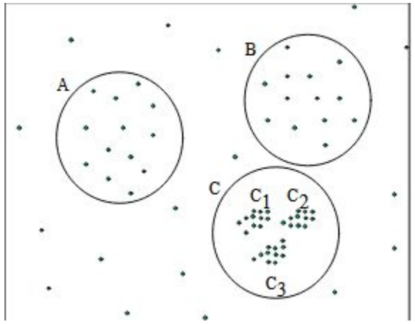


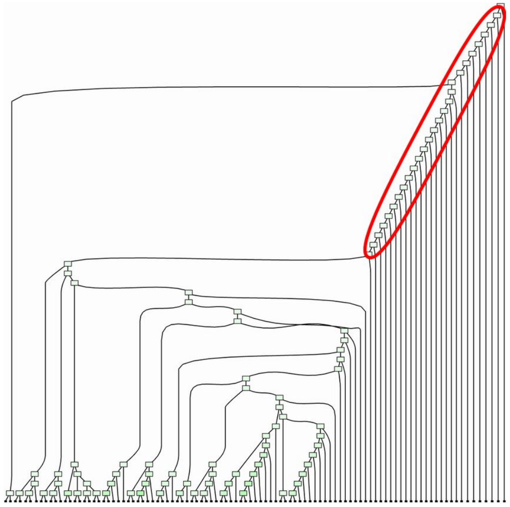

BIRCH



# Предпосылки к оптиксу

1. Почему мы просто не используем какую глобальную плотность?
    a. Потому, что мы таким образом утверждаем, что у нас все кластеры одинаковой примерно плотности, и возникает проблема нахождения этой лучшей плотности
2. Использовать другие методы, например single-link
    a. Конкретно single-link, имеет плохую привычку строить очень длинные, но тонкие кластеры (хотя в некоторых случаях это может быть полезно)
    b. И эвристические методы в целом сложнее анализировать



# Определения

$\epsilon$ — максимальное расстояние для создания кластера

$MinPts$ — минимальное кол-во точек в $\epsilon$-окрестности

$N_\epsilon(p)$ — точки в $\epsilon$-окрестности

$D$ — множество всех точек

---

Directly Density-Reachable

Объект $p$ напрямую достижим из объекта $q$, если:

1. $p \in N_\epsilon(q)$
2. $|N_\epsilon(q)| \geq MinPts$ (условие ядрового объекта)

Density-Reachable

Объект $p$ достижим из объекта $q$,

если существует набор $p_1, \ldots, p_n$, таких, что:

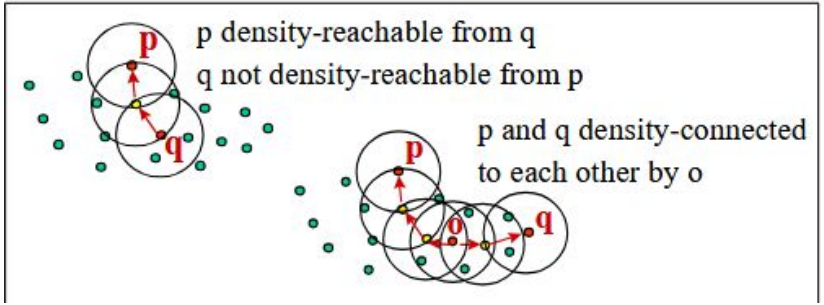
1. $p_1 = p, p_n = q$
2. $p_i$ достижим из $p_{i+1}$

Density-Connected

Объект $p$ связан с объектом $q$, если:

существует такой объект $o$,

что из $o$ достижим и $p$, и $q$



# Определения

Cluster and noise

Множество $C$ называется кластером, если:

1. $\forall p, q \in D$, если $p \in C$ и $q$ достижимо из $p$, то $q \in C$
2. $\forall p, q \in C$, $p$ и $q$ связаны

Любой объект, не принадлежащий ни одному кластеру является шумом

Core Distance

$$cd_{\epsilon,MinPts}(p) = \begin{cases} UNDEFINED, |N_\epsilon(p)| < MinPts \\ distance_{MinPts}(p), \text{otherwise} \end{cases}$$

Где $distance_{MinPts}(p)$ дистанция до $MinPts$–ближайшей точки от $p$

Ядровая дистанция - по факту минимальный эпсилон, при котором P могла быть ядровой точкой

Reachability Distance

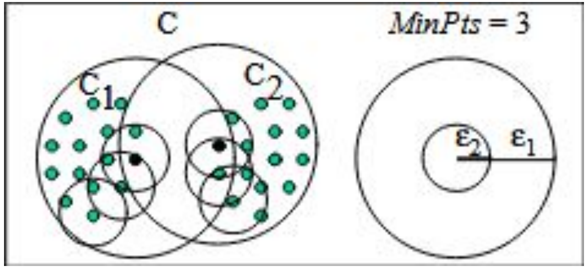
$$rd_{\epsilon,MinPts}(p, o) = \begin{cases} UNDEFINED, |N_\epsilon(o)| < MinPts \\ max(cd(o), distance(o, p), \text{otherwise} \end{cases}$$

Reachability distance - по факту минимальный эпсилон, при котором P могла быть напрямую достижима из O

# Ordering

Идея OPTICS в том, что кластеры с меньшим эпсилон (большей плотностью) будут полностью подкластерами кластеров с меньшей плотностью.

Из этого следует идея, о том, что мы можем как-нибудь упорядочить точки, чтобы в итоге мы рассматривали кластеры с наибольшей плотностью раньше, чтобы например потом при использовании измененного DBSCAN’а, мы могли с любой плотностью создавать кластеры



# Сам алгоритм O(n * |время обработки одной окрестности|). В худшем: O(n^2)

```
function OPTICS(DB, ε, MinPts) is
    for each point p of DB do
        p.reachability-distance = UNDEFINED
    for each unprocessed point p of DB do
        N = getNeighbors(p, ε)
        mark p as processed
        output p to the ordered list
        if core-distance(p, ε, MinPts) != UNDEFINED then
            Seeds = empty priority queue
            update(N, p, Seeds, ε, MinPts)
            for each next q in Seeds do
                N' = getNeighbors(q, ε)
                mark q as processed
                output q to the ordered list
                if core-distance(q, ε, MinPts) != UNDEFINED do
                    update(N', q, Seeds, ε, MinPts)
```

```
function update(N, p, Seeds, ε, MinPts) is
    coredist = core-distance(p, ε, MinPts)
    for each o in N
        if o is not processed then
            new-reach-dist = max(coredist, dist(p,o))
            if o.reachability-distance == UNDEFINED then // o is not in Seeds
                o.reachability-distance = new-reach-dist
                Seeds.insert(o, new-reach-dist)
            else                // o in Seeds, check for improvement
                if new-reach-dist < o.reachability-distance then
                    o.reachability-distance = new-reach-dist
                    Seeds.move-up(o, new-reach-dist)
```

# Extract

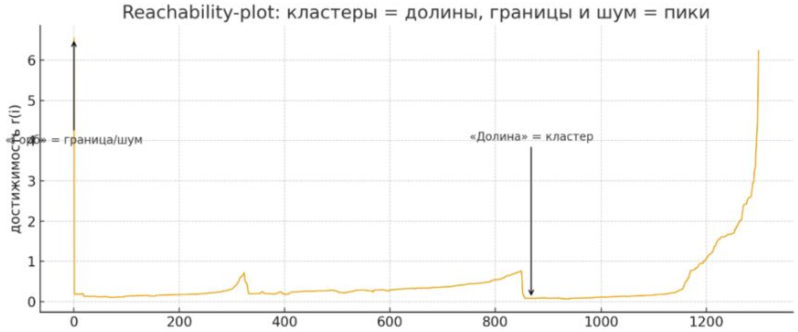
```
ExtractDBSCAN-Clustering (ClusterOrderedObjs,ε', MinPts)
// Precondition: ε' ≤ generating dist ε for ClusterOrderedObjs
   ClusterId := NOISE;
   FOR i FROM 1 TO ClusterOrderedObjs.size DO
      Object := ClusterOrderedObjs.get(i);
      IF Object.reachability_distance > ε' THEN
         // UNDEFINED > ε
         IF Object.core_distance ≤ ε' THEN
            ClusterId := nextId(ClusterId);
            Object.clusterId := ClusterId;
         ELSE
            Object.clusterId := NOISE;
      ELSE    // Object.reachability_distance ≤ ε'
         Object.clusterId := ClusterId;
END; // ExtractDBSCAN-Clustering
```

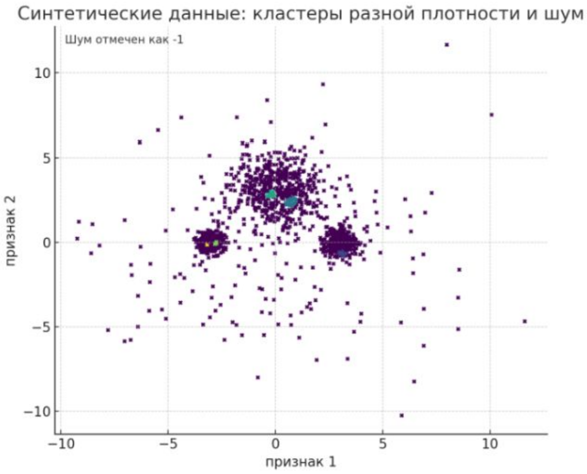
# Визуализация кластерной структуры данных методом OPTICS и производные графические техники

## 1. Что выдаёт OPTICS ?

**упорядочивание** объектов $o : \{1..n\} \to DB$;

и **значения достижимостей** $r : \{1..n\} \to \mathbb{R}_{\geq 0}$.



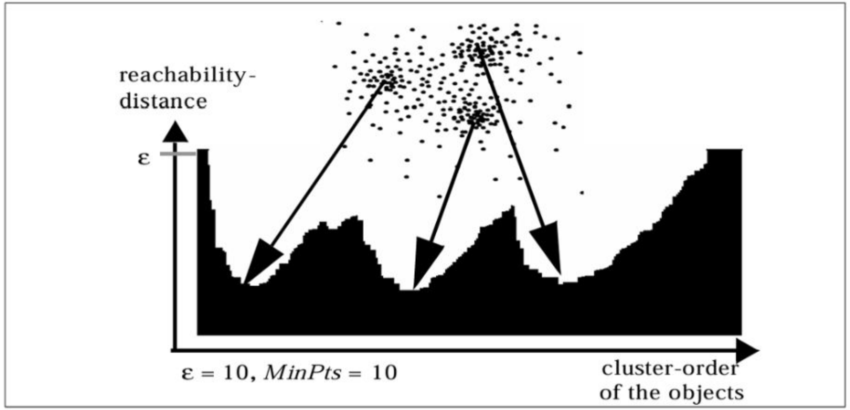
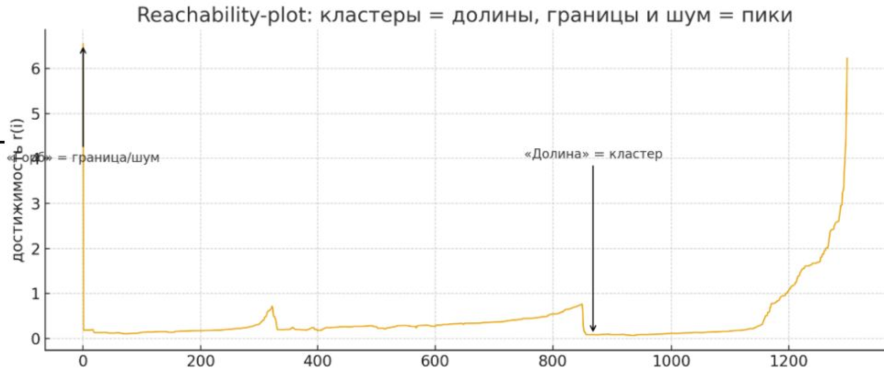




**Figure 9. Illustration of the cluster-ordering**

## 2. Параметры и их влияние



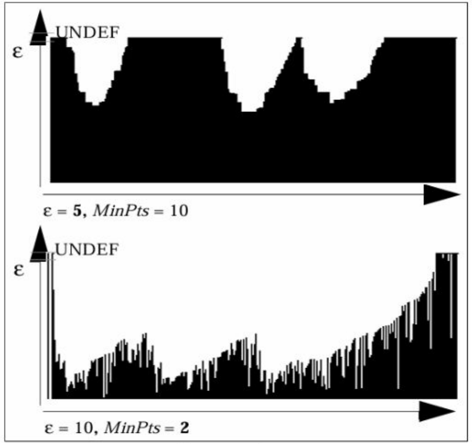




**Figure 10. Effects of parameter settings on the cluster-ordering**

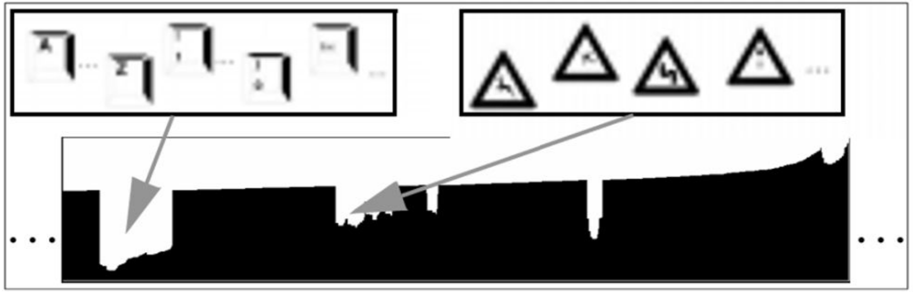


**Figure 11. Part of the reachability-plot for 1,024-*d* image data**

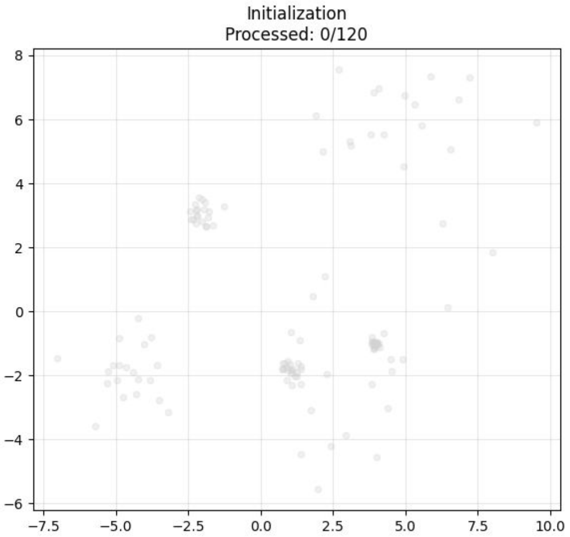
$\varepsilon$ — максимальный радиус

**MinPts** — минимальное число точек в окрестности

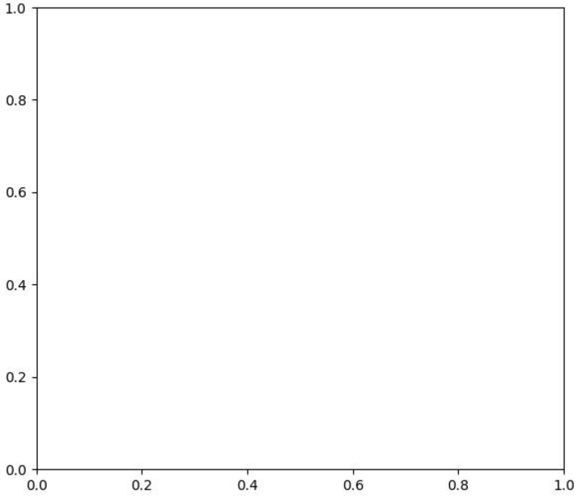
$$Volume_{S(r)} = \frac{\sqrt{\pi^d}}{\Gamma(\frac{d}{2}+1)} \times r^d$$

$$r = \sqrt[d]{\frac{Volume_{DS} \times k \times \Gamma(\frac{d}{2}+1)}{N \times \sqrt{\pi^d}}}.$$

$$Volume_S = \frac{Volume_{DS}}{N} \times k$$

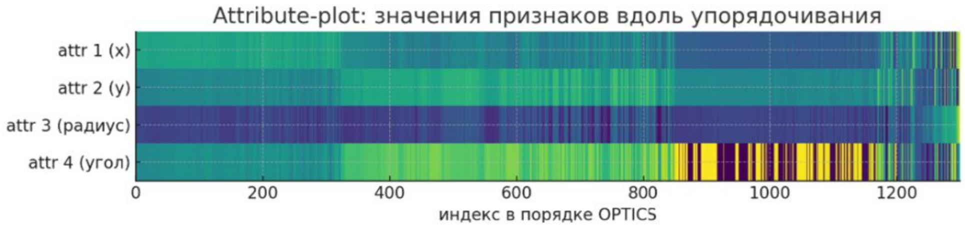


Initialization
Processed: 0/120
8
6
4
2
0
−2
−4
−6
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0

1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

# 3. Attribute-plot



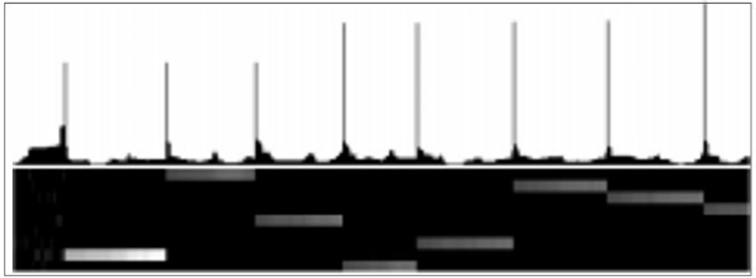



**Figure 13. Reachability-plot and attribute-plot for 9-*d* data from weather stations**

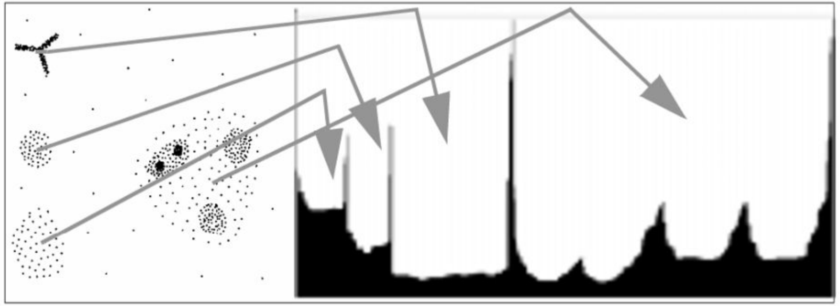


**Figure 12. Reachability-plots for a data set with hierarchical clusters of different sizes, densities and shapes**

## 4. Большие данные и *Circle Segments*

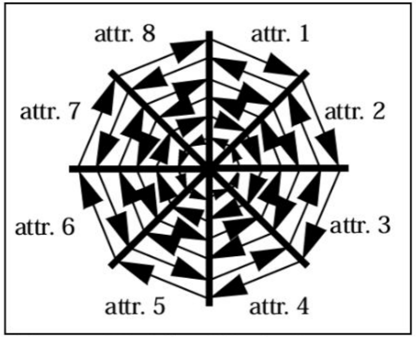



**Figure 14. The Circle Segments technique for 8-*d* data**

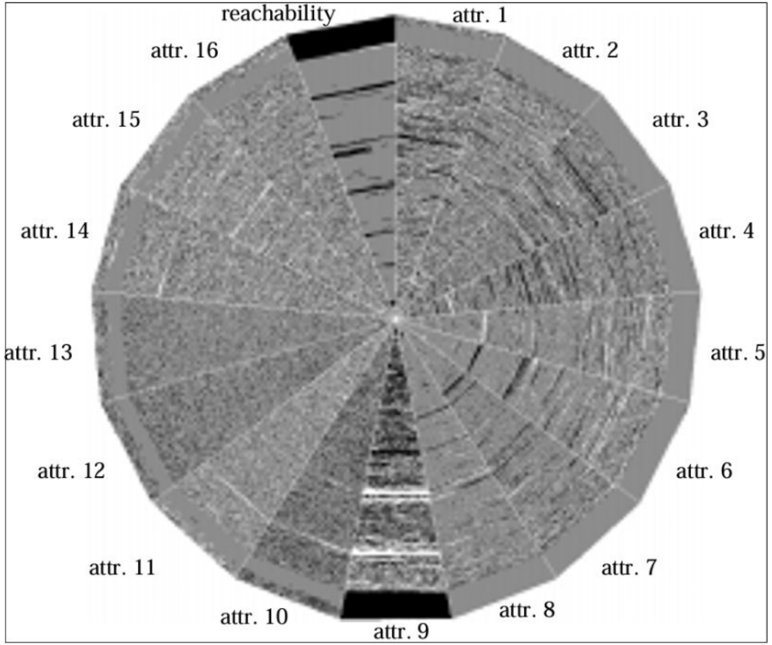



**Figure 15. Clustering structure of 30,000 16-*d* objects**

1) круг разбивают на сегменты — по одному на атрибут (рис. 14);
2) упорядочивание берут из OPTICS (это и есть «сортировка»);
3) значения (DV) кодируют яркостью/цветом;
4) вводится параметр Resolution;
5) используется градация серого, чтобы прогрессия яркости коррелировала с достижимостью (тёмное — малые расстояния, т.е. плотные «внутри кластера» участки).

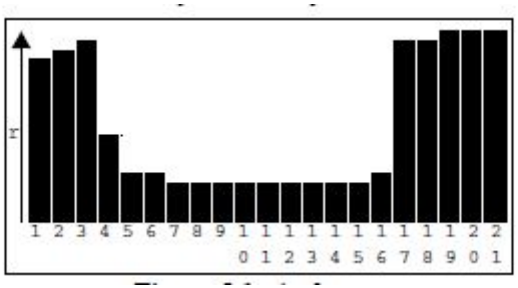
# Automatic Techniques in OPTICS

- OPTICS строит *reachability plot* — график достижимости.

- Кластеры выглядят как «долины» — низкие значения → высокая плотность.

- Раньше кластеры выделяли вручную.

- Цель — научить алгоритм делать это автоматически.



# Formal Cluster Boundaries: Steep Points and Areas

- Если точка резко падает вниз по значению достижимости — это называется ξ-steep downward point, то есть точка резкого спада. Это признак начала кластера — график уходит вниз, значит плотность растёт.

- Если точка наоборот резко поднимается вверх, это ξ-steep upward point — точка резкого подъёма. Это значит, что плотность уменьшается, и мы выходим из кластера.

- Steep Area — последовательность таких точек.

- Кластер = область между спадом и подъёмом.

**Definition 9:** (ξ-steep points)

A point $p \in \{1 \ldots n-1\}$ is called a ξ-steep upward point iff it is ξ% lower than its successor:

$$UpPoint_{\xi}(p) \Leftrightarrow r(p) \le r(p+1) \times (1-\xi)$$

A point $p \in \{1 \ldots n-1\}$ is called a ξ-steep downward point iff $p$'s successor is ξ% lower:

$$DownPoint_{\xi}(p) \Leftrightarrow r(p) \times (1-\xi) \le r(p+1)$$

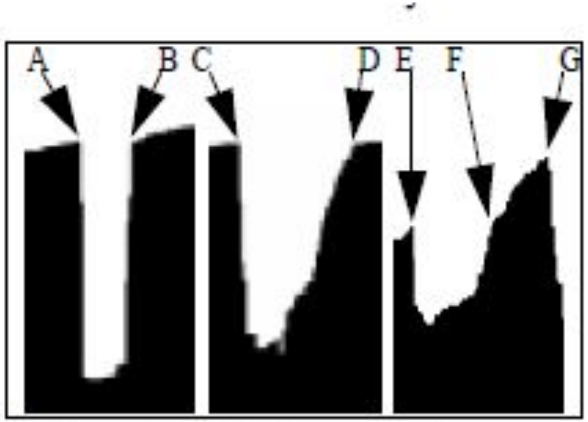



**Figure 17. Real world clusters**

# Formal Cluster Boundaries: Steep Points and Areas

**Definition 10:** ($\xi$-steep areas)

An interval $I = [s, e]$ is called a $\xi$-steep upward area $UpArea_\xi(I)$ iff it satisfies the following conditions:

- $s$ is a $\xi$-steep upward point: $UpPoint_\xi(s)$

- $e$ is a $\xi$-steep upward point: $UpPoint_\xi(e)$

- each point between $s$ and $e$ is at least as high as its predecessor:

$$\forall x, s < x \leq e : r(x) \geq r(x-1)$$

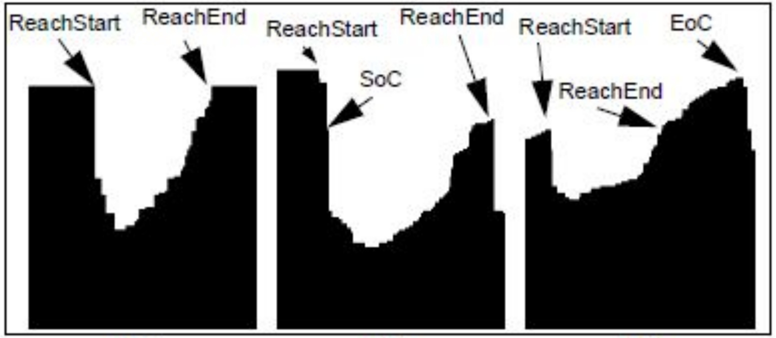
- $I$ does not contain more than *MinPts* consecutive points that are not $\xi$-steep upward:

$$\forall [\bar{s}, \bar{e}] \subseteq [s, e] : ((\forall x \in [\bar{s}, \bar{e}] : \neg UpPoint_\xi(x)) \Rightarrow \bar{e} - \bar{s} < MinPts)$$

- $I$ is maximal: $\forall J : (I \subseteq J, UpArea_\xi(J) \Rightarrow I = J)$

A $\xi$-steep downward area is defined analogously ($DownArea_\xi(I)$).

# ξ-cluster




(a) (b) (c)

**Figure 18. Three different types of clusters taken from an industrial parts data set**

**Definition 11:** (ξ-cluster)

An interval $C = [s, e] \subseteq [1, n]$ is called a ξ-cluster iff it satisfies conditions 1 to 4:

$\text{cluster}_\xi(C) \Leftrightarrow \exists D = [s_D, e_D], U = [s_U, e_U]$ with

1) $DownArea_\xi(D) \wedge s \in D$

2) $UpArea_\xi(U) \wedge e \in U$

3) a) $e - s \geq MinPts$

b) $\forall x, s_D < x < e_U : (r(x) \leq min(r(s_D), r(e_U)) \times (1 - \xi))$

4) $(s,e) =$

$$\begin{cases} (max\{x \in D \mid r(x) > r(e_U + 1)\}, e_U) & \text{if } r(s_D) \times (1 - \xi) \geq r(e_U + 1) & \text{(b)} \\ (s_D, min\{x \in U \mid r(x) < r(s_D)\}) & \text{if } r(e_U + 1) \times (1 - \xi) \geq r(s_D) & \text{(c)} \\ (s_D, e_U) & \text{otherwise} & \text{(a)} \end{cases}$$

# *An Efficient Algorithm To Compute All ξ-Clusters*

Теперь, когда у нас есть формальное определение,
нужно разработать алгоритм, который сможет **найти все такие кластеры автоматически**.

- Алгоритм проходит по графику, ищет пары Down/Up и проверяет условия.
- Благодаря оптимизациям работает эффективно даже на больших данных.

above. The complete algorithm is shown in figure 19. Whenev-

```
SetOfSteepDownAreas = EmptySet
SetOfClusters = EmptySet
index = 0, mib = 0
WHILE(index < n)
  mib = max(mib, r(index))
  IF(start of a steep down area D at index)
    update mib-values and filter SetOfSteepDownAreas(*)
    set D.mib = 0
    add D to the SetOfSteepDownAreas
    index = end of D + 1; mib = r(index)
  ELSE IF(start of steep up area U at index)
    update mib-values and filter SetOfSteepDownAreas
    index = end of U + 1; mib = r(index)
    FOR EACH D in SetOfSteepDownAreas DO
      IF(combination of D and U is valid AND(**)
        satisfies cluster conditions 1, 2, 3a)
          compute [s, e] add cluster to SetOfClusters
  ELSE index = index + 1
RETURN(SetOfClusters)
```

**Figure 19. Algorithm ExtractClusters**

# Experiments

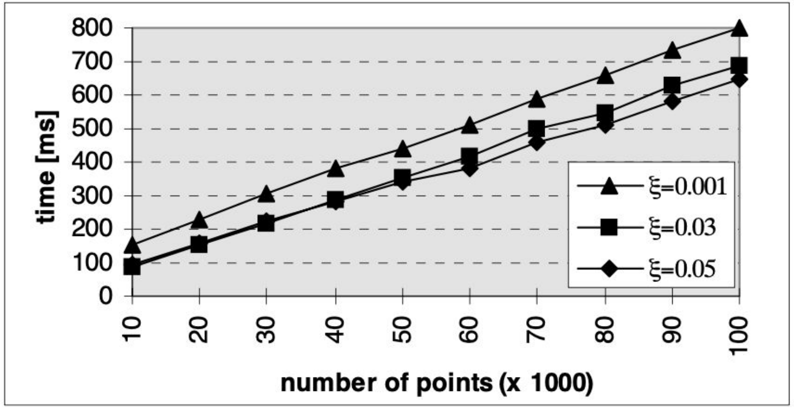



Figure 20. Scale-up of the ξ-clustering algorithm for the 64-*d* color histograms

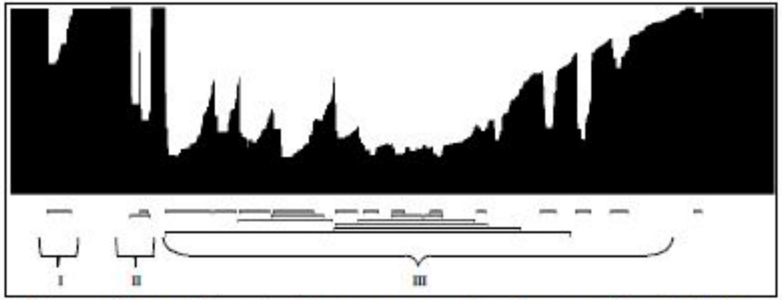



Figure 22. 64-*d* color histograms auto-clustered for ξ=0.02

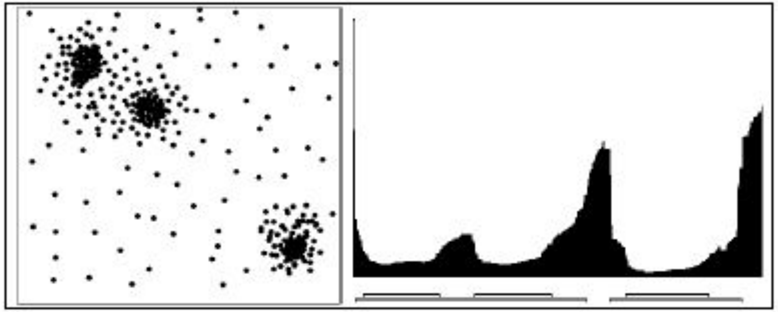


Figure 21. 2-*d* synthetic data set (left), the reachability-plot (right) and ξ=0.09-clustering (below the reachability-plot)