Ludwig-Maximilians-Universität München
Institut für Informatik
Dr. Peer Kröger, Dr. Matthias Schubert
Markus Bundschus

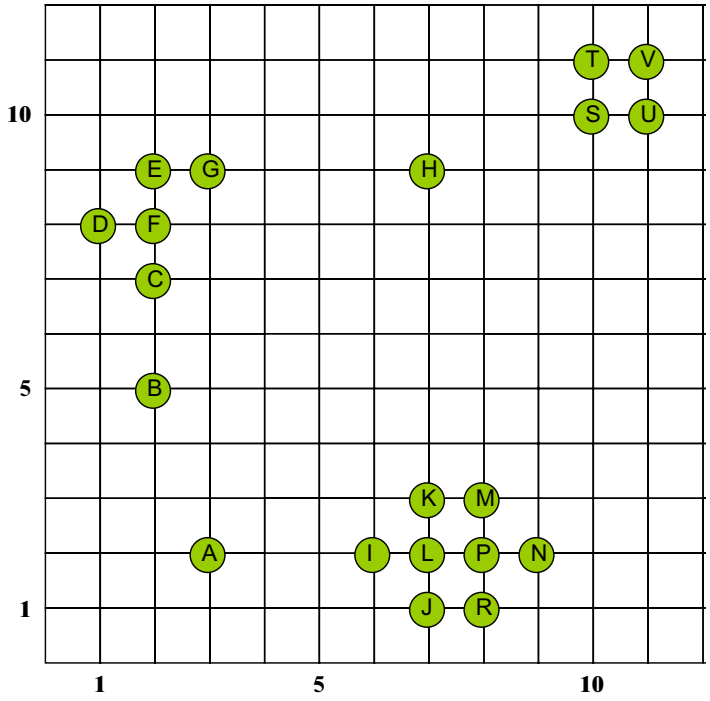
Knowledge Discovery in Databases

WS 2007/08

Übungsblatt 9

Aufgabe 9-1 OPTICS

Gegeben sei der folgende 2-dimensionale Datensatz:



Verwenden Sie als Distanzfunktion zwischen den Punkten wieder die Manhattan-Distanz (L_1-Norm)

Erzeugen Sie mit OPTICS (Pseudocode am Ende des Übungsblattes) jeweils ein Erreichbarkeitsdiagramm für die folgenden Parameter:

(a) $\epsilon = 5$ und $MinPts = 2$

(b) $\epsilon = 5$ und $MinPts = 4$

(c) $\epsilon = 2$ und $MinPts = 4$

(d) $\epsilon = \infty$ und $MinPts = 4$

(e) Diskutieren Sie, welche Auswirkungen die Parameter $MintPts$ und ϵ haben.

Aufgabe 9-2 Outlier Detection

Gegeben der Datensatz und die Distanzfunktion aus Aufgabe 1. Berechnen Sie für die Punkte H und L den LOF-Wert für $MinPts = 3$.

Aufgabe 9-3 Apriori-Algorithmus

Gegeben ist die Menge der Items $I = \{A, B, C, D, E, F, G, H, I, K, L, M\}$.

Weiterhin ist eine Menge von Transaktionen T laut folgender Tabelle gegeben:

Menge der Transaktionen T

Transaktions ID	gekaufte Items
1	B E G H
2	A B C E G H
3	A B C E F H
4	B C D E F G H L
5	A B E K H
6	B E F G H I K
7	A B D G H
8	A B D G
9	B D F G
10	C E F
11	A C E F H
12	A B E G

Bestimmen Sie zum minimalen Support von 30% die häufig auftretenden Itemsets. Verwenden Sie dazu den Apriori-Algorithmus. Geben Sie insbesondere die Kandidatenmengen nach den Join-Schritten und nach den Prune-Schritten an, sowie die häufig auftretenden Itemsets mit ihrem jeweiligen Support.

Pseudocode OPTICS

```
seedlist = ∅   // implemented as a heap
for i = 0 to n-1 do
    if(seedlist = ∅ ) then seedlist = {(random_not_handled_point, ∞)}
    (x, x.reach) = get_and_remove_point_with_min_reach(seedlist)
    x.pos = i
    x.handled = TRUE
    neighbors = rangeQuery(x, ε)
    x.core = nnDist(x, neighbors, MinPts)
    if(x.core < ∞)
        for each y ∈ neighbors with not(y.handled)
            if( y ∉ seedlist)  seedlist = seedlist ∪ {(y, reach-dist(y,x))}
            else
                curr_reach = lookup(seedlist, y)
                update(y, min(curr_reach, reach-dist(y,x)))
        endfor
endfor
```