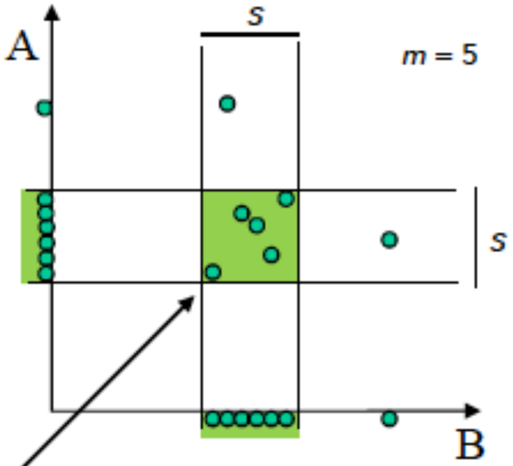
KDD II – Exercise 4

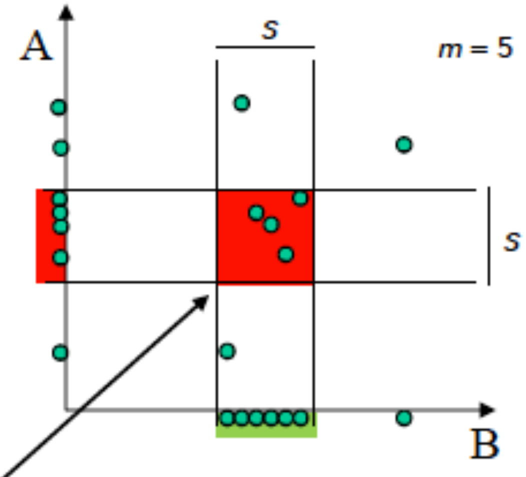
12.06.2017

Downward-closure property: example

- Simple cluster criterion (density of grid cells):
  - If a cell *C* of side length *s* contains more than *m* points, it represents a cluster
- Monotonicity:
  - if *C* contains more than *m* points in subspace *S* then *C* also contains more than *m* points in any subspace $T \subset S$
  - Example: monotonicity (left) and reverse implication (right)



Cell *C* contains more than *m*=5 points in subspace „AB"
=> Also in subspaces „A"⊂ „AB" and „B"⊂ „AB"

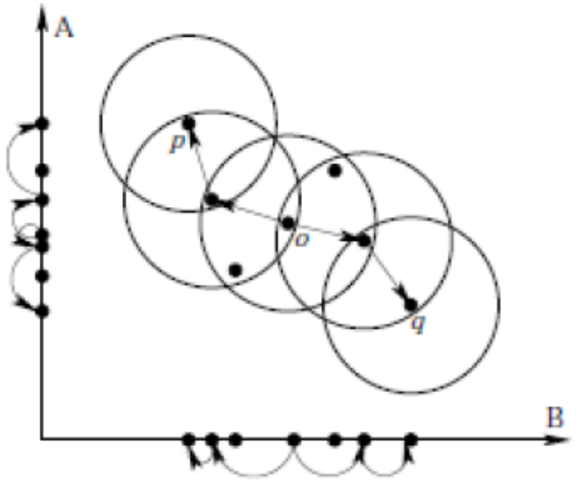


Cell *C* contains less than *m*=5 points in subspace „A"
=> Also in subspace „AB"

SUBCLU: Downward closure of density connected sets 3/6

If C is a density connected set in subspace S then C is a density connected set in any subspace $T \subset S$.

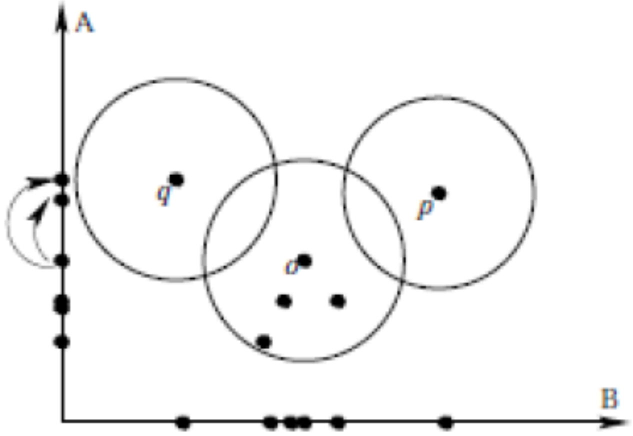
- But, if C is a cluster in S, it need not to be a cluster in $T \subset S$ – maximality might be violated
- All clusters in a higher-dimensional subspace will be subsets of the clusters detected in this first clustering.



(a) p and q are density-connected via o

ε: circles indicate

MinPts = 4



(b) p and q are not density-connected

p and *q* density connected in {A,B}.
Thus, they are also density connected in {A} and {B}

p and *q* not density connected in {B}.
Thus, they are not density connected in{A,B}, although they are density connected in {A}.

SUBCLU: Discussion 6/6

- Algorithm
  - All subspaces that contain any density-connected set are computed using the bottom-up approach (similar to CLIQUE/APRIORI)
  - Density-connected clusters are computed using a specialized DBSCAN run in the resulting subspace to generate the subspace clusters
- Discussion
  - Input: ε and *MinPts* specifying the density threshold
  - Output: all clusters in all subspaces, clusters may overlap
  - Uses a fixed density threshold for all subspaces
  - Advanced but costly cluster model

PREDECON[BKKK04] 1/3

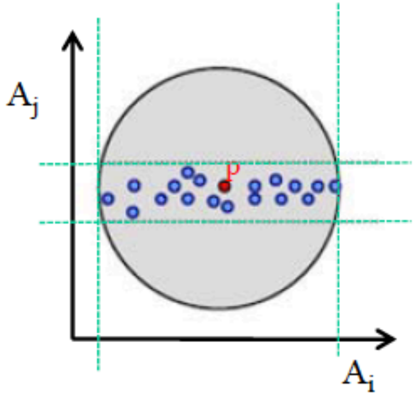
- Instance-based top-down approach: we learn the subspace for each instance
- Extends DBSCAN to high dimensional spaces by incorporating the notion of dimension preferences in the distance function

- For each point p, it defines its subspace preference vector:

$$\bar{\mathbf{w}}_p = (w_1, w_2, \ldots w_d) \qquad w_i = \begin{cases} 1 & if \quad \mathrm{VAR}_i > \delta \\ \kappa & if \quad \mathrm{VAR}_i \leq \delta \end{cases}$$

- VAR_i is the variance along dimension ~~j~~ A_i in $N_\varepsilon(p)$:

$$\mathrm{VAR}_{A_i}(\mathcal{N}_\varepsilon(p)) = \frac{\sum_{q \in \mathcal{N}_\varepsilon(p)} (dist(\pi_{A_i}(p), \pi_{A_i}(q)))^2}{|\mathcal{N}_\varepsilon(p)|}$$



δ, κ (κ>>1) are input parameters

PREDECON[BKKK04] 2/3

- Preference weighted distance function:

w_i

$$dist_p(p,q) = \sqrt{\sum_{i=1}^{d} \frac{1}{w_i} \cdot (\pi_{A_i}(p) - \pi_{A_i}(q))^2}$$

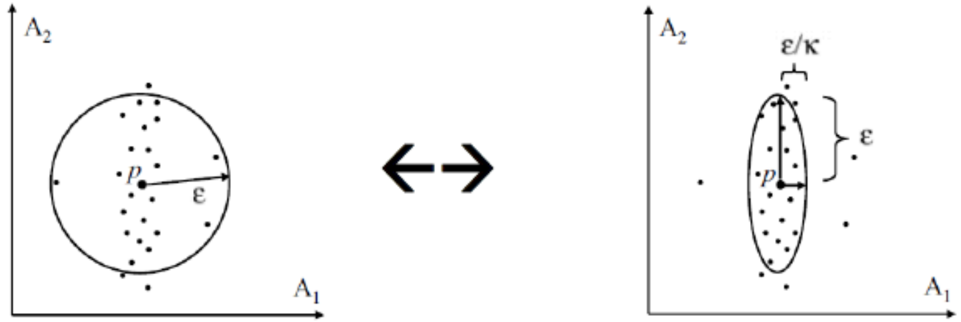
$$dist_{pref}(p,q) = \max\{dist_p(p,q), dist_q(q,p)\}$$

Important dimensions weighted more heavily!

- Preference weighted ε-neighborhood:

$$\mathcal{N}_{\varepsilon}^{\bar{\mathbf{w}}_p}(p) = \{x \in \mathcal{D} \mid dist_{pref}(p,x) \leq \varepsilon\}$$

simple ε-neighborhood



preference weighted ε-neighborhood

PREDECON[BKKK04] 3/3

- Preference weighted core points:

$$\text{CORE}^{\text{pref}}_{\text{den}}(p) \Leftrightarrow \text{PDIM}(\mathcal{N}_\varepsilon(p)) \leq \lambda \wedge |\mathcal{N}_\varepsilon^{\bar{w}_p}(p)| \geq \mu$$

p is core point — Subspace preference dimensionality — Preference weighted neigborhood

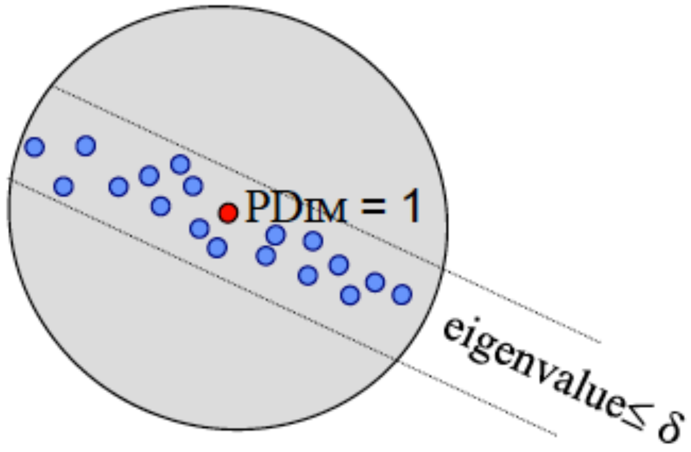
- Direct density reachability, reachability and connectivity are defined based on preference weighted core points
- A *subspace preference cluster* is a maximal density connected set of points associated with a certain subspace preference vector.

4C [BKKZ04]

4C = Computing Correlation Connected Clusters
Idea: Integrate PCA into density-based clustering.

Approach:

- Check the core point property of a point p in the complete feature space
- Perform PCA on the local neighborhood S of p to find subspace correlations



PCA factorizes M_p into $M_p = V\,E\,V^T$

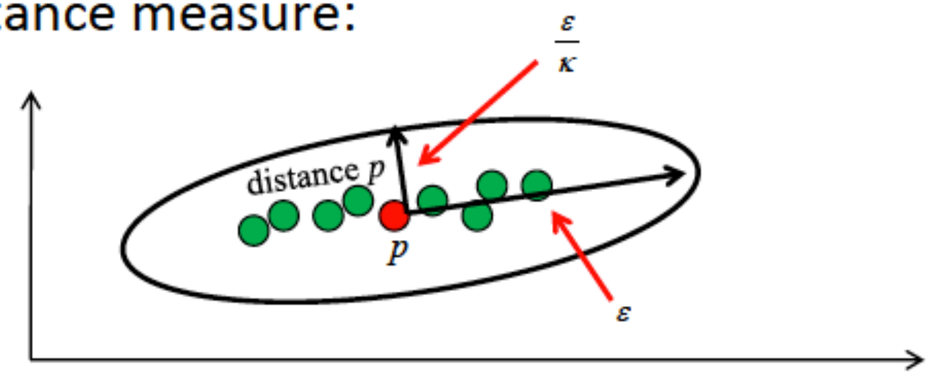
V: eigenvectors

E: eigenvalues

- A parameter δ discerns large from small eigenvalues.
- CorDim(S)=#eigenvalues>δ
- In the eigenvalue matrix of p, large eigenvalues are replaced by 1, small eigenvalues by a value κ >>1 → adapted eigenvalue matrix E'_p

4C: Distance measure

- effect on distance measure:



$$\hat{e}_i = \begin{cases} 1 & if \quad \Omega(e_i) > \delta \\ \kappa & if \quad \Omega(e_i) \leq \delta \end{cases}$$ *where* Ω *is the normalization of the eigenvalues onto* $[0,1]$

- distance of *p* and *q* w.r.t. *p*: $\sqrt{(p-q) \cdot V_p \cdot E'_p \cdot V_p^{\mathrm{T}} \cdot (p-q)^{\mathrm{T}}}$
- distance of *p* and *q* w.r.t. *q*: $\sqrt{(q-p) \cdot V_q \cdot E'_q \cdot V_q^{\mathrm{T}} \cdot (q-p)^{\mathrm{T}}}$

4C: correlation neighbors

- symmetry of distance measure by choosing the maximum:

- *p* and *q* are correlation-neighbors if

$$\max\left\{\begin{array}{l}\sqrt{(p-q)\cdot V_p\cdot E'_p\cdot V_p^{\mathrm{T}}\cdot(p-q)^{\mathrm{T}}},\\ \sqrt{(q-p)\cdot V_q\cdot E'_q\cdot V_q^{\mathrm{T}}\cdot(q-p)^{\mathrm{T}}}\end{array}\right\}\leq\varepsilon$$

algorithm 4C($\mathcal{D}$, ε, μ, λ, δ)

// assumption: each object in $\mathcal{D}$ is marked as unclassified

for each unclassified $O \in \mathcal{D}$ **do**

STEP 1. test $\textsc{Core}^{\text{cor}}_{\text{den}}(O)$ **predicate:**

compute $\mathcal{N}_\varepsilon(O)$;
if $|\mathcal{N}_\varepsilon(O)| \geq \mu$ **then**
compute $\mathbf{M}_O$; ← Covariance matrix $\mathbf{M}_O = \mathbf{V}_P\mathbf{E}_P\mathbf{V}_P^{\mathbf{T}}$
if $\textsc{CorDim}(\mathcal{N}_\varepsilon(O)) \leq \lambda$ **then**
compute $\hat{\mathbf{M}}_O$ and $\mathcal{N}_\varepsilon^{\hat{\mathbf{M}}_O}(O)$; ← Correlation similarity matrix $\hat{\mathbf{M}}_O = \mathbf{V}_P\hat{\mathbf{E}}_P\mathbf{V}_P^{\mathbf{T}}$
test $|\mathcal{N}_\varepsilon^{\hat{\mathbf{M}}_O}(O)| \geq \mu$;

STEP 2.1. if $\textsc{Core}^{\text{cor}}_{\text{den}}(O)$ **expand a new cluster:**

generate new clusterID;
insert all $X \in \mathcal{N}_\varepsilon^{\hat{\mathbf{M}}_O}(O)$ into queue Φ;
while $\Phi \neq \emptyset$ **do**
$Q =$ first object in Φ;
compute $\mathcal{R} = \{X \in \mathcal{D} \mid \textsc{DirReach}^{\text{cor}}_{\text{den}}(Q, X)\}$;
for each $X \in \mathcal{R}$ **do**
if X is unclassified or noise **then**
assign current clusterID to X
if X is unclassified **then**
insert X into Φ;
remove Q from Φ;

STEP 2.2. if not $\textsc{Core}^{\text{cor}}_{\text{den}}(O)$ O **is noise:**

mark O as noise;

end.

CASH [ABKKZ 07]

- Basic idea of CASH (= Clustering in Arbitrary Subspaces based on the Hough transform)
  - Transform each object into a so-called *parameter space* representing all possible subspaces accommodating this object (i.e. all hyper-planes through this object)
  - This parameter space is a *continuum* of all these subspaces
  - The subspaces are represented by a considerably small number of parameters
  - This transform is a generalization of the Hough Transform (which is designed to detect linear structures in 2D images) for arbitrary dimensions

- Transform
  - For each d-dimensional point p there is an infinite number of (d-1)-dimensional hyper-planes through p
  - Each of these hyper-planes s is defined by $(p, \alpha_1, \ldots, \alpha_{d-1})$, where $\alpha_1, \ldots, \alpha_{d-1}$ is the normal vector $\boldsymbol{n}_s$ of the hyper-plane s
  - The function $f_p(\alpha_1, \ldots, \alpha_{d-1}) = \delta_s = \langle p, \boldsymbol{n}_s \rangle$ maps p and $\alpha_1, \ldots, \alpha_{d-1}$ onto the distance δ_s of the hyper-plane s to the origin
  - The parameter space plots the graph of this *function*