PROXIMITY BASED

L'intuizione è che un punto molto lontano dagli altri è un outlier. Questo si basa sull'assunzione che gli outlier siano significativamente distanti dai punti normali.

DISTANCE - BASED

Qui la prossimità è misurata con il vicinato, che se non è abbastanza popolato allora il punto in questione è un outlier.

Per ogni punto o si va a vedere il ε-vicinato(o) dove ε è una threshold in termini di raggio entro cui cercare vicini.

Un oggetto o è un outlier se:

$$\frac{\|\{o' \mid dist(o, o') \leq \varepsilon\|}{\|D\| = n} \leq \pi$$

Ovvero se il numero di punti del suo ε-vicinato in rapporto con la cardinalità di $\|D\|$ è sotto una threshold π impostata dall'utente

KNN

In alternativa possiamo calcolare la distanza fra o e il suo k-esimo vicino più vicino, dove $k = \lceil \pi \cdot \|D\| \rceil$. Se $dist(o, o_k) > z$ allora outlier.

Nested loop algorithm

```
for (∀ oi ∈ D)
{
    Calcola_distanza(oi, tutti gli altri punti)
    Conta gli oggetti nel ε-vicinato(oi)
    if (|ε-vicinato(oi)| ≥ π · M)
        continua
    else
        oi = outlier
}
```

Grid-based algorithm

L'approccio di prima è dispendioso perché calcola tutte le distanze fra gli oggetti e il resto del DB. Perché non confrontare gruppi di oggetti? Ecco cosa fa l'algoritmo Grid-based.

Consiste nel dividere lo spazio dei dati in una griglia dove ogni cella ha diagonale pari a $r/2$.

Tipi di cella:

1) LIVELLO 1: Presa una cella C di riferimento, sono tutte le celle distanti una cella da C. Formalmente $\forall x \in C$ e $\forall y \in C_{LIV1}$: $dist(x, y) \leq r$

2) LIVELLO 2: Presa una cella C di riferimento, sono tutte le celle distanti più di 2 celle da C. Formalmente $\forall x \in C$ e $\forall y \in C_{LIV2}$: $dist(x, y) \geq r$

Regole di pruning

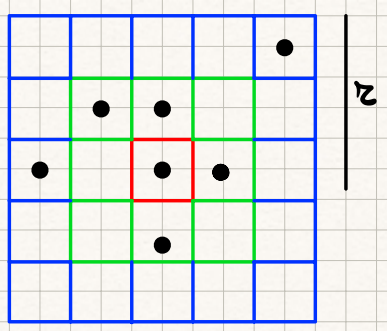
Sia "a" il numero di punti in C, "b_1" in C_{LIV1} e "b_2" in C_{LIV2}.

1) REGOLA LIVELLO 1: Se $a + b_1 \geq \pi M$ allora tutti gli oggetti $o \in C$ NON SONO OUTLIER con r e π specificati. Questo perchè tutti gli oggetti in C_{LIV1} sono dentro l'r-vicinato(o), e ci sono almeno $\lceil \pi \cdot r \rceil$ vicini

2) REGOLA LIVELLO 2: Se $a + b_1 + b_2 < \pi M + 1$ allora tutti gli oggetti $o \in C$ SONO OUTLIER con r e π parametri dati, perchè $\forall o \; |r\text{-vicinato}(o)| < \lceil \pi r \rceil$.

Per le celle che soddisfano una delle due regole possiamo capire se nelle cella sono presenti outlier.

Esempio



Prendo a riferimento C, C_{LIV1} e C_{LIV2}
con $\pi = 1/7$ e $M = 7$

Per il liv 1:

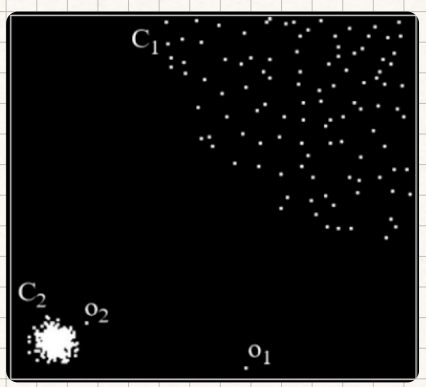
$1 + 4 \geq 1$ ✓

Per il liv 2

$1 + 4 + 2 < 1 + 1$ ✗

LOCAL DISTANCE-BASED OUTLIER DETECT.

Ora dobbiamo venire a patti con le diverse densità che cluster diversi possono presentare nello stesso dataset, e che possono portare a non riconoscere qualche outlier



Esempio

Il punto o_2 ha densità simile al cluster C_1 se paragonato a C_2.

Definiamo:

1) $N_K(x_i)$ i K-NN di x_i
2) $D_K(x_i)$ la distanza media fra x_i e K-NN

$$D_K(x_i) = \frac{1}{K} \sum_{x_j \in N_K(x_i)} |x_i - x_j|$$

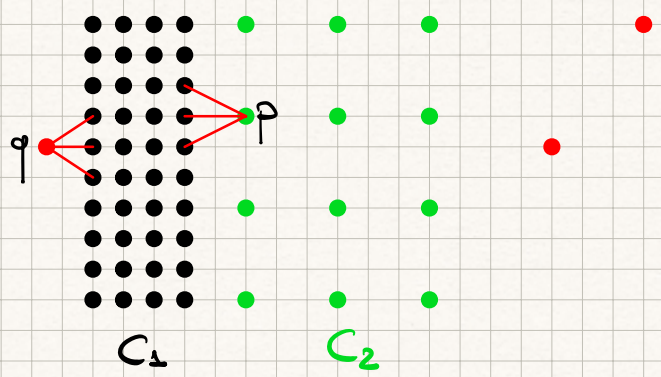
Definiamo ora l'**outlierness**:

$$O_K(x_i) = \frac{D_K(x_i)}{\frac{1}{K} \sum_{j \in N_K(x_i)} D_K(x_j)}$$

Se $O_K(x_i) > 1$ allora i vicini di x_i sono molto lontani

Problema con l'outlierness

Quando diversi cluster sono vicini fra loro, $O_k(x_i)$ può dare risultati non attesi:



In questo caso $O_3(p) > O_3(q)$ perché i punti verdi non fanno parte del 3-vicinato di p perché più distanti rispetto ai punti del del cluster C_1.

DENSITY-BASED OUTLIER DETECTION

L'intuizione è che la densità vicino a un punto outlier sarà molto diversa rispetto alla zona con punti normali.

Il metodo tenta a paragonare la densità nel vicinato dell'oggetto sotto analisi con la densità degli oggetti vicini.

Definiamo:

1) K-distance: $dist_k(o, K\text{-}NN)$

2) K-distance neighborhood of o: $N_k(o)$

$$N_k(o) = \{o' \mid o' \in D, dist(o, o') \leq D_k(o)\}$$

$N_k(o)$ è l'insieme dei K-NN vicini di o.
Possono essere più di K considerando punti equidistanti da o

3) REACHABILITY DISTANCE o' TO o:

$$reachdist(o \leftarrow o') = max\{dist_k(o), dist(o,o')\}$$

Quindi per i K-NN si prende in considerazione $dist_k(o)$
Proprietà: $reachdist(x \rightarrow y) \neq reachdist(y \rightarrow x)$

4) LOCAL REACHABILITY DISTANCE OF O:

$$lrd(o) = \frac{\|N_k(o)\|}{\sum_{o' \in N_k(o)} reachdist(o' \leftarrow o)}$$

LOCAL OUTLIER FACTOR

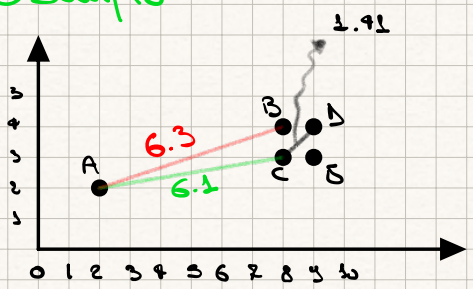
Per ogni oggetto possiamo calcolare questo fattore come:

$$LOF_k(o) = \frac{\sum_{o' \in N_k(o)} \frac{lrd_k(o')}{lrd_k(o)}}{\|N_k(o)\|}$$

Questa formula dipende dal confronto fra $lrd_k(o)$ con $lrd_k(o')$ per tutto il vicinato, più il rapporto si alza, più LOF è alto.

$$\text{Se } lrd_k(o) < lrd_k(o') \Rightarrow LOF \text{ è alto}$$

Esempio



Per $K = 2$:

$N_k(A) = \{B, C\}$

$dist_k(A) = dist(A, B) = 6.3$

Calcoliamo ora:

$$reachdist(A \leftarrow B) = \max(6.3 ; 6.3) = 6.3$$
$$reachdist(A \leftarrow C) = \max(6.3 ; 6.1) = 6.3$$

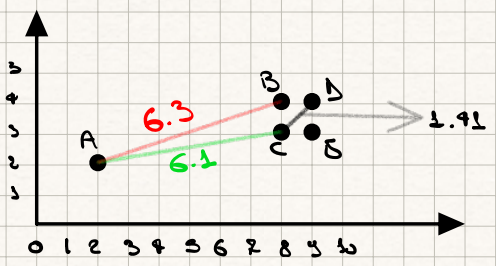
Dato che servirà, dobbiamo calcolare la reachability distance per il vicinato dei vicini di A.

$N_k(B) = \{C, D, E\}$ $dist_k(B) = dist(B, E) = \sqrt{2} = 1,41$

$N_k(C) = \{B, E, D\}$ $dist_k(C) = dist(C, D) = \sqrt{2} = 1,41$

Per B:

$$reachdist(B \leftarrow C) = 1.41$$
$$reachdist(B \leftarrow D) = 1.41$$
$$reachdist(B \leftarrow \delta) = 1.41$$



Per C:

$$reachdist(C \leftarrow B) = 1.41$$
$$reachdist(C \leftarrow D) = 1.41$$
$$reachdist(C \leftarrow \delta) = 1.41$$

Calcoliamo anche l'inverso per $B \leftarrow A$ e $C \leftarrow A$

$$reachdist(B \leftarrow A) = \max(1.41, 6.3) = 6.3$$
$$reachdist(C \leftarrow A) = \max(1.41, 6.1) = 6.1$$

Calcoliamo ora $lrd(A)$, $lrd(B)$ e $lrd(C)$

$$lrd_k(A) = \frac{3}{6.3 + 6.1} = 0.24$$

$$lrd_k(B) = \frac{4}{1.41 + 1.41 + 1.41 + 1.41} = 0.71$$

$$lrd_k(C) = \frac{4}{1.41 + 1.41 + 1.41 + 1.41} = 0.71$$

Da cui ora possiamo calcolare il $LOF_k(A)$

$$LOF_k(A) = \frac{\frac{0.71}{0.24} + \frac{0.71}{0.24}}{3} = 1.97$$

Problemi LOF