

## DM2 - Anomalie

**Outlier:** osservazione completamente differente rispetto alle altre osservazioni presenti in un dataset, tanto da sollevare il sospetto che tale osservazione sia stata generata da un meccanismo differente da quello che caratterizza le altre osservazioni. - Hawkins, 1980

**Global o local?** Un metodo è global se analizza l'intero dataset per decidere se un record è out (DBSCAN), local se solo una parte (KNN).

In generale, porre attenzione alla differenza fra outlier e rumore.

### Metodi statistici

Abbiamo due esempi di *metodi statistici*: il **test di Grubbs** e il **metodo Likelihood**. In Grubbs si assume un G (per ogni punto) come la *deviazione massima dalla media / std(dev)*, mentre nel secondo metodo si calcola la *log likelihood di ogni punto*, spostandolo nel caso dalla distribuzione Maggioritaria a quella Anomala

## Distinction Between Noise and Anomalies

- Noise is erroneous, perhaps random, values or contaminating objects
  - Weight recorded incorrectly
  - Grapefruit mixed in with the oranges
- Noise doesn't necessarily produce unusual values or objects
- Noise is not interesting
- Anomalies may be interesting if they are not a result of noise
- Noise and anomalies are related but distinct concepts

**Probabilistic definition of an outlier:** An outlier is an object that has a low probability with respect to a probability distribution model of the data.

- Usually assume a parametric model describing the distribution of the data (e.g., normal distribution)
- Apply a statistical test that depends on
  - Data distribution
  - Parameters of distribution (e.g., mean, variance)
  - Number of expected outliers (confidence limit)
- Issues
  - Identifying the distribution of a data set
    - Heavy tailed distribution
    - Number of attributes
    - Is the data a mixture of distributions?

## Statistical-based – Grubbs' Test

- Detect outliers in univariate data
- Assume data comes from normal distribution
- Detects one outlier at a time, remove the outlier, and repeat
  - $H_0$ : There is no outlier in data
  - $H_A$ : There is at least one outlier
- Grubbs' test statistic:
 

one-sided test with alpha/N  
two-sided test with alpha/2N
- Reject null hypothesis  $H_0$  of no outliers if:

$$G = \frac{\max |X - \bar{X}|}{S \text{ std dev}}$$

alpha significance  
 t – Student's distribution

$$G > \frac{(N-1)}{\sqrt{N}} \sqrt{\frac{t^2_{(\alpha/2, N-2)}}{N-2+t^2_{(\alpha/2, N-2)}}}$$

degrees of freedom  
 upper critical value of t-distribution

## Statistical-based – Likelihood Approach

- Assume the data set D contains samples from a mixture of two probability distributions:
  - M (majority distribution)
  - A (anomalous distribution)
- General Approach:
  - Initially, assume all the data points belong to M
  - Let  $L_t(D)$  be the log likelihood of D at time t
  - For each point  $x_i$  that belongs to M, move it to A
    - Let  $L_{t+1}(D)$  be the new log likelihood.
    - Compute the difference,  $\Delta = L_t(D) - L_{t+1}(D)$
    - If  $\Delta > c$  (some threshold), then  $x_i$  is declared as an anomaly and moved permanently from M to A

## Statistical-based – Likelihood Approach

- Data distribution,  $D = (1 - \lambda)M + \lambda A$
- M is a probability distribution estimated from data
  - Can be based on any modeling method (naïve Bayes, maximum entropy, etc.)
- A is initially assumed to be uniform distribution
- Likelihood at time t:
 
$$L_t(D) = \prod_{i=1}^N P_D(x_i) = \left[ (1 - \lambda)^{|M_t|} \prod_{x_i \in M_t} P_{M_t}(x_i) \right] \left[ \lambda^{|A_t|} \prod_{x_i \in A_t} P_{A_t}(x_i) \right]$$

$$LL_t(D) = |M_t| \log(1 - \lambda) + \sum_{x_i \in M_t} \log P_{M_t}(x_i) + |A_t| \log \lambda + \sum_{x_i \in A_t} \log P_{A_t}(x_i)$$

## Deviation-based

Se la varianza del set (local o global) è minimizzata rimuovendo quei punti, allora sono outlier. **È quindi calcolata la varianza di un punto  $I$  in relazione a un database.** È un'idea simile ai metodi statistici, **ma indipendente dalla distribuzione.**

## Distance-based

I punti sono giudicati in base alla loro distanza dai vicini. I punti normali sono in aree dense (= tanti vicini), gli outlier no. Due i possibili approcci: si parla di outlier se la distanza è superiore a *tot* da altri specifici punti, oppure consideriamo la distanza dal KNN come punteggio.

**Approccio 1.** Nel modello proposto nel '97 si definiscono il *raggio* e una percentuale  $\pi$ . Un punto è considerato outlier se un  $\pi\%$  ha una distanza dal punto  $<$  raggio. **Ovvero, se il punto è vicino a pochi punti** (o lontano da tanti punti).

Nell'esempio abbiamo fissato il raggio a 2.5, la percentuale a 0.15. A ha 4 punti nel suo raggio, i punti in totale sono 10. Quindi controlliamo se  $4/10 < 0.15$  - non lo è, quindi A non è un outlier.

**Approccio 2.** Un esempio è il KNN. Può verificarsi un problema se abbiamo considerato  $N = 1$  e abbiamo due punti accanto, ma spaiati dal resto del gruppo. Set di punti lontani del resto potrebbero poi essere cluster. Un altro problema si con aree a diversa densità: la distanza non tiene conto della densità dell'area.

La variante **partition-based** partiziona a monte il dataset in piccoli cluster, calcolando gli outlier per ogni cluster. Esiste anche una variante basata sulla **network theory**: ogni punto è un nodo, è presente in edge (link) fra i nodi se il secondo punto è nei KNN del primo punto. Specifichiamo poi un *threshold*: un punto è outlier se ha meno di  $T$  link in entrata. Fissando  $T = 2$ , nell'esempio il punto E è un outlier (0 link in entrata). Tecnicamente si definiscono *Reverse KNN* di un punto i punti con link in entrata a quel punto (ad esempio  $RKNN(C) = A, B, D$ )

**Nell'A2 specifichiamo  $K$ , ovvero a quanti punti il punto è vicino, nell'A1 specifichiamo raggio. A1 ha come output un label, A2 uno score.**

Model [Arning et al. 1996]

- Given a smoothing factor  $SF(I)$  that computes for each  $I \subseteq DB$  how much the variance of  $DB$  is decreased when  $I$  is removed from  $DB$
- With equal decrease in variance, a smaller exception set  $E$  is better
- The outliers are the elements of  $E \subseteq DB$  for which the following holds:  $SF(E) \geq SF(I)$  for all  $I \subseteq DB$

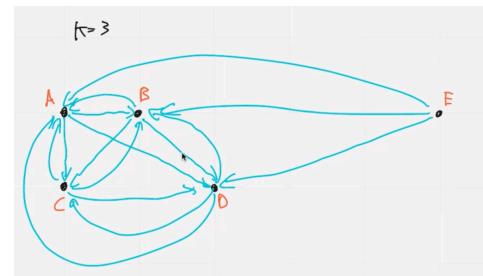
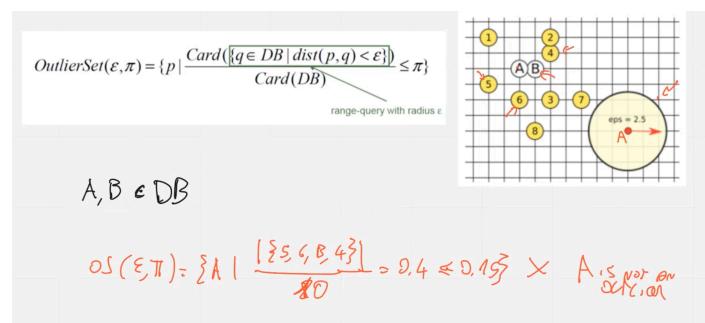
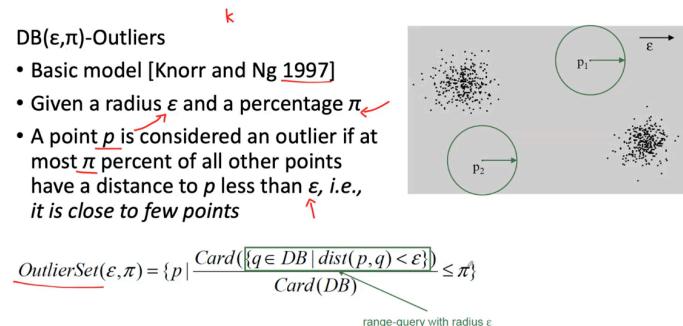
Discussion:

- Similar idea like classical statistical approaches ( $k = 1$  distributions) but independent from the chosen kind of distribution
- Naïve solution is in  $O(2n)$  for  $n$  data objects
- Heuristics like random sampling or best first search are applied
- Applicable to any data type (depends on the definition of  $SF$ )
- Originally designed as a global method
- Outputs a labeling

Approach 1: An object is an outlier if a specified fraction of the objects is more than a specified distance away (Knorr, Ng 1998)

- Some statistical definitions are special cases of this

Approach 2: The outlier score of an object is the distance to its  $k$ -th nearest neighbor



# Strengths/Weaknesses of Distance-Based Approaches

## Pros

- Simple

## Cons

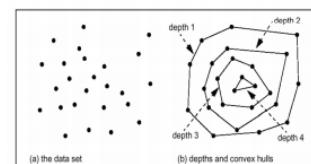
- Expensive –  $O(n^2)$
- Sensitive to parameters
- Sensitive to variations in density
- Distance becomes less meaningful in high-dimensional space

## Depth-based

Si basano sull'assunto che gli outlier si trovino nel bordo del *data space*, indipendentemente dalla distribuzione statistica. Si organizzano i dati in *convex hull layers*: gli outlier sono gli oggetti negli strati più esterni. Per dataset ad alta dimensionalità è computazionalmente costoso creare dei *convex hull*. Una soluzione può essere applicare la PCA (o dei prototipi).

Model [Tukey 1977]

- Points on the convex hull of the full data space have depth = 1
- Points on the convex hull of the data set after removing all points with depth = 1 have depth = 2
- ...
- Points having a depth  $\leq k$  are reported as outliers



## Density-based

Tutti gli approcci sono *local*; cambia però come è definita la densità.

### General idea

- Compare the density around a point with the density around its local neighbors
- The relative density of a point compared to its neighbors is computed as an outlier score
- Approaches differ in how to estimate density

### Basic assumption

- The density around a normal data object is similar to the density around its neighbors
- The density around an outlier is considerably different to the density around its neighbors

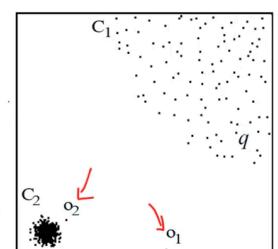
- **Density-based Outlier:** The outlier score of an object is the inverse of the density around the object.
  - Can be defined in terms of the  $k$  nearest neighbors
  - One definition: Inverse of distance to  $k$ th neighbor
  - Another definition: Inverse of the average distance to  $k$  neighbors
  - DBSCAN definition

### Motivation:

- Distance-based outlier detection models have problems with different densities
- How to compare the neighborhood of points from areas of different densities?

### Example

- DB( $\epsilon, \pi$ )-outlier model
  - Parameters  $\epsilon$  and  $\pi$  cannot be chosen so that  $o_2$  is an outlier but none of the points in cluster  $C_1$  (e.g.  $q$ ) is an outlier
- Outliers based on kNN-distance
  - kNN-distances of objects in  $C_1$  (e.g.  $q$ ) are larger than the kNN-distance of  $o_2$



Per ovviare al problema della diversa densità, si considera la **densità relativa**. Senza la densità relativa, nell'immagine a lato è impossibile trovare dei parametri in modo tale che  $o_2$  risulti un outlier, ma i punti in  $C_1$  no. A questi problemi risponde il metodo **Local Outlier Factor**.

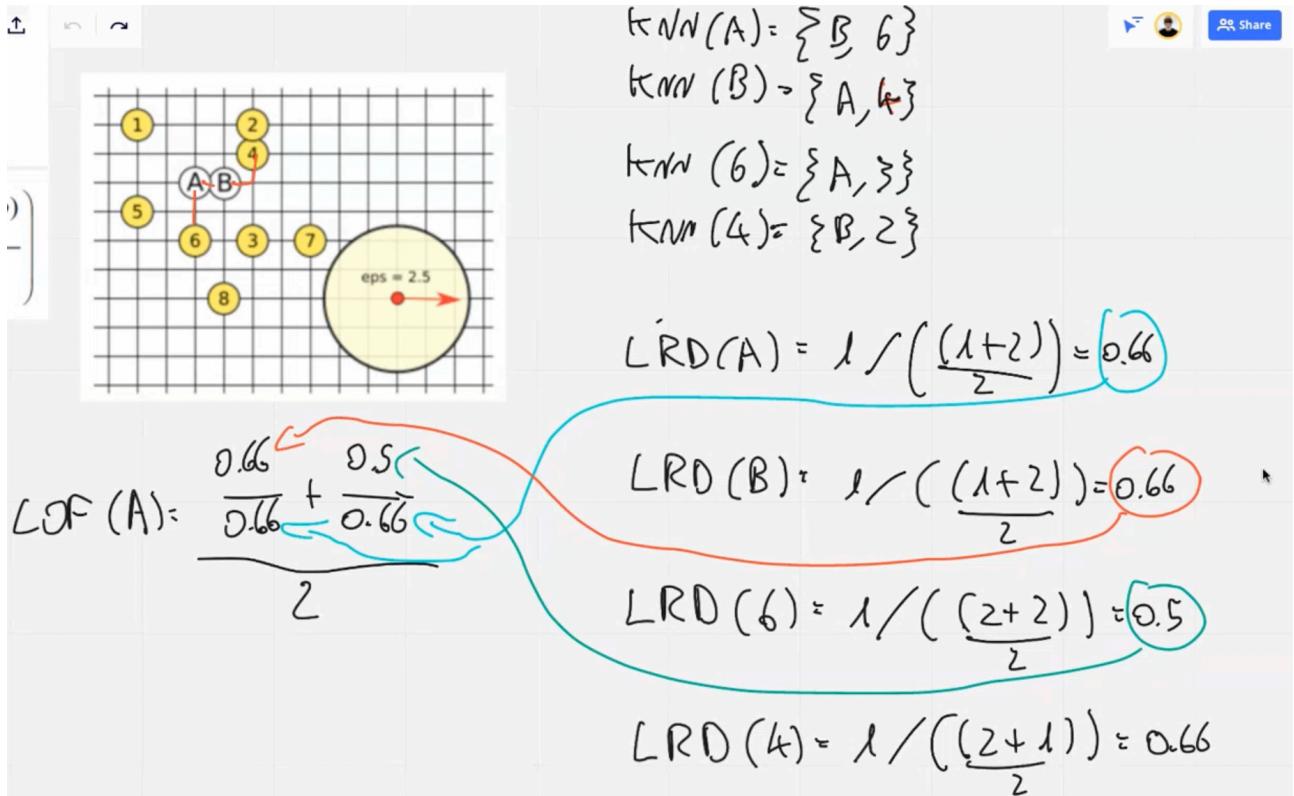
La densità relativa del LOF adotta il concetto di **reachability distance**. O è il punto al centro, P è il punto di cui vogliamo calcolare la distanza, K definisce il KNN.

La **reachability distance** è il massimo fra la **distanza P-O o la k-distance**, cioè la distanza data dal KNN (*If k was 3, the k-distance would be the distance of a point to the third closest point*). In altre parole, se il punto è fuori dai KNN usiamo la distanza fra il punto e O, altrimenti la **k-distance** (le distanze P-O saranno sempre le più grandi e contribuiranno di più alla classificazione come outlier). La **local reach dist** di un punto è l'inverso delle **reach dists medie di quel punto**, ovvero  $(1 / (\text{somma rech dist}) / K)$ . Infine abbiamo il LOF di quel punto.

- Reachability distance
  - Introduces a smoothing factor  
 $\text{reach-dist}_k(p, o) = \max\{k - \text{distance}(o), \text{dist}(p, o)\}$
- Local reachability distance ( $lrd$ ) of point p
  - Inverse of the average reach-dists of the kNNs of p
- Local outlier factor (LOF) of point p
  - Average ratio of  $lrd$ s of neighbors of p and  $lrd$  of p

$$lrd_k(p) = 1 / \left( \frac{\sum_{o \in kNN(p)} \text{reach-dist}_k(p, o)}{\text{Card}(kNN(p))} \right)$$

$$\text{LOF}_k(p) = \frac{\sum_{o \in kNN(p)} lrd_k(o)}{\text{Card}(kNN(p))}$$



Si è proceduto in questo modo, posto  $K == 2$ , per trovare il LOF di A:

- Calcolo dei KNN di A (B, 6) e la loro distanza (B: 1, 6: 2 - si è usata la misura *Manhattan*);
- Calcolo della LRD di A, B e 6:  $1 / ((\text{somme distanze dai KNN}) / K)$
- Calcolo del LOF

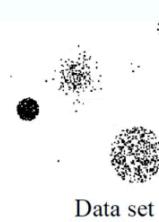
Analogamente si è proceduto con B (KNN: A, 4).

Ottimizzazione: possono essere definiti dei minicluster e calcolare il LOF solo al loro interno,. Si possono eliminando *a priori* i minicluster che non rispettano certi criteri, o eliminare l'insieme di punti nel microcluster se hanno uno score troppo basso per essere outlier.

# Local Outlier Factor (LOF)

## Properties

- $\text{LOF} \approx 1$ : point is in a cluster (region with homogeneous density around the point and its neighbors)
- $\text{LOF} \gg 1$ : point is an outlier



## Discussion

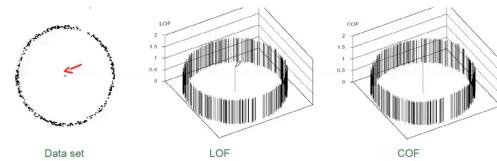
- Choice of  $k$  (MinPts in the original paper) specifies the reference set
- Originally implements a *local* approach (resolution depends on the user's choice for  $k$ )
- Outputs a scoring (assigns an LOF value to each point)

Il LOF ha problemi con l'esempio a fianco - ogni punto ha lo stesso LOF e il punto al centro non è visualizzato come outlier. Il **COF** risolve il problema trattando bassa densità e punti isolati in modo differente.

Un altro problema del LOF può verificarsi se cluster di diversa densità non sono ben separati. A questo problema risponde il metodo **INFLO**, che considera

## Connectivity-based outlier factor (COF) [Tang et al. 2002]

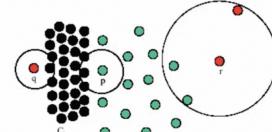
- Motivation
  - In regions of low density, it may be hard to detect outliers
  - Choose a low value for  $k$  is often not appropriate
- Solution
  - Treat "low density" and "isolation" differently
- Example



## Influenced Outlierness (INFLO) [Jin et al. 2006]

### Motivation

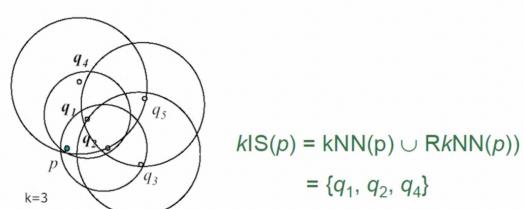
- If clusters of different densities are not clearly separated, LOF will have problems



Point  $p$  will have a higher LOF than points  $q$  or  $r$  which is counter intuitive

### Idea

- Take symmetric neighborhood relationship into account
- Influence space  $kIS(p)$  of a point  $p$  includes its kNNs ( $kNN(p)$ ) and its reverse kNNs ( $RkNN(p)$ )



$$\begin{aligned} kIS(p) &= kNN(p) \cup RkNN(p) \\ &= \{q_1, q_2, q_4\} \end{aligned}$$

## Strengths/Weaknesses of Density-Based Approaches

### Pros

- Simple

### Cons

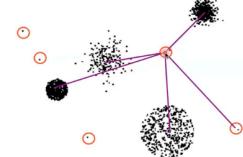
- Expensive –  $O(n^2)$  ←
- Sensitive to parameters
- Density becomes less meaningful in high-dimensional space

### Clustering-based

DBSCAN e OPTICS trovano separatamente dai cluster il "rumore", che potremmo considerare outlier. Questi metodi non sono però ottimizzati per individuare outlier. Con il K-Means, poi, si potrebbero formare dei cluster si potrebbero formare dei "cluster di outlier". Si potrebbero allora definire i cluster con pochi elementi come cluster di outlier, o considerare outlier i punti più diversi dal centroide. La computazione dei cluster con gli outlier porta poi a risultati potenzialmente *biased* e distorti.

**Clustering-based Outlier:** An object is a cluster-based outlier if it does not strongly belong to any cluster

- For prototype-based clusters, an object is an outlier if it is not close enough to a cluster center
- For density-based clusters, an object is an outlier if its density is too low
- For graph-based clusters, an object is an outlier if it is not well connected



Other issues include the impact of outliers on the clusters and the number of clusters

Gli approcci visti finora operano a 2, 3 dimensioni, ma possono risultare problematici a più dimensioni.

### Curse of dimensionality

- Relative contrast between distances decreases with increasing dimensionality
- Data is very sparse, almost all points are outliers
- Concept of neighborhood becomes meaningless

### Solutions

- Use more robust distance functions and find full-dimensional outliers
- Find outliers in projections (subspaces) of the original feature space

Gli approcci a seguire sono adatti a dataset ad alta dimensionalità.

## Angle-based

Partendo da un punto, se molti altri punti sono collocati in una simile direzione, allora il punto è un outlier.

**Non consideriamo la distanza, ma la direzione** (intendendola in senso di *angolo*). Ricorda il depth-based - si suppone che gli outlier siano ai "bordi" della sezione di dataset analizzato, mentre gli inlier sono al centro.

Si parte da un punto  $P$ , poi prendiamo due punti  $X$  e  $Y$  e calcoliamo l'**angolo fra i vettori  $\mathbf{P}X$  e  $\mathbf{P}Y$**  (con la *cosine similarity*). Lo facciamo per tutti gli  $X$  e  $Y$  del dataset. Più lo *spectrum* dei vari angoli sarà piccolo più il punto sarà outlier. **Lo spectrum coincide con osservare la varianza degli angoli:** un outlier avrà una bassa varianza.

## Grid-based

Se lo  $S(C)$  è  $< 0$  ci sono meno punti di quelli che ci si aspetta in quella zona della griglia - i punti lì presenti sono quindi outlier. Il problema è trovare i giusti parametri e "sovrapporre" nel modo giusto la griglia al dataset (per non rompere ad esempio dei cluster).

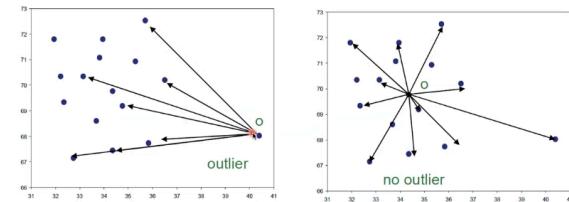
In un certo senso è un *equal binning* tridimensionale.

## Model-based

Dai dati è estratto un modello che viene usato per definire l'*outlier score*. Nel nostro caso, una **Isolation Forest**. Ogni albero estrae un *sample* dei dati, una dimensione a caso (laddove il Decision Tree sceglie quella migliore per lo *split*) e un valore a caso, dove si *splitta* il dataset. L'operazione si ripete finché l'albero non è completo, cioè finché in ogni albero non c'è solo un punto (ma si può porre un *early stopping*). Poi si ripete creando altri alberi. **L'idea è che gli outlier siano punti isolabili con pochi step, perché sono effettivamente isolati nel dataset.** L'approccio è random, per questo va ripetuto più volte. È efficiente, parallelizzabile (gli alberi possono essere costruiti in parallelo) e adatta a dati ad alta dimensionalità.

## ABOD – Angle-based Outlier Degree [Kriegel et al. 2008]

- Angles are more stable than distances in high dimensional spaces (e.g. the popularity of cosine-based similarity measures for text data)
- Object  $o$  is an outlier if most other objects are located in similar directions

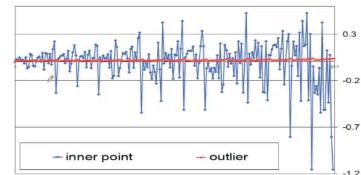
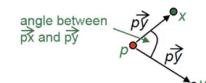


### Basic assumption

- Outliers are at the border of the data distribution
- Normal points are in the center of the data distribution

### Model

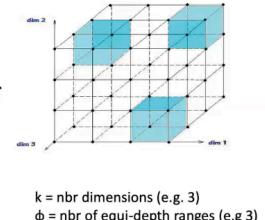
- Consider for a given point  $p$  the angle between any two instances  $x$  and  $y$
- Consider the spectrum of all these angles
- The broadness of this spectrum is a score for the outlierness of a point



## Grid-based Subspace Outlier Detection [Aggarwal and Yu 2000]

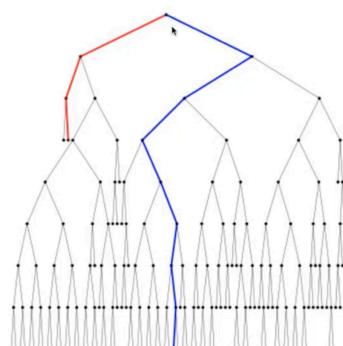
### Model

- Partition data space by an equi-depth grid ( $\phi = \text{number of cells in each dimension}$ )
- Sparsity coefficient  $S(C)$  for a  $k$ -dimensional grid cell  $C$



- where  $\text{count}(C)$  is the number of data objects in  $C$
- $S(C) < 0 \Rightarrow \text{count}(C)$  is lower than expected
- Outliers are those objects that are located in lower-dimensional cells with negative sparsity coefficient

Single Tree scores for  
anomaly (red) and nominal points (blue)



- For each tree:
  - Get a sample of the data
  - Randomly select a dimension
  - Randomly pick a value in that dimension
  - Draw a straight line through the data at that value and split data
  - Repeat until tree is complete

L'Isolation Forest può avere problemi di generalizzazione, assumendo outlier dove invece non ce ne sono. La soluzione è la variante **Extended Isolation Forest**. Piuttosto che selezionare una dimensione, seleziona un vettore che divide il dataset.

For each tree:

- Get a sample of the data
- Randomly select a normal vector
- Randomly select an intercept
- Draw a straight line through the data at that value and split data
- Repeat until the tree is complete

Comparazione fra i risultati di IF ed EIF.

