HIGH DIMENSIONAL CLUSTER

Quando si ha a che fare con dataset dalle grandi dimensioni si deve andare incontro a:

1) Molte dimensioni possono dimostrarsi inutili o addirittura possono mascherare i cluster.
2) Le misure di distanza perdono significato
3) I cluster possono esistere solo in sottospazi

Due tipologie di metodi:

1) SUB-SPACE CLUSTERING
2) DIMENSIONALITY REDUCTION APPROACHES

CURSE OF DIMENSIONALITY

1) I dati in una dimensione sono compatti

2) Se si considerano invece più di una dimensione i dati risultano straccati e molto distanti. Più dimensioni si considerano e peggio è.

3) Le misure di distanza perdono significato

SUBSPACE CLUSTERING

Come detto i cluster possono esistere solo in alcuni sottospazi. In più sottospazi diversi possono avere cluster diversi.

APPROCCIO BOTTOM-UP

Partire da sottospazi con poche dimensioni per poi man mano andare alla ricerca di sottospazi con dimensioni maggiori

APPROCCIO TOP-DOWN

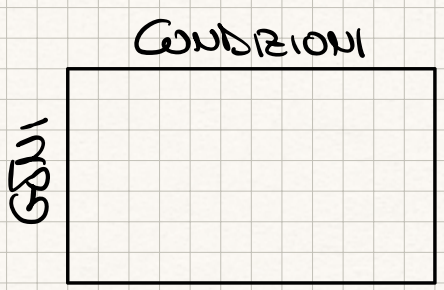
Partire da tutte le dimensioni e piano piano scartare dimensioni che non portano a risultati.

METODO BI-CLUSTER

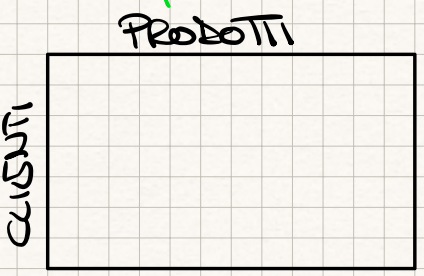
Difatti questo metodo consiste nel clusterizzare sia gli oggetti che gli attributi. Per farlo bisogna rispettare i seguenti requisiti:

1) Solo un piccolo sottoinsieme degli oggetti vengono effettivamente clusterizzati.
2) Un cluster coinvolge solo un sottoinsieme degli attributi
3) Un oggetto può essere in un cluster o in nessuno
4) Un attributo può essere coinvolto in più clusters o non coinvolto affatto in nessuno

Consideriamo $A = \{a_1 \dots a_n\}$ un'insieme di geni e $B = \{b_1 \dots b_m\}$ un insieme di condizioni, per cui costruire una struttura dati matriciale del tipo



Esempio



Un bi-cluster cerca una sottomatrice per cui geni e condizioni seguino un certo pattern

BI-CLUSTER CON VALORI COSTANTI

Per ogni i e j $e_{ij} = \text{COSTANTE}$ costruiscono la sottomatrice.

Esempio

$\forall e_{ij} = 60$:

	⋯	b_6	⋯	b_{12}	⋯	b_{36}
a_1	⋯	60	⋯	60	⋯	60
⋯	⋯	⋯	⋯	⋯	⋯	⋯
a_{33}	⋯	60	⋯	60	⋯	60
⋯	⋯	⋯	⋯	⋯	⋯	⋯
a_{86}	⋯	60	⋯	60	⋯	60

BI-CLUSTER CON VALORI COSTANTI NELLA RIGA

Metto nel bi-cluster tutti gli $e_{ij} = \text{COST} + a_i$ dove a_i rappresenta il valore di scarto per la riga i-esima

Esempio

$e_{ij} = 10 + i$

	b_6	b_8	b_9	b_{15}	b_{20}
a_1	11	11	11	11	11
a_5	15	15	15	15	15
a_8	18	18	18	18	18

BI-CLUSTER CON VALORI COERENTI

Qui prendiamo la sottomatrice dove le colonne e le righe cambiano sincronizzatamente: $e_{ij} = C + a_i + b_j$

Deve valere la regola:

$$\forall i_1, i_2 \in I;\ \forall j_1, j_2 \in J : e_{i_1 j_1} - e_{i_2 j_1} = e_{i_1 j_2} - e_{i_2 j_2}$$

Esempio:

	b_1	b_2	b_3
a_1	10	50	30
a_2	20	60	40
a_3	50	90	70

- $10 - 20 = 50 - 60$
- $10 - 50 = 50 - 90$
- $60 - 90 = 40 - 70$
- ...

BI-CLUSTER CON VALORI COERENTI NELLE RIGHE

$$\forall i_1, i_2 \in I;\ \forall j_1, j_2 \in J : (e_{i_1 j_1} - e_{i_2 j_1}) \cdot (e_{i_1 j_2} - e_{i_2 j_2}) \geq 0$$

Esempio

	b_1	b_2	b_3
a_1	10	50	30
a_2	20	100	50
a_3	50	100	90

- $(10-50)\cdot(20-100) > 0$
- $(10-50)(50-100) > 0$
- $(100-50)(100-90) > 0$

δ - BI - CLUSTER

Per capire questo algoritmo introduciamo delle metriche

1) MEDIA i - ESIMA RIGA

$$e_{iJ} = \frac{1}{|J|} \sum_{j \in J} e_{ij}$$

2) MEDIA j - ESIMA RIGA

$$e_{Ij} = \frac{1}{|I|} \sum_{i \in I} e_{ij}$$

3) MEDIA DI TUTTI GLI ELEMENTI DI UNA MATRICE

$$e_{IJ} = \frac{1}{|I||J|} \sum_{\substack{i \in I \\ j \in J}} e_{ij}$$

4) QUALITÀ DELLA SOTTOMATRICE $I \times J$

$$H(I \times J) = \frac{1}{|I||J|} \sum_{\substack{i \in I \\ j \in J}} \left(e_{ij} - e_{iJ} - e_{Ij} + e_{IJ} \right)^2$$

Definiamo la sottomatrice $I \times J$ come un δ-bi-cluster se $H(I \times J) \leq \delta$ con $\delta \geq 0$.

Per $\delta = 0$ abbiamo un δ-bi-cluster perfetto con valori totalmente coesenti.

Con un $\delta > 0$ l'utente specifica la tolleranza al rumore ammesso.

MAXIMAL- δ-BI-CLUSTER

È un δ-Bi-Cluster $I \times J$ tale per cui non esiste un altro δ-Bi-Cluster $I' \times J'$ che lo contiene.

APPROCCIO EURISTICO

A livello computazionale, è pesante da fare, perciò si usa un'approccio euristico per la ricerca di un ottimo locale.

Due fasi

1) Deletion Phase:

Partendo dall'intera matrice, eliminare riga e colonna iterativamente tale per cui la media quadrata residua è maggiore di δ.

Dove per media quadrata residua si intende:

$$d(i) = \frac{1}{|J|} \sum_{j \in J} \left(e_{ij} - e_{iJ} - e_{Ij} + e_{IJ} \right)^2$$

$$d(j) = \frac{1}{|I|} \sum_{i \in I} \left(e_{ij} - e_{iJ} - e_{Ij} + e_{IJ} \right)^2$$

Si inizia rimuovendo riga e colonna con $d(i)$ e $d(j)$ massimi.

2) Additional phase

Aggiungere righe e colonne nel δ-Bi-cluster finché i requisiti del δ-Bi-cluster vengono mantenuti.
Si aggiungono righe e colonne con $d(i)$ e $d(j)$ minimi.

δ-pCLUSTER

Cerca di valutare una sottomatrice.

Una sotto matrice è un δ-bi-cluster perfetto se $e_{i_1 j_1} - e_{i_2 j_1} = e_{i_1 j_2} - e_{i_2 j_2}$. Questo per via del rumore non è sempre vero.

Per ogni sottomatrice 2×2 di $I \times J$, definiamo il valore p-score come

$$\text{p-score}\begin{pmatrix} e_{i_1 j_1} & e_{i_1 j_2} \\ e_{i_2 j_1} & e_{i_2 j_2} \end{pmatrix} = \left| \left(e_{i_1 j_1} - e_{i_2 j_1}\right) - \left(e_{i_1 j_2} - e_{i_2 j_2}\right) \right|$$

Una sottomatrice $I \times J$ è un δ-pCluster se il p-score di ogni sua sottomatrice 2×2 è maggiore di una certa threshold $\delta \geq 0$.

Questo specifica quando si vuole essere tolleranti rispetto a un δ-pCluster perfetto

Intuizione:

Il p-score controlla il rumore di ogni elemento dentro il bi-cluster, mentre la media controlla il rumore complessivo.

PROPRIETÀ MONOTONICA

Se $I \times J$ è un δ-pCluster anche ogni sua sottomatrice $x \times y$ $(x, y \geq 2)$ è a sua volta un δ-pCluster.

Definiamo così il massimo δ-pCluster come una δ-pCluster $I \times J$ in cui non è possibile aggiungere né colonne né righe tale per cui la nuova matrice sia un δ-pCluster.