Confronto tra Pattern

Sistemi informativi per le Decisioni

Slide a cura di Ing. Marco Patella

Motivazione

- Rilevazione di cambiamenti nei dati
  - Es.: comportamento di acquisto dei clienti nel tempo
- Confronto di collezioni di grandi dimensioni
  - Es.: analisi delle vendite mensili di un supermercato
- Confronto di algoritmi di Data Mining
  - Es.: confronto dei risultati ottenuti da due algoritmi di clustering
- Rilevazione di outlier o pattern “inattesi”
  - Es.: fraud detection

Esempio d’uso

- Un manager di una catena di supermercati vuole analizzare la tendenza delle vendite
- In particolare, si vuole capire se esiste un particolare supermercato le cui vendite differiscano significativamente da quelle degli altri supermercati
- L’analisi va effettuata in 3 direzioni:
  - Confrontando le vendite di ciascun prodotto
  - Confrontando gli scontrini (MBA)
  - Confrontando i prodotti che caratterizzano ogni customer segment

Obiettivo

- Dati due pattern, calcolarne la “similarità”

$$s(p_1,p_2)$$

- Tale definizione dipende dal tipo di pattern
- Intuitivamente, se i due pattern si riferiscono agli stessi dati, la similarità deve essere alta (~1)

Requisiti

- Generalità
  - Soluzione applicabile a qualsiasi tipo di pattern, indipendentemente dalla complessità
- Flessibilità
  - Il criterio di similarità non deve essere univoco
- Semplicità
  - Requisito fondamentale per l'utilizzabilità
- Efficienza
  - Definizione della similarità indipendentemente dai dati

FOCUS (Ganti et al., PODS ’99)

- Un framework per misurare le differenze nelle caratteristiche dei dati
- Calcola una misura di deviazione tra due dataset per quantificare le differenze esistenti tra caratteristiche “interessanti” dei due dataset
- L’idea di base è che una vasta gamma di pattern possa essere descritta in termini di due componenti:
  - componente *struttura*
  - componente *misura*

Esempio: large itemsets

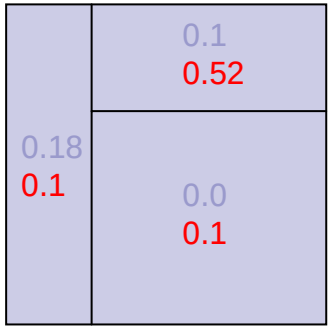
- Struttura: tutti i frequent itemset estratti dal dataset
- Misura: i corrispondenti supporti
- Esempio:

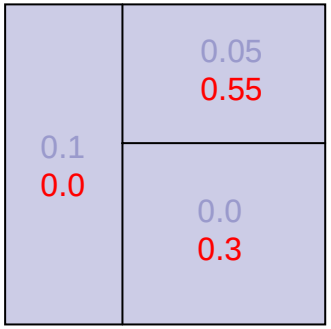
({a}, 0.5)	({b}, 0.3)
({b}, 0.4)	({c}, 0.5)
({a,b}, 0.25)	({b,c}, 0.2)

- Quanto sono simili i dataset originali?

Esempio: decision trees

- Struttura: tutte le regioni associate alle foglie dell'albero
- Misura: le corrispondenti misure (es., frazione di elementi in ciascuna classe)
- Esempio:

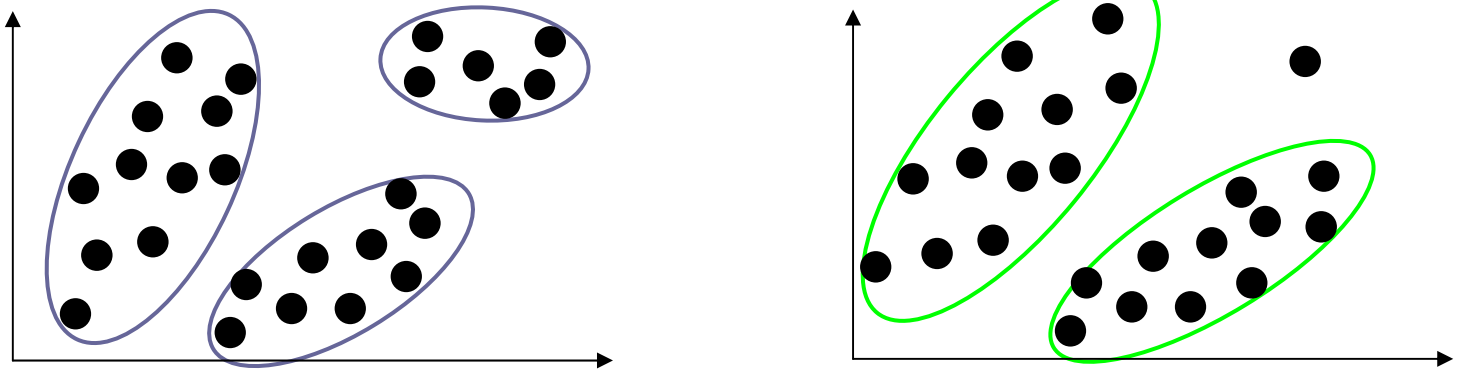




- Quanto sono simili i dataset originali?

Esempio: clustering

- Struttura: tutte le regioni associate ad ogni cluster
- Misura: le corrispondenti misure (es., supporto del cluster)
- Esempio:



- Quanto sono simili i dataset originali?

Come calcolare la differenza?

- Caso base: strutture identiche
- Esempio:

({a}, 0.5)	({a}, 0.3)
({b}, 0.4)	({b}, 0.5)
({a,b}, 0.25)	({a,b}, 0.2)

- Calcolo della differenza tra le componenti “comuni”
  - Es.: |0.5-0.3|=0.2, |0.4-0.5|=0.1, |0.25-0.2|=0.05
- Aggregazione delle differenze in un unico valore
  - Es.: 0.2+0.1+0.05=0.35

… e se le strutture non sono identiche?

- Esempio:

({a}, 0.5)	({a}, 0.3)
({b}, 0.4)	({b}, 0.5)
({a,b}, 0.25)	

- Evidentemente, occorre calcolare il supporto per l'itemset (non large) {a,b} sul secondo dataset
  - Es.: |0.5-0.3|=0.2, |0.4-0.5|=0.1, |0.25-0.1|=0.15

Generalizzando…

- Calcolo del Greatest Common Refinement tra le strutture dei due pattern
- Calcolo delle misure sul GCR per entrambi i dataset
- Calcolo delle differenze nelle misure per i due modelli “indotti”
- Aggregazione delle differenze sui due modelli “indotti”

Esempio: large itemsets

({a}, 0.5)
({b}, 0.4)
({a,b}, 0.25)

({b}, 0.3)
({c}, 0.5)
({b,c}, 0.2)

- Calcoliamo il GCR:
  - {{a},{b},{c},{a,b},{b,c}}
- Calcoliamo i supporti sui dataset originari:

({a}, 0.5)
({b}, 0.4)
({a,b}, 0.25)
({c}, 0.1)
({b,c}, 0.05)

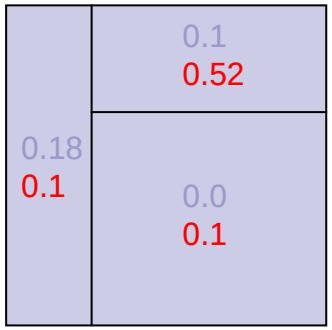
({b}, 0.3)
({c}, 0.5)
({b,c}, 0.2)
({a}, 0.1)
({a,b}, 0.05)

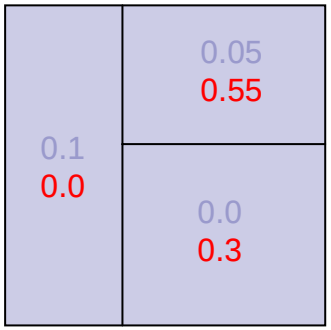
Esempio: large itemsets (cont.)

({a}, 0.5)	({b}, 0.3)
({b}, 0.4)	({c}, 0.5)
({a,b}, 0.25)	({b,c}, 0.2)
({c}, 0.1)	({a}, 0.1)
({b,c}, 0.05)	({a,b}, 0.05)

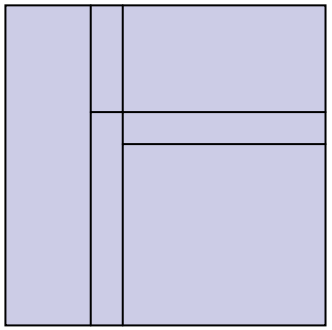
- Calcoliamo le differenze sui nuovi modelli:
  - |0.5-0.1|=0.4, |0.4-0.3|=0.1, |0.1-0.5|=0.4, |0.25-0.05|=0.2, |0.05-0.2|=0.15
- Aggreghiamo le differenze per ottenere il valore finale:
  - 0.4+0.1+0.4+0.2+0.15 = 1.25

Esempio: decision trees

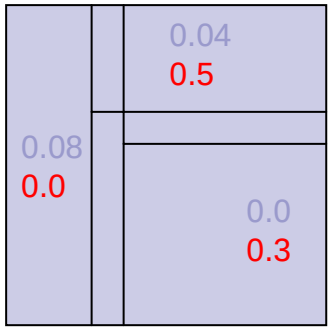


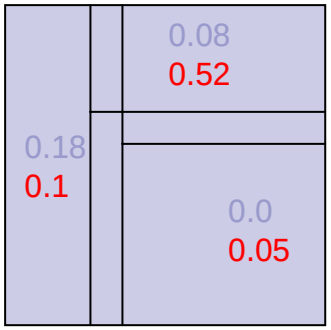


- Calcoliamo il GCR:

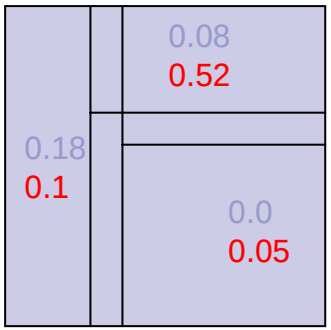


- Calcoliamo le misure sui dataset originari:





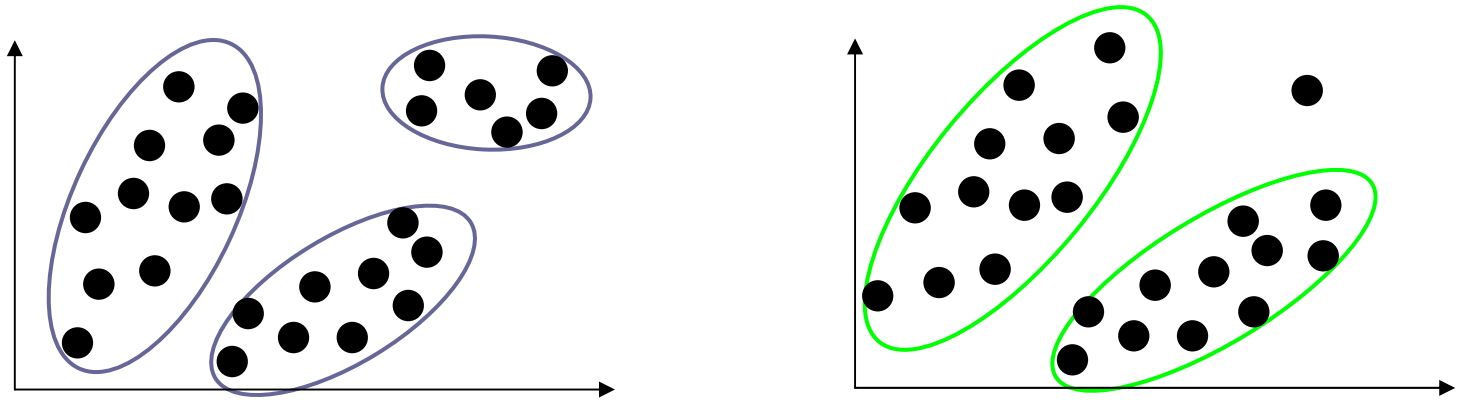
Esempio: decision trees (cont.)



- Calcoliamo le differenze sui nuovi modelli
- Aggreghiamo le differenze per ottenere il valore finale

Esempio: clustering

- Struttura: simile ai decision trees
- Misura: simile alle regole associative



FOCUS: limiti

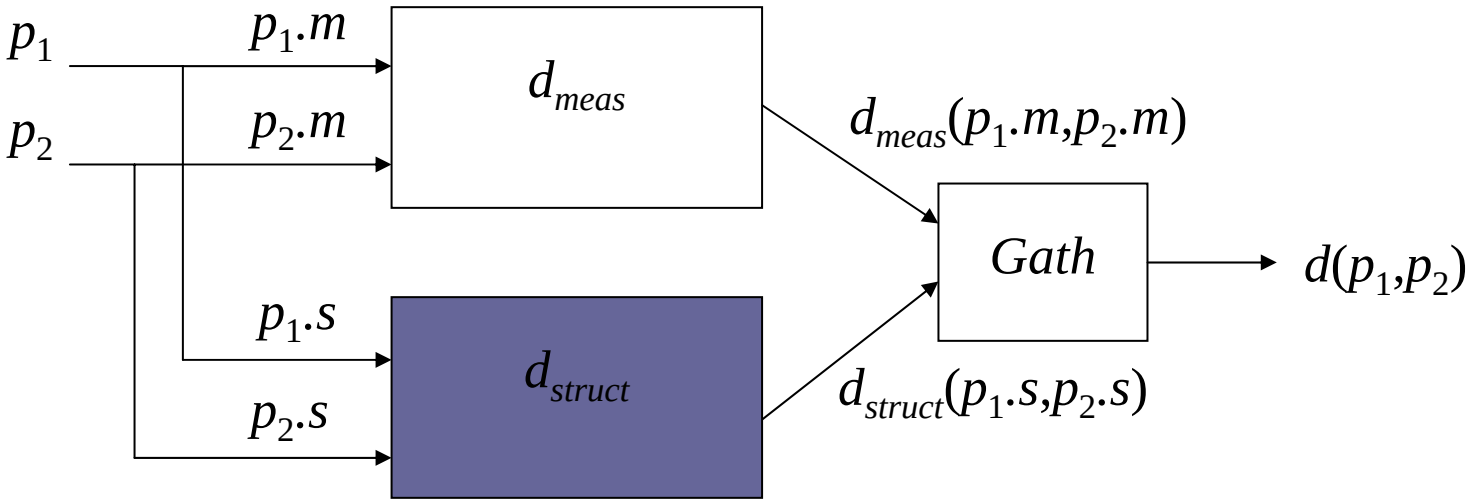
- Scarsa generalità
  - Gestisce solamente pattern “a 2 livelli”
- Scarsa flessibilità
  - I criteri di differenza ed aggregazione possono variare
  - Il criterio GCR è fisso per ogni tipo di pattern
- Scarsa efficienza
  - È necessaria una scansione ulteriore dei dataset per calcolare le misure sul GCR

PANDA framework: (Bartolini, Ciaccia, Patella, Ntoutsi, Theodoridis, PKDD'04)

- Si basa sulla natura a 2 componenti (*struttura* e *misura*) dei pattern
- Gestisce pattern di arbitraria complessità
  - Un pattern complesso è un pattern la cui struttura include altri pattern
- La differenza (distanza) tra 2 pattern tiene conto dei 2 componenti:
  - distanza tra le strutture
  - distanza tra le misure
- La differenza tra pattern complessi tiene conto (ricorsivamente) della differenza tra i pattern componenti

Modello generale

- Un pattern è una rappresentazione compatta e ricca di semantica dei dati grezzi

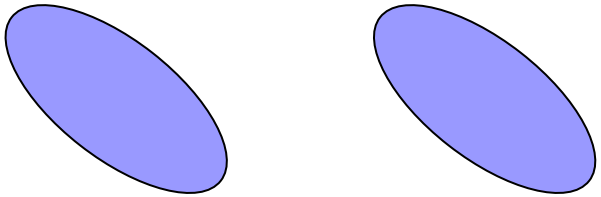


Esempio: large itemset

- Confrontiamo i due pattern
  - ({pane, miele, latte}, 0.1)
  - ({burro, latte}, 0.2)
- Supponiamo che:
  - $d_{meas} = |supp_1 - supp_2|$
  - $d_{struct} = 1 - |struct_1 \cap struct_2| / |struct_1 \cup struct_2|$
  - $d = (d_{meas} + d_{struct})/2$
- $d_{meas} = 0.1$, $d_{struct} = 0.75$, $d = 0.425$

Esempio: cluster

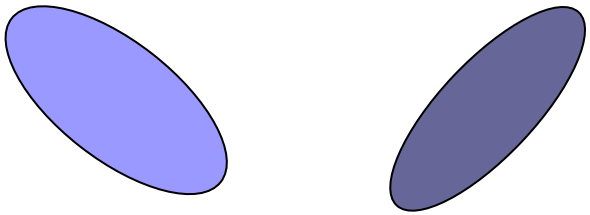
- Stessa struttura: consideriamo la misura



d_{meas} = | 0.1 – 0.15 | = 0.05
d_{struct} = 0

support = 0.1 support = 0.15

- Struttura diversa?



d_{meas} = 0.05
d_{struct} = ?

support = 0.1 support = 0.15

Distanza di Bhattacharyya

- Supponiamo di avere cluster ellittici
- Ogni cluster è definito dal centroide e dalla matrice di covarianza
- Distanza tra i centroidi:

$$(c_1 - c_2)^T (cov_1+cov_2)^{-1}/2 \; (c_1 - c_2)$$

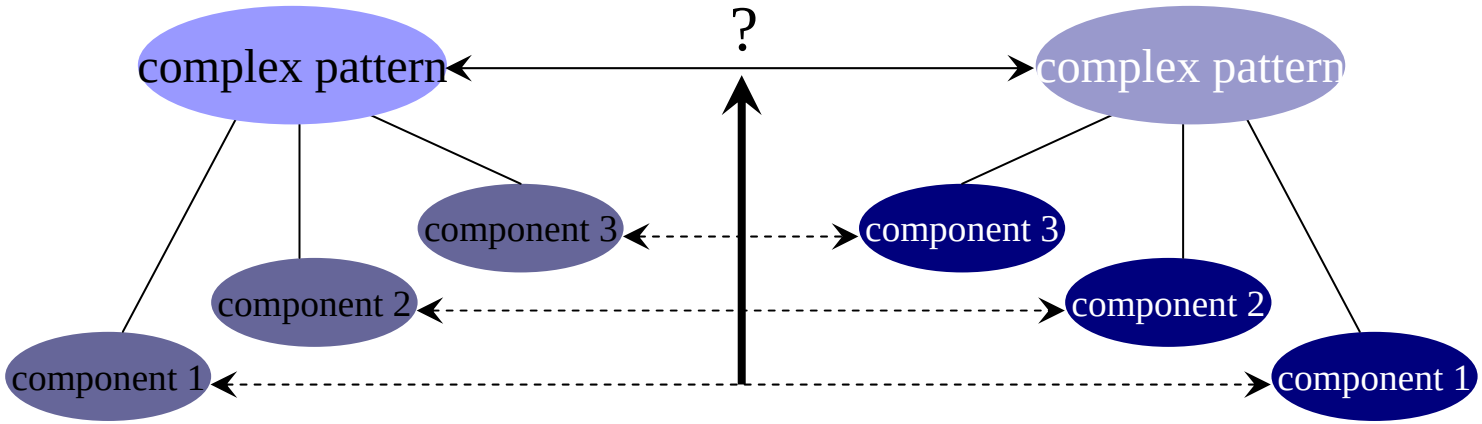
- Distanza tra le matrici:

$$\log(\det((cov_1+cov_2)/2)/\mathrm{sqrt}(\det(cov_1)\det(cov_2)))$$

- Distanza totale = $d_{cen}/8+d_{cov}/2$

Confronto di pattern complessi

- In generale, occorre ricorsivamente confrontare i pattern componenti
- Vengono usate due astrazioni generali:
  - Tipo di accoppiamento
  - Logica di aggregazione



Accoppiamento

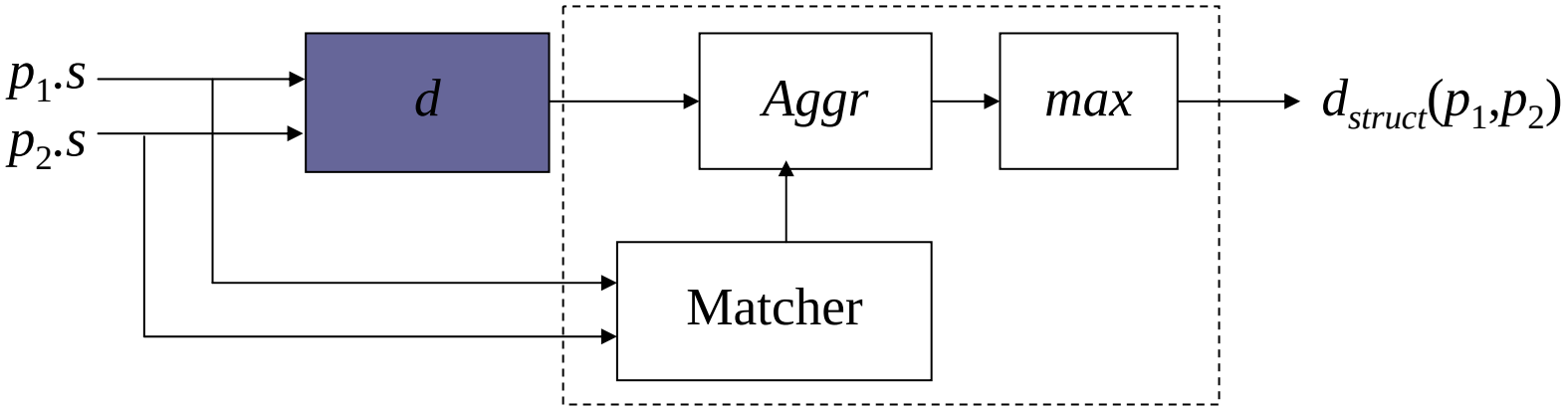
- Definisce come associare i pattern componenti dei due pattern complessi p_1 e p_2
- In generale, equivale a definire una matrice $\mathbf{X}_{nxm} = (x_{ij})$ tale che $x_{ij} \in [0,1]$
- x_{ij} rappresenta l'ammontare dell'accoppiamento tra l'i-esimo pattern componente di p_1 ed il j-esimo pattern componente di p_2

Tipi di accoppiamento

- 1–1
  - $\sum_i x_{ij} \leq 1$, $\sum_j x_{ij} \leq 1$, $\sum_{ij} x_{ij} = \min\{N,M\}$, $x_{ij} \in \{0,1\}$
- N–M
  - Nessun vincolo su X
  - $x_{ij} \in \{0,1\}$

Logica di aggregazione

- Una volta accoppiati i pattern componenti, come calcolare la differenza globale?
- Idea di base: selezionare il "miglior" accoppiamento
- $d_{struct}=\min_{X}\{Aggr(p_1, p_2, X)\}$



Esempio: large itemsets

- Sappiamo come confrontare due itemset
- Vogliamo confrontare set di itemsets
- Es.:

({a}, 0.5)	({b}, 0.3)
({b}, 0.4)	({c}, 0.5)
({a,b}, 0.25)	({b,c}, 0.2)

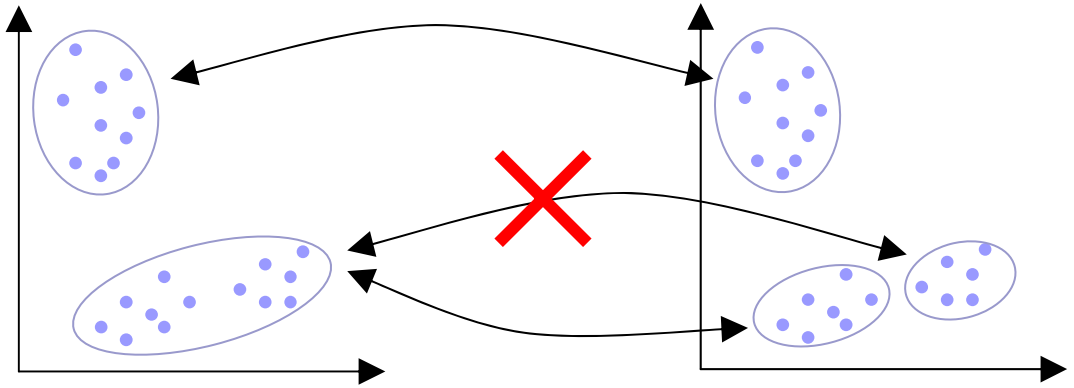
  - d(({b}, 0.4), ({b}, 0.3)= 0.05
  - d(({a}, 0.5), ({c}, 0.5)= 0.5
  - d(({a,b}, 0.25), ({b,c}, 0.2)= 0.19
- d=(0.05+0.5+0.19)/3=0.25

EMD (Earth Mover's Distance)

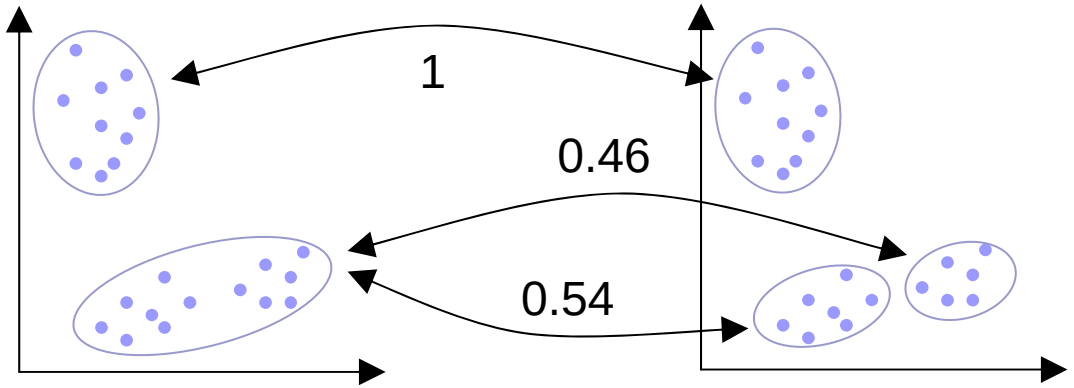
- Permette di accoppiare tra loro "pezzi" dei pattern componenti
- Ad ogni componente deve essere associato un peso (es.: supporto) w_i
  - $\sum_i x_{ij} \leq w_j$, $\sum_j x_{ij} \leq w_i$
  - $\sum_{ij} x_{ij} = \min\{\sum_i w_i, \sum_j w_j\}$,
  - $x_{ij} \in [0,1]$
  - $d_{struct} = \min_{\mathbf{x}}\{\sum_i \sum_j d_{ij} x_{ij}\} / \min\{\sum_i w_i, \sum_j w_j\}$

Esempio: clustering

- 1–1 matching



- EMD matching (w_i=supporto)



Esempio: decision trees

- Confronto tra due foglie dell'albero
  - d_{struct}=1 – ∩aree/∪aree
  - d_{meas}=|$supp_1$ – $supp_2$|
- Logica di aggregazione
  - Una qualsiasi di quelle adottate in precedenza

Confronto PANDA/FOCUS

- Vantaggi PANDA:
  - Diversi criteri di accoppiamento
  - Definizione ricorsiva di pattern complessi
    - Finora abbiamo visto solo 2 livelli di aggregazione
    - Cosa succederebbe con cluster di regole associative?
  - Non è necessario tornare sui dati grezzi per calcolare la differenza tra pattern
- In sommario:
  - + flessibile
  - + generale
  - + efficiente