Data Protection & Privacy

CAHD

Implementation of the algorithm to anonymize Transactional Data

Stefano Ferrazin, Stefano Musante

Università degli Studi di Genova

Anonimizzare Transactional Data tramite CAHD

Transactional Data: dati raggruppati in una matrice sparsa, in cui ci si aspetta di avere un numero maggiore di transazioni (righe) rispetto ai prodotti (colonne).

	Wine	Strawberries	Meat	Cream	Pregnancy Test	Viagra
Bob	X		X			X
David	X		X			
Claire		X		X	X	
Andrea		X	X			
Ellen	X		X	X		

(a) Original Data

	Wine	Meat	Cream	Strawberries	Pregnancy Test	Viagra
Bob	X	X				X
David	X	X				
Ellen	X	X	X			
Andrea		X		X		
Claire			X	X	X	

(b) Re-organized Data

	Wine	Meat	Cream	Strawberries	Sensitive Items
Bob	X	X			Viagra: 1
David	X	X			
Ellen	X	X	X		
Andrea		X		X	Pregnancy Test: 1
Claire			X	X	

(c) Published Groups

Fig. 1. Purchase Transaction Log Example

Si identificano i Sensitive Item che dovranno essere anonimizzati tramite gruppi.

Il grado di privacy del gruppo G sarà uguale alla cardinalità di G, in quanto per l'euristica scelta nel CAHD, in ogni gruppo si può avere al massimo uno per ciascun sensitive item.

→

In general, let $f_1^G \dots f_m^G$ be the number of occurrences for sensitive items $s_1 \dots s_m$ in group G. Then group G offers privacy degree

$$p^G = \min_{i=1\dots m} |G|/f_i$$

Si organizzano i QID del dataset in una matrice a bande (permutando righe e colonne), per aumentare la correlazione tra i QID nei gruppi.

Questo viene fatto per minimizzare l'errore di ricostruzione. Si traduce nel minimizzare la KL divergence.

Vengono infine creati i gruppi anonimizzati..

- In ogni gruppo i sensitive Item sono associati ad una transazione con una probabilità 1/|G|
- Si vuole però che l'associazione di un dato pattern ad un sensitive item sia mantenuta in quanto questa costituisce l'utility di questo tipo di dati.
- Se dopo aver creato i gruppi rimangono delle transazioni, queste sono inserite in un gruppo senza sensitive item, il quale non deve essere anonimizzato.

Implementazione algoritmo

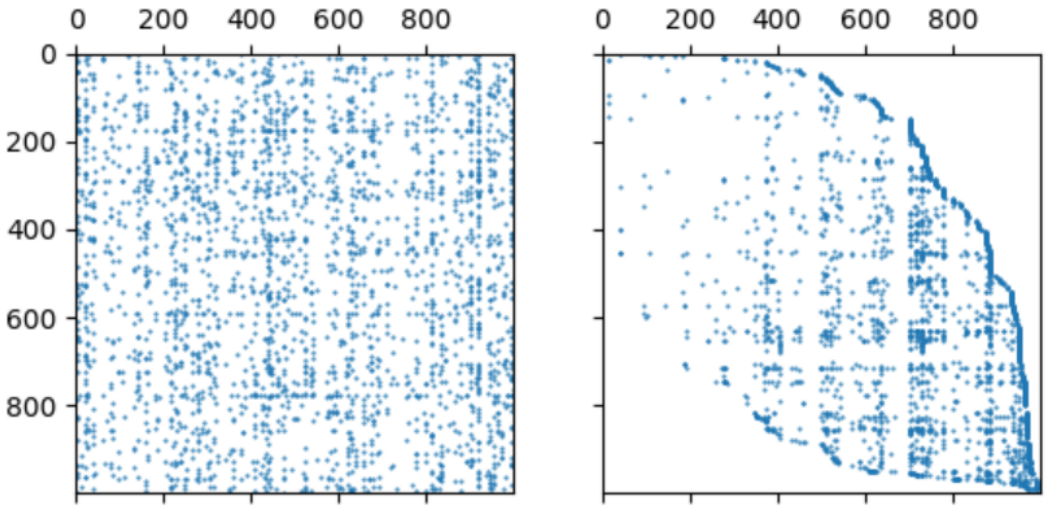
Python 3.9

INIZIO: Si è lavorato sui dataset menzionati nel paper: BMS1 e BMS2. Di questi sono state analizzate 10000 transazioni.

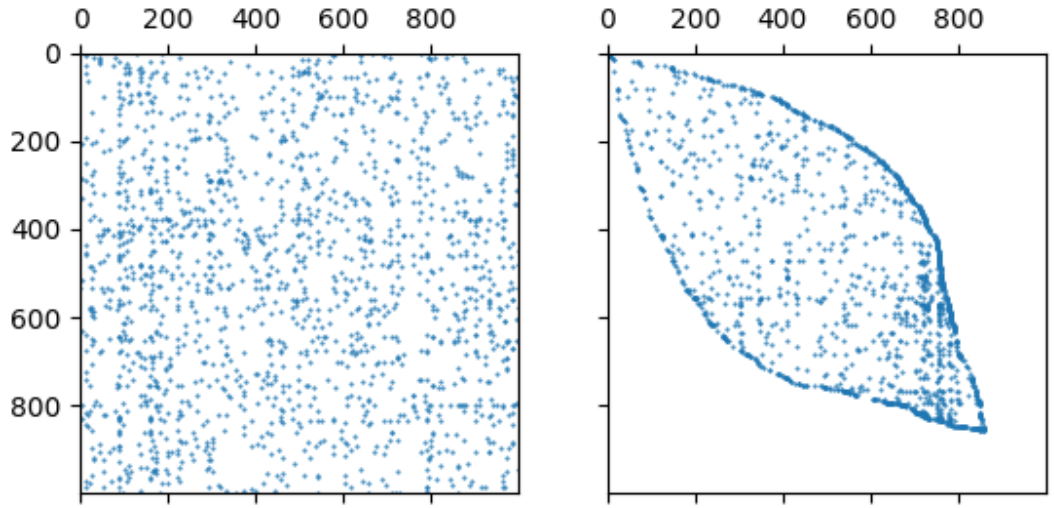
L'algoritmo lavorerà su una transaction matrix e una list item ricavate dai dataset iniziali ***(spmfToCSV.py)***

Per testare l'algoritmo, abbiamo inoltre definito una dimensione massima, per limitare ulteriormente il dataset e rendere più veloce l'esecuzione. ***(main.py)***

BAND-MATRIX: L'algoritmo procederà a creare una band matrix, applicando Reverse Cuthill McKee, partendo dalla transaction matrix e riducendo la bandwidth del dataset.



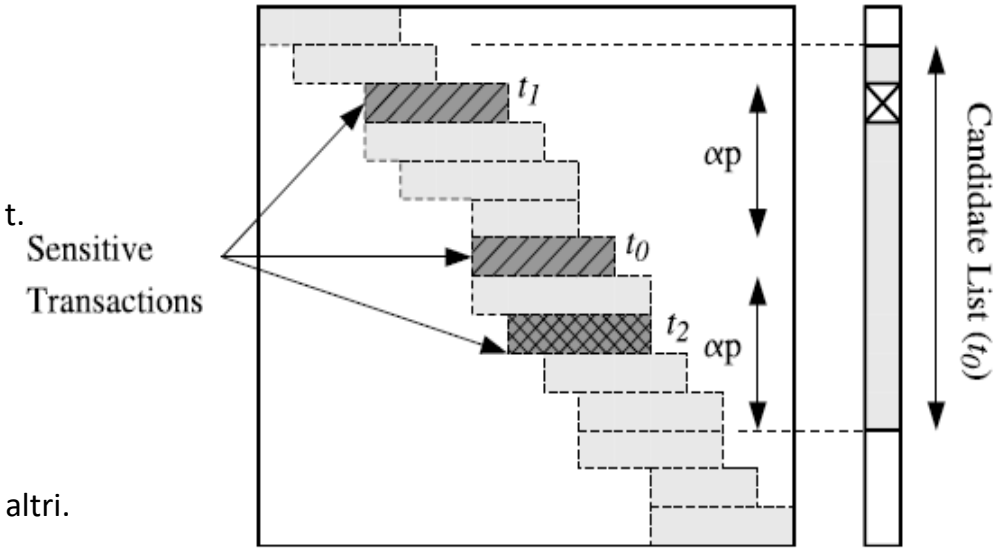
BMS1



BMS2

CAHD

- Si creano due **dizionari**
  1. **sensitive_histogram**: per tenere conto dei sensitive item rimasti da anonimizzare.
  2. **sensitive_rows**: per associare ogni transazione sensibile agli item sensibili in essa.
- Ad ogni iterazione si analizza una **transazione sensibile t**.
  - Si prova a creare una **candidate list CL** partendo dalle αp transazioni precedenti e successive a t.
  - Vanno escluse dalla lista quelle che contengono un'occorrenza di un SI di t.
  - Se non si riesce a creare CL di $2\alpha p$ righe, si prova a crearne una di dimensione almeno $p-1$
  - Riducendo il numero di righe ridurrò anche l'utility
- Si **definiscono** i **gruppi** massimizzando l'utility.
  - Dalla CL estraggo le $p-1$ righe con i QID più simili tra loro.
  - In una lista vengono inserite le p righe che costituiranno il gruppo anonimizzato.
  - Queste righe saranno memorizzate: una volta inserite in un gruppo non potranno fare parte di altri.
- Si **convalida** il **gruppo**
  - Dovranno poter essere anonimizzate le transazioni rimanenti
  - Se non fosse possible si ripristina tutto all'iterazione precedente
  - Bisogna vedere se con le transazioni successive è possibile anonimizzare soddisfacendo i **requisiti di privacy**.
  - Se è possibile il gruppo viene convalidato.
- Si decreta se il **grado di privacy** p è soddisfacibile
  - Se dopo aver analizzato tutte le sensitive transaction non risultano più occorrenze di sensitive item nel sensitive histogram, è possibile salvare i gruppi.
  - Se invece rimangono delle occorrenze bisogna rianalizzare le sensitive transaction e vedere se è possibile anonimizzarle.
  - Se ad una rianalisi il numero di elementi rimanenti nell'istogramma non cambia e non è uguale a zero, si può concludere che il dataset non è soddisfacibile con p.
- Si **salvano** i **gruppi**
  - Vengono estratti i gruppi dalla band matrix.
  - Le righe che non fanno parte di alcun gruppo anonimizzato costituiranno il gruppo non sensitive.



CAHD Group Formation Heuristic

Input: transaction set T, privacy degree p

1. initialize histogram H for each sensitive item $s \in S$
2. $remaining = |T|$
3. **while** $(\exists t \in T | t\ is\ sensitive)$ **do**
4. t = next sensitive transaction in T
5. $CL(t)$ = non-conflicting αp pred. and αp succ. of t
6. $G = \{t\} \cup p-1$ trans. in $CL(t)$ with closest QID to t
7. update H for each sensitive item in G
8. **if** $(\nexists s | H[s] \cdot p > remaining)$
9. $remaining = remaining - |G|$
10. **else**
11. roll back G and continue
12. **end while**
13. output remaining transactions as a single group

KL-DIVERGENCE

Bisogna calcolare la **KL divergence** per misurare la perdita di utility (reconstruction error) avvenuta nella creazione dei gruppi.

- Verrà calcolata rispetto al sensitive item **s** con più occorrenze nel dataset (si può vederlo dall'istogramma iniziale).

$$KL\ Divergence = \sum_{\forall\ cell\ C} Act_C^s \log \frac{Act_C^s}{Est_C^s}$$

Per calcolarla si estraggono r sensitive item in modo casuale e per ogni combinazione (2^r in totale) si computano l'actual pdf e l'estimated pdf.

$$Act_C^s = \frac{Occorrenze\ di\ s\ in\ C}{Occorrenze\ di\ s\ in\ T}$$

Numeratore: occorrenze del pattern associate a **s**.

Denominatore: occorrenze di **s** nella transaction matrix. (Viene utile il metodo **compare** delle **series** di pandas e il **dizionario** creato per le transazioni sensibili per il CAHD).

$$Est_C^s = \frac{{}^{ab}/_{|G|}}{Occorrenze\ di\ s\ in\ T}$$

Numeratore:

- **a:** occorrenze di **s** nel gruppo (per l'euristica adottata sarà 1)
- **|G|:** cardinalità del gruppo.
- **b:** occorrenze del pattern nel gruppo.

Denominatore: occorrenze di **s** nel dataset anonimizzato.

Risorse utilizzate

Python 3.9

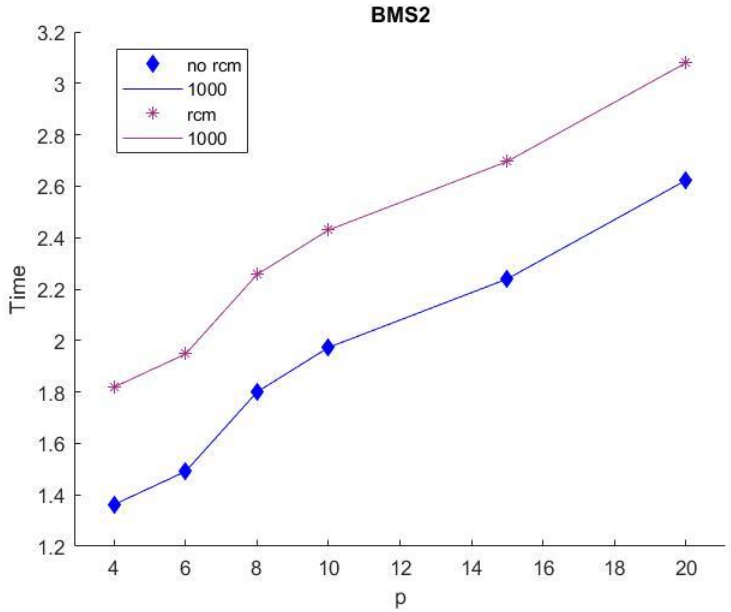
- **Csv**
  - Per estrarre il dataset iniziale e convertirlo in una transaction matrix
- **Pandas**
  - Per visualizzare la transaction matrix come un DataFrame. In questo modo è facile manipolarla lavorando su label di righe e colonne.
- **Numpy**
  - Per gestire gli ndarray (formato in cui vengono restituiti gli indici del pandas DataFrame)
  - Anche per ottenere le permutazioni randomiche.
- **Scipy**
  - Per ottenere un grafo da una matrice delle adiacenze (= transaction matrix)
  - Per applicare il RCM a partire dal grafo.
- **Matplotlib.pylab**
  - Per vedere la differenza tra la transaction matrix iniziale e quella a bande trovata.

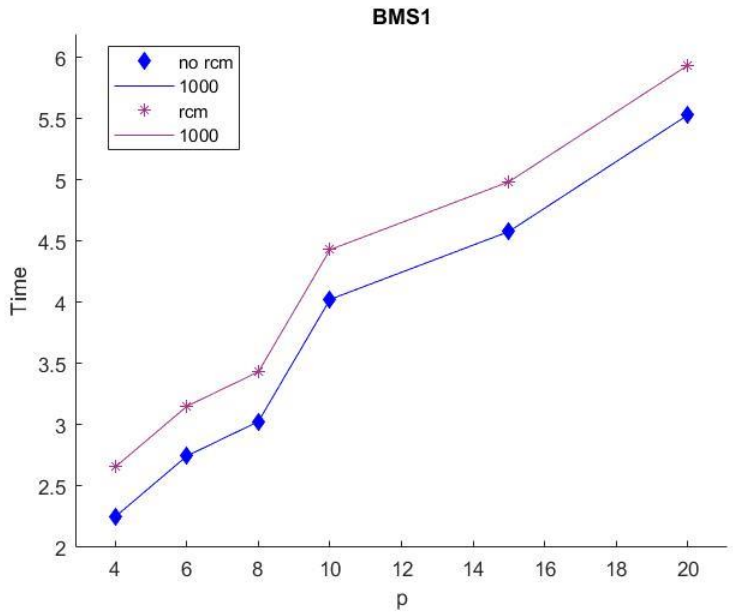
Matlab r2020b

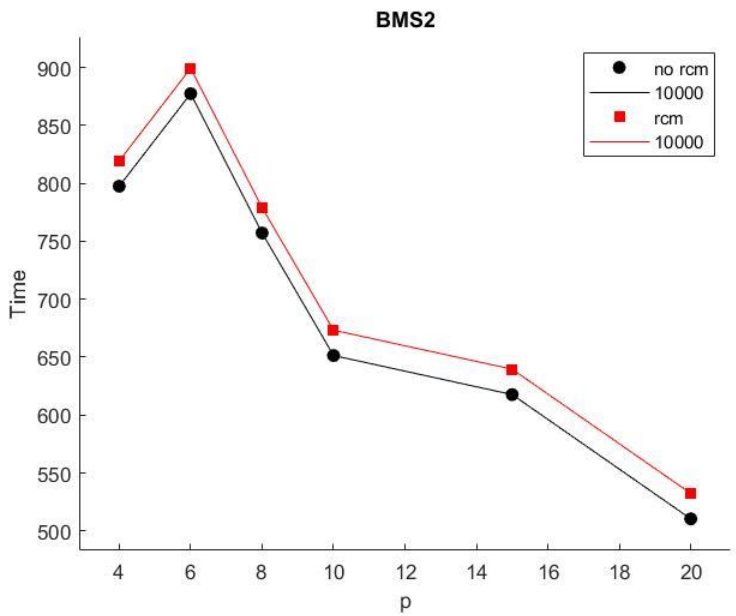
Per plottare i grafici.

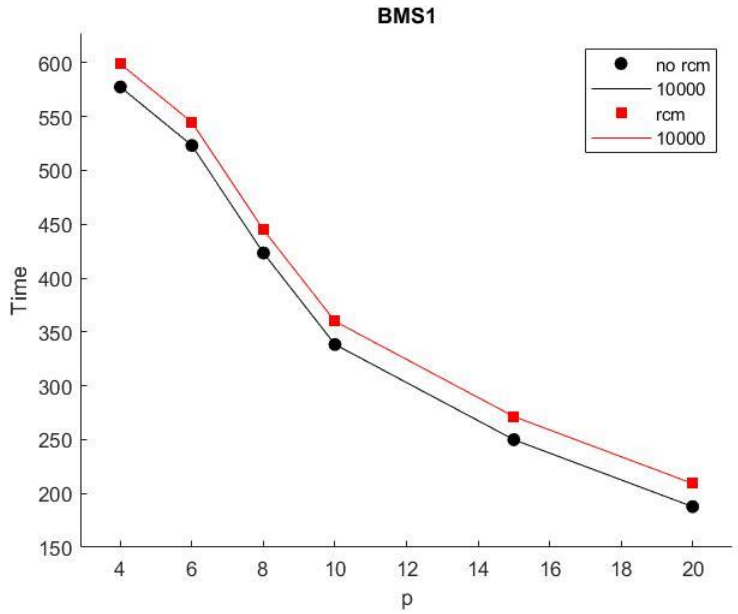
Risultati ottenuti

1000 transazioni e su 10000 transazioni su dataset BMS1 e BMS2. (m = 10, α = 3)









Si nota che con l'aumentare del grado di **privacy**:

- Quando si hanno meno transaction il tempo tende ad **aumentare**
- Quando ce ne sono di più a **diminuire**.

Nel primo caso l'operazione che richiede più tempo è la **creazione della candidate list** che varia in funzione di **p**.

Nel secondo caso invece l'operazione che richiede più tempo è la **creazione del "non sensitive group"**: questo avrà meno righe, se il **p** è più alto.

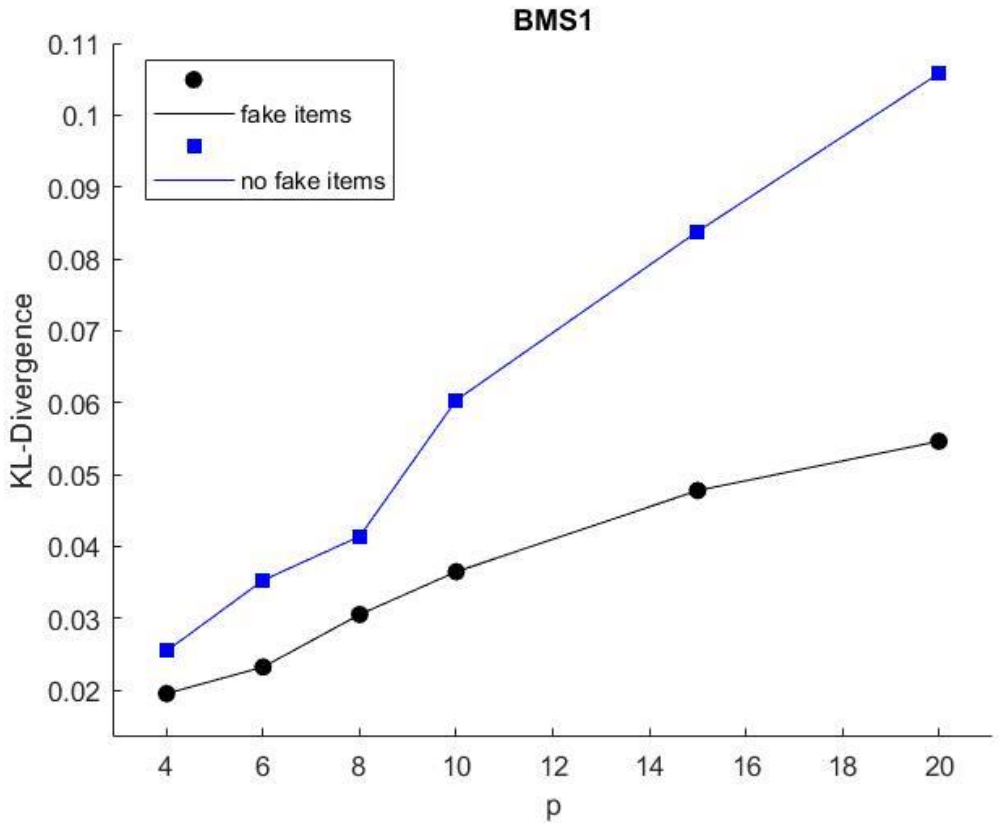
Conclusioni

Misurando la KL divergence abbiamo notato che si ottenevano risultati molto simili tra quelli di BMS1 e BMS2, mentre nel paper differivano di molto.

Questo avviene sia per il fatto che non abbiamo considerato tutto il dataset, ma solo un sottoinsieme, sia perchè abbiamo computato la KL divergence con la band matrix provvista di fake items.

Abbiamo fatto dei test per vedere cosa sarebbe successo se dopo aver creato la matrice a bande avessimo tolto I fake items.

Abbiamo notato per BMS1 che la KL Divergence aumenta in valore.



Test eseguito su BMS1, con 2000 transazioni, m=10, α=3, r=4