Privacy nella Pubblicazione di Dati

Parte I

Indice

Una continua crescita riguardante:

- database governativi e aziendali
- contenuti generati dagli utenti
- informazioni personali identificative collezioante quando un utente crea un account, scarica un'applicazione, ...

La condivisione dei dati serve per:

- studiare le tendenze e fare inferenze statistiche
- condividere conoscenza
- accedere ai servizi online

C'è inoltre l'archiviazione e il calcolo esterno (cloud), che offorno:

- risparmio sui costi e benefici dei servizi
- maggiore disponibilità e protezione da eventuali disastri

Per questa serie di motivazioni è fondamentale garantire che la privacy e l'integrità dei dati siano adeguatamente protette.

Capitolo 1

Privacy nella Pubblicazione dei Dati

Quando si parla del rilascio di informazioni per scopi statistici, è possibile fare una distinzione tra:

- **statistical DBMS:** c'è un'interazione tra client e DBMS, con quest'ultimo che risponde a delle query. Richiede un **controllo a runtime** delle informazioni rilasciate.
- **statistical data:** non c'è un'interazione; il controllo viene fatto prima del rilascio dei dati, tramite delle autorità competenti

1.1 Macrodata, microdata, disclosure

Per **macrodata** si intendono dati aggregati; le tabelle possono essere classificate in due gruppi:

- **Conteggio/Frequenza:** ogni cella contiene il numero o la percentuali di rispondenti che hanno lo stesso valore per gli attributi considerati. Mostrano il numero di volte che un valore compare nei dati (quanti studenti hanno preso un certo voto).
- **Magnitudo:** ogni cella contiene un valore di una *quantità di interesse*. Riportano la somma o media di un valore numerico associato a una cateogoria (somma degli stipendi per dipartimento).

Per **microdata** si intendono dati non aggregati, ovvero dati specifici e individuali; questo tipo di dati sono soggetti a un maggiore rischio di violazione della privacy (attacchi di collegamento).

Rilascio di informazioni

Il rilascio di informazioni si riferisce all'**attribuzione di informazioni sensibili a un rispondente**.

Si può fare una distinzione tra:

- **Identity disclosure:** è quando un terzo può **identificare** un rispondente tramite le informazioni rilasciate; è un problema quando si tratta di microdata, dato che i dati sono dettagliati
- **Attribute disclosure:** è quando **informazioni confidenziali** di un rispondente sono rilasciate o possono essere a lui attribuite, con esattezza o con un grado di precisione inferiore a quello atteso
- **Inferential disclosure:** è quando informazioni sensibili vengono **dedotte con alta certezza dalle proprietà statistiche dei dati rilasciati**.

Tecniche di protezione per macrodata

- **Sampling:** pubblicare solo una porzione della popolazione totale; deve essere rappresentativo e privo di bias
- **Special rules:** si definiscono delle restrizioni sul livello di dettaglio che può essere fornito (ad esempio, non pubblicare o rendere deducibili i redditi sotto un intervallo di 1000$)
- **Threshold rules:** definire una cella come sensibile se il numero di rispondenti è inferiore a un soglia

Tecniche di protezione per microdata

- **Masking:** si trasforma il dataset non rilasciando o modificando i suoi valori. Possono essere:
  - *non-perturbative:* il dataset non viene modificato, ma alcuni dati sono soppressi o alcuni dettagli rimossi (sampling, generalizzazione)
  - *perturbative:* il dataset viene modificato (arrotondamento, swapping); viene introdotto del rumore
- **Dati sintetici:** vengono usati dati plausibili ma sintetici;
  - *fully synthetic:* il dataset contiene solo dati sintetici
  - *partially synthetic:* il dataset contiene sia dati sintetici che dati originali

1.1.1 Anonimity problem

È in continua crescita il numero di record che contengono dati sensibili dei cittadini. Questi record vengono *de-identificati* prima della loro pubblicazione; tuttavia, questo **non è sufficiente**: possono essere usati altri dati per fare dei collegamenti tra identità de-identificate, facendo dunque una **re-identificazione**.

Esempio

SSN	Name	Race	DoB	Sex	ZIP	Marital status	Disease
		asian	64/04/12	F	94142	divorced	hypertension
		asian	64/09/13	F	94141	divorced	obesity
		asian	64/04/15	F	94139	married	chest pain
		asian	63/03/13	M	94139	married	obesity
		asian	63/03/18	M	94139	married	short breath
		black	64/09/27	F	94138	single	short breath
		black	64/09/27	F	94139	single	obesity
		white	64/09/27	F	94139	single	chest pain
		white	64/09/27	F	94141	widow	short breath

Name	Address	City	ZIP	DOB	Sex	Status
.....	.....	.....	.....	.....	.....	.....
.....	.....	.....	.....	.....	.....	.....
Sue J. Doe	900 Market St.	San Francisco	94142	64/04/12	F	divorced
.....	.....	.....	.....	.....	.....	.....

SSN	Name	Race	DoB	Sex	ZIP	Marital status	Disease
	Sue J. Doe	asian	64/04/12	F	94142	divorced	hypertension
		asian	64/09/13	F	94141	divorced	obesity
		asian	64/04/15	F	94139	married	chest pain
		asian	63/03/13	M	94139	married	obesity
		asian	63/03/18	M	94139	married	short breath
		black	64/09/27	F	94138	single	short breath
		black	64/09/27	F	94139	single	obesity
		white	64/09/27	F	94139	single	chest pain
		white	64/09/27	F	94141	widow	short breath

Name	Address	City	ZIP	DOB	Sex	Status
.....	.....	.....	.....	.....	.....	.....
.....	.....	.....	.....	.....	.....	.....
Sue J. Doe	900 Market St.	San Francisco	94142	64/04/12	F	divorced
.....	.....	.....	.....	.....	.....	.....

Classificazione degli attributi in una tabella microdata

- **Identificatori:** attributi che identificano univocamente un rispondente
- **Quasi identificatori:** attributi che linkati ad informazioni esterne possono reidentificare un rispondente, o ridurre l'incertezza sulla loro identità (Data di nascita, ZIP, sesso)
- **Confidenziale:** attributi sensibili
- **Non confidenziale:** attributi non considerati sensibili

Fattori che contribuiscono al disclosure risk

- esistenza di record con *caratteristiche peculiari*
- possibilità di matchare microdata con informazioni esterne

Fattori che diminuiscono il disclosure risk

- le tabelle spesso contengono un sample della popolazione totale
- le tabelle potrebbero non essere aggiornate o esprimere i dati con formati diversi rispetto alle fonti esterne
- le tabelle (anche quelle esterne) contengono rumore

Valutazione del rischio di disclosure

La valutazione del rischio di *disclosure* viene fatta tenendo in considerazione:

- la probabilità che il rispondente di interesse sia presente sulle tabelle di mircodata e sulle tabelle esterne
- la probabilità che le variabili di matching siano registrate in modo linkabile tra microdata e tabella esterna
- la probabilità che il rispondente di interesse è peculiare nella popolazione del file esterno

1.2 k - anonimity

La k - anonimity mira a proteggere l'identità dei rispondenti, tramite generalizzazione e soppressione, rilasciando allo stesso tempo informazioni veritiere.

Cerca di garantire che **ogni combinazione di quasi identificatori sia correlata indistintamente ad almeno k individui**.

Condizione sufficiente per soddisfare la k - anonimity

Ogni combinazione di quasi identificatori deve avere almeno k occorrenze.

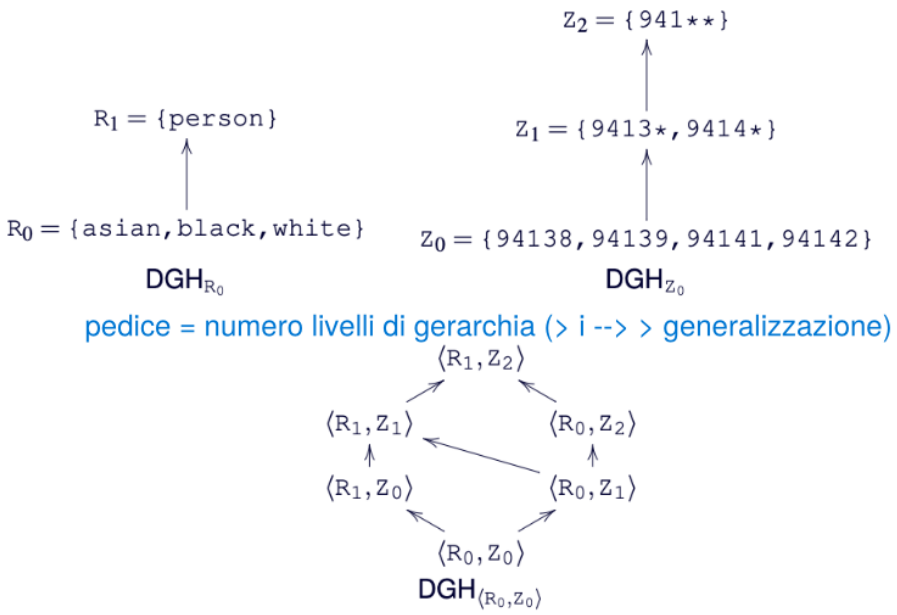
1.2.1 Generalizzazione

Consiste nel sostituire i valori di un dato attributo con dei valori più generali; si basa sulla definizione di una **gerarchia di generalizzazioni**.

Gerarchia di generalizzazione del dominio

- Una relazione di generalizzazione $\leq_D$ definisce un mapping tra il dominio D e le sue generalizzazioni.
- Dati due domini $D_i, D_j \in \text{Dom}$, $D_i \leq_D D_j$ indica che i valori nel dominio D_j sono generalizzazioni dei valori in D_i.
- $\leq_D$ implica l'esistenza, per ogni dominio D, di una gerarchia di generalizzazione del dominio $DGH_D = (\text{Dom}, \leq_D)$:
  - $\forall D_i, D_j, D_z \in \text{Dom} : D_i \leq_D D_j, D_i \leq_D D_z \Rightarrow D_j \leq_D D_z \vee D_z \leq_D D_j$. (*relazione d'ordine totale*)
  - Tutti gli elementi massimali di Dom sono singleton.
- Data una tupla di dominio $DT = \langle D_1, \ldots, D_n \rangle$ tale che $D_i \in \text{Dom}$, $i = 1, \ldots, n$, la gerarchia di generalizzazione del dominio di DT è $DGH_{DT} = DGH_{D_1} \times \ldots \times DGH_{D_n}$.

Esempio



Gerarchia di generalizzazione dei valori

- La relazione di generalizzazione dei valori $\leq_V$ associa ad ogni valore nel dominio D_i un valore unico nel dominio D_j, generalizzazione diretta di D_i.
- Questa relazione implica l'esistenza di una gerarchia di generalizzazione dei valori VGH_D per ciascun dominio D.
- La VGH_D ha una struttura ad albero:
  - **Foglie**: Rappresentano i valori nel dominio D.
  - **Radice**: È il valore più generale, situato nell'elemento massimo di DGH_D.

Esempio

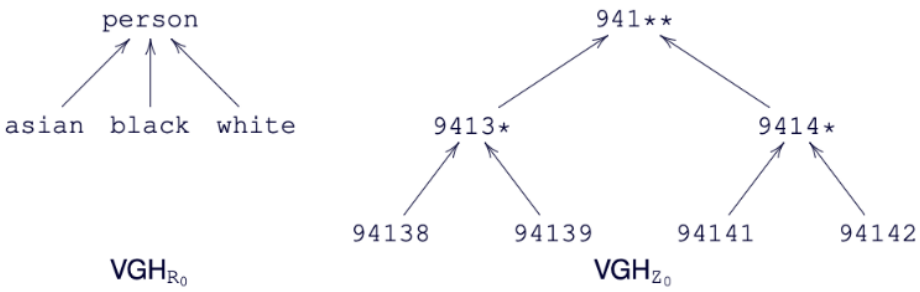


Tabella generalizzata con soppressione

Una tabella T_j è detta una generalizzazione (mediante soppressione di tuple) della tabella T_i ($T_i \preceq T_j$), se soddisfa le seguenti condizioni:

- $|T_j| \leq |T_i|$
- Il dominio $\text{dom}(A, T_j)$ di ogni attributo A in T_j è uguale o è una generalizzazione del dominio $\text{dom}(A, T_i)$ dell'attributo A in T_i.
- È possibile definire una funzione iniettiva che associa ogni tupla t_j in T_j con una tupla t_i in T_i, tale che il valore di ogni attributo in t_j sia uguale o è una generalizzazione del valore dell'attributo corrispondente in t_i.

***k-minimal* generalization con soppressione**

Siano $T_i(A_1, \ldots, A_n)$ e $T_j(A_1, \ldots, A_n)$ due tabelle tali che $T_i \preceq T_j$. Il **vettore di distanza** di T_j da T_i è definito come il vettore

$$DV_{i,j} = [d_1, \ldots, d_n],$$

dove ogni d_z per $z = 1, \ldots, n$ è la lunghezza del percorso unico tra $dom(A_z, T_i)$ e $dom(A_z, T_j)$ nella gerarchia di generalizzazione del dominio DGH_{D_z}.

Siano T_i e T_j due tabelle t.c. $T_i \preceq T_j$, e sia $MaxSup$ la soglia specificata di soppressione accettabile. La tabella T_j è detta una **generalizzazione k-minimale** della tabella T_i se e solo se:

1. T_j soddisfa la k-anonymity garantendo la soppressione minima richiesta se per ogni tabella T_z che soddisfa la k-anonymity e t.c. $T_i \preceq T_z$ e $DV_{i,z} = DV_{i,j}$, allora deve valere $|T_j| \geq |T_z|$.
2. $|T_i|$ - $|T_j| \leq MaxSup$ (non ho cancellato più del consentito)
3. $\forall T_z$ t.c. $T_i \preceq T_z$ e T_z soddisfa le condizioni 1 e 2 $\Rightarrow \neg(DV_{i,z} < DV_{i,j}) \iff DV_{i,z} >= DV_{i,j}$

Esempio

$MaxSup = 2$

Race:R_0	ZIP:Z_0
asian	94142
asian	94141
asian	94139
asian	94139
asian	94139
black	94138
black	94139
white	94139
white	94141

PT

Race:R_1	ZIP:Z_0
person	94141
person	94139
person	94139
person	94139
person	94139
person	94139

$GT_{[1,0]}$

Race:R_0	ZIP:Z_1
asian	9414*
asian	9414*
asian	9413*
asian	9413*
asian	9413*
black	9413*
black	9413*

$GT_{[0,1]}$

1.2.2 Computazione di una generalizzazione preferita

Possono essere applicati diversi criteri di preferenza:

- **Distanza assoluta minima:** minor numeri di passi di generalizzazione
- **Distanza relativa minima:** somma pesata, minimizza il mumero totali di passi relativi
- **Massima distribuzione:** maggior numero di tuple distinte
- **Minima soppressione**

1.2.3 Classificazione di tecniche per *k-anonimity*

Generalizzazione e soppressione possono essere applicate a diversi livelli di granularità:

- **Generalizzazione:** a livello di colonna o di cella
- **Soppressione:** a livello di riga, di colonna o di cella

1.3 Algoritmi per AG_TS e AG_

Computing a k-minimal solution

- Ogni percorso in DGH_{DT} rappresenta una strategia di generalizzazione per PT
- Chiamiamo *locally minimal generalization* il nodo con indice minore in ogni percorso che soddisfa la k-anonymity
- Proprietà sfruttate dall'algoritmo:
  1. Ogni k-minimal gen è localmente minima rispetto a un percorso, ma il contrario non è vero
  2. Salendo nella gerarchia, il # di tuple da rimuovere per garantire la k-anonymity diminuisce
- Se non esiste una soluzione che garantisca la k-anonymity sopprimendo meno di MaxSup tuple all'altezza h, non può esistere una soluzione con altezza inferiore a h che lo garantisca.

L'algoritmo adotta una ricerca binaria sul reticolo dei vettori distanza:

1. Valuta le soluzioni all'altezza $\lfloor \frac{h}{2} \rfloor$
2. Se esiste almeno una soluzione che soddisfa la k-anonymity:
   - Valuta le soluzioni all'altezza $\lfloor \frac{h}{4} \rfloor$
3. Altrimenti valuta le soluzioni all'altezza $\lfloor \frac{3h}{4} \rfloor$
4. Fino a quando l'algoritmo min(h) per la quale esiste un DV che soddisfa la k-anonymity

Per ridurre il costo computazionale, l'algoritmo utilizza una matrice di vettori distanza.

k-Optimize algorithm

- Ordinare gli attributi nel quasi-identificatore (QI) e i valori nei rispettivi domini.
- Associare un indice intero a ciascun valore del dominio, seguendo l'ordine definito.

Ad esempio:

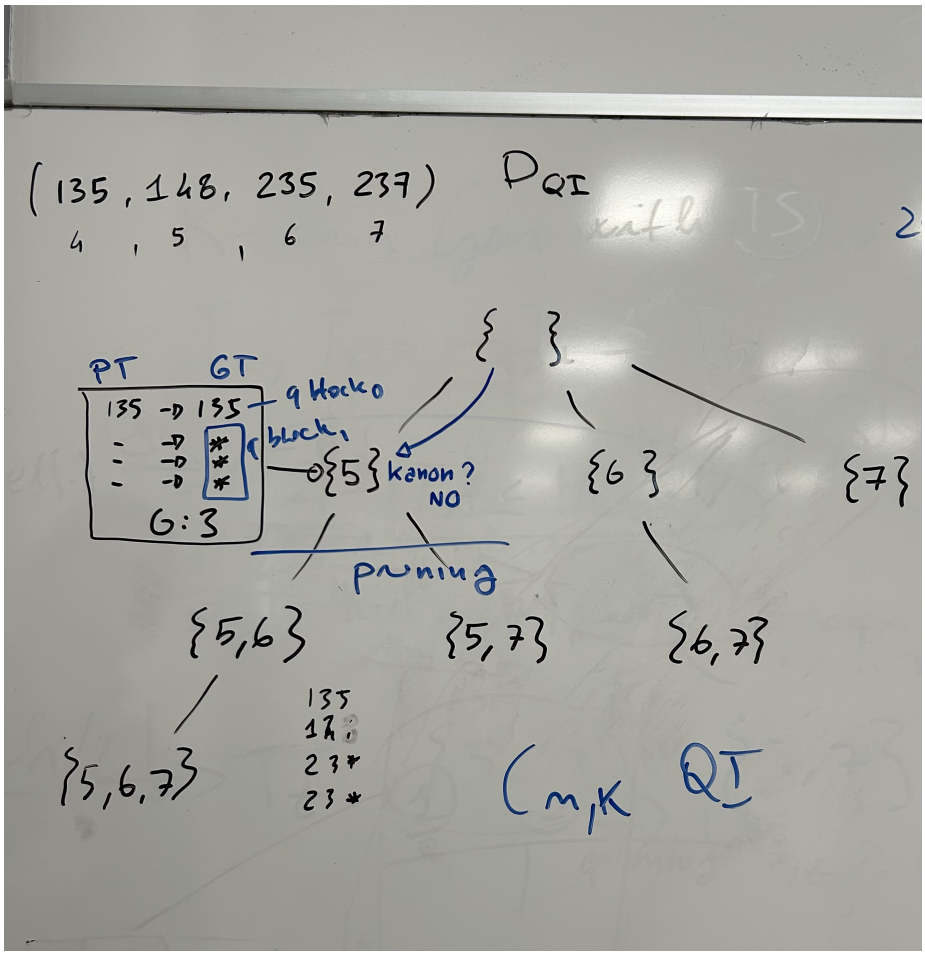
Race	**ZIP**
⟨[asian: 1] [black: 2] [white: 3]⟩	⟨[94138: 4] [94139: 5] [94141: 6] [94142: 7]⟩

- Una generalizzazione è l'unione dei singoli valori di indice.
- Il valore più basso in un dominio di attributi viene omesso. Ad esempio, {6} corrisponde a:
  - **Race**: {1}, cioè: ⟨[asian or black or white]⟩
  - **ZIP**: {4, 6}, cioè: ⟨[94138 or 94139], [94141 or 94142]⟩
- L'ordine dei valori all'interno dei domini ha un impatto sulla generalizzazione.

L'algoritmo **k-Optimize** costruisce un **albero di enumerazione** per l'insieme degli indici I.
La radice dell'albero è l'insieme vuoto $\emptyset$, e i figli di ciascun nodo n sono ottenuti aggiungendo un singolo elemento i dell'insieme I, tale che $\forall i' \in n, i > i'$. Ogni nodo ha un **costo** che riflette la quantità di generalizzazione e soppressione associata all'anomizzazione rappresentata dal nodo.
L'algoritmo cerca l'anonimizzazione con il costo minimo attraverso una **visita dell'albero** tramite **ricerca in profondità**. Tuttavia, poiché l'albero ha $2^{|I|}$ nodi, la visita completa non è praticabile. Quindi viene adottata una strategia di **potatura (pruning)**:

- Un nodo n viene potato se nessuno dei suoi discendenti può fornire una soluzione ottimale.
- Questo si determina calcolando un **lb:limite inferiore** sul costo dei nodi nel sottoalbero radicato in n. Se il limite inferiore è maggiore del miglior costo corrente, il nodo n viene potato.

LO SPIEGHINO FATTO DA NOI:



L'algoritmo considera il dominio degli attributi del quasi-identifier e li indicizza globalmente (indice numerico per ogni valore di ogni attributo, prendendo tutti i valori di tutti gli attributi). A partire da questi ultimi fa un albero nel quale è presente ogni possibile partizionamento degli attributi, corrispondente alla creazione di cluster nella tabella. Ogni cluster avrà la propria generalizzazione e il proprio costo (per costo si intende quanti passi di generalizzaizone servono). Ogni nodo del grafo corrisonde a un partizionamento e ogni figlio è un partizionamento successivo del padre (è un partizionamento della parte destra). L'algoritmo, partendo dalla radice, esegue depth first search valutando per ogni nodo la k-anonimity della tabella: se la k-anonimity non è rispettata viene fatto **pruning** su tutti i nodi figli e il depth firsy search continua sui nodi rimanenti. Se, invece, k-anpnimity è rispettata depth first search continua verso

i figli. Viene infine scelta la soluzione che , soddisfando k-anonimity, ha il costo minore (minor numero di passi di generalizzazione).
Attenzione: se vediamo 6 significa che i cluster sono due: i numeri prima di 6, i numeri dopo 6 compreso il 6. Se invece troviamo 5,6 allora i cluster saranno 3 $\rightarrow$ 4 | 5 | 6 7
Es: a livello 5 i valori vengono divisi in due cluster, il primo con i valori prima di 5, il secondo con i valori dopo 5 (5 compreso). In questa situazione ci troveremo con due cluster, il primo formato solo dal valore 135, il secondo con i valori 148, 235, 237. Il valore 135 essendo da solo non necessita di nessun passo di generalizzazione, a differenza del secondo cluster che, avendo tre valori completamente differenti andrà generalizzato al massimo. L'algorito elabora una soluzione e verifica che rispetto alla nostra tabella, e al k richiesto, la soluzione sia k anonima. Se è k anonima l'algoritmo va a controllare le soluzioni discendenti che potrebbero contenere soluzioni migliori, nel caso non lo sia applica la tecnica del **pruning** e va a tagliare tutte le soluzioni sottostanti poichè non conterranno soluzione.

Incognito Algorithm

L'algoritmo **Incognito** verifica **k-anonimity** con riferimento a un adeguato sottoinsieme del QI.
Esso adotta un approccio **bottom-up** per visitare le gerarchie di generalizzazione dei domini (DGHs). La condizione di k-anonimity rispetto a un sottoinsieme di QI è necessaria, ma non sufficiente per garantire la k-anonimity rispetto a tutto il QI. Il processo iterativo dell'algoritmo procede come segue:

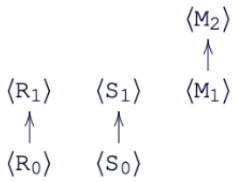
- **Iterazione 1**: si controlla la **k-anonimity** per ciascun attributo singolo in QI, scartando le generalizzazioni che non soddisfano la k-anonimity.
- **Iterazione 2**: si combinano le generalizzazioni rimanenti in coppie, verificando la **k-anonimity** per ciascuna coppia ottenuta. Scartando le coppie che non soddisfano la k-anonimity.
- **Iterazione n**: si considerano tutte le n-uple di attributi ottenuti dalle generalizzazioni che soddisfavano la k-anonimity nell'iterazione $i-1$, scartando le soluzioni che non la rispettano.
- ...
- **Iterazione** $|\mathbf{QI}|$: restituisce il risultato finale, che rappresenta una generalizzazione che soddisfa la k-anonimity rispetto all'intero quasi-identificatore (QI).

L'algoritmo procede dunque costruendo progressivamente soluzioni, partendo da singoli attributi e combinandoli in gruppi via via più grandi fino a considerare tutti gli attributi del quasi-identificatore.

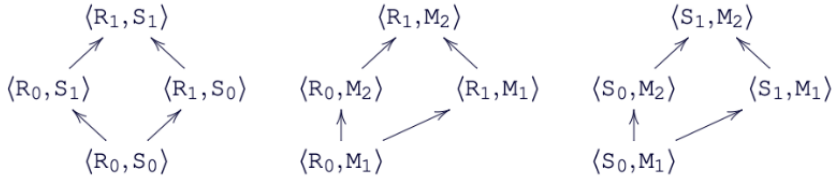
Esempio

Race	Sex	Marital status
asian	F	divorced
asian	F	divorced
asian	F	married
asian	M	married
asian	M	married
black	F	single
black	F	single
white	F	single
white	F	widow

Iteration 1



Iteration 2



In questo caso vogliamo un $k = 2$; per questa ragione l'ultima riga della tabella (*widow*) viene scartato poiché c'è solo un rispondente per tale valore.

È per questo che M_0 non è presente; viene sopppresso e si parte da M_1.

1.4 Algoritmi per _CS e CG_

1.4.1 Mondrian Multidimensional Algorithm

L'algoritmo **Mondrian Multidimensional** si basa su una rappresentazione spaziale delle tuple e dei quasi-identificatori:

- Ogni attributo nel quasi-identificatore (**QI**) rappresenta una dimensione.
- Ogni tupla nel set di dati privati (**PT**) rappresenta un punto nello spazio definito da **QI**.
- Le tuple con lo stesso valore di **QI** sono rappresentate assegnando una molteplicità ai punti.
- Lo spazio multidimensionale viene partizionato dividendo le dimensioni in modo tale che ogni area contenga almeno k occorrenze di valori dei punti.
- Tutti i punti in una regione vengono generalizzati a un valore unico.
- Le tuple corrispondenti sono sostituite dalla generalizzazione calcolata.

L'algoritmo Mondrian è flessibile e può operare:

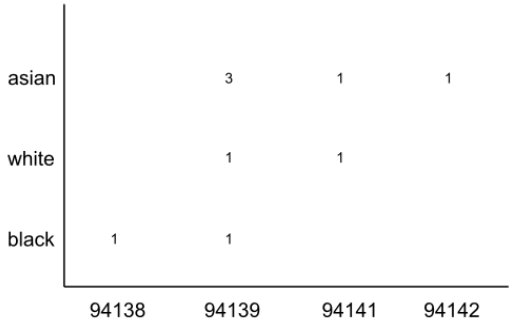
- **Su un numero diverso di attributi:**
  - *Single or Multi-dimension.*
- **Con diverse strategie di generalizzazione:**
  - *Global or Local recoding*: colonna o cella.
- **Con diverse strategie di partizionamento:**
  - *Strict or Relaxed partitioning*: senza o con possibili sovrapposizioni (con *relaxed* due occorrenze uguali possono appartenere a cluster diversi).
- **Utilizzando metriche diverse per determinare come dividere ogni dimensione.**

Esempio

Wished $k = 3$

Race	ZIP
asian	94142
asian	94141
asian	94139
asian	94139
asian	94139
black	94138
black	94139
white	94139
white	94141

PT

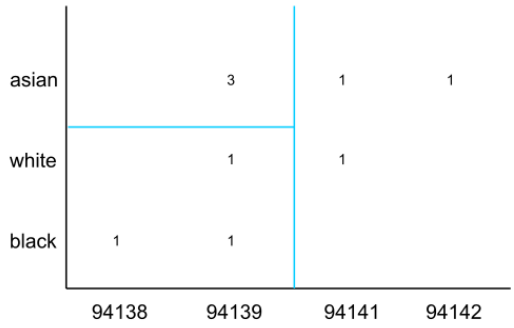


Ogni tupla è un punto nello spazio; in questo caso ogni taglio deve rispettare $k = 3$. C'è un'idea di ordinamento dei valori; con i numeri ha senso con altri valori è forzato.

Le tuple vengono divise in cluster; le tuple di ogni cluster vengono rese uguali generalizzandole.

Race	ZIP
asian or white	9414*
asian or white	9414*
asian	94139
asian	94139
asian	94139
black or white	9413*
black or white	9413*
black or white	9413*
asian or white	9414*

GT



1.4.2 k-anonymity Revisited

La **k-anonymity** cambia a seconda del livello di generalizzazione applicato:

- **AG:** Ogni n-upla di quasi-identificatori deve apparire almeno k volte.
- **CG:** La condizione di avere almeno k occorrenze è sufficiente ma non necessaria. È possibile utilizzare un requisito meno restrittivo:

1. Per ogni sequenza di valori pt in $PT[QI]$, ci devono essere almeno k tuple in $GT[QI]$ che contengono una sequenza di valori che generalizzano pt.
2. Per ogni sequenza di valori t in $GT[QI]$, ci devono essere almeno k tuple in $PT[QI]$ che contengono una sequenza di valori per cui t è una generalizzazione.

La generalizzazione a livello di cella permette una maggiore flessibilità rispetto alla gen a livello di attributo.

1.5 Attribute disclosure

La *k-anonimity* è vulnerabile a diversi attacchi.

- **Omogeneità dei valori di attributi sensibili:** se tutti gli appartenenti ad un gruppo hanno la stessa informazione sensibile, mi basta conoscere che appartieni a quel gruppo per sapere che hai quella informazione sensibile

Race	DOB	Sex	ZIP	Disease
...	...	...	...	...
black	64	F	941**	short breath
black	64	F	941**	short breath
...	...	...	...	...

- **Conoscenza pregressa:** è conoscenza a livello di istanza che posso avere e mi permette di scartare alcune possibilità; nella figura, se so che corri due ore al giorno deduco che non hai il fiato corto.

Race	DOB	Sex	ZIP	Disease
...	...	...	...	...
white	64	F	941**	chest pain
white	64	F	941**	short breath

1.5.1 ℓ-Diversity

Definiamo un *q-block* come un gruppo di tuple con lo stesso *quasi-identifier*; diciamo che un *q-block* è ℓ-diverse se contiene almeno ℓ valori **differenti** e ***ben rappresentati*** per l'attributo sensibile.

Questo implica che un attaccante deve eliminare almeno $\ell-1$ valori possibili per inferire un valore sensibile di un rispondente.

Una tabella è ℓ-diverse se tutti i suoi *q-block* sono ℓ-diverse; questo implica che:

- l'attacco di omogeneità non è possibile
- l'attacco di conoscenza pregressa è più difficile (devo eliminare ℓ-1 possibilità)

ℓ-diversity può lasciare spazio a degli attacchi basati sulla distribuzione dei valori all'interno dei *q-block*.

Skewness attack (attacco di distorsione)

Avviene quando un *q-block* ha una distribuzione diversa da quella del mondo reale; quando faccio i gruppi devo sia avere dei valori diversi che dei valori simili a quelli attesi.

Attacco di similarità

Avviene quando un *q-block* contiene dei valori che sono diversi ma semanticamente simili (ad esempio, ulcera allo stomaco/gastrite).

t-closeness

Diciamo che un *q-block* rispetta *t-closeness* se la distanza tra la distribuzione dei valori degli attributi sensibili nel *q-block* e quella della popolazione di riferimento è minore di *t*.

Una tabella rispetta *t-closeness* se tutti i suoi *q-blocks* rispettano *t-closeness*.

1.5.2 Tipi di conoscenza pregressa

Le conoscenze possono riguardare:

- l'individuo target
- altri individui, il che potrebbe comunque rivelare informazioni sensibili
- famiglie di valori uguali, come informazioni genomiche che collegano un gruppo di persone.

1.5.3 Rilasci multipli

I dati potrebbe essere soggetti a rilasci multipli, come aggiornamenti o pubblicazioni ricorrenti. Con il rilascio multiplo di multiplo ci si espone ad attacchi di intersezione,

1.5.4 *m-invariance*

Per affrontare il problema dei rilasci longitudinali, una sequenza $T_1, ..., T_n$ di tabelle di microdati rilasciate soddisfa la proprietà di *m-invariance* se:

- ogni classe di equivalenza contiene almeno m tuple;

- nessun valore sensibile appare più di una volta in ciascuna classe di equivalenza;
- per ogni tupla t, le classi di equivalenza a cui appartiene t nella sequenza sono caratterizzate dallo stesso insieme di valori sensibili.

Ciò implica che la correlazione delle tuple in $T_1, ..., T_n$ non permette a un destinatario malevolo di associare meno di m valori sensibili differenti a ciascun rispondente.

1.6 *k-anonimity* in altre applicazioni

1.6.1 Social Networks

In una rete sociale ciò che ti può rendere peculiare è il numero di connessioni che hai (esempio influencer); si cerca di avere ogni nodo uguale ad almeno altri k, dove k è il grado di protezione che voglio ottenere.

Per fare questo si è possibile sopprimere o aggiungere archi.

1.6.2 Data Mining

Il *k-anonymous data mining* mira a garantire che i risultati del data mining non violino i requisiti di *k-anonymity* sui dati originali. Alcuni esempi di tecniche per compromettere la k-anonymity sfruttando il data mining includono:

- **Association Rule Mining**: tecniche per trovare regole di associazione possono compromettere la k-anonymity.
- **Classification Mining**: tecniche di classificazione possono portare a minacce per la privacy.

1.6.3 Location-Based Services

Bisogna preocupparsi del fatto che la locazione di un individuo potrebbe rivelare la sua identità. Così come si generalizza il valore dei dati per aumentare il numero degli utenti ed avere più incertezza, lo stesso viene fatto con la posizione.

Si può adottare il concetto di *k-anonimity* come segue:

- Considerare solo le aree che contengono almeno k individui
- Ingrandire l'area per includere almeno altri $k-1$ utenti (*k-anonymity*)
- Obfuscazione delle aree (*location privacy*) per ridurre la precisione o la confidenza dei dati; magari non si può semplicemente ingrandire perché l'utente si troverebbe al centro
- Protezione del percorso degli utenti (*trajectory privacy*) tramite modifica delle traiettorie

1.7 Privacy Sintattica e Semantica

- **Syntactic Privacy:** le definizioni di privacy sintattiche misurano il grado di protezione di una persona nei dati con un valore numerico.

  Ad esempio:

  – Ogni rilascio di dati deve essere indistinguibilmente associato ad almeno un certo numero di individui nella popolazione.

- **Semantic Privacy:** le definizioni di privacy semantiche soddisfano un requisito di privacy semantico.

  Ad esempio:

  – Il risultato di un'analisi eseguita su un dataset rilasciato non deve essere influenzato dalla presenza o assenza di una singola tupla nel dataset.

1.8 Differential Privacy

La *Differential Privacy* mira a prevenire che un attaccante sia in grado di stabilire la presenza o l'assenza di un individuo in un dataset. È un tipo di *semantic privacy*.

Definizione informale

La distribuzione della probabilità sui dati pubblicati deve essere *essenzialmente la stessa* indipendentemente dal fatto che un individuo sia incluso o meno nel dataset.

Formally:

- A randomized function K gives ε-differential privacy if for all data sets D and D' differing on at most one row, and all $S \subseteq \text{Range}(K)$, $\Pr[K(D) \in S] \leq e^{\varepsilon} \times \Pr[K(D') \in S]$

L'obiettivo della funzione randomizzata è quello di aggiungere rumore; c'è da tenere in considerazione il *trade-off* tra introduzione del rumore e utilizzabilità dei dati.
e^{ϵ} indica il *privacy budget*, che diminuisce man mano che il dataset viene interrogato; quando si esaurisce non si potrà più interrogare quel dataset.

- $\epsilon = 0 \rightarrow$ non mi dice niente, risposta qualsiasi
- $\epsilon = 1 \rightarrow$ dato preciso, utile ma poca privacy
- $0 < \epsilon < 1 \rightarrow$ c'è del rumore ma ho dati utili

La *differential privacy* può essere applicata in due scenari:

- **Interattivo:** valutazione a run-time delle query (statistical DBMS)
- **Non interattivo:** rilascio di tabelle pre-computate (statistical data)

Viene rinforzata aggiungendo del rumore casuale, a discapito della veridicità dei dati.

k-anonimity* vs *differential privacy

- ***k-anonimity***
  - rappresenta bene il mondo reale
  - protezione non completa
- ***differential privacy***
  - garantisce una miglior protezione
  - non garantisce protezione completa, è più complicato fare *enforce*

Capitolo 2

Alcuni esempi di altri problemi di privacy

2.1 Distribuzione di valori sensibili

Riesco ad inferire informazioni non dalla singola tupla, ma dall'insieme di tuple.

Esempio: Soldiers' Medical Records

- I record individuali non sono sensibili.
- La distribuzione dell'età dei soldati in una località può indicare il tipo di località:
  - Soldati giovani suggeriscono tipicamente un campo di addestramento.
  - Funzionari più anziani indicano un quartier generale.

2.2 Dati del genoma

Le informazioni genomiche presentano opportunità in medicina ma sollevano anche diversi problemi di privacy:

- Il genoma umano può identificare il suo proprietario
- Contiene informazioni sensibili sulla provenienza etnica, predisposizione a malattie e altri tratti fenotipici
- I dati genomici possono rivelare informazioni sui parenti e sui discendenti sulla base del genoma (non solo tua)

2.3 Social Media

Le nostre attività sui social media e i "like" possono rivelare informazioni sensibili.

È importante notare che i social media condividono frequentemente i nostri dati con terze parti, come inserzionisti e aziende di analisi, il che può portare a violazioni della privacy.

2.4 Dati Biometrici

La privacy dei dati biometrici solleva ulteriori preoccupazioni; sono sistemi in grado di identificare gli utenti senza il loro consenso.