Differential Privacy

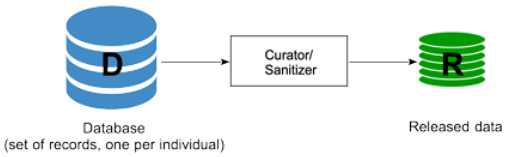
Parte IV

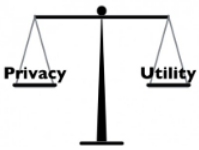
Indice

Introdotto nel 2006, ha avuto un'esplosione di utilizzo di questo concetto negli ultimi anni.
L'idea è la protezione dei dati, i dati sono privati e non basta che siano semplicemente anonimizzati.

Capitolo 1

Basic Scenario





1. **D:** Bisogna avere molti dati (se no D.P. non funziona bene), pensiamo di avere tanti record e ciascun record riguarda dati sensibili di individui.
2. **Curator/Sanitizer:** è una parte fidata che può accedere in chiaro ai dati
3. **R:** Dati rilasciati

1.1 L'intuizione classica per la privacy

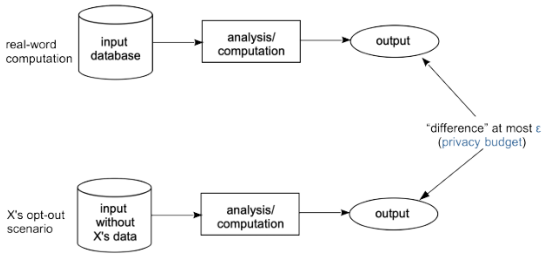
Cosa vuol dire mantenere la privacy degli individui?

- Ciò che viene rilasciato non dipende dai miei dati
- le informazioni apprese su un individuo dai risultati pubblicati R non sono più di quelle che possiamo apprendere su quell'individuo senza avere accesso a R.

Problema: Se gli individui non hanno impatto sui risultati allora i risultati non avrebbero utilità, serve equilibrio.

1.2 Differential privacy– Intuition

Quello che cerca di fare D.P. è che la presenza o l'assenza di un utente non cambi di molto il rischio di compromettere la sua privacy. Chiaramente il rilascio di informazioni impone che un minimo cambiamento ci sia, però sia minimale. Le inferenze che io posso fare su un individuo da un rilascio R deve essere lo stesso che io posso inferire utilizzando i dati di qualunque altro utente.



Più questo ϵ è piccolo più la differenza dei dati con o senza un determinato individuo è minimale. Tutto questo viene fatto iniettando del rumore all'interno della mia computazione. Deve essere rumore casuale (NON deterministico)
Due principali complicazioni:

- Capire quanto rumore iniettare
- Il risultato della computazione eseguita più volte sullo stesso dataset è sempre diverso, poichè ogni volta viene inserita una quantità di rumore diversa (se viene eseguita la stessa computazione troppe volte si riesce a inferire il valore corretto, da questo nasce la necessità del **Privacy budget**)

1.3 Definizione formale di Differential Privacy

Siano due database D e D' due databse vicini (differiscono per un singolo individuo)

Un algoritmo A soddisfa ϵ-Differential Privacy se per tutte le coppie di database vicini D, D', e per tutti gli outputs o:
$$P[A(D) = o <= e^{\epsilon}]\ P[A(D') = o]$$

N.B.:

- La definizione si riferisce a una computazione (non ai dati) quindi ad un algoritmo A, ed è l'algoritmo che soddisfa la definizone di D.P.
- D.P. è soddisfatta quando qualunque sia la coppia di dataset D e D' che vengono considerati (basta che siano vicini), quando applicando una determinata computazione su D, la probabilità di ottenere un determinato risultato più o meno deve essere uguale alla probabilità di ottenere lo stesso risultato applicando lo stesso algoritmo su D'

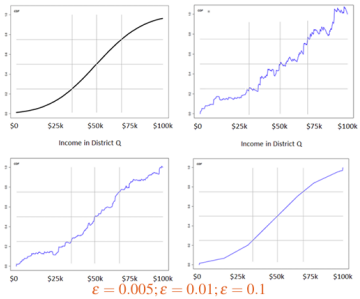
- La variazione dipende da questo parametro ϵ
- Per un attaccante è impossibile capire se un determinato risultato viene prodotto da D o da D'
- Tutto questo funziona bene se abbiamo grandi quantità di dati, se sono piccole si rischia di dover inserire troppo rumore e che i dati non siano più utili

1.4 Pivacy budget ϵ

Determina quanto **rumore** viene inserito nella computazione, un trade-off tra utilità e privacy.

- **piccolo ϵ:** più privacy, meno utilità
- **grande ϵ:** meno privacy, più utilità

Se $\epsilon = 0$ allora avremmo due distribuzioni di probabilità esattamente identiche avremmo la privacy maggiore in assoluto, ma bassissima utilità dei dati poichè il risultato della computazione non dipenderebbe dai dati di nessun indviduo (quindi non dipende dall'input, risultato casuale). Man mano che ϵ aumenta aumenta anche la diversità e quindi l'impatto di ogni singolo individuo, minore privacy ma maggiore utilità. Il miglior valore di ϵ non si sa con certezza, si considerano buoni valori di ϵ valori minori di uno.



1.5 Global sensitivity

Il problema di questa tecnica è capire quanto rumore inserire nel risultato della computazione per garantire il soddisfacimento della definizione, l'idea è quella di calibrare la quantità di rumore sulla base di qual è l'influenza sui dati di un singolo individuo sul risultato della computazione.

- **Sensitività globale:** caratterizza l'entità dell'influenza di un singolo individuo (caso peggiore), e quindi la quantità di rumore che si deve aggiungere

Esempio 1(scelto anche esempio di conteggio poichè sono quelli che funzionano meglio con questa tecnica)

Supponiamo di avere questi dati da un ospedale e di voler calcolare quanti pazienti hanno l'influenza:

Sex	Height	DoB	Disease	Drug X
M	6'2"	1960-03-25	Obesity	3.5
F	5'3"	2001-05-05	Diabetes	2.3
F	5'9"	1998-11-13	Healthy	1.0
M	5'3"	2000-10-05	Flu	3.7
M	6'7"	1995-02-22	Flu	2.2
...	...	...	...	...

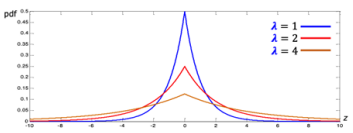
Immaginiamo che nel DB D ci sono 50 pazienti con l'influenza, bisogna capire quanto rumore iniettare per soddisfare D.P. Togliendo uno qualunque dei 50 pazienti ne avremmo 49, quindi noi vogliamo mascherare il fatto che anche solo togliendone uno questo ha un impatto.
Nel caso peggiore il nostro impatto è pari a 1, se uno di questi individui con influenza c'è abbiamo 50 pazienti, se no sono 49.
Per questo motivo in questo esempio la **global sensitivity** è pari a 1.
Per qualsiasi tipo di conteggio la global sensitivity sarà sempre 1.

Esempio 2 (sulla stessa tabella)

Supponiamo di voler calcolare la somma di quanto di un determinato mediciale è stato somministrato ai pazienti. In questo caso la global sensitivity, togliendo i dati di un individuo, raramente sarà pari a 1. Inoltre noi vogliamo stimare il **caso peggiore**.
Avendo solo i pazienti visibili in tabella, il paziente numero 4 sarebbe il caso peggiore ossia la G.S. sarebbe pari a 3.7, ma nel caso in cui venisse aggiunto un nuovo paziente più impattante la G.S. cambierebbe quindi non è semplice capire come calcolare la G.S, questo esempio per capire che già per una computazione piccola è difficile stimarla.
Nella slide infatti viene assunto che i valori di questo medicinale possono essere compresi solamente tra 1 e 4, in questo modo calcoare il caso peggiore è semplificato.

1.6 Tecniche di generazione di rumore

Per esempio si può usare la Distribuzione di Laplace, caratterizzata dal parametro λ che nel nostro caso è ottenuto come Global sensitivity/ϵ



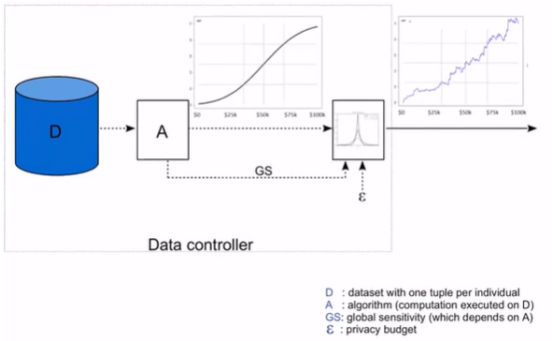
$\text{Lap}(z,\lambda)=P(z \mid \lambda) = \frac{1}{2\lambda} e^{\frac{-|z|}{\lambda}}$, $\lambda = \frac{GS(A)}{\epsilon}$

Più è grande la Global sensitivity più è grande λ ed è inversamente proporzionale ad ϵ

- $\lambda = 1$**:** alta probabilità di aggiungere una piccola quantità di rumore, l'area maggiore è nell'intorno dello zero
- λ **aumenta:** più lambda aumenta più le curve sono flat, sopratutto allontanandosi dallo zero. Quindi più λ è grande maggiore sarà la probabilità di aggiungere tanto rumore

1.7 Global picture

Mettiamo insieme tutto:



Applicare Differential privacy vuol dire:

1. Parto da una collezione di dati D, contenenti informazioni sensibili
2. Sui quali vogliamo eseguire una computazione A
3. In base alla computazione che vogliamo eseguire calcoliamo la Global sensitivity
4. Applico la computazione A sullla collezione di dati D, ottengo il risultato
5. Sporco il risultato con il **rumore** applicando delle tecniche tipo quella di Laplace
6. Restituisco il risultato sporcato

Capitolo 2

Proprietà di Differential Privacy

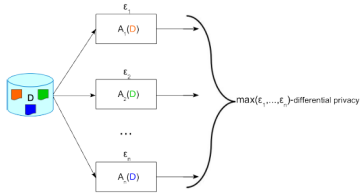
2.1 Chiusura del post-processing

Resiliente a operazioni di post-procesing.
C'è una fase di post-processing da fare sui risultati aggiunti di rumore prima di rilasciarli.
Questa proprietà ci dice che non vi è differenza tra i risultati prima e dopo il post-processing, entrambi rispettano la proprietà di Differential Privacy

2.2 Composizione parallela

"Differential privacy si compone bene con se stessa" ma cosa vuol dire?

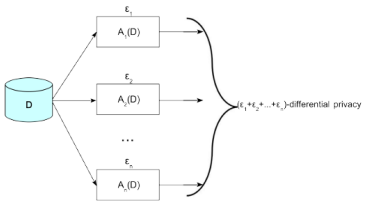
Composizione parallela: sequenza di m computazioni su sottoinsiemi disgiunti di una base di dati D



Da una parte abbiamo i dati D, a bbiamo tante computazioni fatte su questi dati. Successivamente i risultati di queste computazioni vengono rilasciate (sempre sugli stessi dati), rilasciando più computazioni è sempre più probabile inferire qualcosa sui dati iniziali.
Con D.P. possiamo quantificare precisamente quanto varia la privacy degli individui ma mano che rilascio i risultati delle computaioni, e posso definirlo in modo preciso:
Se io eseguo varie computazioni so precisamente quali parte del DB vanno a toccare, quindi è vero che eseguo m computazioni ma non è detto che ci siano intersezioni tra di esse. Tutte le informazioni sono protette con un ϵ che è il massimo degli ϵ che è stato usato

2.3 Composizione sequenziale

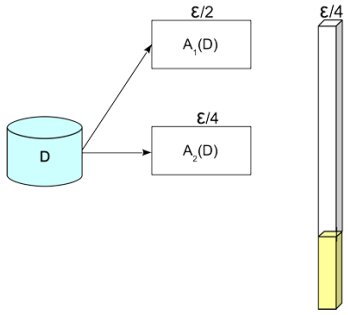
Composizione sequanziale: sequenza di m computazioni su database D con risultati sovrapposti



La differenza rispetto al caso precedente è che riguardano individui non disgiunti, più computazioni sugli stessi individui.
In questo caso rilasciare tutte queste informazioni equivale a rilasciarne una contenente tutte le informazioni, e ϵ verrà calcolato sommando i singoli ϵ utilizzati nelle singole computazioni

2.4 Perchè ϵ è chiamato "Privacy budget"?

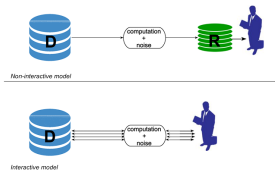
ϵ rappresenta il livello di protezione che voglio mantenere, ogni volta che eseguo una computazione, è come se utilizzassi parte di questo budget.
Quando il budget si esuarisce, non posso più eseguire computazioni poichè ho raggiunto l' ϵ che mi ero prefissato



Capitolo 3

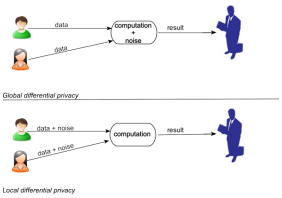
Modelli di Differential Privacy

3.1 Interattivo VS non Interattivo



- **Interattivo:** valutazione delle query run-time
- **Non interattivo:** chi ha in mano i dati decide cosa rilasciare a priori, rilascio di dati pre-computati in macrotabelle

3.2 Globale VS Locale



- **Globale:** c'è un'entità fidata che esegue la computazione sui dati proteggendo il risultato della computazione con differential privacy

- **Locale:** non c'è più una parte fidata che riceve i dati degli utenti, c'è qualcuno che colleziona i dati ma non essendo fidato non riceve i dati originale ma dati già sporcat di rumore

3.2.1 Definizione formale di Local Differential Privacy

Un algoritmo randomizzato K soddisfa ϵ-local differential privacy se e solo se per ogni input x,x' e output o di A:

$$P[K(x) = o] <= e^{\epsilon}\ P[K(x') = o]$$

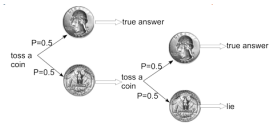
Al posto di avere D e D' che sono collezioni di dati, ma sono i dati di un individuo, tutto il resto si applica allo stesso modo della Global.

Capitolo 4

Differential Privacy nel mondo reale

4.1 Privacy in practice

Differential privacy basata sul **Lancio della moneta** molto usata in Google e Apple.
si basa sull'idea di lanciare la moneta



Questo stesso meccanismo è usato per proteggere la privacy dei miei dati prima che finiscano in mano alla terza parte non fidata, viene protetto ogni singolo bit in questo modo.
La protezione ottenuta è pari a $\epsilon = ln(3)$

4.2 Esempio dei censimenti

L'Ufficio del Censimento degli Stati Uniti ha distribuito OnTheMap, un'applicazione basata sul web che mostra dove i lavoratori sono occupati e dove vivono.
E' basato su un'idea diversa di differential privacy, ovvero:

$$(\epsilon, \delta)\text{-}\textit{differential privacy}$$

- ϵ è il privacy budget
- δ ci dice quanto il risultato rilasciato è maggiore rispetto all'ϵ che mi ero prefissato

Capitolo 5

Problemi con differential privacy

Ha rivoluzionato molto ma ha anche delle problematiche:

- Finchè si tratta di conteggi va tutto molto bene, ma con altre computazioni è difficile stabilire la global sensitivity
- A volte viene inserito troppo rumore per proteggere gli outlier
- Come settare il valore di ϵ, viene suggerita la soluzione che quando il privacy budget è esaurito allora le computazioni verranno eseguire su dati sintetici(difficile)