ETICA SOCIETA' E PRIVACY

Lezione 1

Cos'è la privacy

Il primo cenno alla privacy fa riferimento al primo procedimento penale negli stati uniti. Nel 1890 warren e brandeis scrivono the right to pricacy.

La privacy veniva vista come il diritto a rimanere soli e inizia ad essere intesa come la possibilità di una persona a partecipare ad una società senza che altri individui potessero collezionare informazioni che gli riguardassero. Successivamente fu intesa come un sistema per limitare l'accesso alle persone.

▪ Godkin (1880): "nothing is better worthy of legal protection than private life, or, in other words, the right of every man to keep his affairs to himself, and to decide for himself to what extent they shall be the subject of public observation and discussion."

▪ Bok (1989): privacy is "the condition of being protected from unwanted access by others—either physical access, personal information, or attention."

La Privacy ha il controllo su tutte le informazioni collegate all'individuo.

▪ Westin and Blom-Cooper (1970): "privacy is the claim of individuals, groups, or institutions to determine for themselves when, how, and to what extent information about them is communicated to others."

▪ Fried (1968): "Privacy is not simply an absence of information about us in the minds of others; rather it is the control we have over information about ourselves."

Ciò che veniva deciso a livello di stantard internazionale riguardo alla privacy

- FIRST PUB 41: Diritto di un'entità di determinare il livello a cui questa entità interagisce con l'ambiente e quello di voler condividere informazioni con gli altri.
- ISO: la privacy viene definita come il diritto di controllare o influenzare quali informazioni riguardanti l'individuo possono essere memorizzate e a chi possono essere distribuite.
- DEFINIZIONE ADOTTATA: La privacy è l'abilità di una persona di controllare la disponibilità delle informazioni e le esposizioni su di essa. Questa definizione collegata al bisogno di poter funzionare nella società in maniera anonima.

Perché la privacy è così importante?

La privacy riguarda un sentimento dell'individuo che può sentirsi a disagio nel momento in cui delle informazioni che lo riguardano vengono diffuse nella comunità. Un altro sentimento legato all'individuo è quello dell'insicurezza su come le informazioni vengono utilizzate ed in quali ambiti.

Quindi il compito delle aziende sarà quello di mettere a proprio agio i clienti nel trattamento dei dati personali in modo tale da assumere una posizione di fiducia da parte del cliente.

Un aspetto molto importante riguarda i dati personali dall'uso alla raccolta di essi. La prima preoccupazione è quella di capire come le informazioni vengono usate e cosa ne fanno.

Quali tipi di privacy conosci?

Quelle più importanti sono:

- la privacy politica, ovvero la scelta dei propri rappresentanti in maniera anonima.
- la privacy che riguarda le abitudini di consumo.
- la privacy medica, sicuri di poterci curare senza che altri lo sappiano.
- La propietà privata dove ognuno può racchiudere i propri effetti personali
- La information technology end-user privacy è quella più importante e racchiude tutte le precedenti

DATA PRIVACY

Un problema di data privacy esiste ogni qual volta un dato che sono univocamente identificabili per un gruppo di persone, sono memorizzati in forma digitale o in forma cartacea.

Un esempio sono i dati delle cartelle sanitarie, indagini per la giustizia criminale, i tratti biologici e materiale genetico, ecc..

La sfida principale riguardante la privacy è quella dell'utilizzo, cioè la condivisione dei dati, cercando di proteggere le informazioni che riguardano le persone.

L'idea di condividere i dati in forma aggregata nasce per non permettere la diffusione di esse, permettendo l'anonimato delle persone.

Il diritto alla privacy cambia da paese a paese, infatti gli stati uniti, canada, unione europea hanno concezioni diverse anche se l'idea di fondo è la stessa.

Negli stati uniti non esiste una legge unica sulla privacy e tutti gli aspetti riguardanti, come nel caso dell'Italia il GDPR.

Negli stati uniti, quando uno in cui qualcuno entra in contatto con dei dati personali, quest'ultimo può farne ciò che vuole.

Esistono delle eccezioni regolamentare in singoli esempi di legge che trattano degli aspetti molto specifici della privacy negli stati uniti

Una legge è quella di Health Insurance Portability and Accountability Act (HIPAA), ovvero le assicurazioni sulla salute e la possibilità di trasferire questa su altre compagnie e sulla responsabilizzazione di essa.

Questa legge aveva un obiettivo, cioè quello di stabilire delle procedure per l'esercizio al diritto della privacy riguardanti le informazioni sulla salute, attraverso procedure precise per l'accesso ad esse. L'altro è quello dell'utilizzo della diffusione delle informazioni riguardanti l'individuo attraverso una dichiarazione di utilizzo.

Quindi, la cosa importante è quella di utilizzare un meccanismo di autenticazione per l'accesso ai dati.

Il mezzo con cui una persona può essere autentificato riguardo la propria salute è SSN che permette di correlare qualsiasi informazione riguardo la sua salute ed è valido in tutto il

paese. Grazie a questo sono possibili molte operazioni che è possibile farle sono tramite SSN.

Un altro provvedimento è quello che riguarda i minori che si applica alla raccolta dei dati personali sotto l'eta dei 13 anni, chiamato children's online privacy protection act (COPPA)

Questa legge però non regolamentava i siti web alla quale loro potevano avere accesso. Quello che viene fatto attualmente è la richiesta dell'età al momento dell'iscizione.

Un'altra legge è Fai credit reporting act (FCRA) che cerca di stabilire delle regole che assicurano l'accuratezza delle informazioni riguardanti i clienti che potevano essere contenute nei siti. L'idea era quella di proteggere i cittadini da informazioni non accurate o poco veritiere riguardante i credit reports. Questa legge ha permesso di eliminare o diminuire il più possibile la proliferazione di alcune tipologie di credit reports che riguardava informazioni molto importanti, a livello di gossip, per un individuo facilmente identificabile. Accesso per i prestiti e le assicurazioni.

Computer security and privacy laws: riguardante la privacy online.

- 1974 Family Educational Rights and Privacy Act (FERPA)
- 1986 U.S. Electronic Communications Privacy Act (ECPA)
- 1988 U.S. Video Privacy Protection Act
- 2001 U.S. Provide Appropriate Tools Required to Intercept and Obstruct Terrorism (PATRIOT) Act

Privacy law in Canada: PIPEDA specifica le regole che permettono la diffusione delle informazioni personali, riconosce la privacy dell'individuo e stabilisce delle regole sull'utilizzo di queste informazioni. Nata per adattarsi alla legge europea.

Privacy law in the European Union; a differenza degli stati uniti la privacy è molto regolamentata e viene presa in esempio per gli altri paesi.

Regolamentata dall'articolo 8 che afferma l'esistenza un diritto fondamentale visto come protezione della vita privata, famiglia, casa e tutto cio che ne riguarda.

La corte europea ha data un'interpretazione vasta, soprattutto per i censimenti pubblici, sicurezza nazionale, esempio impronte digitali e fotografia, o raccolta di dati medici.

la privacy possiamo individuarla in due fasce di tempo:

- la prima che va da 1995-2018 (Data Protection Directive (95/46/EC))
- la seconda dal 2018 ad oggi (General Data Protection Regulation (GDPR) (EU) 2016/679).

Data Protection Directive (95/46/EC) tentativo di armonizzare le leggi per la protezione del dato per i paesi dell'unione europea, tramuta in legge da ogni paese.

Si basava su 8 principi

- L'utilizzo legale ai principi di legge legali;
- Il processamento di dati con scopi limitati;
- L'utilizzo di dati in quantità adeguata pertinenti per le attività raccolte;
- Accuratezza del dato;
- devono essere memorizzare per il tempo necessario all'utilizzo;

- devono essere trattati in accordo con i diritti dei data subject;
- i dati devono essere processati in modo sicuro;
- possono essere traferiti in altri paesi solo se vengono resi sicuri;

DEFINIZIONE (Dati personali)

Nella direttiva del 95 i dati venivano definiti come una persona identificata, che è detto data subject, in maniera diretta o indiretta in riferimento ad un ID o ad uno o più aspetti specifici della sua identità fisica, sociale, economica e mentale. Quindi tutto ciò che riguarda questo è identificabile come dato personale.

DEFINIZIONE (Trattamento del dato)

Ogni operazione o un insieme di operazioni attuata su dati personali in maniera automatica o non, come per esempio, la raccolta dei dati, la memorizzazione, l'organizzazione, l'archiviazione, la ricerca, l'utilizzo e la diffusione attraverso la comunicazione, l'allineamento, il blocco e la distruzione.

DEFINIZIONE (Resposabilità)

La responsabilità per l'adeguamento sulla normativa della privacy viene gestita dai "controller", che sia di natura artificiale, pubblica autorità, agenzie o ogni altro corpo che insieme determina gli obiettivi, gli scopi e i mezzi per il trattamento dei dati personali

Il responsabile del trattamento è una qualunque persona che tratta i dati per un titolare del trattamento.

Il principio alla base è che i dati personali non devono essere processati tranne queste 3 categorie di condizioni si verificano:

- Trasparenza: il data subject ha il diritto di essere informato nel momento in cui essi vengono utilizzati, il titolare ha anche l'obbligo di fornire tutte le informazioni richieste per assicurare che il trattamento sia fornito in maniera corretta.
- Scopo legittimo: i dati personali possono essere trattati solo per scopi legittimi e non possono essere trattati per scopi che vanno oltre questo scopo.
- Proporzionalità: se c'è un obiettivo vuol dire che deve essere perseguito utilizzando solo i dati necessari, quindi devono essere pertinenti e usare solo quelli che sono utili al fine dell'obiettivo e non devono essere eccessivi.

TRASPRENZA

I dati possono essere processati se almeno una delle seguenti condizioni si verifica:

- l'interessato ha dato il suo consenso.
- l'elaborazione era necessaria per l'esecuzione di un contratto.
- il trattamento era necessario per l'adempimento di un obbligo legale.
- il trattamento era necessario al fine di proteggere gli interessi vitali della persona interessata.
- il trattamento era necessario per l'attività svolta nell'interesse pubblico.
- il trattamento era necessario ai fini degli interessi legittimi.
- il trattamento era necessario ai fini degli interessi legittimi perseguiti dal responsabile del trattamento, tranne nel caso in cui tali interessi fossero sovrascritti dagli interessi per i diritti e le libertà fondamentali dell'interessato.

Riguardo alla trasparenza il data subject ha il diritto di accedere a tutti i dati trattati e che lo riguardano e può richiedere la modifica dei dati la cancellazione e il blocco nel caso in cui siano incompleti e che non vengono processati secondo le regole del dato.

PROPORZIONALITÀ

- I dati devono essere accurati e, ove necessario, aggiornati.
- I dati non devono essere conservati in una forma che consenta l'identificazione delle persone più a lungo del necessario.
- Gli Stati membri devono prevedere garanzie adeguate per i dati personali conservati per periodi più lunghi per uso storico, statistico o scientifico.
- Ulteriori restrizioni applicate ai dati personali sensibili (credenze religiose, opinioni politiche, salute, orientamento sessuale, razza, appartenenza a organizzazioni passate).
- L'interessato può opporsi in qualsiasi momento al trattamento dei dati personali a fini di marketing diretto.
- Una decisione che produce effetti legali o influisce in modo significativo sull'interessato potrebbe non basarsi esclusivamente sul trattamento automatizzato dei dati.
- Una forma di ricorso deve essere fornita quando vengono utilizzati i processi decisionali automatici.

Ogni stato membro deve dotarsi di un’autorità della privacy:

- Deve monitorare il livello di protezione dei dati in quello stato membro.
- fornire consulenza al governo in merito a misure e regolamenti amministrativi.
- Avviare un procedimento legale in caso di violazione della normativa sulla protezione dei dati.

Il responsabile del trattamento deve comunicare all'autorità di controllo le seguenti informazioni:

- il nome e l'indirizzo del responsabile del trattamento.
- le finalità del trattamento.
- una descrizione delle categorie dell'interessato e dei dati o delle categorie di dati ad essi correlati.
- i destinatari a cui i dati potrebbero essere comunicati.
- proposta di trasferimenti di dati verso paesi terzi.
- una descrizione generale delle misure adottate per garantire la sicurezza del trattamento.

Queste informazioni sono conservate in un registro pubblico.

PRIVACY IN ITALIA

In Italia, la Direttiva 95/46 / CE sulla protezione dei dati (Direttiva sulla protezione dei dati) è stata recepita dal decreto legislativo n. 196/2003, che conteneva il Codice italiano sulla protezione dei dati personali (Codice in materia di protezione dei dati personali) fino al 2018 che prende tutte le direttive dello scorso decreto.

Il regolamento generale sulla protezione dei dati (GDPR) (regolamento (UE) 2016/679) è un REGOLAMENTO (non solo una direttiva) con cui i membri dell'UE intendono rafforzare e unificare la protezione dei dati per tutti gli individui all'interno dell'Unione europea (UE)

▪ Gli obiettivi primari del GDPR sono di restituire a cittadini e residenti il controllo dei propri dati personali e di semplificare il contesto normativo per le imprese internazionali unificando il regolamento all'interno dell'UE

▪ Il regolamento è stato adottato il 27 aprile 2016

▪ Si applica dal 25 maggio 2018 dopo un periodo di transizione di due anni e, a differenza di una direttiva, non richiede alcuna legislazione che consenta di approvare i governi nazionali.

Con il GDPR cambiano le definizioni, anche se di poco.

DEFINIZIONE (DATI PERSONALI)

qualsiasi informazione relativa a una persona fisica identificata o identificabile ("soggetto dei dati"); una persona fisica identificabile è una persona che può essere identificata, direttamente o indirettamente, in particolare facendo riferimento a un **identificatore** come un nome, un numero di identificazione, dati sulla posizione, un identificatore online o uno o più fattori specifici della posizione geografica, della identità genetica, mentale, economica, culturale o sociale di quella persona fisica.

DEFINIZIONE (ELABORAZIONE)

Elaborazione: qualsiasi operazione o insieme di operazioni eseguite su dati personali o su insiemi di dati personali, anche con mezzi automatizzati, quali raccolta, registrazione, organizzazione, strutturazione, conservazione, adattamento o alterazione, recupero, consultazione, uso, divulgazione mediante trasmissione, diffusione o altrimenti messa a disposizione, allineamento o combinazione, limitazione, cancellazione o distruzione.

Vengono introdotte due nuove definizioni:

DEFINIZIONE (PSEUDONIMIZZAZIONE)

Pseudonimizzazione: il trattamento dei dati personali in modo tale che i dati personali non possano più essere attribuiti a una specifica persona interessata senza l'uso di informazioni aggiuntive, a condizione che tali informazioni aggiuntive siano conservate separatamente e siano soggette a misure tecniche e organizzative per garantire che i dati personali non sono attribuiti a una persona fisica identificata o identificabile.

DEFINIZIONE (VIOLAZIONE DEI DATI PERSONALI)

Violazione dei dati personali: una violazione (accidentale o illegittima) della sicurezza che porta alla distruzione, perdita, alterazione, divulgazione o accesso non autorizzati ai dati personali trasmessi, archiviati o altrimenti trattati.

Il GDPR elenca i diritti dell'interessato, ovvero la persona i cui dati personali vengono elaborati

Questi diritti rafforzati offrono alle persone un maggiore controllo sui propri dati personali, anche attraverso:

▪ la necessità del consenso chiaro dell'individuo al trattamento dei dati personali, in modo che il data subject possa dare un consenso diretto.

▪ un accesso più facile da parte del soggetto ai suoi dati personali.

▪ i diritti di rettifica, cancellazione e "da dimenticare".

▪ il diritto di opposizione, compreso l'uso dei dati personali a fini di "profilazione".

▪ il diritto alla portabilità dei dati da un fornitore di servizi a un altro.

PRINCIPIO DEL PRIVACY BY DESIGN AND BY DEFAULT

Il GDPR incoraggia l'utilizzo del principio del Privacy by design e by default.

PRIVACY DI DEFAULT

Il setting della privacy deve essere impostato al massimo livello per default.

Il responsabile del trattamento dovrebbe prendere misure tecniche e procedurali al fine di garantire che l'elaborazione, durante l'intero ciclo di vita dell'elaborazione, sia conforme al regolamento, inoltre i responsabili del trattamento dovrebbero implementare meccanismi per garantire che i dati personali siano trattati solo quando necessario per ogni scopo specifico.

PRIVACY BY DESIGN

La privacy per default è una attuazione più drastica di quella by design (ad esempio l'intero processo di business deve tenendo a mente i principi di protezione del dato).

Privacy by Design [Ann Cavoukian, 2012, ricercatrice canadese] è un approccio all'ingegneria dei sistemi che tiene conto della privacy durante l'intero processo di ingegneria. E' basato su 7 principi fondamentali:

1- Proattivo non reattivo, preventivo non correttivo, prevedere delle misure di protezioni del dato che prevengano degli abusi sul dato.
2- privacy come impostazione predefinita,
3- privacy incorporata nel design, dell'intero servizio.
4- funzionalità completa - somma positiva, non somma zero, ovvero, bisogna fornire un servizio che serva a qualcosa per l'utente sempre in rispetto della privacy.
5- sicurezza end-to-end - protezione completa del ciclo di vita,
6- visibilità e trasparenza - mantenerla aperta,
7- rispetto della privacy dell'utente - mantenerlo incentrato sull'utente.

PSEUDOANONYMIZATION

Il GDPR si riferisce alla pseudonimizzazione come un processo che trasforma i dati personali in modo tale che i dati risultanti non possano essere attribuiti a un soggetto specifico senza l'uso di informazioni aggiuntive, Ad esempio, crittografando i dati localmente, mantenendo le chiavi di decrittazione separatamente dai dati crittografati

Se i dati personali sono pseudonimizzati con adeguate politiche e misure interne, sono considerati effettivamente anonimizzati e non soggetti a controlli e sanzioni del GDPR. Quindi se si garantisce la suddivisione e l'anonimato tra il dato e il riferimento all'individuo di esso, allora non verrà sottoposto a sanzioni tramite GDPR.

Il regolamento non riguarda il trattamento di informazioni ritenute anonime, anche a fini statistici o di ricerca.

Le politiche e le misure che soddisfano i principi della protezione dei dati in base alla progettazione e alla protezione dei dati di default dovrebbero essere considerate adeguate a questo scopo.

RESPONSABILITÀ E RESPONSABILIZZAZIONE

il responsabile del trattamento dei dati deve essere capace di dimostrare in qualsiasi momento che il trattamento viene fatto in compliance con il GDPR e che siano stati implementati i principi di privacy by design e privacy by default

È responsabilità del responsabile del trattamento implementare misure efficaci ed essere in grado di dimostrare la conformità delle attività di trattamento

▪ Gli utenti devono essere chiaramente informati dell'entità della raccolta dei dati, della base giuridica per il trattamento dei dati personali, per quanto tempo i dati vengono conservati, se i dati vengono trasferiti a terzi e / o al di fuori dell'UE e la divulgazione di qualsiasi il processo decisionale che viene preso su una base esclusivamente algoritmica

▪ Gli utenti devono ricevere i dettagli di contatto del responsabile del trattamento e del responsabile della protezione dei dati designato, ove applicabile

▪ Gli utenti devono anche essere informati dei loro diritti alla privacy ai sensi del GDPR

▪ Le valutazioni d'impatto sulla protezione dei dati (DPIA) devono essere condotte quando si verificano rischi specifici per i diritti e le libertà degli interessati

▪ È richiesta la valutazione e la mitigazione del rischio è richiesta l'approvazione preventiva delle autorità nazionali per la protezione dei dati (DPA) per i rischi elevati

▪ Devono essere conservati registri delle attività di elaborazione che includano finalità del trattamento, categorie coinvolte e termini previsti

▪ Le registrazioni devono essere rese disponibili all'autorità di controllo su richiesta

RESPONSABILE DELLA PROTEZIONE DEI DATI (DPO)

▪ Il GDPR stabilisce la figura del Responsabile della protezione dei dati (DPO), una persona con conoscenza esperta della legge e delle pratiche di protezione dei dati che dovrebbe aiutare il responsabile del trattamento dei dati a monitorare la conformità interna al presente regolamento

▪ Ci si aspetta inoltre che il responsabile della protezione dei dati sia competente nella gestione dei processi IT, della sicurezza dei dati (compresa la gestione degli attacchi informatici) e di altri problemi critici di continuità aziendale relativi alla conservazione e al trattamento di dati personali e sensibili

▪ Le autorità pubbliche e le imprese le cui attività principali sono incentrate sul trattamento regolare o sistematico di dati personali, devono assumere un DPO(l'università lo fa in quanto tratta i dati personali di noi studenti ma anche dei dipendenti).

VIOLAZIONE DEI DATI

Il titolare del trattamento ha l'obbligo legale di informare l'autorità di controllo senza indebito ritardo, a meno che sia improbabile che la violazione comporti un rischio per i diritti e le libertà delle persone. Vi sono al massimo 72 ore dopo essere venuti a conoscenza della violazione dei dati per effettuare la segnalazione

Gli individui devono essere informati se si determina un impatto negativo

Il responsabile del trattamento dei dati dovrà informare il responsabile dei dati senza indebito ritardo dopo essere venuto a conoscenza di una violazione dei dati personali

Tuttavia, la comunicazione agli interessati non è richiesta se il responsabile del trattamento ha implementato adeguate misure di protezione tecniche e organizzative che rendono incomprensibili i dati personali a chiunque non sia autorizzato ad accedervi, come la crittografia

SANZIONI a spaventato le aziende

Possono essere imposte le seguenti sanzioni:

▪ un avvertimento per iscritto in caso di prima non conformità intenzionale

▪ audit periodici sulla protezione dei dati

▪ un'ammontare fino a 10 milioni di euro o fino al 2% del fatturato mondiale annuo dell'esercizio precedente nel caso di un'impresa, in caso di violazione di alcuni obblighi

▪ un'ammontare fino a 20 milioni di euro o fino al 4% del fatturato mondiale annuo dell'esercizio precedente nel caso di un'impresa, se si sono verificate gravi violazioni dei principi.

Anche il trasferimento dei dati da un paese ad un altro sono stati definiti da delle regole.

SAFE HARBOR PRIVACY PRINCIPLES

Fu stato sviluppato tra il 1998 e il 2000 al fine di impedire alle organizzazioni private all'interno dell'UE o degli Stati Uniti di divulgare o perdere accidentalmente informazioni personali.

La commissione europea decise che i principi definiti dagli stati uniti erano consoni con quelli dell'EU e quindi vennero adottati i medesimi. Ma dopo che un cliente si è lamentato del fatto che i suoi dati di Facebook non erano sufficientemente protetti, la Corte di giustizia della Commissione europea ha dichiarato nell'ottobre 2015 che la decisione Safe Harbor non era valida. La Commissione ha tenuto ulteriori colloqui con le autorità statunitensi su "un quadro rinnovato e solido per i flussi di dati transatlantici"

Il 2 febbraio 2016 la Commissione europea e gli Stati Uniti hanno concordato di stabilire un nuovo quadro per i flussi di dati transatlantici, noto come "**Scudo UE-USA per la privacy**"

Lezione 2

Privacy nell'era dei Big Data.

Percentuali molto alte di persone che hanno sperimentato un'invasione della privacy già dalla fine degli anni '70. Col passare del tempo tali percentuali sono salite anche grazie alla maggiore diffusione delle tecnologie informatiche.

Si può parlare di un compromesso tra convenienza e privacy? Cioè mantenere un certo livello di privacy per avere determinati servizi come geolocalizzazione, raccomandazioni su amazon ecc, mantenendo alta la compliance con la privacy?

Quali sono i processi informatici che lo permettono?

Ad esempio, il principio di **Privacy by default, Privacy by design**: un servizio non deve avere somma 0, cioè non si deve avere un servizio inefficiente per preservare la privacy.

Cosa intendiamo per Privacy?

Non intendiamo solo il concetto di nascondere delle cose. Essa riguarda il possedere le info che ci riguardano, in questo modo siamo in grado di controllare quanto di noi stessi può essere svelato agli altri (**Self-Possession**); questo è il contrario di quello che succede nei social.

Tutto questo riguarda anche l'autonomia (**Autonomy**) di pensiero, di movimento e anche l'integrità delle info che ci riguardano: se le nostre info raggiungono le altre persone in maniera poco integra si può fornire a queste ultime una visione non corretta su noi stessi che può influenzare le decisioni che potranno essere prese nei nostri confronti in futuro (es. concessione di un prestito).

Privacy collegata ai Diritti Umani: la privacy deve essere garantita sempre in quanto può accadere che, in certi contesti, se viene a mancare è possibile che qualcuno che abbia intrapreso determinate azioni rischia di essere discriminato.

Diritto alla Privacy

"The right to be left alone"

Ormai la privacy viene riconosciuta come un diritto molto importante anche nelle leggi, costituzioni e trattati internazionali.

Problema: ricorda Cambridge Analytica e cosa ha comportato.

Uno dei fattori principale al giorno d'oggi è la grande quantità (abbondanza) di dati personali disponibili come registri di carte di credito, telefonate, mail, account social…

Legge delle conseguenze non intenzionali

Una tecnologia viene descritta per una determinata azioni, ma poi viene utilizzata anche per altri scopi: ad esempio il telefono inizialmente veniva utilizzato solo per telefonare ed inviare messaggi, mentre adesso lo si può utilizzare anche per effettuare pagamenti. Il codice SSN negli stati uniti cambiò il suo scopo infatti ora è un codice per indentificare gli studenti.

Big Data

Con questo termine ci riferiamo all'acquisizione e analisi di grandi collezioni di info, talmente grandi che talvolta non si riescono neanche ad analizzare: questo può essere un vantaggio, ma in contrapposizione abbiamo che non si sa neanche come proteggerli.

Alcune problematiche riguardo ad essi sono:

- **Datafication**: Tendenza tecnologica che trasforma molti aspetti della nostra vita in dati che vengono successivamente trasferiti in informazioni realizzate come una nuova forma di valore. Vengono raccolti dati senza sapere precisamente cosa farne, ma con l'idea di utilizzarli in futuro;
- **Dati enormi non strutturati e difficili da analizzare**: avendo dati sporchi non sappiamo a che info si riferisce ogni dato. Talvolta non c'è nemmeno un modo per interpretarli;
- **Pattern:** attraverso queste grandi collezioni di dati si possono analizzare dei pattern che altrimenti passerebbero inosservati. Ad esempio, la ricerca di determinate parole su un motore di ricerca riguardo determinati sintomi comportava che la persona avesse un certo problema di salute;
- **Memorizzazione indiscriminata:** raccolti anche dati che non servono per uno scopo dichiarato. In parte si possono usare anche per Intelligence, per effettuare analisi più o meno lecite;
- **Memorizzazione per tempi indefiniti:** in questo modo si può favorire l'estrazione di pattern interessanti riguardanti diversi aspetti delle persone in questione.

Perdita dell'informational privacy

Con questa espressione indichiamo l'abilità di determinare quali info riguardano noi stessi, quelle che gli altri possono memorizzare e cosa possono farci con esse.

Nessuno degli scenari mostrati in precedenza può avvenire senza la perdita di controllo sui nostri dati.

Si può pensare all'utilizzo di tali dati in forma anonima, cioè nei dati raccolti si sostituisce con un identificativo lo username della persona.

Problema: **deanonimizzazione.**

Quando Netflix ha pubblicato un dataset anonimo con i voti rilasciati dai suoi utenti è stato possibile risalire agli utenti stessi collegando le recensioni pubbliche rilasciate dagli stessi su ImDB, collegando quindi alle valutazioni espresse uno username. Questo è stato un problema in quanto se un utente aveva votato un film per adulti su Netflix, conscio del fatto di essere protetto dall'anonimato, attraverso il processo di deanonimizzazione quella valutazione veniva associata all'utente stesso con tutte le conseguenze del caso.

Quindi l'anonimizzazione non può solo consistere nella sostituzione del nome perché andando ad incrociare i dati da fonti diverse si possono trovare molti più dati di quanti se ne immagini, anche perché le abitudini/gusti sono molto facili da predire attraverso tecniche di AI.

Lezione 3

Sistemi Informativi/Informatici

I sistemi informativi sono sistemi formali, socio-tecnici e organizzativi progettati per raccogliere, elaborare, archiviare e distribuire (anche comunicazione verso l'esterno o trai i vari reparti dell'organizzazione) informazioni utilizzate da un'organizzazione.

I sistemi informatici sono la parte automatizzata del sistema informativo.

Dei componenti software e hardware lavorano insieme per acquisire, memorizzare ed elaborare info con lo scopo di supportare:

- Le attività;
- Le decisioni prese dai vertici dell'organizzazione: in questo caso le attività di supporto sono analisi, controllo, coordinazione, visualizzazione e reporting.

I sistemi informativi lavorano grazie allo scambio di dati/informazioni: questa operazione di scambio può essere modellata come un set di processi interconnessi dove l'output di uno è l'input di un altro.

I dati prodotti appartengono a diverse entità come impiegati, clienti, utenti, fornitori… Essi sono di diverso tipo:

- Dati personali identificativi;
- Contenuti generati dagli utenti;
- Mail scambiate a diverso livello;
- Log dei processi automatizzati e non;
- Dati di transazioni.

Questi dati scambiati possono essere pubblici e privati (in alcuni casi anche sensibili).

Come rafforzare la privacy in questo contesto?

La privacy può essere applicata a diversi livelli:

- Educare dipendenti e manager in quanto principali attori del sistema;
- Rafforzare le misure di sicurezza informatica: impiegare cifratura e altre tecniche protettive;
- Proteggere le reti e le infrastrutture dei server: uso di firewall;
- Limitare l'accesso fisico alle aree sensibili;
- Usare VPN per il lavoro remoto;
- Tenere aggiornato il software: si evita di lasciare vulnerabilità presenti in versioni precedenti.

Per applicare questi concetti possiamo definire 4 procedure utili:

Sistemi informativi e il GDPR

1. ***Autorizzazione basata sugli attributi***: un meccanismo di autorizzazione dichiarativo e completamente tracciabile per l'accesso a tutti i dati, in circostanze predefinite;

2. ***Anonimizzazione/pseudonimizzazione*** dei dati: meccanismi sicuri per la pseudonimizzazione o anonimizzazione delle informazioni degli archivi;
3. ***Tracciabilità***: tenere un registro di chi, ha creato, modificato o cancellato informazioni, quando e per quale scopo;
4. ***Cancellazione dei dati***: meccanismi di sicurezza per garantire il "diritto all'oblio" senza compromettere l'affidabilità del sistema.

Autorizzazione basata sugli attributi

Controllo degli accessi

Limitare l'accesso per le risorse del computer, in particolare nelle impostazioni di condivisione dei dati multiutente.

I requisiti di privacy e sicurezza da mantenere nei sistemi di informazione insieme alla complessità associata, aumentano la difficoltà nel realizzare schemi di controllo degli accessi efficienti.

Sistemi molto complessi richiederanno sicuramente un controllo degli accessi molto elaborati.

Ad esempio, un medico può accedere al fascicolo elettronico di un suo paziente se questo gli ha fornito l'accesso. Tuttavia, in caso di emergenza è importante dare l'accesso al fascicolo del paziente anche ad altri dottori.

RBAC (Role Based Access Control)

Messo a punto dal NIST (US National Institute of Standards and Technology) nel '92. Utilizzato da aziende di piccole dimensioni (max 500 impiegati).

L'idea alla base è quella di usare i ruoli degli utenti all'interno dei processi del sistema piuttosto che l'ID. Viene quindi effettuata una mappatura tra utente e ruolo e poi una mappatura tra ruoli e risorse.

L'accesso può essere fornito in lettura o scrittura.

In sostanza gli amministratori assegnano il permesso di accesso in base al ruolo, i quali possono essere assegnati anche ai singoli utenti. N.B.: un utente può avere più ruoli.

Funzionamento di RBAC

Innanzitutto, gli amministratori assegnano i ruoli ai singoli utenti, ogni utente puo avere piu ruoli con diversi sistemi di accesso, possono aggiornare i ruoli o accedere alle autorizzazioni assegnando gli utenti (o rimuovendo gli utenti da) ai ruoli appropriati.

Limiti

- RBAC fornisce configurazioni di controllo degli accessi a grana grossa, predefinite e statiche;
- Non riesce a fornire un meccanismo flessibile attraverso il quale clienti utenti e le istituzioni possano esprimere le loro esigenze: non potrebbe rimanere al passo con un'evoluzione del sistema molto veloce;
- Inoltre, non riesce a catturare lo scopo per il quale i dati verranno divulgati ai vari stakeholder.

Oggi, la definizione di politiche di autorizzazione basate esclusivamente sul ruolo di un utente non è abbastanza adeguata. Anche il contesto che circonda tale utente, i suoi dati e l'interazione tra i due sono importanti per fornire accesso all'utente corretto, al momento giusto, nella corretta posizione geografica e soprattutto soddisfacendo le regole della normativa.

- Definendo il contesto tra utenti e attributi, è possibile definire solide politiche specificando che sebbene un utente possa creare ordini di acquisto e approvarli, è in grado di approvare solo ordini di acquisto a cui quell'utente non è assegnato;
- Conoscere il ruolo di un utente non è più sufficiente per garantire un controllo dell'accesso sicuro e protetto;
- Sono richiesti il contesto e le informazioni sulla relazione tra le varie entità coinvolte nel sistema;
- Il controllo degli accessi in base agli attributi consente a un'azienda di estendere i ruoli esistenti utilizzando attributi e criteri.

Per questo si è passati all' **Attribute Based Access Control (ABAC).**

Aggiungendo il contesto, le decisioni di autorizzazione possono essere prese in base:

- al ruolo dell'utente;
- per chi o cosa l'utente si è collegato;
- A cosa l'utente ha bisogno di accedere;
- Da dove l'utente ha bisogno di accedere;
- Quando quell'utente ha bisogno di accedere;
- Come l'utente accede a tali informazioni.

Quindi diventano molto importanti il contesto e le informazioni di relazione tra le varie entità che fanno parte del sistema.

ABAC consente ad un'azienda di estendere i ruoli esistenti utilizzando attributi e politiche aziendali. Inoltre, utilizza politiche basate sugli attributi individuali usando il linguaggio naturale. Ad esempio, viene descritta una regola specificando che una certa figura può fare determinate azioni su una certa risorsa.

Creando una policy che è facile da capire, con il contesto attorno a un utente e ciò a cui dovrebbe avere accesso, il controllo degli accessi diventa molto più solido.

Esso è simile a RBAC nel senso che adotta anche un approccio basato su criteri. Utilizza attributi di soggetti, oggetti e ambiente (anziché ruoli). Gli attributi vengono utilizzati per esprimere policy molto ricche/complesse. Toglie la necessità di essere registrati nel sistema per poter accedere risorse condivise (ad esempio se un dipendente di un'azienda

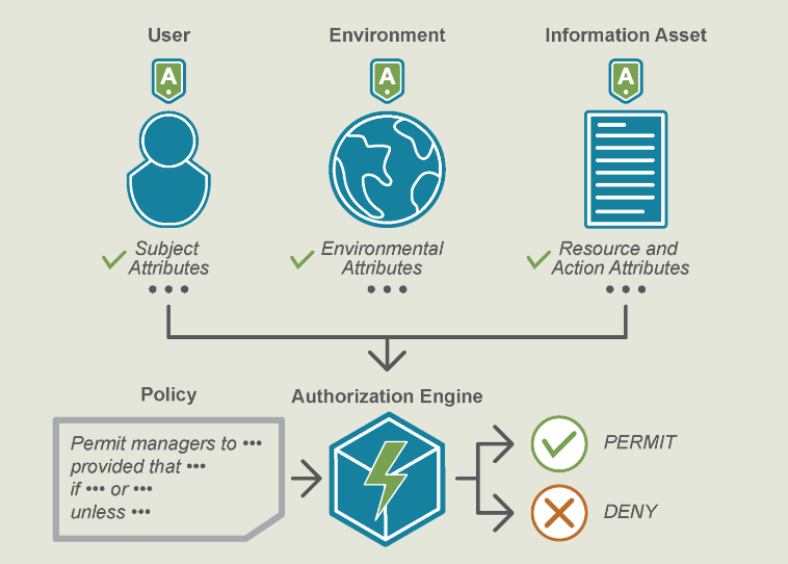
effettua un ordine di un prodotto per l'azienda e in quel caso lui risulta il responsabile per l'approvazione degli ordini, si approverà l'ordine da solo, e chiaramente questo non funziona).

Come funziona ABAC

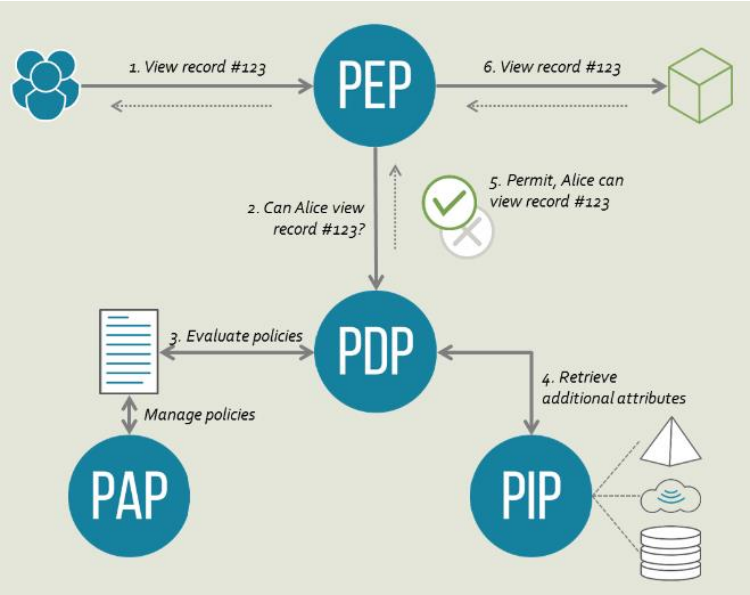
Le 3 componenti principali sono descritte da alcuni attributi per identificarle.

1- Gli utenti che sono i subject attributes, cioè colui che fa l'acesso ad un determinato attributo. Gli attributi definibili sul soggetto sono il ruolo/i, gruppi di appartenenza, dipartimento/azienda, livello manageriale, certificazioni/competenze in possesso, user ID…
2- L'azione è il compito che deve essere fatto sulla risorsa a cui si vuole accedere: può essere Read o Write; l'azione può anche essere descritta da un insieme di attributi. Le risorse invece sono gli oggetti coinvolti dalle azioni. Gli attributi, sistema di gestione dei documenti, di solito corrispondono a metadati o tag relativi ai documenti
3- L'ambiente identifica il contesto in cui l'accesso è stato richiesto. I suoi attributi sono l'ora e la posizione geografica, il canale di comunicazione, la robustezza dei protocolli di crittografia.

Queste caratteristiche vengono processate dall'**Authorization Engine**, confronta le richieste che sono state effettuate, accede alle policy e verifica che vengano soddisfatte: in base alle policy e alla richiesta decide se consentire o negare l'accesso alle risorse.



Authorization Engine



È composto da 4 diversi componenti:

1. **PEP** (Policy Enforcement Point): punto in cui arriva la richiesta, viene inoltrata al PDP;
2. **PDP** (Policy Decision Point): essa prende la decisione guardando la richiesta e cercando le policy applicabili. Successivamente valuta se queste sono rispettate e restituisce la decisione al PEP che a sua volta verra comunicato all'utente. Si avvale degli attributi nel PIP e PAP;
3. **PIP** (Policy Information Point): componente che memorizza tutti gli attributi che riguardano risorse e soggetti;
4. **PAP** (Policy Administration Point): al suo interno sono memorizzate tutte le policy.

Dopo aver effettuato le opportune verifiche viene permesso/negato l'accesso.

Gli scambi di info vengono effettuati con XACML stantard di ABAC (dialetto di XML, si usa anche JSON) per esprimere le richieste di accesso ai dati.

Se ho due policy, il sistema ABAC le accetta tutte tranne in alcuni casi se vanno in conflitto tra di loro.

Esempio di richiesta di accesso

Policy 1) ovvero quella dell'istituto medico. La policy innanzitutto definisce il target di utenti che possono accedere, la risorsa a cui possono accedere e le azioni che possono fare. Successivamente vengono definite delle regole per permettere di accedere o meno a quel dato.

Policy 2) ovvero quello dei pazienti. La policy opera nello stesso modo di quella precedente, rispettando le richieste di accesso ai dati fornite dai clienti e salvandole nel loro database.

In questo caso si parla di un ricercatore che sta conducendo uno studio per valutare l'efficacia del farmaco sperimentale sui sintomi dell'epatite C ogni mese nel primo anno

dopo il trattamento. Il ricercatore ha bisogno di accedere all'uso passato di droghe del paziente e anche informazioni sui sintomi a livello mensile.

```
<Request REQ1>
<Attributes>
      :access-subject   :Subject-id     :Carol
      :access-subject   :Role           :Researcher
      :access-subject   :GDPR Comp     :Yes
      :resource         :Type           :HealthData
/* Content */
      :resource      :Content        :Drug-usage
      :resource      :Content        :Symptoms
      :resource      :Content        :Disease
      :resource      :Content        :Duration
      :resource      :Content        :Date
/* Comparing Attribute Values */
      :content-selector :resource.Disease  :HepC
      :content-selector :resource.Duration :Monthly
      :content-selector :resource.Date :
                  resource.Date+365>Current Date
      :action       :Action-id           :Release
      :purpose      :Purpose-id          :DrugEffect
</Attributes>
</Request REQ1>
```

L'unica regola da soddisfare è la GDPR Comp: se il campo in questione è Yes allora la richiesta è conforme alle norme GDPR e quindi il ricercatore può accedere alle info da lui richieste.

N.B.: potrebbe essere che si effettui una richiesta a più fonti di info. Se le due richieste hanno esiti diversi ed è presente la clausola **'permit-overrides'** allora nel caso in cui ci siano contraddizioni la richiesta accettata (permit) prevarrà su quella rifiutata (deny).

Lezione 4

Pseudononymization

La pseudonimizzazione è una procedura di gestione e de-identificazione dei dati mediante la quale i campi di informazioni di identificazione personale all'interno di un set di dati vengono sostituiti da uno o più identificatori artificiali o pseudonimi. Un singolo pseudonimo sostituito per ogni campo o una raccolta di campi sostituiti rende il record di dati meno identificabile rimanendo idoneo per l'analisi/elaborazione dei dati.

I dati pseudonimizzati possono essere ripristinati al loro stato originale attraverso una procedura di restore, con l'aggiunta di informazioni che consentono quindi alle persone di essere nuovamente identificati.

La differenza tra pseudonimizzazione è che il dato di per se è già anonimo, e per ritornare al data subject originale vengono usate procedure di re-introduzione dei dati. Mentre i dati anonimi non possono mai essere ripristinati al loro stato originale.

Se la pseudonimizzazione viene fatta bene e la chiave viene protetta, allora anche questi dati possono essere considerati anonimi.

Esistono 2 approcci:

- **Column encryption:** è una funzionalità progettata per proteggere i database in cui sono archiviati dati sensibili, consente ai client di crittografare i dati sensibili all'interno delle applicazioni client e di non rivelare mai le chiavi di crittografia al motore di database. Di conseguenza, la crittografia fornisce una separazione tra coloro che possiedono i dati (e possono visualizzarli) e coloro che gestiscono i dati (ma non dovrebbero avere accesso). Se il database non puo decifrare a sua volta i

dati vuol dire che quei dati sono veramente al sicuro, poiché in seguito ad un accatto al database non troveranno la chiave utili per la cifratura dei dati. Quando si configura la crittografia per una colonna, è possibile specificare le informazioni sull'algoritmo di crittografia e le chiavi crittografiche utilizzate per proteggere i dati nella colonna. Abbiamo 2 tipi di chiavi

1. Column encryption keys: utilizzate per crittografare i dati in una colonna crittografata; (una chiave per colonna)
2. Column masters keys: una chiave master permette di cifrare una o piu colonne.

Il DBMS archivia la configurazione della crittografia per ogni colonna nei metadati del database (le chiavi non vengono mai archiviate o utilizzate in un testo normale degli scambi. Il DBMS memorizza solo i valori crittografati delle chiavi di crittografia della colonna e le informazioni sulla posizione delle chiavi master della colonna, che sono archiviate in archivi di chiavi attendibili esterni.

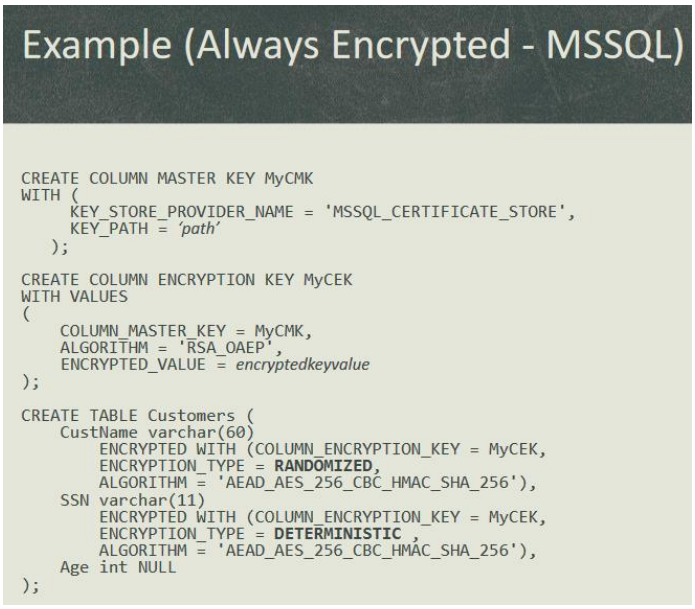
Il DBMS supporta alcune query sui dati crittografati, a seconda del tipo di crittografia per la colonna.

Abbiamo due tipi di cifrature encryption dei dati:

I tipi di **crittografia deterministici** generano lo stesso valore per qualsiasi valore presente nel database (ad esempio un CF verrà cifrato sempre nello stesso modo), consentendo quindi di recuperare i singoli valori cifrati, indicizzare le colonne e raggruppare i dati stessi. Potrebbe inoltre consentire a utenti non autorizzati, con l'uso della brute force, di trovare i valori crittografati esaminando i modelli nella colonna crittografata (essendo cifrati tutti nello stesso modo, quindi cifrandone uno è possibile riuscire a cifrare anche gli altri), specialmente se l'insieme dei valori è molto piccolo, come True / False o regione Nord / Sud / Est / Ovest.

Invece, la **crittografia randomizzata** oltre a cifrare i dati utilizzando una chiave, utilizza un metodo che crittografa i dati in un modo meno prevedibile. La crittografia casuale è più sicura, ma impedisce la ricerca, il raggruppamento, l'indicizzazione e l'unione(join) su colonne crittografate.

Esempio Db Microsoft



- **Dynamic Data Masking**: evita l'estrazione dei dati offuscando i dati. Funzione incorporata in MS SQL, ma può essere facilmente implementata in qualsiasi DBMS (ad es. Utilizzando stored procedure). Ad esempio, il numero della carta di credito può diventare "xxxx-xxxx-xxx-x815" lasciando intendere che sia una carta di credito ma nascondendone la maggior parte. I dati nel DB sono in chiaro, il masking viene messo in atto quando si effettua una query. Alcune funzioni di data masking sono:
  - **Default**: in base al tipo di dato effettua il mascheramento;
  - **Email**: lascia il primo carattere, poi inserisce tutte x;
  - **Random**: sostituisce i valori con altri caratteri indicati nella funzione;
  - **Custom**: si lasciano solo la prima e l'ultima lettera e un padding.

Come ci si comporta con i dati mascherati?

I proprietari del db possono vedere i dati non mascherati, tutti gli altri invece no. Tuttavia, esistono dei privilegi (MASK/UNMASK) che possono essere concessi a una persona o gruppi per poter vedere in caso dati non mascherati. Inoltre, quando si effettua una query, i dati non vengono mostrati ma rimangono mascherati: con l'uso di una Select i dati nella nuova tabella non saranno visibili.

Limiti del Data Masking

Bypassare il mascheramento è possibile usando tecniche di inferenza o forza bruta.

Il Dynamic Data Masking è progettato per semplificare lo sviluppo dell'applicazione limitando l'esposizione dei dati in una serie di query predefinite utilizzate dall'allineamento dell'applicazione.

Quando si accede direttamente a un database di produzione, utenti non privilegiati con query ad hoc e le giuste autorizzazioni possono applicare tecniche per ottenere l'accesso ai dati effettivi.

Se è necessario concedere tale accesso ad hoc, bisogna revisionare (Auditing) e monitorare tutte le attività del database per mitigare questo scenario.

Conclusioni

Il Dynamic Data Masking non dovrebbe essere usato come misura isolata per proteggere completamente i dati sensibili dagli utenti che eseguono query ad hoc sul database.

È appropriato per prevenire l'esposizione accidentale di dati sensibili, ma non proteggerà da intenti maliziosi di inferire i dati sottostanti.

È importante gestire correttamente le autorizzazioni sul database e seguire sempre il principio delle autorizzazioni minime richieste.

È importante abilitare **Auditing** per tenere traccia di tutte le attività che si svolgono sul database.

Tracciabilità attraverso l'auditing del database

Cos'è l'auditing?

Audit / auditing: processo di esame e validazione di documenti, dati, processi, procedure, sistemi.

Log di audit: documento che contiene tutte le attività che vengono verificate in ordine cronologico.

Obiettivi di audit: insieme di regole aziendali, controlli di sistema, regolamenti governativi, o politiche di sicurezza.

Altre definizioni a riguardo sono:

Auditor: una persona autorizzata all'audit**.**

Procedura di audit: serie di istruzioni per il processo di audit.

Rapporto di audit: un documento che contiene i risultati dell'audit.

Esistono 2 tipi di audits:

- **Interno**: esame delle attività condotte da membri del personale dell'organizzazione sottoposta a processo di auditing.
- **Esterno** esame delle attività dell'organizzazione condotte da una terza parte indipendente. Esso può dare luogo a delle sanzioni se l'ente esterno verifica che ci sono degli inadempimenti all'interno dell'azienda.

Attività di auditing

Individuare le problematiche di sicurezza che devono essere affrontate.

Definire piani, politiche e procedure per lo svolgimento degli audit.

Organizzare e condurre gli audit interni.

Assicurare che tutte le voci contrattuali siano soddisfatte dall'organizzazione che si sta verificando.

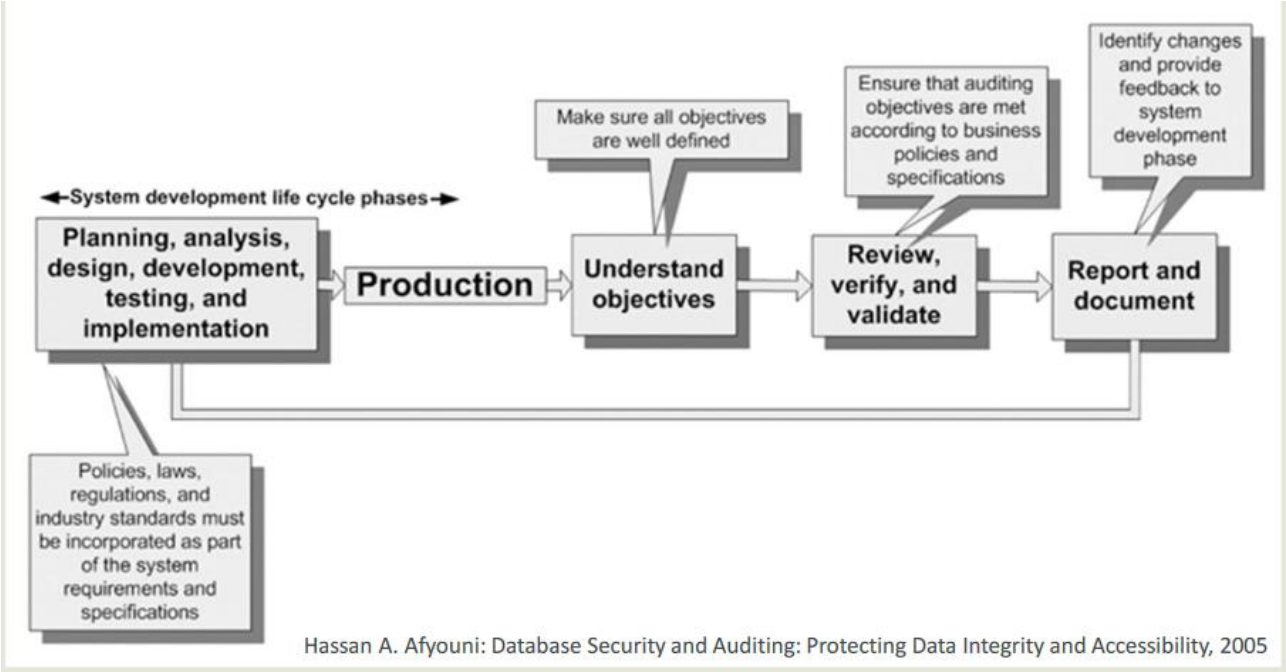
Agire come collegamento tra la società e il gruppo di controllo esterno.

Fornire consulenza al Dipartimento legale.

Processo di auditing: assicura che il sistema funzioni e sia conforme alle politiche, ai regolamenti e alle leggi. Il processo di controllo viene eseguito dopo la messa in servizio del prodotto.

Controllo di vita del processo di auditing:

- Ciclo di vita dello sviluppo del sistema
- Processo di audit:
    - Comprendere gli obiettivi
    - Riesaminare, verificare e convalidare il sistema
    - Documentare i risultati



Quello che interessa principalmente è il **data audit**. Esso è un record cronologico delle modifiche ai dati archiviate nel file di registro o nell'oggetto tabella database.

Database auditing: è un record cronologico delle attività del database.

Gli obiettivi di questo processo sono le seguenti verifiche:

- Integrità dei dati.
- Utenti e ruoli dell'applicazione.
- Riservatezza dei dati.
- Controllo degli accessi.
- Modifiche ai dati.
- Modifiche alla struttura dei dati.
- Disponibilità del database o dell'applicazione.
- Controllo dei cambi.

- Relazioni sull’auditing.

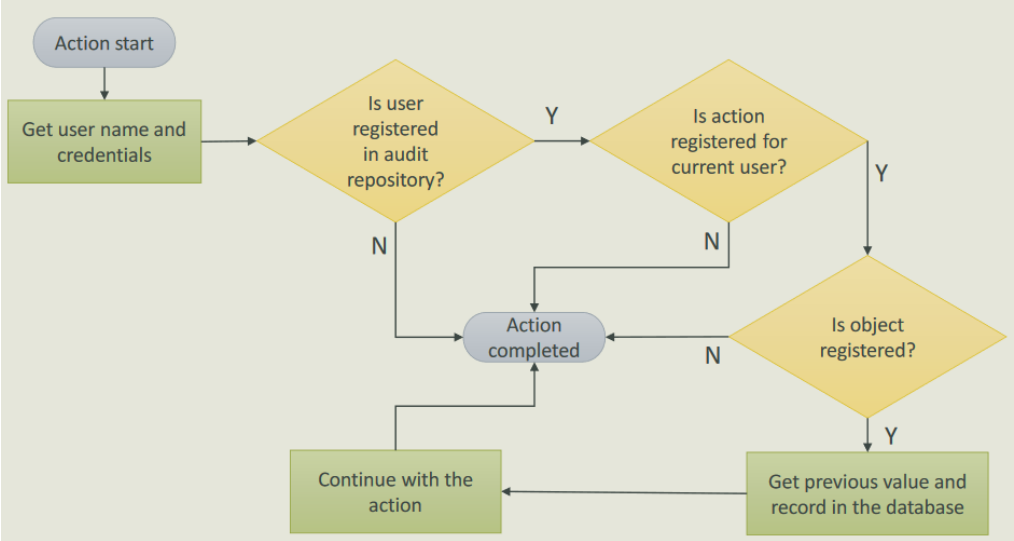
Modelli di auditing

Può essere implementato con funzionalità integrate o con il proprio meccanismo, utilizzando PLSQL o altri linguaggi.

Le informazioni registrate sono:

- Stato dell'oggetto prima che l'azione fosse intrapresa
- Descrizione dell'azione che è stata eseguita.
- Nome dell'utente che ha eseguito l'azione
- Ip address della macchina dell’utente.

Un esempio di modello può essere il seguente:



Conclusioni

Per rendere i sistemi informativi conformi al GDPR, è necessario prendere in considerazione più misure di sicurezza:

- Controllo degli accessi basato sull'attributo (ABAC)
- Crittografia (crittografia sempre crittografata / crittografia trasparente dei dati)
- Usare Dyamic data masking
- Implementare l’auditing

Ma è tutto questo necessario per proteggere la privacy dei dati memorizzati?

Sicuramente il livello di protezione è alto, ma comunque non ci garantisce una sicurezza totale.

Lezione 5

Raccolta e divulgazione dei dati

Al giorno d'oggi Internet offre opportunità per la raccolta e la condivisione di informazioni sensibili sulla privacy dei suoi utenti soprattutto attraverso i dispositivi di cui ognuno è munito (smartphone, smartwatch) e i social media. La preoccupazione quindi è quella di mantenere al sicuro questi dati.

La protezione della privacy richiede l'investigazione di molti aspetti diversi, incluso il problema della protezione delle informazioni rilasciate contro i trasferimenti e linking attacks. Queste collezioni di dati possono essere analizzate attraverso tecniche di:

- Machine learning
- Data mining
- Statistical inference
- Big Data analytics.

Spesso i dati statistici vengono rilasciati attraverso dataset o (linked) open data per incentivare la loro analisi per produrre informazione da essi.

La disclosure (divulgazione) può avvenire in diversi modi:

- Dati rilasciati;
- Risultato della combinazione tra dati rilasciati e informazioni pubbliche;
- Combinazione di dati rilasciati con altre fonti di dati esterne che potrebbero essere disponibili al pubblico (social media).

N.B.: quando si rilasciano dati, il rischio di divulgazione dei dati rilasciati dovrebbe essere molto ridotto!

Micro vs Macrodata

In passato i dati venivano rilasciati in forma tabellare (macro) attraverso DB statistici.

Col passare del tempo invece si è passati al rilascio di specifici set di dati (micro) i quali sono facilmente soggetti a rischi di attacchi esterni.

<u>Macrodata</u>

Le tabelle macrodata possono essere classificate nei seguenti gruppi:

- Tabelle di Conteggio / frequenza: ciascuna parte della tabella contiene il numero di intervistati (conteggio) o la percentuale di intervistati (frequenza) che devono essere analizzati per la maggior parte degli attributi di analisi associati alla tabella
- Dati di tipo Magnitude: sono i dati che contengono un valore aggregato di una quantità di interesse misurata.

Gli intervistati sono aziende, autorità, singole persone, ecc., da cui vengono raccolti dati e informazioni associate per la compilazione di statistiche.

Tabelle esempio

Count

	Benefit						
County	$0-19	$20-39	$40-59	$60-79	$80-99	$100+	Total
A	2	4	18	20	7	1	52
B	-	-	7	9	-	-	16
C	-	6	30	15	4	-	55
D	-	-	2	-	-	-	2

Magnitude

	Hypertension	Obesity	Chest pain	Short Breath	Tot
M	2	8.5	23.5	3	37
F	3	30.5	0	5	38.5
Tot	5	39	23.5	8	75.5

Microdata

Sono dati molto più precisi e dettagliati come possiamo vedere dalla tabella seguente.

N	Child	County	Educ	Salary	Ethnicity
1	John	Alfa	very high	201	AfrAm
2	Jim	Alfa	high	103	Caucas
3	Sue	Alfa	high	77	AfrAm

In questo caso ogni riga di colonna si riferisce ad uno specifico respondent, mentre nel caso precedente la tabella faceva riferimento ad un gruppo di respondent.

Come detto il rilascio di tali dati è molto rischioso. I principali accorgimenti da prendere in considerazione sono quelli per proteggere l'identità, attributi ed evitare il rischio di inferenza.

Nel corso degli anni sono state date diverse definizioni di disclosure: quella più attinente è "attribuzione impropria di informazioni a un responder, dove i dati riguardano singole persone o organizzazioni".

I rischi dei microdati sono:

Information disclosure: in questo caso abbiamo la disclosure quando

- Un respondent viene identificato dai dati rilasciati (**identity disclosure**). Si ha quando una parte terza è in grado di identificare un soggetto a causa dei dati rilasciati. Tuttavia, bisogna precisare che rivelare se un individuo è un respondent di una collezione di dati potrebbe/non potrebbe violare i requisiti di confidenzialità;
  - **Macrodata**: rivelare l'identità non è generalmente un problema, a meno che l'identificazione non porti a rivelare informazioni riservate (attribute disclosure, di solito avvengono sempre insieme);
  - **Microdata**: l'identificazione è generalmente considerato un problema, poiché i record dei microdati sono molto dettagliati;
- Con il rilascio di alcuni dati vengono rivelate anche info sensibili riguardo il respondent (**attribute disclosure**), ed inoltre è possibile attribuire tali info ad esso. Può verificarsi anche quando le informazioni riservate vengono rivelate esattamente o quando possono essere stimate accuratamente;
- I dati rilasciati consentono di determinare il valore di alcune caratteristiche di un respondent in modo accurato attraverso deduzioni, cosa che in caso contrario non darebbe stata possibile (**inferential disclosure**). Si verifica quando le informazioni possono essere dedotte con elevata sicurezza dalle proprietà statistiche dei dati rilasciati. È difficile prendere in considerazione questo tipo di divulgazione per due ragioni:
  - Se la divulgazione equivale all'inferenza, non è possibile rilasciare dati;
  - Le inferenze sono progettate per prevedere il comportamento aggregato, non i singoli attributi, e sono quindi spesso predittori inadeguati dei singoli valori dei dati.

<u>Restricted data/access</u>

La scelta dei metodi di limitazione statistici della disclosure dipende dalla natura dei dati prodotti la cui riservatezza deve essere protetta.

Questo può avvenire andando a rimuovere dati sensibili come nome, codice fiscale, e-mail… per preparare i microdati al “rilascio”. Inoltre, i dati possono essere protetti andando a fornire un accesso limitato ad essi oppure fornendo il rilascio di un insieme di dati ristretto, utilizzando tecniche definite precedentemente.

Tecniche di protezione

Le tecniche di protezione comprendono:

- Campionamento: la riservatezza dei dati è protetta mediante la conduzione di un censimento a campione del censimento;
- Regole speciali: progettate per tabelle specifiche, impongono restrizioni al livello di dettaglio che può essere fornito in una tabella;
- Regola di soglia: le regole sono destinate alle celle sensibili attraverso tecniche di soppressione delle celle, arrotondamento casuale, arrotondamento controllato e modifica confidenzialità.

Problema dell'anonimato nei dati

Diverse agenzie, istituzioni, uffici e organizzazioni rendono i dati (sensibili) che coinvolgono persone disponibili al pubblico

- microdati termici utilizzati per l'analisi;
- spesso richiesto e imposto dalla legge.

Per proteggere la privacy, i microdati vengono "igienizzati"

- gli identificatori espliciti (CF, nome, numero di telefono) vengono rimossi.

Ma non è sufficiente per preservare la privacy

- suscettibile a linking attacks.
- i database disponibili pubblicamente (elenchi degli elettori, elenchi delle città) possono rivelare l'identità "nascosta".

Cos'è il **linking attack**?

Tipologia di attacco che modella l'abilita di collegare i dati del database pubblicato a informazione esterna in possesso, che permette la re-identificazione di (alcuni degli) individui rappresentati nei dati. Nei database relazionali questo tipo di attacco è possibile grazie ai "*quasi-identifier*", cioè attributi come sesso e data di nascita, che combinati, possono identificare unicamente una persona. Gli attributi restanti rappresentano l'informazione privata che potrebbe essere violata attraverso l'attacco.

Per rafforzare la privacy nei microdati occorre identificare 3 diversi tipi di attributi:

- **identificatori espliciti**: vengono sempre rimossi;
- **quasi identifier**: usati per re-identificare gli individui;
- **attributi sensibili**: potrebbero non esistere, ovvero quelli che portano i dati sensibili.

identifier	quasi identifiers			sensitive
Name	Birthdate	Sex	Zipcode	Disease
Andre	21/1/79	male	53715	Flu
Beth	10/1/81	female	55410	Hepatitis
Carol	1/10/44	female	90210	Brochitis
Dan	21/2/84	male	02174	Sprained Ankle
Ellen	19/4/72	female	02237	AIDS

goal of privacy preservation (rough definition)
de-associate individuals from **sensitive information**

Quasi identifier

Sono attributi che non rappresentano id univoci, ma se combinati con altri quasi identifier possono creare un identificatore unico.

Questo tipo di identifier è stato alla base di diversi attacchi su alcuni dati rilasciati:

- Linking attack che è riuscito a collegare le info disponibili alla cartella medica del governatore del Massachussets (a scopo di ricerca);
- Linking attack che ha identificato gli utenti di netflix in base ai dati rilasciati.

Definizione 1: dato A = {a1, …, an} un set di attributi e D un dataset (tabella) definito su A → Un quasi-identifier di D è un set di attributi QI ⊆ A il cui rilascio deve essere controllato.

Ad esempio, data una tabella in cui sono annotati problemi fisici relativi ad alcune persone con i relativi dati, un quasi-identifier può essere il seguente:

- QI1= {ZIP, Date of Birth, Ethnicity};
- QI2= {ZIP, Date of Birth, Gender, MaritalStatus}.

Lezione 6

k-anonimity

Questo framework, proposto nel 1998, insieme alla sua applicazione attraverso la **generalizzazione e soppressione**, ha proposto un approccio per proteggere le identità dei respondent, pur fornendo informazioni veritiere.

Def formale: Sia T un set di dati sull'insieme A = {a1, ..., an} di attributi e QI_T l'insieme di quasi-identificatori associati a T. T soddisfa la k-anonimity se per ogni quasi-identifier QI ∈ QI_T e ogni sequenza esistente di valori degli attributi QI appaiono almeno con k occorrenze in T.

Cosa significa avere un dataset k-anonimizzato?

Partiamo dal presupposto che ogni rilascio di dati deve essere tale che ogni combinazione di valori di quasi-identifier possa essere indistintamente abbinata ad almeno k respondent in modo tale da:

- Si nasconde ogni individuo tra altri k-1 individui;
- Il linking attack non può essere eseguito con una confidenza > 1/k;

La k-anonimity richiede che, nel rilascio della tabella stessa, i respondent siano indistinguibili rispetto a un insieme di attributi.

Le condizioni necessarie per il requisito di k-anonimity sono:

- Ogni set di valori di quasi-identifier nella tabella rilasciata deve avere almeno k occorrenze;
- Attributi sensibili non sono considerati.

Come ottenere la k-anonimity

L'idea è di non sporcare i dati. Possiamo ottenerla in 2 modi:

1. **Generalizzazione:** i valori di un dato attributo vengono sostituiti utilizzando valori più generali. Tuttavia, se i dati vengono troppo generalizzati si rischia di rappresentare dati diversi allo stesso modo e quindi non è possibile distinguere i dati tra di loro;

2. **Soppressione**: è una tecnica ben nota che consiste nel proteggere le informazioni sensibili rimuovendole. Ad esempio rimuovere tuple molto rare. Sopprimere molte tuple rischia di compromettere la realtà dei dati.

Un esempio di uso di queste tecniche è il seguente

Birthdate	Sex	Zipcode
21/1/79	male	53715
10/1/79	female	55410
1/10/44	female	90210
21/2/83	male	02274
19/4/82	male	02237

original microdata

	Birthdate	Sex	Zipcode
group 1	*/1/79	person	5****
	*/1/79	person	5****
suppressed	1/10/44	female	90210
group 2	*/*/8*	male	022**
	//8*	male	022**

2-anonymous data

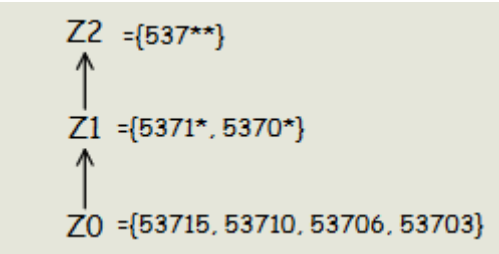
La terza tupla viene rimossa completamente in quanto outliner (facilmente identificabile rispetto alle altre) anche perché avendo solo una tupla diversa dalle altre non è possibile anonimizzarla.

N.B.: i dati anonimizzati possono essere rilasciati per analisi statistiche, e in caso fosso troppo anonimizzato diventerebbe inutile. Ciò significa che non bisogna anonimizzare troppo.

Gerarchie di generalizzazione del dominio

Una **gerarchia di generalizzazione di dominio** DGH_D di un attributo A è un ordine parziale sull'insieme di domini Dom A = {D0, ..., Dn} che soddisfa le seguenti condizioni:

1. Ogni dominio D_i ha al massimo un dominio generalizzato diretto;
2. Tutti gli elementi massimali di Dom sono singleton (per assicurare che tutti i valori in ciascun dominio possano essere eventualmente generati su un singolo valore)



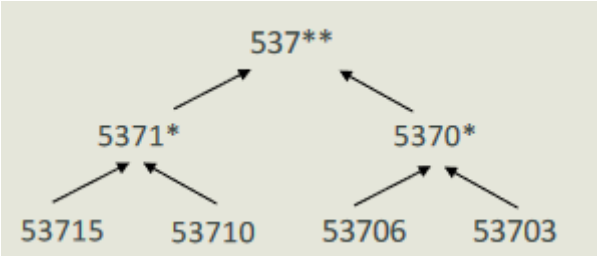
Gerarchie di generalizzazione del valore

Una **relazione di generalizzazione del valore** associa a ciascun valore vi del dominio Di un valore univoco vj nel dominio Dj, dove Dj è una generalizzazione diretta di Di.

Questa relazione implica l'esistenza, per ciascun dominio D, di una **gerarchia di generalizzazione del valore**, indicata con VGH_D

VGH_D è un albero, dove le foglie sono i valori in D e la radice (cioè il valore più generale) è il valore nell'elemento massimo in DGH_D.

Esempi di albero e di gerarchia di generalizzazione.

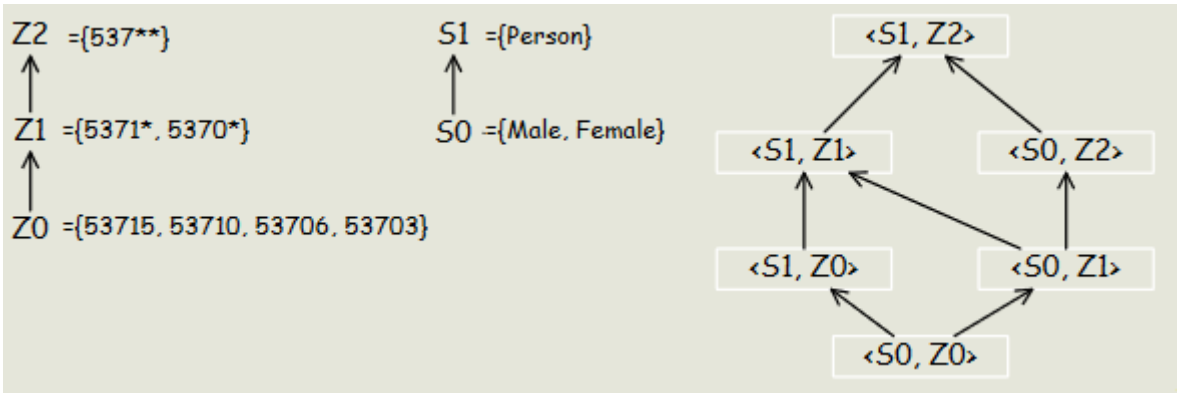


Reticolo di generalizzazione

Offre un percorso/strategia di generalizzazione.

Data una tupla di dominio DT = <DA1, ..., DAn> come DAi nel DomAi, la gerarchia di generalizzazione del dominio di DT è DGH_{DT}= DGH_{DA1} x ... x$DGHD_{An}$.

Inoltre, DGH_{DT} definisce un reticolo il cui elemento minimale è DT.



Il reticolo rappresenta le azioni di generalizzazione che possono essere effettuate, ma non aiuta a raggiungere l'obiettivo di generalizzazione con un senso per il dataset.

Tabella generalizzata con soppressione

Dato un insieme di attributi = {A1, ..., An} e due tabelle Ti e Tj definite su A. La tabella Tj è una **generalizzazione (con soppressione della tupla)** della tabella Ti, indicata con Ti ≤ Tj, sse:

1. Il dominio di ciascun attributo Ax in Tj è uguale o una generalizzazione del dominio di Ax in Ti;
2. Ogni tupla tj in Tj ha una tupla ti corrispondente in Ti t.c. per ogni attributo Ax, tj[Ax] è uguale o una generalizzazione di ti[Ax].

N.B.: non tutte le generalizzazioni sono le stesse perché potrebbero avere diversi livelli di k-anonimity in base all'attributo che si sceglie di generalizzare. E infatti, in alcuni otteniamo dei risultati migliori scegliendo un livello di k-anonimity minore anziché uno maggiore, in quanto quest'utlimo risultato molto generalizzato è quindi un dato inutilizzabile.

Distance Vector (DV)

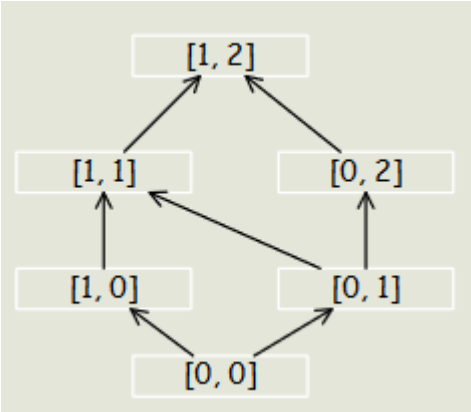
Siano Ti e Tj (generalizzazione di Ti) due tabelle definite sullo stesso insieme di attributi A = {A1, ..., An} tale che Ti ≤ Tj. Il **distance vector** di Tj da Ti è il vettore DVij = [d1, ..., dn] dove ogni dx è la lunghezza del percorso univoco tra dom (Ax, Ti) e dom (Ax, Tj) nella gerarchia di generalizzazione del dominio DGHDx.

Considerando gli esempi precedenti abbiamo il seguente D.V. in cui si possono vedere annotati i passi di generalizzazione effettuati. Ad esempio la coppia di valori [1,0] indica che è stato effettuato un passo di generalizzazione sul domino di di sx ovvero quello relativo al sesso della persona.

Criterio di Generalizzazione:

Prese due generalizzazione l'idea è di costruire il reticolo di generalizzazione dell'intero QI delle due generalizzazioni. Più saliamo nel reticolo più la generalizzazione aumenta e più assicuro la k-anonymity. Obiettivo è ottenere la minima generalizzazione che soddisfa la k-anonymity ovvero che minimizzi il distance vector.

Generalizzazione k-minimal con soppressione



Siano Ti e Tj (generalizzazione di Ti) due tabelle in modo tale che Ti≤ Tj e MaxSup sia la soglia specificata di soppressione accettabile. Tj è una generalizzazione k-minima di Ti sse:

1. Tj soddisfa la k-anonimity;
2. La soppressione è minima per ogni Tx: Ti≤ Tx, DVix=DVij, Tx è k-anonimo → |Tj| ≥ |Tx|;
3. |Tj| -|Ti| <= MaxSup;
4. Per ogni Tx: Ti ≤ Tx, Tx soddisfa le condizioni 1, 2 and 3 → DVix ≥ DVij.

In sostanza, date 2 generalizzazioni che soddisfano la k-anonimity, si terrà quella che minimizza il vettore delle distanze ed il numero di tuple soppresse.

Problema NP-Hard. Complessità determinata dal numero di atributi, se i quasi-identifier sono pochi gli algoritmi possono trovare facilmente utilizzo, altrimenti si richiede l'utilizzo di euristiche.E

Classificazione tecniche di k-anonimizzazione

La generalizzazione e la soppressione possono essere applicate a diversi livelli di granularità

La generalizzazione può essere applicata:

- a livello di **singola colonna**: un passaggio di generalizzazione generalizza tutti i valori nella colonna;
- a livello di **singola cella**: per una colonna specifica, la tabella può contenere valori a diversi livelli di generalizzazione;

La soppressione può essere applicata:

- a livello di **riga**: l'operazione di soppressione rimuove un'intera tupla;
- a livello di **colonna**: l'operazione di soppressione oscura tutti i valori di una colonna;
- a livello di **singola cella**: solo determinate celle di una data tupla / attributo possono essere cancellate.

Algoritmi per il problema della k-anonimity

Esplorare tutte le soluzioni possibili e trovare delle tabelle minime k-anonime con generalizzazione degli attributi e soppressione delle tuple è un problema NP-hard (problema difficile non deterministico in tempo polinomiale). Questa difficoltà riguarda il fatto che bisogna esplorare diversi livelli degli alberi o dei reticoli forniti andando ad aumentare la complessità dell'algoritmo stesso.

Lezione 7

Algoritmo di Samarati

Considerazioni iniziali:

- Ogni percorso nel reticolo DGH_{DT} rappresenta una strategia di generalizzazione che va dalla parte bottom alla parte top del reticolo stesso;
- Ci si basa sulla **generalizzazione locale minima** ovvero il nodo più basso di ogni percorso che soddisfa la k-anonimity per ogni percorso;
- Altre proprietà:
    - Ogni generalizzazione K-minima è localmente minima rispetto al percorso, tuttavia non è vero il contrario.
    - Man mano che si sale nella gerarchia il numero di tuple che devono essere rimosse per garantire la k-anonymity diminuisce: conseguenza del fatto che sono state rimosse le cosiddette tuple outliner che potrebbero compromettere la k-anonimity del dataset.
- Se non c'è una soluzione che garantisce la k-anonimity sopprimendo meno tuple di MaxSup all'altezza h, non può esistere una soluzione con altezza < di h che la garantisca.

L'ultima proprietà viene sfruttata con un approccio dicotomico sul reticolo dei distance vectors. L'algoritmo quindi sarà:

1. Valutare soluzioni all'altezza h/2;
2. Se esiste almeno una soluzione che soddisfa la k-anonimity:
    a. Allora valuta le soluzioni ad altezza $\lfloor h/4 \rfloor$;
    b. Altrimenti valuta le soluzioni ad altezza $\lfloor 3h/4 \rfloor$.
3. Proseguire fin quando l'algoritmo non raggiunge l'altezza minima per cui c'è un distance vector che soddisfa la k-anonimity.

Per ridurre il costo computazionale, L'algoritmo adotta una matrice dei vettori di distanza la quale evita il calcolo esplicito di ogni tabella generalizzata.

Example h=1 (k=2 and MaxSup=2)

- Compute first solution at height 1 ($GT_{[1,0]}$ and $GT_{[0,1]}$)

Ethn.: E_1	Zip: Z_0
person	*94142(*)*
person	94141
person	94139
person	94139
person	94139
person	*94138(*)*
person	94139
person	94139
person	94141

Ethn.: E_1	Zip: Z_0
asian	9414*
asian	9414*
asian	9413*
asian	9413*
asian	9413*
AfrAm	9413*
AfrAm	9413*
caucas(*)	*9413**
caucas(*)	*9414**

- Both the generalized table satisfy 2-anonymity

Nell'esempio abbiamo che, partendo da altezza 1 del reticolo, si generalizza prima un campo della tabella e poi l'altro. Entrambe le tabelle soddisfano la 2-anonimity perché abbiamo almeno 2 combinazioni di valori delle tuple. Tuttavia, dobbiamo rimuovere le tuple outliner per garantire la k-anonimity, ciò è possibile perché il num max di soppressioni MaxSup è 2.

Algoritmo Incognito

Idea: la k-anonimity, rispetto ad un sottoinsieme proprio di QI, è una condizione necessaria, ma non sufficiente per la k-anonimity, rispetto a QI. Con questo vogliamo dire che se prendiamo un sottoinsieme dell'insieme dei QI e al suo interno non è rispettata la k-anonimity allora non sarà rispettata anche nell' insieme generale.

Funzionamento:

- **Iterazione 1**: controlla la k-anonimity per ciascun attributo in QI, scartando la generalizzazione che non la soddisfa;
- **Iterazione 2**: combina le restanti generalizzazioni in coppie e controlla la k-anonimity per ogni coppia ottenuta;
- **Iterazione i**: combina tutte le i-uple di attributi ottenute combinando generalizzazioni che soddisfacevano la k-anonimity all'iterazione i –1. Elimina le soluzioni non anonime;
- **Iterazione | QI |**: restituisce il risultato finale.

Approccio Bottom-Up: parte da elementi meno generalizzanti fino ad elementi più generalizzanti per la visita del DHGs. Inoltre, questo algoritmo adotta un approccio a priori, cioè basato sulla generalizzazione dei candidati.

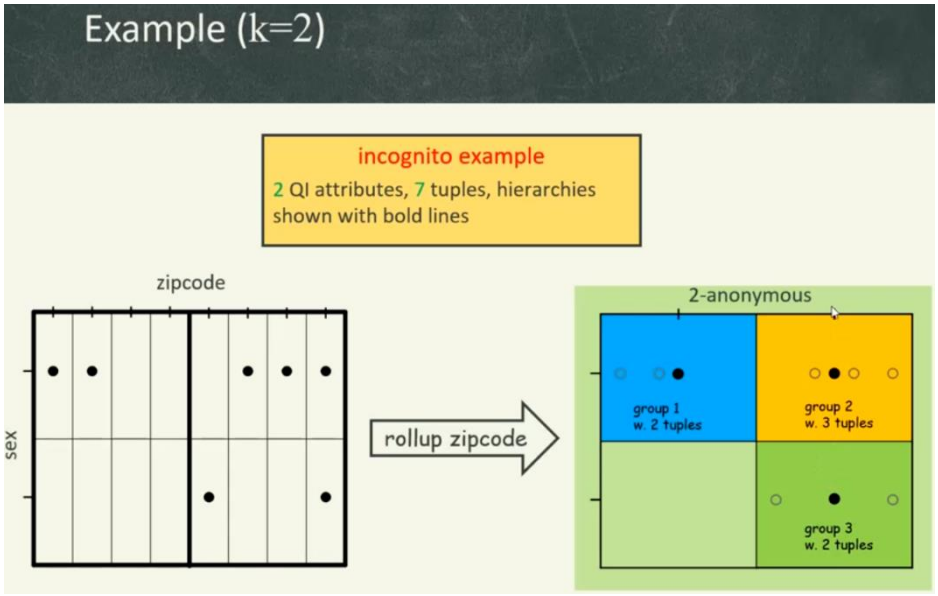
Inoltre, sfrutta le proprietà di monotonicità relative alla frequenza delle tuple nel reticolo (ricorda le gerarchie OLAP e il frequente mining del set di elementi).

Considerando sempre l'esempio sugli attributi S e Z e andiamo a calcolare l'insieme di generalizzazione. Definiamo le seguenti proprietà per l'algortimo incognito:

Proprietà di generalizzazione (~rollup in OLAP): se per un nodo in una gerarchia vale la k-anonimity, allora essa varrà anche per i suoi nodi antenati (a livello superiore). Se un nodo non è k-anonimo, allora non possiamo dire nulla sui suoi antenati in quanto, con opportune generalizzazioni, potrebbero diventarlo.

Proprietà dei subset (~apriori in OLAP): se per un set di QI non vale la k-anonimity, allora non può valere neanche per nessuno dei suoi sovrainsiemi anche se ci aggiungo un attributo, anzi vado a peggiorare i dati.

Incognito considera insiemi di attributi QI di cardinalità crescente (~ apriori) e elimina i nodi nel reticolo utilizzando le due proprietà elencate andando quindi a considerare solo i percorsi che sono promettenti dal punto di vista della k-anonimity.



Algoritmo Mondrian

Considerazioni iniziali:

- Ogni attributo in QI rappresenta una dimensione;
- Ogni tupla in PT (tabella privata) è rappresentata da un punto nello spazio definito da QI;
- Le tuple con lo stesso valore QI sono rappresentate associando il numero di occorrenze con dei punti;
- Lo spazio multidimensionale è partizionato dividendo le dimensioni in modo tale che ogni area contenga almeno k occorrenze dei valori dei punti;
- Tutti i punti in una regione sono generalizzati ad un valore unico;
- Le tuple corrispondenti sono sostituite da generalizzazioni calcolate.

Per raggruppare occorre utilizzare un criterio.

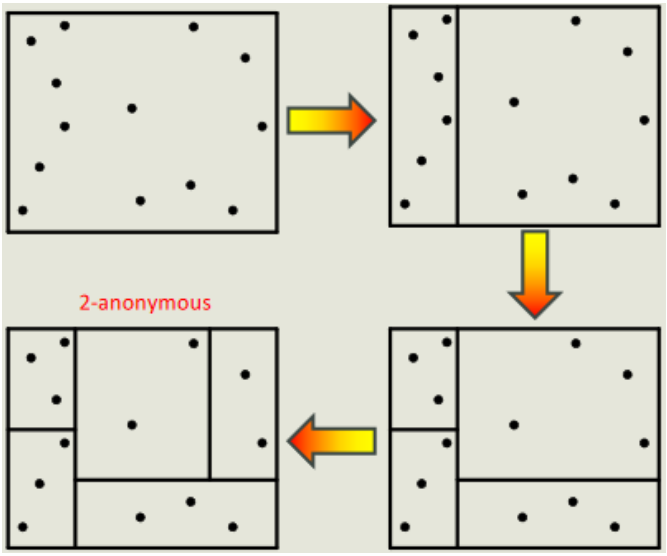
A differenza degli algoritmi visti in precedenza samarati e incognito (lavorano per minimizzare il livello di generalizzazione), tale algoritmo si usa la **discernability metric**.

Essa penalizza ogni tupla con le dimensioni del gruppo a cui appartiene (per grandezza intendiamo quanti valori distinti vengono rappresentati nel gruppo stesso).

L'idea alla base di Mondrian è quella di avere un raggruppamento ideale, in cui tutti i gruppi hanno la stessa dimensione, nel nostro caso k.

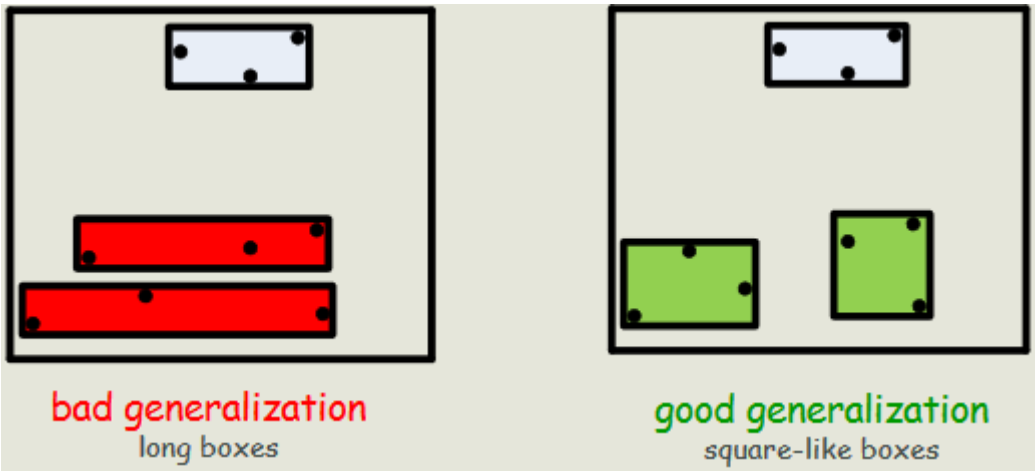
L'algoritmo quindi cerca di costruire gruppi di uguale dimensione.

Ipotizzando di lavorare su un dataset, l'algoritmo costruirà dei gruppi in base al livello di k-anonimity che si vuole avere andando a partizionare il dataset come possiamo vedere nell'esempio. Come detto prima, si cerca di partizionare e creare gruppi della stessa dimensione.



La qualità dei gruppi dipende dalla cardinalità del gruppo stesso, ovvero, in questo caso, ogni volta che viene creato un gruppo si cerca sempre di soddisfare la 2 k-anonimity.

Un buon gruppo contiene tuple con valori di QI simili. Tuttavia, tale proprietà non viene garantita da questo algoritmo. Per questo motivo viene definita una nuova metrica, ovvero la **NCP (Normalized Certainty Penalty)**, la quale misura il perimetro del gruppo per usarlo come penalty. Attraverso questa metrica possiamo vedere il vantaggio principale che apporta in quanto si vanno a raggruppare elementi simili tra loro.



Algoritmo Topdown

Fasi:

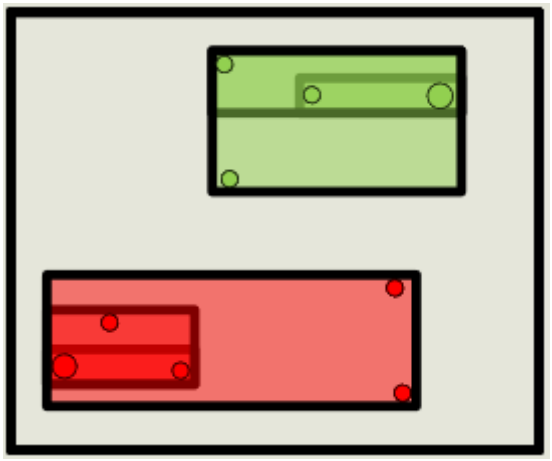
1. Si inizia con l'intero dataset;
2. Iterativamente si divide il dataset in 2;
3. Si continua fin quando non sono rimasti dei gruppi che contengono un numero meno di 2*k – 1 tuple.

Inizialmente trova i 2 punti più lontani (semi) che genereranno i 2 gruppi di split

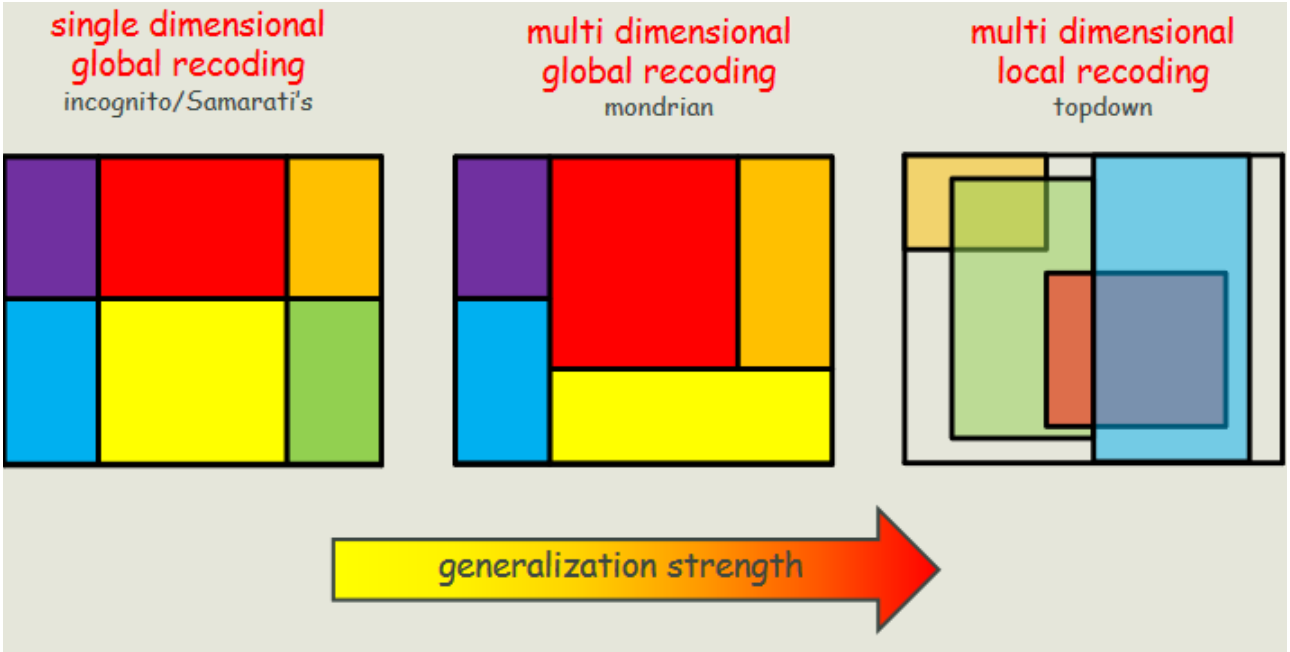
• ricerca quadratica euristica, non completa;

• i semi diventeranno casualmente i 2 punti divisi.

Va prima a trovare nell'unico spazio disponibile gli elementi con maggiore diversità, successivamente cerca gli elementi più vicini agli ultimi punti trovati e li aggrega in un gruppo in modo da costruire un perimetro. Quindi l'obiettivo è sempre quello di ingrandire i rettangoli cercando sempre di minimizzare il perimetro.



Generalization Strength



Quindi possiamo concludere:

1- L'algoritmo samarati/incognito in sostanza, ottengono un single dimensional global recoding, ovvero data una dimensione lo split è uguale su tutte le altre dimensioni di x ed y.
2- Per mondrian, dato un gruppo permette di splittare le varie dimensioni sui vari gruppi presenti.
3- Per topdown permette di effettuare dei raggruppamenti locali, ovvero minimizzando la misura del perimetro permette di ottenere della soluzione che si sovrappongono aumentando quindi la generalizzazione.

Quindi possiamo dire che i dataset ottenuti con gli algoritmi topdown e mondrian saranno molto più utilizzabili dal punto di vista della qualità del dato rispetto agli algoritmi di incognito/samarati che rischiano di generalizzare troppo avendo un solo raggruppamento per ogni dimensione a differenza degli altri due algoritmi.

Lezione 8

Problemi con la k-anonimity

Homogeneity attack

Può accadere che quando tutte le tuple con lo stesso valore quasi-identifier in una tabella K-anonimity hanno lo stesso valore nell'attributo sensibile.

Ad esempio, se un malintenzionato conosce i dati di una persona e sa anche che i dati della stessa si trovano in una certa tabella sulle malattie, allora egli potrà inferire che la persona soffre di una certa malattia.

Background knowledge attack

Basato su una conoscenza precedente che consiste di alcune informazioni esterne aggiuntive.

Ad esempio, sapendo che una persona in passato faceva una certa attività fisica, quindi si sottoponeva periodicamente a controlli, allora nella tabella che riporta i malati di cuore e di infezioni virali, posso escludere la prima ipotesi in quanto la persona in questione era sempre controllata sotto quel punto di vista.

Positive and negative disclosure

Date le conoscenze di base dell'"avversario", una tabella pubblicata GT (tabella generalizzata) potrebbe divulgare le informazioni in due modi:

- **disclosure positiva**: la pubblicazione della tabella GT derivata da PT (tabella privata) produce una divulgazione positiva se l'avversario è in grado di identificare correttamente il valore di un attributo sensibile con alta probabilità;
- **disclosure negativa**: la pubblicazione della tabella GT derivata da PT produce una divulgazione negativa se l'avversario può eliminare correttamente alcuni possibili valori dell'attributo sensibile (con alta probabilità). Inoltre, se iterata può portare alla disclosure positiva, perché eliminando tutti man mano gli attributi riuscirò ad attribuire quei dati a quel soggetto.

Definizioni

- Sia q un valore di un attributo non sensibile Q in PT;
- Lascia che q* sia il valore generalizzato di q in GT;
- Selezioniamo un possibile valore dell'attributo sensibile S;
- Lascia n (q*, s) il numero di tuple t* ∈ GT dove t* [Q] = q* e t* [S] = s;
- Lascia che un blocco q* (aka **classe di equivalenza**) sia l'insieme di tuple in GT i cui valori non sensibili di Q si generalizzano in q*.

Lack of diversity (mancanza di diversità)

La mancanza di diversità in un attributo sensibile S si verifica quando:

$$\text{per ogni } s' \neq \text{da } s \; n(q^*, s') << n(q^*, s)$$

In questo caso Pr(t[S]=s | t*[Q]=q*) ≈ 1 quindi in teoria conoscendo qual è il Q* block riesco a dedurre il valore dell'attributo S sulla stessa tupla con una probabilità uguale a 1. ed anche i valori sensibili della persona identificata da t[Q] = q possono essere determinati con alta precisione dall'attaccante, a condizione che egli sappia che la persona è nella PT.

Principio di l-diversity

Per garantire la diversità, un blocco q* dovrebbe avere almeno l (elle) ≥ 2 diversi valori sensibili in modo tale che i valori più frequenti (nel blocco q*) abbiano all'incirca la stessa frequenza.

Così un blocco q* è **ben rappresentato** da l valori sensibili. Quindi in poche parole, se ci sono l valori sensibili e questi sono tutti alla stessa frequenza, difficilmente potro capire quali di questi valori fanno riferimento a quella persona rappresentati da un blocco q*. In sostanza, se ci sono almeno l valori sensibili rappresentati da un blocco q* l'attaccante ha bisogno di l – 1 pezzi di conoscenza (info potenzialmente dannose) di base per eliminare l-1 i valori sensibili possibili della persona e inferire una disclosure positiva.

Impostando il parametro l, l'editore di dati può determinare il livello di protezione fornita contro le conoscenze di base (anche se queste conoscenze di base non sono note all'editore).

Quindi in sostanza se abbiamo un valore di k abbastanza alto e un valore di l abbastanza alto per garantire la diversità all'intero del Q* block, allora possiamo affermare che abbiamo un dataset abbastanza sicuro da non subire attacchi.

Proprietà locale: un blocco q* è l-diverso se contiene almeno l valori ben rappresentati per l'attributo sensibile S. Proprietà globale: una tabella è l-diversa se ogni blocco q* è l-diverso.

4-anonymous data

Id	Zipcode	Age	National.	Disease
1	130**	<30	*	Heart Disease
2	130**	<30	*	Heart Disease
3	130**	<30	*	Viral Infection
4	130**	<30	*	Viral Infection
5	1485*	≥40	*	Cancer
6	1485*	≥40	*	Heart Disease
7	1485*	≥40	*	Viral Infection
8	1485*	≥40	*	Viral Infection
9	130**	3*	*	Cancer
10	130**	3*	*	Cancer
11	130**	3*	*	Cancer
12	130**	3*	*	Cancer

4-anonymous / 3-diverse data

Id	Zipcode	Age	National.	Disease
1	1305*	≤40	*	Heart Disease
4	1305*	≤40	*	Viral Infection
9	1305*	≤40	*	Cancer
10	1305*	≤40	*	Cancer
5	1485*	>40	*	Cancer
6	1485*	>40	*	Heart Disease
7	1485*	>40	*	Viral Infection
8	1485*	>40	*	Viral Infection
2	1306*	≤40	*	Heart Disease
3	1306*	≤40	*	Viral Infection
11	1306*	≤40	*	Cancer
12	1306*	≤40	*	Cancer

Entropia l-diversity

Una tabella è l-diversa se rispetta questa entropia, cioè se per ogni blocco q* abbiamo che:

$$-\sum(s \in S)\ p(q*, s)\ \log(p(q*,s)) \geq \log(l)$$

dove $p(q*, s)$ = (n(q*, s) / $\sum(s' \in S)$ n($q*$, s')) è la frazione di tuple nel blocco q* con valore dell'attributo sensibile uguale a s.

Teorema: monotonicità della l-diversity

L'entropia soddisfa la proprietà di monotonicità: se una tabella GT soddisfa l'entropia l-diversity, allora qualsiasi generalizzazione di GT la soddisfa.

Di conseguenza ogni algoritmo di k-anonimity può essere esteso per rinforzare la proprietà di l-diversity.

Problemi l-diversity:

Se abbiamo 2 blocchi q* diversi ma con attributi sensibili simili si può essere soggetti ad alcuni attacchi, poiché se i so che quel soggetto ha quella particolare caratteristica so anche che fa parte di quel blocco di valori. Ciò rende la l-diversity vulnerabile ad attacchi basati sulla distribuzione dei valori interna ai blocchi (classi di equivalenza).

I 2 principali tipi di attacchi sono:

- **skewness attack**: succede quando la distribuzione in un blocco q è diversa dalla popolazione originale. Avendo quindi accesso a queste distribuzioni di dati diverse posso in qualche modo fare inferenza;
- **similarity attack**: si verifica quando un blocco q ha valori diversi ma semanticamente simili per l'attributo sensibile. (esempio di una tabella 2-anonimity ma con due attributi sensibili simili)

Principio di t-closeness

Si dice che una classe di equivalenza (blocco q*) abbia t-closeness se la distanza tra la distribuzione di un attributo sensibile in questa classe e la distribuzione dell'attributo nell'intera tabella non è maggiore di una certa soglia t.

Una tabella si dice t-close (per avere t-closeeness) se tutte le classi di equivalenza hanno t-closeness.

Come si misura la distanza tra le 2 distribuzioni?

Richiedere che P e Q siano vicini limiterebbe la quantità di informazioni utili rilasciate, in quanto limita le informazioni sulla correlazione tra attributi di quasi-identificatore e attributi sensibili.

Tuttavia, questo è esattamente ciò che è necessario limitare: se un osservatore ottiene un'immagine troppo chiara di questa correlazione, si verifica la disclosure degli attributi.

Il parametro t nella t-closeness consente di scambiare tra utilità e privacy.

Possibili distanze (da calcolare)

Date due distribuzioni P = {p1, p2, ..., pm} e Q = {q1, q2, ..., qm} due distanze ben note sono:

$$D[\boldsymbol{P},\boldsymbol{Q}] = \sum_{i=1}^{m} \frac{1}{2}|p_i - q_i|$$ **Distanza Variazionale**

$$D[\boldsymbol{P},\boldsymbol{Q}] = \sum_{i=1}^{m} p_i \log \frac{p_i}{q_i}$$ **Kullback-Leibler (KL distance)**: essa può essere vista come una sorta di cross entropy tra i valori.

Queste 2 distanze non tengono in considerazione le distanze di tipo semantico.

Per questo la t-closeness utilizza la Earth Mover Distance (EMD) la quale si basa sulla minima quantità di lavoro necessaria per trasformare una distribuzione in un'altra spostando la massa di distribuzione tra loro. In sostanza, per capire se sto calcolando una distanza tra distribuzioni molto alta è necessario che attributi semanticamente diversi tra di loro vadano ad avere distribuzioni diverse.

EMD (attributi numerici)

Let the attribute domain be $\{v_1, v_2, \ldots v_m\}$

The (normalized) distance between two numerical values is

$$dist(v_i, v_j) = \frac{|i-j|}{m-1}$$

Let $r_i = p_i - q_i$, the EMD distance can be computed as follows

$$D[\boldsymbol{P},\boldsymbol{Q}] = \frac{1}{m-1}(|r_1| + |r_1 + r_2| + \cdots + |r_1 + r_2 + \cdots + r_{m-1}|)$$

i.e.,

$$D[\boldsymbol{P},\boldsymbol{Q}] = \sum_{i=1}^{m} \left| \sum_{j=1}^{i} r_j \right|$$

EMD (attributi categorici)

- For **flat attribute domains** (without a domain hierarchy)

$$D[\boldsymbol{P},\boldsymbol{Q}] = \sum_{i=1}^{m} \frac{1}{2}|p_i - q_i|$$

- For **attributes with a domain hierarchy**, EMD is

$$D[\boldsymbol{P},\boldsymbol{Q}] = \sum_{N} cost(N)$$

where N is a non-leaf node and cost(N) is the cost of moving between N's children branches

Esempio

microdata

id	Zipcode	Age	Salary	Disease
1	47677	29	3K	Gastric ulcer
2	47602	22	4K	Gastritis
3	47678	27	5K	Stomach ulcer
4	47905	43	6K	Gastritis
5	47909	52	11K	Flu
6	47906	47	7K	Bronchitis
7	47605	30	8K	Bronchitis
8	47673	36	9K	Pneumonia
9	47607	32	10K	Stomach ulcer

3-anonymous data

id	Zipcode	Age	Salary	Disease
1	4767*	≤40	3K	Gastric ulcer
2	4767*	≤40	5K	Stomach ulcer
3	4767*	≤40	9K	Pneumonia
4	4790*	>40	6K	Gastritis
5	4790*	>40	11K	Flu
6	4790*	>40	8K	Bronchitis
7	4760*	≤40	4K	Gastritis
8	4760*	≤40	7K	Bronchitis
9	4760*	≤40	10K	Stomach ulcer

0.167-close w.r.t. Salary and 0.278-close w.r.t. Disease

Problema t-closeness

- Sappiamo che un problema derivante dalla k-anonimity e che l'attaccante possiede delle informazioni di cui noi non sappiamo nulla a riguardo, non conosciamo la probabiltà e quale algortimo di inferenza verrà usato.
- Ogni attributo è un potenziale quasi-identifier. Inoltre, bisogna aggiungere che è facile ottenere informazioni esterne sulle persone. Quindi generalizzando i gli attributi sui quasi-identifier avrò un dataset la cui utilità è molto scarsa.

Membership disclosure: riguarda il fatto che una tabella contiene (in modo anonimizzato) o meno le info di una certa persona. Dato che i quasi-identifier, con alta probabilità, sono unici all'interno di una popolazione, anche andando a generalizzare/sopprimere si può dire che una certa persona a cui è associato un quasi-identifier è presente nella tabella.

Possiamo quindi dire che la k-anonymity potrebbe non nascondere se una determinata persona si trova nel dataset.

A questo proposito è stata introdotta la **δ-presence** che va a limitare la probabilità che se che un attaccante riesce a capire se una persona è presente o meno nel dataset. Va affiancata alla k-anonimity.

Data una tabella pubblica esterna T e una privata PT, diciamo che la presenza δ vale per una generalizzazione GT di PT, con δ = (δmin, δmax) se:

$$\delta min \leq P(t \in PT \mid GT) \leq \delta max \quad \forall t \in T$$

Quindi in un dataset che rispetta questa proprità, diremo che ogni tupla $t \in T$ è δ-present in PT. Il risultato sarà il range di probabilità (min, max) che una certa tupla si trovi nella tabella ovvero $P(t \in PT \mid GT)$. Diremo quindi che GT è (δmin, δmax) -generalization di PT. (esempio slide 31). Cosa succede se abbiamo la probabilità minima, possiamo dire con alta probabiltà di escludere la persona e quindi questa informazione puo essere usata anche per scopi malevoli, sapendo che quel soggetto non è presente in quel dataset.

Monotonicità della δ-presence: se una generalizzazione GT2 di una tabella non è δ-present rispetto ad una tabella pubblica T e alla tabella privata PT allora non lo è neanche GT1. (N.B.: GT2 generalizzazione di GT1). Tale proprietà può essere usata negli algoritmi k-anonimity ad esempio per potare percorsi nel reticolo che non rispettano la δ-presence.

Scegliere un buon δ:

Example: release of a dataset related to diabetes patients

Let I_p be the event that person p has diabetes

Since the rate of diabetes in all US population is public information, any adversary will have a prior belief b_r on I_p given the public dataset T

$$b_r = P(I_p) = 0.07$$

The private dataset PT is a subset of the set of all diabetes patients in T. Seeing some anonymization GT of PT, attacker will have a posterior belief b_o on I_p:

$$b_o = P(I_p|GT) =$$
$$= P(I_p|p \in PT) \cdot P(p \in PT|GT) + P(I_p|p \notin PT) \cdot P(p \notin PT|GT)$$
$$= 1 \cdot P(p \in PT|GT) + \frac{P(I_p) \cdot |T| - |PT|}{|T| - |PT|} \cdot (1 - P(p \in PT|GT))$$
$$= P(p \in PT|GT) + \frac{|T| \cdot (1 - b_r)}{|T| - |PT|} + \frac{b_r \cdot |T| - |PT|}{|T| - |PT|}$$

Lo svolgimento dei passaggi si ottiene grazie a Bayes. Cherry pick = “selezione accurata dell’impiegato”

We need to ensure that the company can’t “cherry-pick” employees known **not to be** in the database

Thus the posterior belief should not be arbitrarily low

If we let the cost c = \$200, then we must ensure

$$b_o \cdot d - b_r \cdot d \geq c \qquad b_r - b_o \leq \frac{c}{d} = \frac{2}{100}$$

and we obtain

$$\delta_{min} = P(p \in PT|GT) \geq 0.02$$

We consider an acceptable cost due to misuse: assume a hiring decision, and that a \$100 annual difference in total cost of employee is noise (difference in productivity, taking an extra sick day, salary negotiation, etc.)

If expected annual cost of medical treatment of diabetes based on misuse of the database is c < \$100, the risk of misuse is acceptably small

The total cost of diabetes per person is around d = \$10,000

We must ensure

$$b_o \cdot d - b_r \cdot d < c \qquad b_o - b_r < \frac{c}{d} = \frac{1}{100}$$

If $|PT| \cong 0.04|T|$, we obtain

$$\delta_{max} = P(p \in PT|GT) \leq 0.05$$

Conclusioni

- **Protezione contro le misure di utilità**: sono necessarie ricerche per sviluppare misure che consentano agli utenti di valutare, oltre alla protezione offerta dai dati, l'utilità dei dati rilasciati;
- **Algoritmi efficienti**: il calcolo di una tabella che soddisfa la k-anonimity garantendo la minimalità è un problema NP-hard;
- **Nuove tecniche**: la proprietà della k-anonimity non sono legate ad una tecnica specifica;
- **Fusione di diverse tabelle e viste**: la proposta originale di k-anonimity e la maggior parte dei lavori secondari ipotizzano:
  - L'esistenza di una singola tabella da rilasciare;
  - La tabella rilasciata contiene al massimo una tupla per ogni intervistato;
- **Conoscenza esterna**: la k-anonimity non ha modella la conoscenza esterna che può essere ulteriormente sfruttata per fare inferenza ed esporre i dati dall'identità o la disclosure degli attributi.

K-anonimity soffre di **Curse of dimensionality (problema)**

La generalizzazione si basa fondamentalmente sulla località spaziale

- Ogni record deve avere k vicini vicino

I set di dati reali sono molto scarsi

- Molti attributi (dimensioni)
  - Dataset Netflix Prize: 17.000 dimensioni
  - Record dei clienti Amazon: diversi milioni di dimensioni
- Il “vicino più vicino” è molto lontano

La proiezione a dimensioni ridotte perde tutte le informazioni → dataset k-anonimizzati sono inutili.

Quindi in caso di dataset molto grandi la k-anonimity viene scartata.

Lezione 9

Data Mining

Insieme di tecniche e metodologie che hanno per oggetto l'estrazione di informazioni utili da grandi quantità di dati (es. banche dati, datawarehouse ecc...), attraverso metodi automatici o semi-automatici (es. apprendimento automatico) e l'utilizzo scientifico, aziendale/industriale o operativo delle stesse.

Quello che il data mining produce sono:

- Regole di associazione;
- Classificatori;
- Cluster.

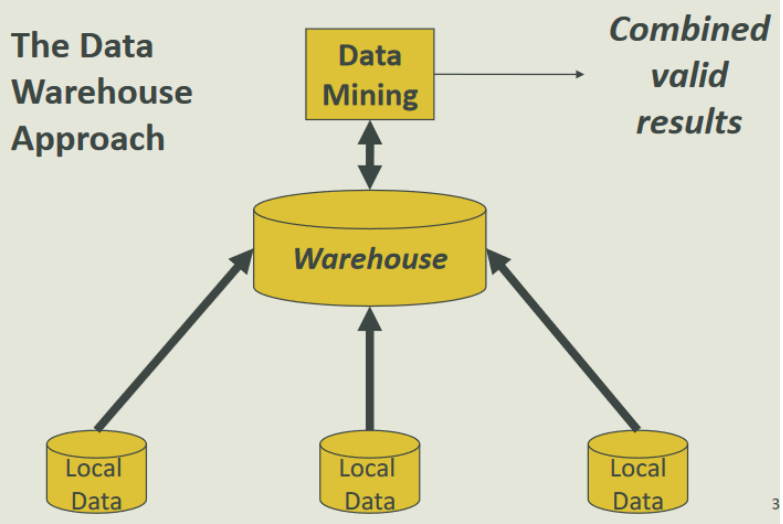
I risultati in sé non violano la privacy in quanto non contengono valori identificabili individualmente e rappresentano risultati complessivi.

Il problema è estrarre i risultati senza l'accesso ai dati.

Metodo standard

L'approccio tradizionale al mining di dati da fonti distribuite è stato quello di raccogliere tutti i dati in un data warehouse centrale, quindi eseguire algoritmi di data mining contro il warehouse. Esistono numerose applicazioni che non sono realizzabili in base a tale vincolo, il che comporta la necessità di un data mining distribuito, in modo da garantire alcune caratteristiche di base:

- Performance;
- Connettività;
- Eterogeneità e privacy delle fonti.



Private Distributed Mining

La semplice creazione di un algoritmo di data mining che estrae i dati direttamente dalle fonti locali non risolve questi problemi. Gli impedimenti per la creazione di un data warehouse esistono ancora, vengono semplicemente uniti al data mining.

Una possibile soluzione può essere quella di effettuare un mining locale sulle fonti locali di dati. Poi un aggregatore metterà insieme i risultati ottenuti in un unico risultato.

Principalmente questa operazione interessa a:

- Governi, agenzie pubbliche (Asl): in caso di epidemia potrebbero usare questi dati per agire di conseguenza;
- Industrie, aziende che lavorano nello stesso settore: potrebbero usare il mining per identificare quali sono le azioni migliori da intraprendere per aiutare i membri di questi trade groups, tuttavia alcuni processi sono segreti e non possono essere svelati.
- Multinazionali: un'azienda vorrebbe estrarre i suoi dati per risultati validi a livello globale. Ma le leggi nazionali potrebbero impedire la condivisione oltre i propri confini dei dati;

- Uso pubblico di dati privati: il mining di dati consente studi di ricerca su popolazioni di grandi dimensioni. Ma queste popolazioni sono riluttanti a rilasciare informazioni personali.

Classi di soluzioni

1. Data Obfuscation: nessuno vede i dati reali (nascosti/rumorosi);
2. Summarization: vengono mostrati solo i fatti necessari;
3. Data Separation.

Data Obfuscation

L'obiettivo è quello di nascondere l'informazione protetta. Per farlo esistono diversi approcci come modificare in maniera random i dati, scambiare i valori tra record oppure effettuare modifiche controllate per nascondere i dati.

Tuttavia, questa tecnica porta ad alcune problematiche come ad esempio se i dati vengono realmente nascosti o se è comunque possibile ricavare info ed imparare dai risultati.

Summarization

L'obiettivo in questo caso è rendere disponibili solo dei piccoli riassunti di dati innocui (non possono essere usati da terzi inconsapevolmente).

Gli approcci in questo caso sono molteplici: effettuare statistiche complessive dei dati e usare una possibilità di effettuare query limitata.

Anche in questo caso si possono avere alcune problematiche: la possibilità di dedurre dati dalle statistiche e se le info sono sufficienti.

Data Separation

L'obiettivo prevede che i dati rimangano sempre in locale e sono condivisi in maniera parziale e/o cifrata con una terza parte (fidata) per l'analisi.

Gli approcci possibili sono: dati tenuti dai propri creatori/proprietari; effettuare un rilascio limitato ad una parte terza fidata che effettuerà operazioni/analisi.

Problemi: la parte terza è in grado di effettuare le analisi sui dati? I risultati delle analisi rilasciano info private?

Alberi di decisione

Es 1 – Uso di metodi di data obfuscation

Scenario

Assumiamo che gli utenti saranno predisposti:

- Fornisce valori reali di determinati campi;
- Fornisce valori modificati per determinati campi (campi che potrebbero richiedere dati sensibili).

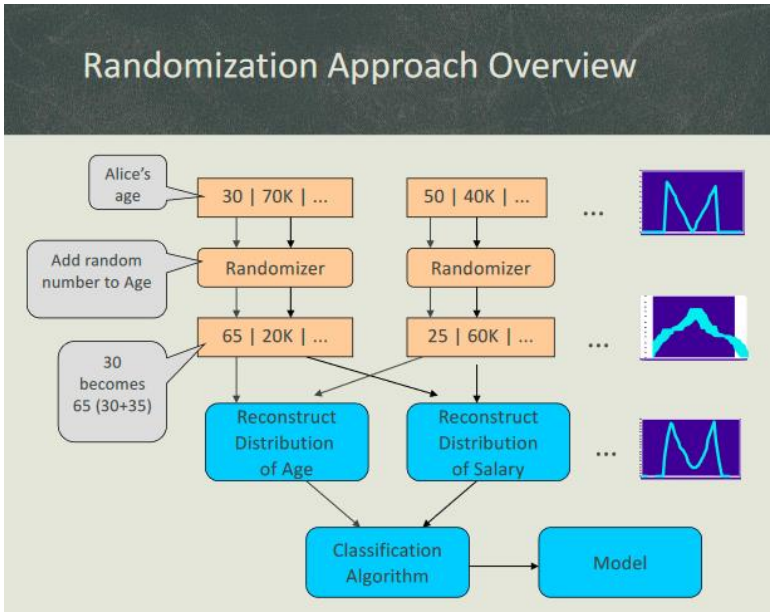
A questa valutazione si aggiunge che:

- 17% si rifiuta di partecipare a indagini e quindi a fornire dati;

- 56% sono disposti a rispondere, basta che venga mantenuta la privacy;
- 27% sono disposti, con una leggera preoccupazione per la privacy.

Si procede quindi a perturbare i dati con la distorsione dei valori degli attributi:

l'utente fornisce un certo dato Xi che non sarà memorizzato come tale, ad esso si aggiungerà un valore random r. r potrà essere estratto da una distribuzione uniforme [-alfa, alfa] oppure può essere un rumore di tipo Gaussiano che segue una distribuzione normale con media μ=0 e varianza σ.



Nell'immagine possiamo vedere che i dati forniti dall'utente vengono sporcati andando a modificare la sua distribuzione; così facendo si hanno dei nuovi valori che proteggono quelli originali. Sfruttando i dati di tutti gli utenti è possibile ricostruire le distribuzioni dei vari parametri, poi con le distribuzioni è possibile costruire un algoritmo di classificazione che va a creare un modello predittivo attraverso cui andare a ricavare la distribuzione originale (sarà simile ma non uguale) dai dati randomizzati conoscendo il metodo usato per perturbare i dati.

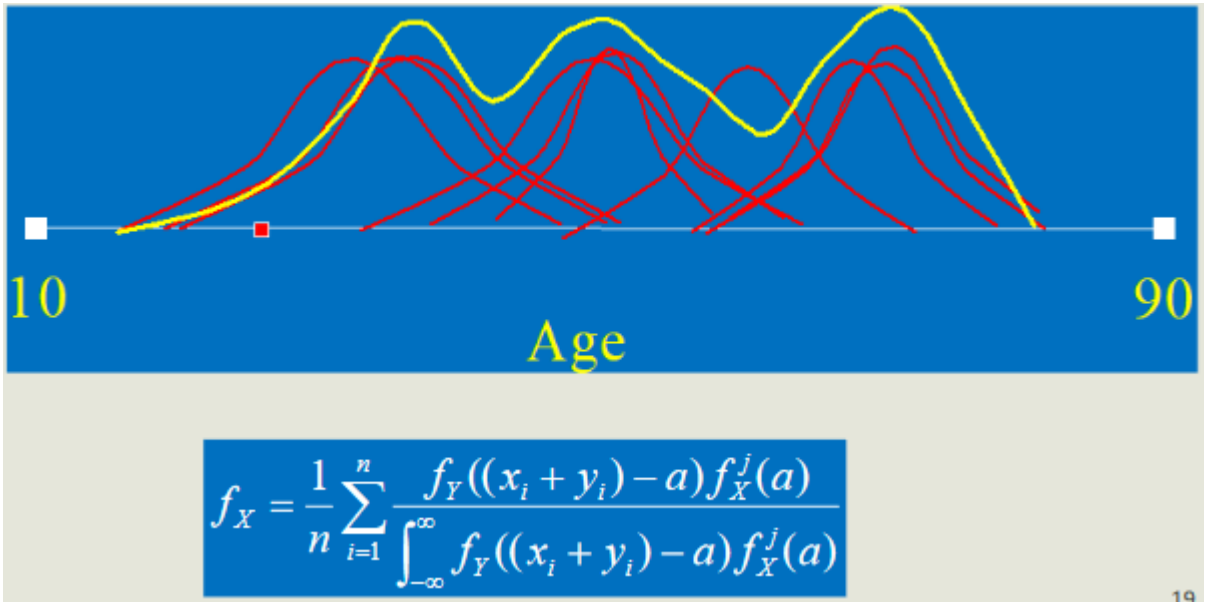
Original Distribution Reconstruction: Def formale

Dati x1, x2,…, xn valori degli n dati originali, usando la **distorsione dei valori**, avremo che i valori diventeranno w1 = x1 + y1, …, wn = xn + yn dove le y sono dei valori random identicamente distribuiti che seguono Y cioè la distribuzione del rumore.

Il **problema** consiste nello stimare la funzione di densità Fx dati i valori distorti wi e la funzione di densità Fy.

Come si può fare ciò?

Regola di Bayes per le funzioni di densità: viene effettuata una stima probabilistica del valore originale. L'obiettivo è rendere la stima il più possibile vicina all'originale. Inoltre, combinando diverse stime in base ai valori V che sono a disposizione posso avere una stima molto più completa e fedele all'originale. Quindi usando la regola di Bayes per le funzioni di densità avremo che Fx sarà:



Bootstrapping method

In questo metodo la stima della funzione di densità è

$$f'_X(a) = \frac{1}{n}\sum_{i=1}^{n} \frac{f_Y(w_i - a) f_X(a)}{\int_{-\infty}^{\infty} f_Y(w_i - z) f_X(z) dz}$$

avendo un buon numero di campioni, posso stimare la funzione con l'utilizzo della appena citata con una buona confidenza. Tuttavia, non si conosce Fx quindi non posso usare direttamente quella scrittura, ma iterativamente posso raffinare la stima di Fx usando il valore precedente di ogni computazione

$$f_X^{j+1}(a) = \frac{1}{n}\sum_{i=1}^{n} \frac{f_Y(w_i - a) f_X^j(a)}{\int_{-\infty}^{\infty} f_Y(w_i - z) f_X^j(z) dz}$$

iniziando da una stima iniziale che per Fx può essere la distribuzione uniforme, proseguendo poi fin quando non ho una convergenza del risultato.

Il problema di questo algoritmo è l'integrale al denominatore, in quanto la sua stima è molto costosa. Per questo viene usato un partizionamento dei valori possibili nell'intervallo di valori che devono essere stimati per velocizzare il calcolo. Più il partizionamento sarà fine, più sono i campioni di dati di cui si avrà bisogno e viceversa.

La formula utilizzata è simile alla precedente, semplicemente si calcola la probabilità all'iterazione j+1 che x appartenga ad un certo intervallo I. Questo avviene usando la regola di Bayes applicata al discreto.

$$Pr^{j+1}(X \in I_p) = \frac{1}{n}\sum_{s=1}^{k} N_{I_s} \frac{f_Y(m_{I_s} - m_{I_p}) Pr^j(X \in I_p)}{\sum_{t=1}^{k} f_Y(m_{I_s} - m_{I_t}) Pr^j(X \in I_t)}$$

where

- k is the number of intervals,
- N_I is the number of points that lie in the interval I,
- m_I is the mid point of the interval I.
- The denominator can be evaluated independently of I_p ➔ $O(n^2)$

Criteri di stop (convergenza): test del chi-quadro attraverso cui possiamo verificare che i dati siano distribuiti in maniera simile.

Il metodo funziona abbastanza bene (dimostrato).

<u>Classificazione alberi di decisione</u>

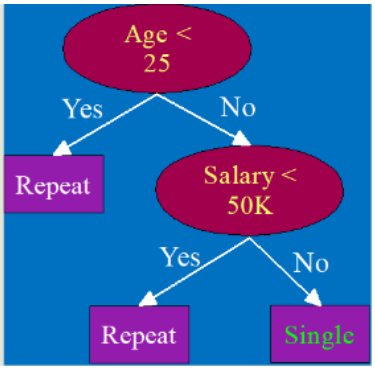
Idea: partizionare ricorsivamente i dati fino a quando ogni partizione contiene principalmente esempi della stessa classe.

Lo sviluppo avviene in 2 fasi:

1. Growth Phase (costruzione albero):
    a. I dati vengono partizionati secondo i valori di ogni singolo attributo (gini index, entropia, ...)
    b. Utilizzare il best split (massimizza gli indici) per partizionare i dati in due nodi
    c. Ripetere per ciascun nodo, se i punti non sono della stessa classe
2. Prune Phase:
    a. Generalizzare l'albero per evitare overfitting e predire bene esempi non ancora visti
    b. Rimuovere il rumore statistico

Esempio utilizzo dell'albero

Age	Salary	Repeat Visitor?
23	50K	Repeat
17	30K	Repeat
43	40K	Repeat
68	50K	Single
32	70K	Single
20	20K	Repeat



Nel dataset in questione abbiamo come info l'età e lo stipendio di alcune persone. Il primo nodo di split è l'età in cui abbiamo una condizione: se verificata si andrà nella classe repeat e si dovrà ripetere la procedura altrimenti ci sarà un secondo nodo di split.

L'idea è sempre quella di lavorare con la ricostruzione sui valori distorti.

Split Points

- Durante la fase di ricostruzione, i valori degli attributi vengono suddivisi in intervalli per velocizzare le interpretazioni
- I valori al limite degli intervalli sono i candidati ad essere split points

Data partitioning

- La fase di ricostruzione fornisce una stima del numero di punti in ciascun intervallo
- L'idea è partizionare i dati in S1 e S2 dove S1 contiene i punti dati con valori più bassi

Quando e come vengono ricostruite le distribuzioni? Abbiamo 3 strategie

Global:

- Ricostruisci per ogni attributo una volta all'inizio.
- Costruisci l'albero di decisione utilizzando i dati ricostruiti

ByClass:

- Prima dividi i dati di addestramento,
- ricostruisci ogni classe separatamente.
- Costruisci l'albero di decisione usando i dati ricostruiti

Local:

- Prima suddividi i dati di training
- Ricostruisci ogni classe separatamente
- Ricostruisci su ciascun nodo mentre costruisci l'albero

Per verificare il funzionamento dell'algoritmo si possono usare le misure di accuracy.

Privacy metric

Dal nome si può capire che viene usate per capire il livello di privacy.

Se, da dati perturbati, si può stimare che il valore originale x sia compreso tra [x1, x2] con confidenza c%, quindi la riservatezza al livello di confidenza c% è correlata con l'ampiezza dell'intervallo x2 - x1.

	Confidence		
	50%	95%	99.9%
Discretization	0.5 x W	0.95 x W	0.999 x W
Uniform	0.5 x 2α	0.95 x 2α	0.999 x 2α
Gaussian	1.34 x σ	3.92 x σ	6.8 x σ

Example

- Salary 20K - 150K
- 95% Confidence
- 50% Privacy in Uniform
- 2α = 0.5*130K / 0.95 = 68K

Considerando l'esempio appena descritto, ci si trova nella colonna della confidence al 95%. Considerato il valore di Privacy che vogliamo mantenere del 50% lo si va a moltiplicare per l'ampiezza dell'intervallo 150k – 20k = 130k, il risultato poi sarà diviso per 0.95 in modo da ottenere 2α che sarà 68k.

Problemi

- Se vogliamo valori alti di privacy, la discretizzazione fornirà un modello povero cioè scarsamente predittivo.
- La gaussiana invece fornisce un livello di privacy più alto ma ad alti livelli di confidence.

Quantificazione della privacy

Spiegazione: se è possibile stimare il valore originale con confidenza c% nell'intervallo [α1, α2], allora l'ampiezza dell'intervallo (α2- α1) definisce la quantità di privacy al livello di confidenza c%.

Es Ampiezza intervallo 2α

- Il livello di confidence del 50% garantisce un livello di privacy α.
- Il livello di confidence del 100% garantisce un livello di privacy 2α

Data Separation

I dati vengono mantenuti solo da parti trusted e non devono essere condivisi in modo da poterne fare un utilizzo sicuro.

La soluzione per attuare ciò è un approccio **Secure Multiparty Computation**.

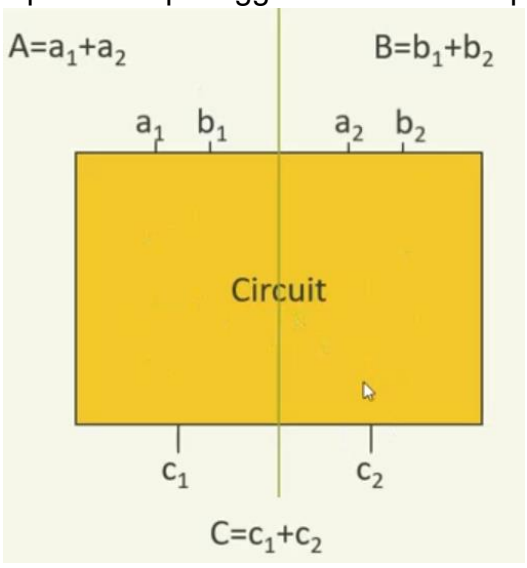
L'obiettivo della SMC è quello di calcolare una funzione che dia un risultato attendibile/fedele solo quando ogni parte abbia una porzione dell'input. L'idea è quella di usare in modo combinato le parti di input per avere poi il valore reale.

Es. Problema del milionario

Ci sono due persone che vogliono sapere chi è la più ricca, ma senza divulgare il loro patrimonio. Questo calcolo può essere effettuato in modo sicuro se la funzione viene rappresentata come un circuito.

Come funziona?

Ogni parte fornisce il suo input al circuito. A questo input viene poi aggiunto del rumore per distorcere il valore fornito. Il valore viene poi passato al circuito, suddiviso nei valori iniziali che saranno poi raggruppati: come si può vedere dall'immagine nessuna delle due parti avrà accesso all'informazione completa. Successivamente sarà calcolato C = c1 + c2, dove c1, c2 sono calcolati nella rispettiva porzione di circuito indipendente. Questo avviene tramite una XOR che permette di effettuare la somma localmente. Un'altra possibilità è quella di usare un AND gate: si usa un protocollo **Oblivious transfer** che permette all'altra parte di ottenere il valore corretto mediante una tabella di



verità. Il calcolo avviene secondo la seguente tabella:

value of (a_2,b_2)	(0,0)	(0,1)	(1,0)	(1,1)
OT-input	1	2	3	4
value of output	$c_1+a_1b_1$	$c_1+a_1(b_1+1)$	$c_1+(a_1+1)b_1$	$c_1+(a_1+1)(b_1+1)$

Oblivious transfer

- What is it?
  - *A* has inputs a_i
  - *B* makes choice
  - *A* doesn't know choice, *B* only sees chosen value.
- How?
  - *A* sends public key *p* to B
  - B selects 4 random values *b*
    - encrypts (only) b_{choice} with f_p, sends all to *A*
  - *A* decrypts all with private key, sends to B: $c_i = a_i \oplus e(f_p^{-1}(b_i))$
  - B outputs $c_{choice} \oplus e(b_{choice}) = a_{choice} \oplus e(f_p^{-1}(f_p(b_{choice}))) \oplus e(b_{choice})$

N.B.: "e" indica la fase di encription.

Il vantaggio è che una parte effettua le scelte e l'altra gli input.

Il SMC può essere usato per costruire dei decision tree privacy preserving.

Presupposti: si effettua un partizionamento orizzontale a 2 parti, in cui ogni parte ha lo stesso schema e conosce il set di attributi, mentre le tuple del sistema sono divise tra prima e seconda parte: ognuna non conosce il contenuto dell'altra.

L'idea è di apprendere un classificatore decision tree chiamato ID3 che sarà anche il modello di riferimento, a cui aggiungeremo le funzionalità di SMC.

Assunzioni/Limiti dell'algoritmo

- Il protocollo usa il modello semi-onesto: cioè segue fedelmente il protocollo, ma tiene traccia di tutti i suoi calcoli intermedi.
- Viene considerato solo casi a due parti trusted: l'estensione a più parti non è banale.
- Calcola un'approssimazione ID3 al posto del classificatore reale.
  - Protocollo per il calcolo dell'$ID3_\delta \in ID3_\delta$ (si cerca un approx tra tutte le δ-approx possibili; δ = grado di approx di ID3).
  - δ - approssimazione of ID3
  - δ ha implicazioni sull'efficienza
- Funziona solo con attributi categorici.

ID3 Decision tree classifier

Abbiamo 3 parametri:

- R – set di attributi
- C – attributo di classe, var target che vogliamo predire.
- T – set di transactions, cioè tuple descritte dall'insieme R.

Procedimento:

1. Calcola l'entropia di ogni attributo utilizzando l'insieme di dati T. Se si ottiene un'entropia alta, allora l'attributo R non discrimina sufficientemente per l'attributo C. Al contrario R è un buon predittore per la classe C;
2. Dividi l'insieme T in dei sottoinsiemi utilizzando l'attributo per cui l'entropia risultante (dopo la divisione) è minima (o, equivalentemente, il guadagno di informazione è massimo) e quindi ottengo una distribuzione più pulita delle classi nelle partizioni;
3. Crea un nodo dell'albero decisionale contenente quell'attributo;
4. Ripeti i passaggi sui sottoinsiemi utilizzando gli attributi rimanenti fin quando non raggiungo nodi puliti o finiscono le possibilità di split.

Privacy Preserving ID3

Step 1 (si associa la var di class al nodo foglia): Se R è vuoto, restituisce un nodo foglia con il valore di classe assegnato alla maggior parte delle transazioni in T, quindi significa che se arrivo al nodo foglia posso fare la predizione.

Da considerare che il set di attributi è pubblico, quindi entrambe le parti sanno se R è vuoto.

Invece non sono note le porzioni del dataset: bisogna eseguire il protocollo Yao in cui viene usato l'approccio SMC per calcolare le seguenti funzionalità:

- Input (| T1 (c1) |, ..., | T1 (cL) |), (| T2 (c1) |, ..., | T2 (cL) |) che rappresenta la cardinalità delle tuple T1, T2 della classe c1 fino a cL.
- Output i (var di classe) dove | T1 (ci) | + | T2 (ci) | è il più grande;

Step 2: Se T è costituito da transazioni che hanno lo stesso valore c per l'attributo di classe, restituisce un nodo foglia con il valore c.

- I nodi foglia con più di una classe (nel set di transazioni), sono rappresentati con un simbolo fisso diverso da ci,
- Si forzano le parti a inserire il simbolo fisso o ci
- Controllare l'uguaglianza per decidere se nel nodo foglia prevale la classe ci
- Approcci vari per il controllo dell'uguaglianza (anche SMC per preservare la privacy di alcuni dati).

Step 3: (a) determina l'attributo che classifica meglio le transazioni in T cioè l'attributo da usare per lo split: ipotizziamo sia A (essenzialmente calcolato x * (ln x)).

(b, c) Chiamare in modo ricorsivo ID3δ (cioè la sua approx) per gli attributi rimanenti sugli insiemi di transazioni T (a1), ..., T (am) dove a1, ..., am sono i valori dell'attributo A. Poiché i risultati dello step 3 (a) e il i valori degli attributi sono pubblici, entrambe le parti possono partizionare individualmente il database e preparare i propri input per le chiamate ricorsive.

Il problema rimane quello di calcolare x*ln(x).

Si usa il protocollo di Yao per ottenere un'approx grezza di ln(x) in quanto non si può usare la SMC per il calcolo. Quindi si espande e calcola la serie di taylor di ln(1 + ε) per raffinare il calcolo.

Conclusioni

I vincoli sulla privacy e sulla sicurezza possono essere impedimenti al mining di dati

- Problemi con l'accesso ai dati
- Restrizioni sulla condivisione
- Limitazioni sull'uso dei risultati

Soluzioni tecniche possibili:

- Randomizzazione / scambio di dati non impedisce l'apprendimento di buoni modelli
- Non abbiamo bisogno di condividere i dati per apprendere i risultati globali

Non completamente sicuro

- Query multiple

Lezione 10

Differential Privacy

DIFFERENTIAL PRIVACY (Lezione 10)
Uno dei approcci allo stato dell'arte nella privacy e data protection nei database statistici e anche nel machine learning/data analysis, molti di questi algoritmi sono stati riproposti nella variente differential privacy.
Nuovo paradigma: ci stiamo spostando da una visione di anonimizzazione che abbiamo introdotto nei metodi precedenti ad un paradigma completamente nuovo, criticato ma rimane un paradigma valido nell'ottica nelle privacy reservation.

Nuovo paradigma sulla privacy
Come rendere i dati riservati ampiamente disponibili per avere una occurata data analysis, senza ricorrere a camere bianche, dati contratti di utilizzo, piani di protezione dei dati o restrizioni visualizzazioni?
La Legge fondamentale per il recupero delle informazioni afferma che le risposte troppo accurate a troppe query distruggeranno privacy. Se anche una signola query non consente di sapere molto riguardo su un certo dataset, se io posso eseguire diverse query ad un certo punto sarà possibile una divulgazione dei fatti che non volevo che fossero divulgati.

Come può essere interpretato questo nuovo paradigma poi modellato dalla differential planning?
È possibile riuscire ad utilizzare delle info riguardo una popolazione in maniera utile senza imparare nulla di più sui singoli individui.

Esempio
▪ un database medico può insegnarci che il fumo provoca il cancro,
questo database incide sul punto di vista di una compagnia assicurativa sui costi medici dei fumatori
▪ i suoi premi assicurativi (il suo prezzo) possono aumentare, se l'assicuratore sa che lui fuma
▪ il fumatore può anche trarne vantaggio: apprende i suoi rischi per la salute, entra un programma per smettere di fumare
▪ La privacy del fumatore è stata compromessa?

Queste info prese da un database medico ha impatto sulla società.

Se questo individuo non avesse partecipato allo studio avremmo saputo qualcosa in più su di lui ? Forse NO.
L'impatto sul fumatore è lo stesso indipendentemente dal fatto che fosse o meno nello studio. Sono le conclusioni raggiunte nello studio che influenzano il fumatore, non la sua presenza o assenza nel set di dati (lo studio è stato effettuato su un insieme di individui). La sua partecipazione al dataset non ha nessun impatto sulle conseguenze che i risultati di questo studio hanno sulla sua persona e sulla sua salute

Differential privacy (è una definizione, non un algoritmo)
garantisce che vengano raggiunte le stesse conclusioni, ad esempio il fumo che provoca il cancro, indipendentemente dal fatto che un individuo partecipi o meno allo studio, indipendentemente che un individuo abbia optato nell'essere escluso o incluso dallo studio.

In particolare, garantisce che qualsiasi sequenza di risultati (risposte alle query) abbia "essenzialmente" la stessa probabilità di verificarsi, indipendentemente dalla presenza o assenza di qualsiasi individuo. L'output non permette di capire se l'individuo è presente o meno nel dataset.

Qual è l'intuizione dietro il Differential Privacy?
Assumiao un individuo/curatore (che amministra il database) che possiede i dati degli individui in un database D (con n righe) → Obbiettivo: proteggere ogni signola tupla permettendo l'utilizzo dal punto di vista statistico per intero (non è possibile individuare singoli attributi di singoli individui).

DP può essere ottenuto in maniera offline: creo database sintetico/igenizzato che rispetta la DP.
Oppure interattivo(online): database mantenuto in maniera chiara ma viene modificato l'output della query.

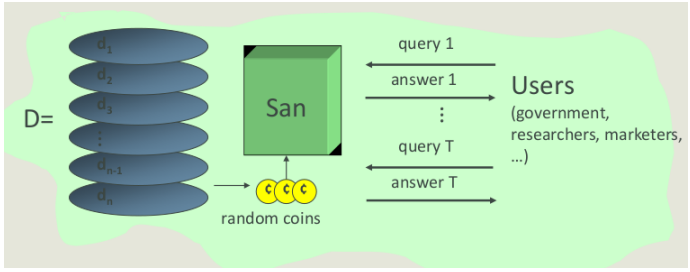
Esempio: (come sporcare il risultato della query → DP online, risposta randomizzata)
Un primo esempio di privacy mediante un processo randomizzato è la risposta randomizzata, una tecnica sviluppata nelle scienze sociali per raccogliere informazioni statistiche sull'imbarazzo o comportamento illegale, catturato da una proprietà P
▪ Ai partecipanti allo studio viene chiesto di riferire o meno
avere la proprietà P come segue:
Lanciare una moneta.
Se esce croce, rispondi in modo veritiero.
Se esce testa, lancia una seconda moneta e rispondi "Sì" se esce testa e "No" se esce croce

Algoritmo randomizzato
Dato un set discreto B, il simplex di probabilità su B,
denotato Δ(B) è definito così:
SPOILER: Ad ogni elemento di B va ad associare la sua probabilità

$$\Delta(B) = \left\{ x \in \mathbb{R}^{|B|} \mid \forall i, x_i \geq 0 \wedge \sum_{i=1}^{|B|} x_i = 1 \right\}$$

In generale, un algoritmo randomizzato con dominio A e intervallo (discreto) B sarà associato a una mappatura da A al simplex di probabilità su B, indicato con Δ (B).
Più formalmente:
Un algoritmo randomizzato M con dominio A e intervallo discreto B è associato a una mappatura M: A → Δ (B).
All'ingresso a in A, l'algoritmo M genera M (a) = b con probabilità $(M(a))_b$ per ogni b in B.

Setting base:



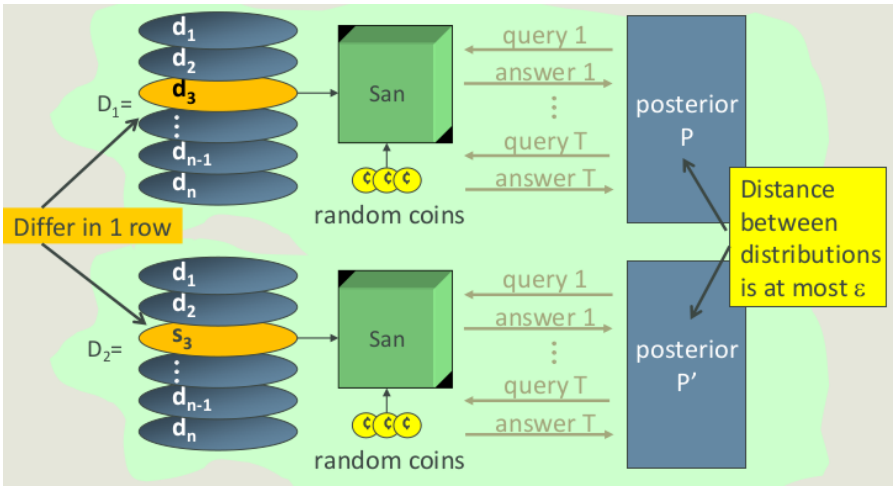
Distanza tra database
Norma l1: $||D1||_1$= D (number of record of D)
The l 1 distance between two databases D1 and D2 is $||D1 - D2||_1$

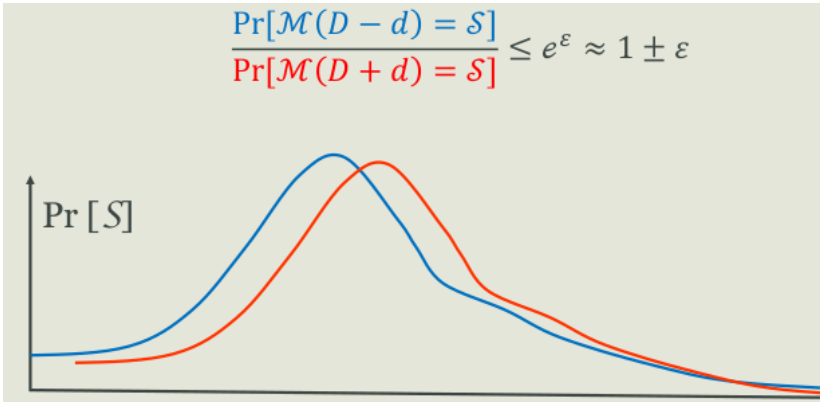
Ɛ-differential privacy (def)
Un algoritmo randomizzato M con dominio D è Ɛ-differenzially private se per tutti gli S contenuti in Range(M) e per tutti D1, D2 contenuti in D (solo D2 contenuto in D) tale che $\|D\ 1 - D\ 2\ \|_1 \leq 1$:

$$\frac{\Pr[\mathcal{M}(D_1) \in \mathcal{S}]}{\Pr[\mathcal{M}(D_2) \in \mathcal{S}]} \leq e^{\varepsilon}$$

La prob che la risposta ottenuta dall'algoritmo randomizzato provenga da D1 diviso prob che la risposta ottenuta dall'algoritmo randomizzato provenga da D2 sia minure uguale a $e^{Ɛ}$ (Il più possibile vicino ad 1). Non riusciamo a distinguere se la risposta è prodotta da un dataset o l'altro (Indistinguishability). Devo trovare Ɛ affinché il rapporto sia 1 (privacy budget).

Se l'algoritmo rispetta la Ɛ DP la distanza tra p e p^1 sarà al massimo di Ɛ.





Comportamento desiderato: le distribuzioni devono essere molto vicine e simili tra loro

Altra definizione del differential privacy:
Un algoritmo randomizzato M con dominio D è (Ɛ,δ)-differenzially private se per tutti gli S contenuti in Range(M) e per tutti D1, D2 contenuti in D (solo D2 contenuto in D) tale che $\|D\ 1 - D\ 2\ \|_1 \leq 1$:

$$\Pr[\mathcal{M}(D_1) \in \mathcal{S}] \leq e^{\varepsilon} \Pr[\mathcal{M}(D_2) \in \mathcal{S}] + \delta$$

Differenze definizioni:

▪ (Ɛ, 0) - la DF garantisce che, per ogni esecuzione del meccanismo M (D), l'output osservato sia (quasi) altrettanto probabile che venga osservato su ogni database adiacente, contemporaneamente
▪(Ɛ,δ) - la DF dice che per ogni coppia di database vicini D1, D2 (DB che differiscono da una sola tupla), è estremamente improbabile che il valore osservato M (D1) sia molto più o meno probabile che venga generato quando il database è D 1 rispetto a quando il database è D2. Nuovo grado di incertezza introdotto da δ

Esiste un DF che si applica a gruppi: la differenza tra D1 e D2 è di k tuple, il rapporto della prima def adesso è minore o uguale a $e^{k\varepsilon}$.

$$\frac{\Pr[\mathcal{M}(D_1) \in \mathcal{S}]}{\Pr[\mathcal{M}(D_2) \in \mathcal{S}]} \leq e^{k\varepsilon}$$

Esempio: utile per questionari che includono più membri della stessa famiglia: famiglia con 4 membri e usiamo una group DP che tutelano la privacy di famiglie di 4 membri.

Garanzie DF
▪ La DF promette di proteggere le persone da qualsiasi danno aggiuntivo che potrebbero subire a causa del fatto che i loro dati si trovavano nel database privato D che non avrebbero affrontato se i loro dati non facessero parte di D
▪ Sebbene gli individui possano effettivamente subire danni una volta che i risultati M(D) di un meccanismo differenzialmente privato M sono stati resi noti, la privacy differenziale promette che la probabilità di danno non è stata aumentata in modo significativo dalla scelta di partecipare
DF NON garantisce
▪ La privacy differenziale non garantisce che ciò che si ritiene essere il proprio segreto rimarrà segreto
▪ È molto probabile che le conclusioni tratte dall'indagine possano riflettere informazioni statistiche su un individuo
▪ Se il sondaggio afferma che specifici attributi privati sono fortemente correlati con attributi osservabili pubblicamente, ciò non costituisce una violazione della DF → se fumo la gente sa che fumo dato che mi vede o i vestiti puzzano di fumo quindi in qualche modo non è un segreto, ma una persona sa che se fumo ho più probabilità di contrarre delle malattie e quindi di conseguenza pur non sapendo la probabilità che io mi ammali sa che ho un rischio molto alto di contrarre malattie cardiocircolatorie.
▪ la stessa correlazione sarebbe osservata con quasi la stessa probabilità indipendentemente dalla presenza o assenza di qualsiasi intervistato.

Risposta randomizzata Esempio base
Consideriamo un semplice meccanismo di risposta randomizzato
▪ Ai partecipanti allo studio viene richiesto di segnalare se hanno o meno una determinata proprietà P e di rispondere come segue:

- Lanciare una moneta
- Se esce croce, rispondi in modo veritiero
- Se esce testa, lancia una seconda moneta e rispondi "Sì" se esce testa e "No" se la coda

La versione della risposta randomizzata sopra descritta è ln (3) -differenzialmente privata
Proof:
Perdiamo il caso che Pr [Risposta = Sì | Verità = Sì] = ¾ (sarebbe 1/2+(1/2*1/2))
In particolare, quando la verità è "Sì" il risultato sarà "Sì" se la prima moneta esce croce (probabilità 1/2) o la prima e la seconda testa si presentano (probabilità ¼))

Invece, Pr [Response = Sì | Verità = No] = 1/4 (il primo si avvicina alla testa e il secondo arriva la coda → probabilità ¼)

$$\frac{\Pr[Response = Yes|Truth = Yes]}{\Pr[Response = Yes|Truth = No]} = \frac{3/4}{1/4} = 3 = e^{\varepsilon}$$

Sensitivity di una query function (altro tassello per la progettazione di algoritmi che siano differenzialmente privati)

Usiamo query numeriche, cioè funzioni
f: D ⟶ R^k
che sono uno dei tipi più fondamentali di query di database che associano database a k numeri reali

The ℓ_1 **sensitivity** of a function $f: \mathcal{D} \longrightarrow \mathbb{R}^k$ is

$$\Delta f = \max_{\substack{D_1, D_2 \in \mathcal{D} \\ \|D_1 - D_2\|_1 = 1}} \|f(D_1) - f(D_2)\|_1$$

Stiamo calcolando quanto vale la differenza di una certa funzione su un DB quando io calcolo questa funzione su Databases che differiscono di una singola tupla. Qual è la variazione che io mi aspetto di avere sulla funzione quando do in input a questa funzione due db che differiscono di una tupla. Il massimo di questa variazione è la nostra deltaF.
La sensibilità l_1 cattura la magnitudine/l'ampiezza secondo cui un dato singolo può cambiare il risultato della funzione f nel caso pessimo

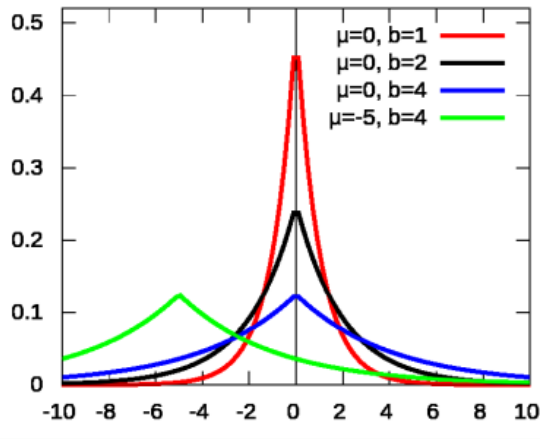
Come si passa dal calcolo della sensitività alla randomizzazione?
La sensibilità ci da un upper bound su quanto dobbiamo perturbare il suo output per preservare la privacy → se io riesco a scrivere un alg rand che mi perturba il dataset tenendo conto della sensitività della funzione io riesco al contempo di garantire la privacy ma anche a fre un modo che il calcolo dia un risultato simile a quello reale. Una distribuzione che porta automaticamente alla DP è una distribuzione di rumore secondo la distribuzione di Laplace.

Abbiamo una altissima probabilità di avere valori vicini allo zero rispetto ad avere valori 'lontani' da zero.

We say that a random variable is distributed according to Laplace distribution centered in μ with scale b ($X \sim Lap(\mu, b)$)

$$p(x) = \frac{e^{-\frac{|x-\mu|}{b}}}{2b}$$

- Variance: $\sigma^2 = 2b^2$
- We will consider $X \sim Lap(b)$



Dobbiamo dimensionare b in modo tale che il nostro rumore distribuito con Laplace garantisca effettivamente la DP.
Immaginiamo una funzione f: D ⟶ R^k, il meccanismo di Laplace è definito come

$$\mathcal{M}_{\mathcal{L}}(D, f(\cdot), \varepsilon) = f(D) + Y_1, \ldots, Y_k$$

where Y_i are i.i.d. random variables drawn from $\mathrm{Lap}(\Delta f/\varepsilon)$, with $\Delta f = \max_{D_1, D_2 \in \mathcal{D}} \|f(D_1) - f(D_2)\|_1$ $\|D_1 - D_2\|_1 = 1$

Queste Y_i vanno a sporcare una delle conponenti (la iesima) dell'output della funzione f.

Esiste un teorema che dice che il meccanismo di laplace (la nostra M) preserva la ε-DP.
DeltaF, la sensibilità, piccoli = minore distorsione.
Se abbiamo ε (privacy budget) piccolo, per aumentare la privacy dobbiamo aumentare la distorsione.
Se io volgio aumentare la privacy devo aggiungere più rumore.

Esempi:

1. Query di conteggio → quanti respondents nel database sono femmine?
2. Query di istogramma →quanti respondents ricadono nelle categoria data?
3. Query per calcolo di medie
4. Classificatori → predire il fatto che una persona sia fidabile, basandosi sulle proprie caratteristiche

1. Query conteggio
Il conteggio è una primitiva estremamente potente: acquisisce tutto ciò che è possibile apprendere nel modello di apprendimento delle query statistiche, nonché molte attività di datamining standard e statistiche di base
▪ La sensibilità di una query di conteggio è 1 (l'aggiunta o la cancellazione di un singolo individuo può modificare un conteggio al massimo 1)
▪ La privacy ε-differenziale può essere ottenuta aggiungendo il rumore tratto da una distribuzione di Lap(1 / ε) NB 1 è il deltaF e 1/ε è la nostra b
▪ La distorsione o errore previsti è 1 / ε, indipendentemente dalla dimensione del database.

Conteggi multipli
▪ Un fisso e arbitrario elenco di m query di conteggio può essere visto come una query a valori vettoriali
▪ Il caso peggiore legato alla sensibilità di questa query a valori vettoriali è m, poiché un singolo individuo potrebbe cambiare ogni conteggio
▪ In questo caso è possibile ottenere una privacy ε differenziale aggiungendo rumore ridimensionato a m / ε alla risposta vera a ciascuna query

2. Query di istogramma
È un caso molto più comune.
▪ Nel caso speciale (ma comune) in cui le query (I risultati delle query) sono strutturalmente disgiunte, non dobbiamo necessariamente lasciare che il rumore si riduca con il numero di query → Quante persone sono impiegati, quante sono studenti.. questi conteggi, il fatto di aggiungere o togliere una persona di questo conteggio va a variare una delle categoria di questa query multipla (se io aggiungo uno studente aumentero di una unità il conteggio degli studenti e le altre categorie non vengono toccate)
▪ Nella query dell'istogramma, l'universo D è partizionato in celle e la query chiede quanti elementi del database si trovano in ciascuna delle celle
▪ Le celle sono disgiunte

- l'aggiunta o la rimozione di un singolo elemento del database può influire sul conteggio esattamente in una cella
- La differenza rispetto a quella cella è limitata da 1

- Le query dell'istogramma hanno la sensibilità 1 (caso conteggio) e si può rispondere aggiungendo rumore Laplaciano (1 /ε) al conteggio reale in ogni cella

3. Query media
Conteggio su una quantità uniformemente distribuita su un intervallo [alfa,beta], immaginiamo un db con N oggetti (esemoio), la sensibilità comune della media è (beta-alfa)/n → aggiungere o tolgiere un elemento può far variare la media al massimo di (beta-alfa)/n
Di conseguenza la ε-DP può essere raggiunta aggiungendo rumore Laplaciano ((beta-alfa)/n ε)

Caso su sensitivity con f specifica come media, conteggio, non funziona.
F di **utilità.**
Più è alto il valore U (utilità) piu i risultati della funzione sono effettivamente di utilità per lo scopo prefissato.
Laplace a volte anche se aggiungo pochissimo rumore alla funzione f(D) questo può corrispondere ad una modifica elevata della funzione di utilità.
Laplace solo query numeriche → vogliamo applicare un meccanismo a qualsiasi query e con un risultato di tipo R (numeri, boolean, categorico..)

Quando non è possibile calcolare una funzione di utilità che vada dai res della funzione f ai Reali affettivamente Laplce non è più utilizzabile.

ES:

Il prezzo che massimizza il ricavato è 3.01

Consider the database D, in which each record contains the name of a possible buyer of a product and the maximum amount he is willing to spend to buy such a product: the goal is to set the price that maximizes the income of the seller.

Customer	Max Amount
A	1
B	1
C	3.01

The **optimal price** is the result of the function $f: \mathcal{D} \longrightarrow \mathbb{R}$ defined as

$$f(D) = r \text{ such that } r \text{ maximizes } u,$$

where $u: \mathcal{D} \times \mathbb{R} \longrightarrow \mathbb{R}$ is the utility function defined as

$$u(D, r) = r * |\{d \in D \mid amount(d) \geq r\}|.$$

Applichiamo Laplace

The optimal price for database D is $f(D) = 3.01$.

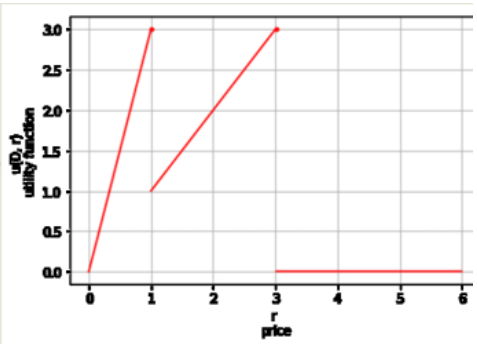
Suppose $\mathcal{M}(f, D) = 3.02$.

The mechanism adds only a little amount of noise:
$|f(D) - \mathcal{M}(f, D)| = 0.01$

But our earning is destroyed, because

$u(D, f(D))$ = 3.01 (maximum earning possible), while

$u(D, \mathcal{M}(f, D))$ = 0.



M(f,D) è il db+Rumore

Qualsiasi rumore distrugge tutto (u=0)

Come faccio DP in questi caso?
Spostare il problema sul calcolo di sensitività su u e non su f.
Noi desideriamo un output M(f,D) tale per cui l'u calcolata in funzione di db DB e di M(f,D) e l'u calcolata sullo stesso db e per la funzione non sporcata f(D) siano il più possibile vicini. Come viene calcolata la sensitività di u:

$$\Delta u := \max_{r \in \mathcal{R}} \max_{\substack{D_1, D_2 \in \mathcal{D} \\ \|D_1 - D_2\|_1 = 1}} |u(D_1, r) - u(D_2, r)|.$$

NB D1 e D2 differiscono di una tupla (tutti i possibili db che differiscono di una tupla). R risultato di f.
Possiamo sfruttare questa sensitività per applicarla ad un meccanismo alternativo a Laplace (ancora più generico di Laplace → query non numeriche): **Meccanismo Esponenziale:**

Given any utility function $u: \mathcal{D} \times \mathcal{R} \longrightarrow \mathbb{R}$, the **exponential mechanism** is defined as

$$\mathcal{M}_E(D, u, \varepsilon) = R$$

where R is a random variable with values in $\mathbb{R}$ such that

$$P(R = r) = e^{\frac{\varepsilon\, u(D,r)}{2\Delta u}}.$$

Produce un output r appartenente allo spazio delle risposte della nostra query, con una prob proporzionale a u(D,r), cioè più è alta la f di utilità più è probabile che sia 0proprio r il valore che viene mandato in output. Non succede che un elemento r in output in un meccanismo esponenziale ha uno score di utilità inferiore alla massima utilità.
Thm: meccanismo exp preserva la ε-DP.

Possibile **combinare** diversi meccanismi, spesso necessario (voglio usare più volte query) → garantisco ancora la DP, dobbiamo lavorare molto sui ε e delta per ottenere risultati soddisfacenti
Thm: se noi usiamo una funzione di composizione che sfrutta i meccanismi di DP quello che otteniamo rimane differezialmente privato, valido sia per computazione indipendenti che sequenziali.
Thm: in caso di computazioni/composizioni **parallele** la DP è garantita

4. Classificatore lineare
La CL consiste nel trovare un vettore w che separa positivi e negativi (2 categorie, per più categorie parliamo di iperpiani e vettori a più dimensioni), consideriamo una versione

parametrica del vettore, bisogna trovare i parametri giusti in modo che il vettore separi per bene le due classi.
Come è possibile ottenere un risultato privato sfruttando esempi di CL?
Empirical Risk Minimization: dataset costituito da esempi di tipo d_i e c_i variabili di classe. Sarebbe un CL basato sulla seguente funzione:

Si può dimostrare che se la norma del vettore è <=1 e se la loss è una funzione con crescità limitata allora qualsiasi D1 e D2 vale il seguente calcolo del valore di sensitività

$$f(D) = \operatorname{argmin}_w \frac{1}{2}\lambda\|w\|^2 + \frac{1}{n}\sum_{i=1}^{n} L(c_i w^T d_i)$$

Regularizer (model complexity) **Risk** (training error)

Theorem

If $\|d_i\| \leq 1$ and L is 1-Lipschiz, then, for any D_1, D_2 with $\|D_1 - D_2\|_1 \leq 1$,

$$\|f(D_1) - f(D_2)\|_2 \leq \frac{2}{\lambda n}$$

Come privatizzare ERM
Possiamo perturbare l'output calcolando il res esatto f(D) e aggiungo rumore basato sulla sensitività (es con Laplace), oppure si può mandare in out un valore con una probabilità proporzionale con la funzione obiettivo (utility function), oppure si cambia la f obiettivo introducendo il rumore e alla fine il risultato deve minimizzare la funzione ob iettivo perturbata.
Esempio:

$$\frac{1}{2}\lambda\|w\|^2 + \frac{1}{n}\sum_{i=1}^{n} L(c_i w^T d_i) + \frac{1}{n} b^T w$$

Regularizer (model complexity) **Risk** (training error) **Perturbation** (privacy)

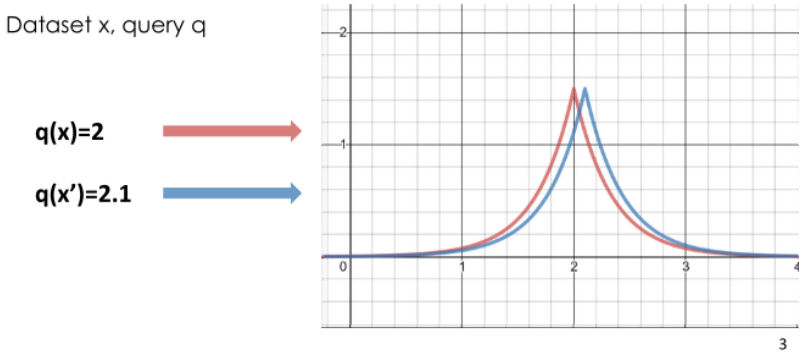
b vettore di rumore calcolato sulla base della sensitività della funzione.

Possiamo utilizzare diverse funzioni di Loss: dalla Logistic loss (private Logistic Regression) alla Huber Loss (Support Vector Machines)

Commenti
Laplace e exp non sono gli unici meccanismi: ci sono quello Gaussiano, sample eggregate, objective perturbation (se un problema di ottimizzazione è convesso posso renderlo privato).
La scelta di ε è un problema, ci sono pubblicazioni che ne parlano.

DP usata da all'ufficio del censimento degli USA (quali sono i percorsi di chi pendola per motivi di lavoro), Google (software di telemetria, statistiche storiche del traffico), Apple con Siri (ma usava ε molto alti → la DP era rotta, differenza tra distribuzioni troppo alte)

SEMINARIO FINALE
Differential privacy con sensitività locale
• La differential privacy è una tecnica che permette di calcolare il valore di una query q su un dataset x senza rivelare informazioni sui singoli record che compongono x.
• Lo scopo della differential privacy è quello di permettere di divulgare analisi accurate eseguite su dati sensibili, senza compromettere la privacy della persone a cui i dati sensibili appartengono.
Rilasciare il risultato di query aggregate (più query), ma non rilasciare il vero risultato ma un valore molto simile (distorsione dev'essere piccola per garantirmi che il res finale è simile a quello vero, ma allo stesso tempo dev'essere sufficientemente grande per garantirmi la privacy).



X e x^1 sono due dataset diversi ma simili (adiacenti, noi, invece, consideriamo DB che differiscono di una tupla -> coi garantiamo la DF solo su quella tupla).

ε-DF
• $\mathcal{X}^n$: insieme di tutti i dataset con n record, da un data universe $\mathcal{X}$
• $Q = \{f: \mathcal{X}n \rightarrow \mathbb{R}\}$ query a valori reali
Un meccanismo $\mathcal{M}$: $\mathcal{X}^n \times Q \longrightarrow \mathbb{R}$ rispetta la $\mathcal{E}$-differential privacy se, per ogni coppia di dataset adiacenti $x, x' \in \mathcal{X}n$ e per ogni $r \in \mathbb{R}$, si ha:

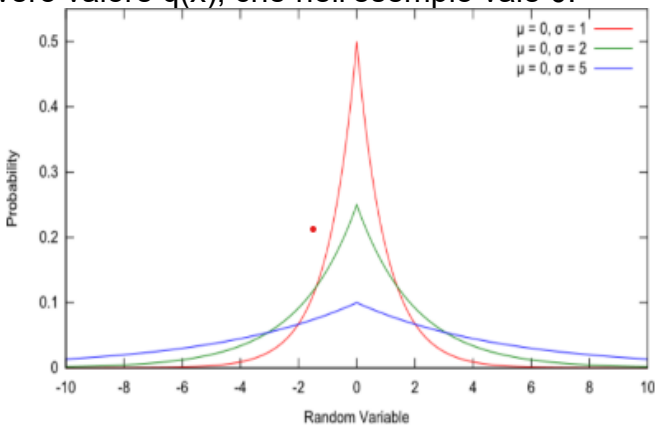
$$e^{-\varepsilon} \leq \frac{P(\mathcal{M}(x,q) = r)}{P(\mathcal{M}(x',q) = r)} \leq e^{\varepsilon}$$

la $\mathcal{E}$ decide la distanza tra le due distribuzioni di prob.
Attenzione a non considerare solo la privacy ma anche l'accuratezza del nostro meccanismo (è facile ottenere meccanismi che rispettano la DF ma che non servono a niente).

Come fare a migliorare l'accuratezza di Laplace?
Il primo sviluppato, il più sviluppato e il più semplice.

Dato un dataset x e una query q, il meccanismo di Laplace restituisce un valore estratto dalla distribuzione di Laplace centrata nel vero valore q(x), che nell'esempio vale 0.

$$\mathcal{M}_{Lap}: (x, q) \longmapsto Lap(q(x), GS_q/\mathcal{E})$$



Il nostro desiderio è scegliere la curva con il parametro sigma più basso possibile.

GS_q è la **sensitività globale** della funzione q, cioè la misura di quanto varia al massimo q se la calcolo in due dataset adiacenti x, x'.

$$GS_q = max_{(x,x')adiacenti}|q(x) - q(x')|$$

Il meccanismo di Laplace con parametro GS_q/ε garantisce la ε-differential privacy e, se invece scegliamo un parametro minore, non è più garantita (questa è la miglior forma che garantisce la DP).
Problemi sensitività globale:
• Alcune funzioni q hanno GSq molto grande: la distribuzione di Laplace calibrata sulla sensitività globale diventa molto piatta (quindi rischiamo di avere risultati lontanissimi da quello vero)
• Addirittura alcune funzioni hanno GSq = ∞, quindi il meccanismo di Laplace non può essere utilizzato.
• Può essere molto complicato calcolare analiticamente GS_q

Nuova definizione con **(ε, δ)-DP**

Un meccanismo $\mathcal{M}: \mathcal{X}^n \times Q \longrightarrow \mathbb{R}$ rispetta la $(\mathcal{E}, \delta)$-differential privacy se, per ogni coppia di dataset adiacenti $x, x' \in \mathcal{X}^n$ e per ogni $r \in \mathbb{R}$, si ha

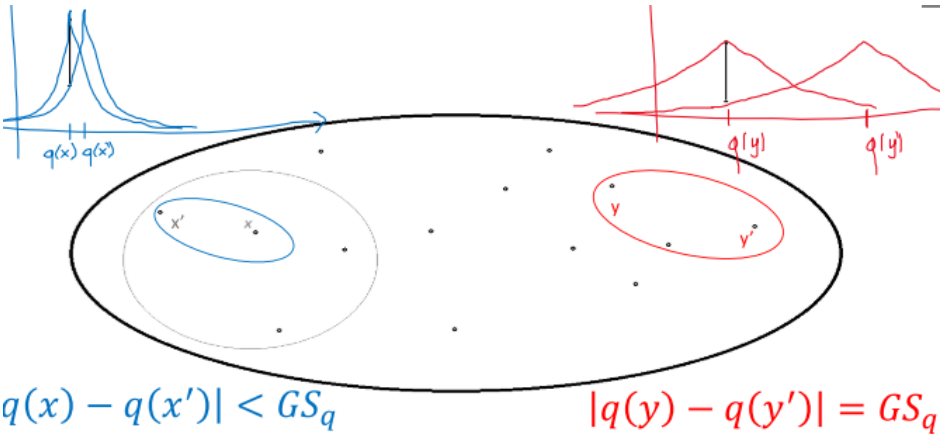
$$P(\mathcal{M}(x, q) = r) \leq e^{\mathcal{E}} \cdot P(\mathcal{M}(x', q) = r) + \delta$$

TEOREMA: $\mathcal{M}$ rispetta la ($\mathcal{E}$, δ)-differential privacy se esiste un evento E con $P(E) > 1 - \delta$ tale che $\mathcal{M}$ rispetta la $\mathcal{E}$-differential privacy quando E si verifica.

Come facciamo ad usare Laplace se la sensitività globale della funzione q è troppo grande?

• La differential privacy NON richiede che sia impossibile trarre alcuna informazione sul dataset x né che sia impossibile discriminare tra x e qualunque altro dataset y.

• Di fatto noi dopo aver visto il risultato distorto della query q(x), possiamo farci un'idea di quale possa essere il dataset di partenza x anche se non possiamo discriminare tra x e un suo vicino x'.



NB: segmento nero di dx e sx è lo stesso.
Ci troviamo ad avere una distribuzione peggiore (quella rossa) di due DB y e y^1 di cui non ci interessa.
Idea: non andiamo a guardare tutte le forme di distribuzione di laplace tra tutte le coppie di dataset adiacenti ma solo localmente intorno a x.
Formalmente:

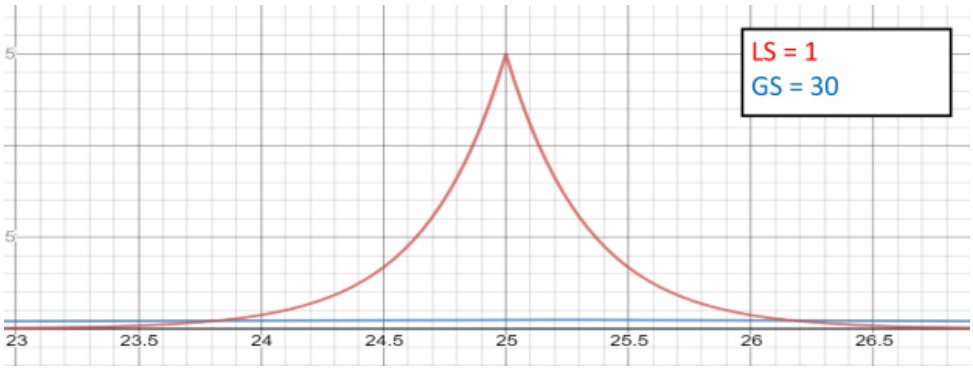
$$LS_q(x) = max_{x' \, t.c. \, (x,x')adiacenti} |q(x) - q(x')|$$

$$GS_q = max_{(y,y')adiacenti} |q(y) - q(y')|$$

GS=LS quando la coppia di dataset che da il risultato peggiore contiene x, in tutti gli altri casi LS<GS (GS è la precedente definizione di global sensivity)
Con LS al posto di GS in Laplace il meccanismo è più accurato.

ES: Mediana
Abbiamo un dataset contenente l'elenco dei voti presi da ogni studente ad un esame.
Supponiamo che gli studenti siano 5,
x = [$x1$ $x2$ $x3$ $x4$ $x5$]
dove ogni xi è un voto da 0 a 30. Supponiamo che i valori siano già stati sistemati in ordine crescente, dal più basso al più alto. La mediana di x è il valore centrale $x3$.

• Due dataset sono adiacenti se cambia un solo voto tra l'uno e l'altro. Il caso con la massima variazione si ha tra i due dataset adiacenti
x =[0 0 0 30 30]
x′ = [0 0 30 30 30]
• Mediana(x) = 0, Mediana(x') = 30 $\Longrightarrow GSmediana$ = 30
Dataset assolutamente strani
LS invece è uguale a 1: la massima variazione che possiamo ottenere tra x e i suoi adiacenti è 1.



LS su Laplace:
Primo metodo: usiamo il meccanismo di Laplace con parametro
$\sigma = LSq(x)/\varepsilon$ fisso. Cioè la forma della distribuzione di Laplace è fissa ed è decisa a partire da $LSq(x)$.
Quindi:
•$\mathcal{M}\ (x, q) = Lap\ (q\ (x)\ , \sigma)$
• $\mathcal{M}\ (x', q) = Lap\ (q\ (x')\ , \sigma)$
NB: NON rispetta la DP. Ma perchè fissarlo questo parametro σ?.

Secondo metodo: usiamo il meccanismo di Laplace con parametro $LSq(\cdot)/\varepsilon$ calcolato di volta in volta. Cioè la forma della distribuzione di Laplace varia al variare del dataset.
Quindi: • $\mathcal{M}\ (x, q) = Lap\ (q\ (x)\ , LSq(x)/\varepsilon)$
• $\mathcal{M}\ (x', q) = Lap\ (q\ (x'), LSq(x')/\varepsilon)$

NO DP, il metodo precendente costruiva due curve identiche e la forma era scelta in maniera tale da mantenere la distanza tra le due curve minore di quella cnsentiva, con questo metodo le curve non hanno più la stessa forma, la forma dipendere dalla sensitività locale del DB a partire dal quale sono calcolate e ci potrebbero essere casi dove la distanza è troppo per garantire la DP.

Lidea di usare LS per aumentare la accuratezza è una buona idea, ma non posso usarla in modo diretto come fatto tutt'ora : bisogna costruire dei meccanismi che sono un po più complessi che ci permettono di diminuire Il rumore e garantire la DP.
Ne vedremo 3 di questo tipo:

1. **Propose test release**
   • Il meccanismo di Laplace con sensitività locale fissa e uguale a $LSq(x)$, dove x è il vero dataset segreto, dà risultati privati: il problema è che in questo modo viene rivelata $LSq\ (x)$ e questo potrebbe permettermi di capire qual è il vero x (perché ad esempio è l'unico con quella sensitività locale tra i suoi vicini).
   • Idea: non usiamo direttamente $LSq(x)$ per calibrare la forma della Laplaciana ma un valore b tale che $LSq\ (x) \leq b \leq GSq$ -> in questo modo soddisfiamo la DP senza rivelare la LS.
   ES:
   Nello scenario in cui ci troviamo ci sono 2 attori:
   • C: curatore fidato del dataset segreto x
   • A: analista che vuole sapere il risultato della query q(x), senza però avere accesso a x.
   A conosce la query q e quindi conosce anche la sua sensititvità globale (molto alta, supponiamo). Quindi sa che non si fiderà del risultato del meccanismo di Laplace, se calibrato su GSq. Al contrario, sa dire qual è il valore massimo di distorsione che è disposto ad accettare per fidarsi del risultato.

   1. A propone un valore di sensitività b con cui impostare la forma della distribuzione di Laplace
   2. C calcola la distanza d di x dal più vicino dataset y avente $LSq\ (y) > b$
   Se la distanza vale 0 allora io non devo cambiare nessun record di x per avere un LS > b, in caso contrario posso calcolare in maniera DP.
   3. Se $d = 0$, risponde «non posso darti il risultato»
   4. Altrimenti, restituisce un valore estratto dalla distribuzione $Lap(q\ (x)\ , b/\varepsilon)$

Attenzione: Il rischio è che ci siano due dataset adiacenti x e x' dove uno ha sensitività minore di b e l'altro ha sensitività maggiore di b -> NO DP

Idea vincente: invece di calcolare la vera distanza d da x al più vicino dba vente LS troppo alta lui calcola questa distanza in maniera distorta (con Laplace). Introduciamo rumore anche nel test

1. A propone un valore di sensitività b con cui impostare la forma della distribuzione di Laplace
2. C calcola la distanza d di x dal più vicino dataset y avente $LSq\ (y) > b$ e la distorce con il meccanismo di Laplace: $d(\text{distorto}) = Lap(d, 1/\varepsilon)$
3. Se $d(\text{distort}) > 0$, C restituisce un valore estratto dalla distribuzione $Lap\ (q\ (x)\ , b/\varepsilon)$
4. Altrimenti C risponde «non posso darti il risultato»

Solo un caso dove non rispetto la DP: quando d=0 e C restituisce un risultato. Limitare probabilità che accada questo.
FIX modificando il test (punto 3):controllando s $d(\text{distorto}) > \ln(1/2\delta)/\varepsilon$
Adesso rispetto la DP (ε, δ)

Punti critici:
• Peggiori garanzie di privacy: ε, δ -differential privacy invece che εdifferential privacy
• L'algoritmo funziona bene se siamo in grado di calcolare efficientemente $LSq(x)$. Se q è facile da calcolare e $|\mathcal{X}|$ è piccolo, possiamo farlo in $n(\ \mathcal{X}\ - 1)$ valutazioni di q.
• L'algoritmo potrebbe non dare risposta (se la soglia alla local sensitivity proposta è molta bassa)

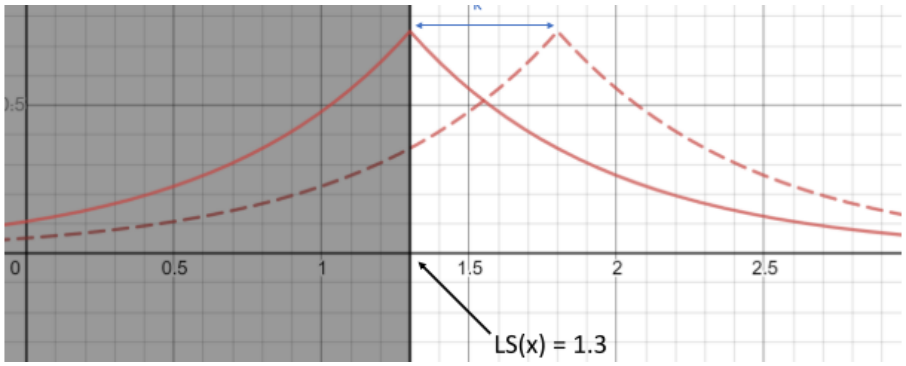
2. Privately bounding LS

• Il problema della sensitività locale di x è che rivela essa stessa informazioni su x: potremmo allora calcolarla in modo privato, con il meccanismo di Laplace!
• Infatti $LSq(x)$ è una funzione di x, quindi avrà una sua sensitività globale GS_{LSq}
• Possiamo calcolare una stima privata $l(\text{segnato})$ di $LSq(x)$ con il meccanismo di Laplace e poi usare il meccanismo di Laplace con questa sensitività per calcolare $q(\text{segnato})$

- $\hat{l} = Lap(LS_q(x), \frac{GS_{LS_q}}{\varepsilon})$
- $\hat{q} = Lap(q(x), \frac{\hat{l}}{\varepsilon})$

PROBLEMA: $\hat{l}$ ha ½ di probabilità di essere minore di $LS_q(x)$

Con questo meccanismo solo la metà delle volte garantisco la DP



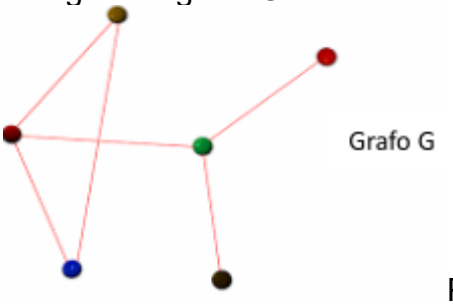
Questo è il primo passaggio dell'algoritmo.
Come prima se scelgo un k pari a un determinate valore garantisco la (ε, δ) DP

- $\hat{l} = Lap\left(LS_q(x), \frac{GS_{LS_q}}{\varepsilon}\right) + \ln\left(\frac{1}{2\delta}\right)\frac{GS_{LS_q}}{\varepsilon}$
- Restituisce $\widehat{q(x)} = Lap\left(q(x), \frac{\hat{l}}{\varepsilon}\right)$

Punti critici:
• Peggiori garanzie di privacy: ε, δ -differential privacy invece che εdifferential privacy
• L'algoritmo funziona bene se siamo in grado di calcolare efficientemente $LSq(x)$. Se q è facile da calcolare e $|\mathcal{X}|$ è piccolo, possiamo farlo in $n(\ |\mathcal{X}| - 1)$ valutazioni di q.
• l(segnato) potrebbe venire negativa (non possiamo creare una distribuzione di Laplace con varianza negativa) oppure si potrebbe avere $l(\text{segnato}) > GS$ e quindi aggiungiamo più rumore di prima!
• Questo metodo ha senso se siamo in grado di calcolare GS_{LSq} questa quantità è bassa

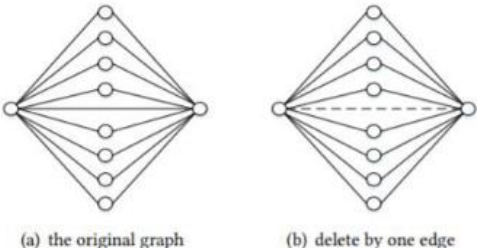
ES: nell'esempio della mediana sui voti è completamente inutilizzabile
ES2:
Sia G = G(V,E) un grafo con |V|=n nodi. Consideriamo adiacente a G un grafo G' se G e G' si differenziano per un solo arco (che c'è in G e non in G' o viceversa).

Vogliamo calcolare privatamente la funzione T(G), che conta il numero di triangoli nel grafo G.



Proviamo con laplace: calcolo il GS ovvero quanto varia il numero di triangoli al max di un grafo se aggiungo o tolgo un arco.

- $GS_T = n - 2$



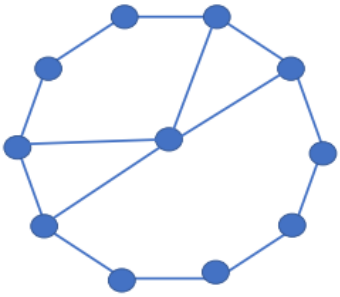
• $LS_T\ (G)$ = massimo numero di vicini comuni tra due nodi di G
• $GS_{LS\,T}$ = 1: se due nodi u e v in un grafo G hanno al massimo k vicini in comune, aggiungendo un arco posso aumentare al più di 1 il numero di nodi vicini comuni a u e v.

Caso specifico:

- n= 100
- T(G) = 2

- $GS_T = 100 - 2 = 98$
- $LS_T(G) = 1$

ESEMPIO ($\varepsilon = 1, \delta = .1$) :
- $\hat{T}(x)$ = 1,7 (PBLS)
- $\hat{T}(x)$ = 293 (LapGS)



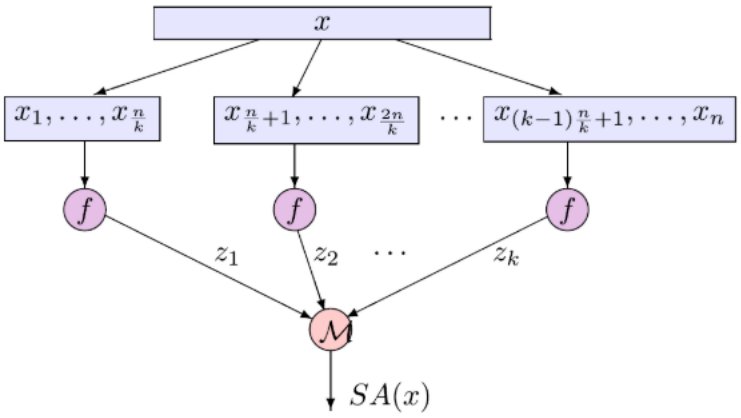
3. Smooth sensitivity

Poiché $LS_q(x)$ rivela informazioni su x, proviamo a calibrare la forma della laplaciana non su di essa ma su un'altra definizione di sensitività:

$$SS_q(x) = max_{y \in X^n} \left(LS_q(y) \cdot e^{-\frac{\varepsilon d(x,y)}{2\ln\left(\frac{2}{\delta}\right)}} \right)$$

• Questa funzione ha due buone qualità:
  • ha bassa sensitività globale
  • $LSq\ (x) \leq SS_q\ (x) \leq GS_q$ -> OK accuratezza e DP (per il fatto che SS è maggiore di LS)
• Si dimostra che il meccanismo di Laplace con parametro σ = 2 $SS_q(x)/\ \varepsilon$ rispetta la (ε, δ)-differential privacy.

• In generale non è semplice calcolare la smooth sensitivity per una generica funzione q: questo limita l'applicabilità del meccanismo.
• Esistono però delle funzioni per cui esiste un algoritmo per calcolare efficientemente $SS_q(x)$: una di questa è la mediana.
• IDEA: calcolare la funzione q su dei campioni estratti uniformemente dal dataset x e poi aggregare i risultati utilizzando la mediana. Il meccanismo di Laplace (con SS) verrà utilizzato nel calcolo della mediana.



Idea di costruire un framework chiamato Sample and Aggregate. Noi abbiamo un db x e vogliamo calcolare f(x) e invece di calcolare subito f(x) noi possiamo suddividere x in k sottocampioni, dopo di che calcolo f su ogni valore (avremo k stime). Dopo aggreghiamo le k stime con una funzione (es:mediana) usando la DP solo in questo passaggio (di aggregazione)

Punti critici:
• La funzione q si deve prestare a questo tipo di framework (es. se devo calcolare la somma di tutti i valori di campo del database, non ha senso usare questo metodo!)
• Il numero di record in x deve essere sufficientemente grande
• Se i dati sono molto ‘instabili’, l’accuratezza di questo metodo potrebbe essere bassa.