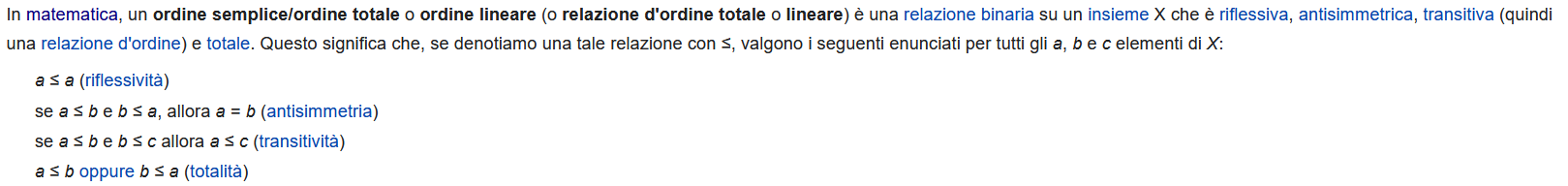
**Mandalorian Multidimensional K-Anonymity**

**Totally-ordered domains: **

* **X:** Quasi-Identifier Attribute
* **DX:** Il Dominio dell’attributo X
* ***CDM*:** Discernability metric (capi 1.2)
* ***CAVG\**:** Normalized average equivalence class size
* **Classe di Equivalenza:** Una tabella *T* è costituita da un multinsieme di tuple. Un classe di equivalenza per *T* rispetto agli attributi *X1, … , Xd* è l’insieme di tuple in *T* contenentivalori identici

(*x1, …, xd*) per *X1, … , Xd*. In SQL, questo è come una query GROUP BY su *X1, … , Xd*

**1.2. General-Purpose Quality Metrics**

Il processo di anonimizzazione dovrebbe generalizzare o perturbare i dati orifinali il minimo necessario per soddisfare i requisiti di k-anonimity. Qui consideriamo alcune metriche per misurare la qualità.

La più semplice metrica per misurare la qualità è basata sulle dimensioni delle classi di equivalenza *E* in *V*. Intuitivamente, la *discernability metric (CDM)* assegna ad ogni tupla *t* in *V* una penalità, che è determinata della dimensione della classe di equivalenza *E* contenente t.  
.

Come alternativa, proponiamo anche la *normalized average equivalence class size metric (CAVG).*

**Multidimensional GLOBAL Recoding**

* In a relational database, each attribute has some domain of values. We use the notation *DX* to denote the domain of attribute *X*. A **global recoding** achieves anonymity by mapping the domains of the quasi-identifier attributes to generalized or altered values.
* ***single-dimensional global recoding***is defined by a function *ϕii* : *DXi → D’ for each attribute Xi* of the quasi-identifier. An anonymization *V* is obtained by applying each *ϕi* to the values of *Xi* in each tuple of *T*.
* ***multidimensional global recoding***is defined by a *single* function *ϕ* : *DX1 x … x DXn : D’*, which is used to recode the domain of value *vectors* associated with the set of quasi-identifier attributes. An anonymization *V* is obtained by applying *ϕ* to the vector of quasiidentifiervalues in each tuple of *T*.

**Single-dimensional Partitioning:**

* Assumendo che sia associato un ordine totale ad ogni quasi-identificatore *Xi*. Un **single-dimesional partitioning** definisce, per ogni *Xi*, un insieme di non sovrapponibili intervalli che coprono *DXi. ϕi* mappa ogni x appartenete a *DXi*  to some *summary statistic* for the interval in which it is contained.
* Questo modello è facilmente estendibile al *multidimensional recording*. Assumendo un ordine totale per ogni *DXi*. Un *multidimensiona region* è definito da una coppia di d-tuple (p1, …, pn), (v1, …, vn) appartenenti a DX1, …, DXn tali che .

**Strict Multidimensional Partitioning:**

Un strict multidimensional partitioning definisce un set regioni di non-overlapping che coprono DX1, …, DXd. *ϕ* mappa ogni tupla (x1, …, xd) [appartenente a DX1, …, DXd] in una statistica riassuntiva per la regione in cui è contenuta.

When *ϕ* is applied to table T (assuming each region is mapped to a unique vector of summary statistics), the tuple set in each non-empty region forms an equivalence class in V .

**Multidimensional LOCAL Recoding:**

A differenza del global recording, il *local recording model mappa* (non-distinct) valori individuali in valori generallizati. Formally, a local recoding function, which we will denote *ϕ\* ,* maps each (non-distinct) tuple *t* ∈ *T* to some recoded tuple *t’*. *V* is obtained by replacing each tuple *t* ∈ *T* with *ϕ\** (*t*). In this section, we describe one such model that relaxes the requirements of strict multidimensional partitioning.

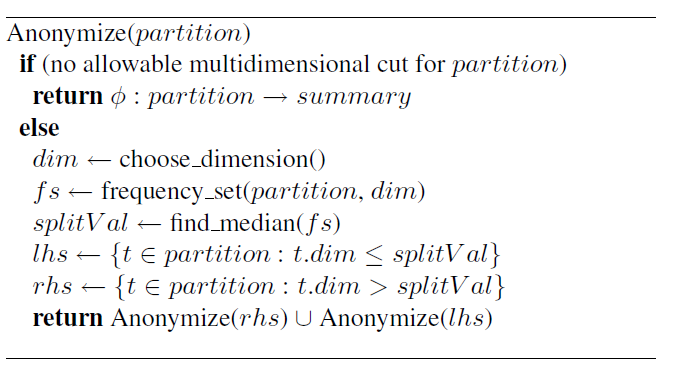
**Relaxed Multidimensional Partitioning:**

Un relaxed multidimensional parittioning per una tabella *T* definisce un inseime di regioni multidimensionali (potenzialmente sovrepponibili) che coprono DX1, …, DXd. Local recoding function *ϕ\** mappa ogni tuple (x1, …, xd) appartenent a *T* to a summary statistic per una delle regioni a cui appartien/è contenuto.

Sotto questo relaxed model, un partizionamento non è necessariamente definito da un taglio binario(binary cut). Invece, un insieme di punti viene partizionato definendo 2(possibly overlapping) regioni multidimensionali P1 e P2, per poi mappare ogni punto sia in P1 o P2(ma non entrambi). In this case, the upper-bound on the size of a minimal partition (one that cannot be divided without violating k-anonymity) is 2k - 1.

**4. A Greedy Partitioning Algorithm**

Usando un partizionamento multidimensionale, un k-anonymization viene generato in 2 step. Nel 1° step, le regioni multidimensioni sono definite in modo che coprano lo spazio di dominio(multidimensional regions are defined that cover the domain space). E nello 2° step, recoding functions sono costruite usando summary statistics per ogni regione.

In questa sezione mostreremo un semplice algoritmo scalabile, che ricorda quelli usati per costruire kd-tree, che può essere adattato sia a partizioni *strict* or *relaxed*. Il 2° step è descritto dettagliatamente nella sezione 5.

Per ogni iterazione bisogna sceglere la dimensione e i valori su cui partizionare. The split value è la mediana della partizione proiettata su dim(la dimensione scelta).

Se è consentito un taglio multidimensionale(multidimensional cut) per una partizione P perpendicolare a qualche asse *Xi* , allore il taglio perpendicolare a *Xi* nella mediana è consentito.

Abbiamo una certa flessibilità nello scegliere la dimensione su cui partizionare. Finché effettuiamo un taglio consentito quando ne esiste uno, questa scelta non influisce sul limite superiore della dimensione della partizione. Una euristica, utilizzata nella nostra implementazione, sceglie la dimensione con l'intervallo di valori più ampio (normalizzato). In alternativa, potrebbe essere possibile scegliere una dimensione in base a un carico di lavoro previsto.

L’algoritmo di partizionamento è facilmente adattabile ad un partizionamento *relaxed*. In particolare, i punti uguali alla mediana(dove t.dim = splitVal) vengono separati equamente tra *lhs\_child*  e *rhs\_child*, in modo che |*lhs\_child*| == |*rhs\_child*| (+1 se dispari).

In fine, una strategia *greedy* siamile per il partizionamento multidimensionel può essere usata per categorical attributes in presenza di una generalizazione gerarchia definita dell’utente(user-defined generalization hierarchies).

**4.1. Bounds on Quality**

Utilizzando i nostri limiti superiori sulla dimensione della partizione, è facile calcolare i limiti per le metriche di uso generale descritte nella Sezione 1.2 e gli attributi totalmente ordinati. Da definizione, K-anonymity richiede che ogni classe di equivalenza contenga come minimo k record. Per questa ragione, il valore ottimale raggiungibile di , e .

Per un partizionamento multidimensionale *strict*, assumendo che i punti in ogni partizionamneto distinto sono mappati i un unico vettore di summary statistics. Abbiamo mostrato che utilizzando l’algoritmo greedy, per ogni equivalenza *E*, , dove *m* è il massimo numero di copie di ogni tupupala di quasi-identificatori.

*CDM* è massimizato quando le tuple aono divise nel maggiore numero possibile di classi di equivalenza, quindi . Quindi,

Allo stesso modo, *CAVG* è massimizata quando le tuple sono divise nel numero maggiore possibile di classi di equivalenza, quindi .

Si osserva che per le costanti *d* e *m*, l’algoritmo greedy strict è un constant-factor approximation, come misurato da *CDM* e *CAVG*. Se *d* varia, ma *m=k* è costante, l’approssimazione è *O(d)*.

Per il partizionamento multidimensionale relaxed, l’algoritmo greedy produce 2- approximation, perché , e .

**4.2. Scalability**

Quando la tabella *T* da essere anonimizzata è più grande della memoria disponibile, il problema principale da affrontare è trovare la mediana di un attributo per una data prtizione. (NON PESO SIA UN NOSTRO PROBLEMA QUINDI NON HO SBATTI DI STARLO A TRADURRE)

**5.Workload-Driven Quality Measurement**

Il prosito generale delle metriche descritte nella sezioni 1.2 sono un buon punto di partenza quando l'applicazione che alla fine consuma i dati anonimi è sconosciuta. Pero’, in alcuni casi, l'editore potrebbe voler considerare un carico di lavoro anticipato o rispondere a una serie di query aggregate. Questa sezione introduce quest'ultimo problema, inclusi esempi in cui la ricodifica multidimensionale fornisce la flessibilità necessaria.

Considera un set di queries che selezionano predicati(per uguaglianza o intervallo) della forma *attribute <oper> constant* e una funzione di aggregazione(COUNT, SUM, AVG, MIN, and MAX). La nostra capacità di rispondere a questo tipo di queries da dati anonimizati dipende da due fattori principali:

La tipologia della summary statistic(s) in relazione per ogni attributo e

Il grado in cui i predicati di selezione nel carico di lavoro corrispondono ai limiti dell'intervallo nei dati anonimi. (the degree to which the selection predicates in the workload match the range boundaries in the anonymous data)

La scelta della summary statistic influenza la nostra abilità di calcolare varie funzioni aggregate(various aggregate functions). In questo paper, consideriamo di rilasciare(we consider releasing) 2 summary statistic per ogni attributo A e classe di equivalenza E:

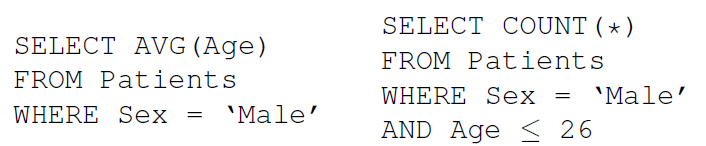
* **Range Statistic (R):** Fino ad ora, tutti i nostri esempi hanno coinvolto un singolo *summary statistic* definito dal range dei valori per *A* che appaiono in *E*, che consente un facile calcolo degli aggregati MIN e MAX
* **Mean Statistic (M):** Abbiamo anche considerato una *summary statistic* definita dalla media dei valori di *A* che compaiono in *E*, che consente il calcolo di AVG e SUM.

Quando si sceglie una *summary statistic,* è importatnte considerare potenziali strade per inferenza. Da notare che in alcuni casi usando remplicemente min-max range consente alcune inferenze sulla distribuzione dei valori all'interno di una classe di equivalenza. Per esempio, consideriamo un attributo *A*, e *k=2.* Supponiamo che una classe di equivalenza dell'anonimizzazione rilasciata(of the released anonymization) contenga 2 tuple, e *A* è una *summarized* in un range [0, 1]. E’ facile dedurre(infer) che in una delle tuple originali *A=0*, a nell’altra *A=1.*

Questo tipo di *deduzioni* riguardanti la distribuzione(che possono avvenire anche in single-dimensional recoding) è probabile che costituisca un problema nel prevenire joining attacks perché, senza background knowledge, è ancora possibile per un avversario distinguere le tuple all'interno di una classe di equivalenza l'una dall'altra.

Il secondo fattore che influenza la nostra capacità di rispondere a query aggregate è il grado in cui i predicati di selezione nel dato carico di lavoro "corrispondono" ai confini delle statistiche di intervallo nell'anonimizzazione rilasciata(released anonymization). Per molti versi, questo è analogo alla corrispondenza di indici e predicati di selezione nell'elaborazione delle query tradizionali.

**Predicate-Range Matching**

Un predicato *Pred* concettualmente divide la tabella originale *T* in 2 insiemi di tuple, e (quelli che soddisfano il predicato e quelli che non lo soddisfano). Quando *range statistic* sono pubblicate, noi diciamo che un anonimizazione *V* ‘corrisponde’ a un predicato booleano *Pred* se ogni tupla *t* ∈ è mappato in una classe di equivalenza in *V* contenente nessuna tupla da .

Per illustrare questa idea, considera un workload contenente 2 queries: