Εθνικο Μετσοβιο Πολυτεχνειο
Σχολη Ηλεκτρολογων Μηχανικων και Μηχανικων Υπολογιστων
Τομεας Τεχνολογιας Πληροφορικης και Υπολογιστων

Διασφάλιση Ιδιωτικότητας σε Δυναμικά Δεδομένα

Διπλωματικη Εργασια

του

ΓΕΩΡΓΙΟΥ ΓΡΑΤΣΙΑ

Επιβλέπων: Τιμολέων Σελλής
Καθηγητής Ε.Μ.Π.

Εργαστηριο Συστηματων Βασεων Γνωσεων και Δεδομενων
Αθήνα, Ιούλιος 2008

Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Τομέας Τεχνολογίας Πληροφορικής και Υπολογιστών
Εργαστήριο Συστημάτων Βάσεων Γνώσεων και Δεδομένων

Διασφάλιση Ιδιωτικότητας σε Δυναμικά Δεδομένα

Διπλωματικη Εργασια

του

ΓΕΩΡΓΙΟΥ ΓΡΑΤΣΙΑ

Επιβλέπων: Τιμολέων Σελλής
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 10η Ιουλίου 2008.

.....	.....	.....
Τιμολέων Σελλής	Ιωάννης Βασιλείου	Νεκτάριος Κοζύρης
Καθηγητής Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.	Αναπλ. Καθηγητής Ε.Μ.Π.

Αθήνα, Ιούλιος 2008

.....

Γεωργιος Γρατσιας
Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Τομέας Τεχνολογίας Πληροφορικής και Υπολογιστών
Εργαστήριο Συστημάτων Βάσεων Γνώσεων και Δεδομένων

Περιεχόμενα

Κεφάλαιο 1

Εισαγωγή

Αρκετοί οργανισμοί, διαχειρίζονται σήμερα ένα σημαντικό μέγεθος δεδομένων, αποθηκευμένα σε κάποιο σύστημα βάσεων δεδομένων. Επιθυμία, αυτών οργανισμών, αποτελεί πολλές φορές η δυνατότητα χρήσης αυτής της πληροφορίας από τρίτους (ή άλλα υπό-τμήματα του ίδιου του οργανισμού) για την εξαγωγή χρήσιμων συμπερασμάτων ή επιπλέον πληροφορίας. Παράλληλα όμως, οι οργανισμοί, επιθυμούν να αποκρύπτονται κάποια στοιχεία, για την διασφάλιση της ανωνυμίας του χρήστη.

Για παράδειγμα, ένας οργανισμός υγείας, όπως ένα νοσοκομείο, είναι δυνατόν να επιθυμεί να δημοσιεύσει την βάση με το ιστορικό των ασθενών οι οποίοι έχουν νοσηλευτεί σε αυτό. Παράλληλα όμως ο οργανισμός επιθυμεί η δημοσίευση της βάσης, να αποκρύπτει σε ποιον ασθενή ανήκει το κάθε ιστορικό. Μία λύση θα μπορούσε να είναι η αφαίρεση των αναγνωριστικών του ατόμου, όπως ο αριθμός ταυτότητας, όνομα και άλλα. Αυτό όμως στην πραγματικότητα δεν είναι αρκετό. Είναι πιθανόν με βάση κάποια attributes κάποιος εξωτερικός χρήστης να ανακαλύψει σε ποιο άτομο ανήκει αυτή η εγγραφή. Για παράδειγμα, έστω ότι έχουμε ένα δημοσιευμένο πίνακα, ο οποίος μεταξύ άλλων περιέχει την ηλικία, ταχυδρομικό κώδικα, επάγγελμα και φύλο. Πόσο πιθανόν είναι να βρούμε κάποιον, ο οποίος μένει σε μία συγκεκριμένη περιοχή, είναι συγκεκριμένου φύλου και ηλικίας και έχει συγκεκριμένο επάγγελμα; Η πιθανότητα, μπορεί άλλοτε να είναι μεγάλη άλλοτε μικρή και εξαρτάται από διάφορους παράγοντες. Πρόσφατη μάλιστα εργασία έδειξε ότι το 87 τοις εκατό του πληθυσμού των Ηνωμένων Πολιτειών της Αμερικής μπορεί να προσδιοριστεί μοναδικά από τον ταχυδρομικό κώδικα, φύλο και ημερομηνία γέννησης.

Patient Data			
Age	Sex	Zipcode	Disease
25	Male	53771	Prostate Cancer
25	Female	53772	Hepatitis
26	Male	53771	Prostate Cancer
27	Male	53710	Broken Arm
27	Female	53712	AIDS
28	Male	53711	Lung Cancer

(α′)

Voter Registration Data			
Name	Age	Sex	Zipcode
Ahmed	25	Male	53771
Alice	28	Female	55410
Claire	31	Female	90210
Dave	19	Male	02174
Evelyn	40	Female	02237

(β′)

Σχήμα 1.1: Ο δημοσιευμένος πίνακας και η εξωτερική πηγή δεδομένων.

Ας δούμε για παράδειγμα τον πίνακα του Σχήματος 1.1. Ένα νοσοκομείο αποφασίζει να εκδώσει τον πίνακα με το ιστορικό των ασθενών του, αφαιρώντας τα id. Ο Bob όμως βρίσκει τον πίνακα των ψηφοφόρων και κάνει join τους δύο πίνακες. Μπορεί να παρατηρήσει κανείς ότι ο Bob μπορεί να ανακαλύψει με πιθανότητα ίση με 1 την ασθένεια του Ahmed. Το

παράδειγμα είναι αληθινό, για την ακρίβεια σε μία εργασία ο ερευνητής χρησιμοποιώντας τον πίνακα του ιστορικού των ασθενειών ενός νοσοκομείου και τον διεθνή κατάλογο ψηφοφόρων των Ηνωμένων Πολιτειών κατάφερε με ακριβώς πιθανότητα 1 να ανακαλύψει την ασθένεια του κυβερνήτη της Μασαχουσέτης.

1.1 k-anonymity

Στόχος λοιπόν του privacy preservation είναι να μειώσει την πιθανότητα να ανακαλύψουμε ποια εγγραφή ανήκει σε ποιο άτομο. Πιο συγκεκριμένα, η προσπάθεια ανακάλυψης της ταυτότητας μία εγγραφής μπορεί να γίνει με join κάποιου εξωτερικού πίνακα (ή γενικότερα κάποια εξωτερική πληροφορία) με την νέα δημοσιευμένη βάση. Το privacy preservation επιθυμεί να υπάρχει μία μέγιστη πιθανότητα το πολύ k, ένας αντίπαλος (εξωτερικός παράγοντας) με κάποιο join να μπορέσει να ανακαλύψει σε ποιο άτομο (ή περισσότερα όπως μία ολόκληρη κοινότητα) ανήκει μία εγγραφή ή σύνολο εγγραφών. Η μεθοδολογία η οποία έχει προταθεί για την επίλυση αυτού του προβλήματος είναι το k-anonymity, η οποία εξασφαλίζει ότι η πιθανότητα ανακάλυψης της ταυτότητας μίας εγγραφής είναι το πολύ $1/k$.

2-anonymity			
Age	Sex	Zipcode	Disease
[25 – 26]	Male	53771	Prostate Cancer
[25 – 27]	Female	53772	Hepatitis
[25 – 26]	Male	53771	Prostate Cancer
[27 – 28]	Male	[53710 – 53771]	Broken Arm
[25 – 27]	Female	53712	AIDS
[27 – 28]	Male	[53710 – 53771]	Lung Cancer

Σχήμα 1.2: Η anonymization του αρχικού πίνακα.

Ας δούμε ξανά τον πίνακα του Σχήματος 1.1 και ας δούμε ξανά τον εκδιδόμενο πίνακα του Σχήματος 1.2. Αν ο Bob πάρει τον εξωτερικό πίνακα και πάρει τον δημοσιευμένο πίνακα τότε μπορεί να κατασκευάσει με πιθανότητα 1/2 οποιαδήποτε εγγραφή. Ο λόγος που συμβαίνει αυτό είναι γιατί ο πίνακας έχει χωριστεί σε διάφορα υποσύνολα έτσι ώστε σε κάθε υποσύνολο τουλάχιστον κάθε δύο εγγραφές να έχουν ίδια τιμή για την age, το sex και το zipcode. Επειδή αυτό συμβαίνει ανά δύο εγγραφές ο πίνακας ικανοποιεί το 2-anonymity. Γενικότερα θα λέμε Quasi-Identifier το ελάχιστο σύνολο από ιδιότητες $Q = X_1, \ldots, X_d$ με το οποίο ένας πίνακας T μπορεί να γίνει join με κάποιες εξωτερικές πληροφορίες για να αναγνωριστούν ατομικές εγγραφές. Το k-anonymity είναι μία μεθοδολογία η οποία χωρίζει τον πίνακα μας σε κλάσεις ισοδυναμίας τουλάχιστον μεγέθους k, έτσι ώστε σε κάθε κλάση οι εγγραφές να έχουν το ίδιο Quasi-identifier.

1.2 l-diversity

Στόχος όμως του privacy preservation δεν είναι μόνο η ασφάλεια της ταυτότητας μίας εγγραφής. Είναι και η διασφάλιση, ότι από ένα σύνολο εγγραφών δεν θα μπορούμε να βρούμε εύκολα κάποια συγκεκριμένα στοιχεία για ένα άτομο. Πιο συγκεκριμένα, ας δούμε ξανά τον πίνακα του Σχήματος 1.2. Ενώ ο πίνακας μας ικανοποιεί το 2-anonymity δεν είναι σε θέση να προστατέψει σε κάποιες περιπτώσεις τις εγγραφές μας. Για παράδειγμα ο Bob μπορεί να μην ξέρει αν ο Ahmed ανήκει στην πρώτη ή στην τρίτη εγγραφή, όμως γνωρίζει σίγουρα ότι ο πάσχει από prostate cancer. Το k-anonymity δεν είναι σε θέση να μας εξασφαλίσει ότι ο αντίπαλος δεν θα μπορέσει να εξάγει με επιτυχία κάποιες πληροφορίες για κάποιες εγγραφές.

Οι ιδιότητες, τις οποίες δεν θέλουμε να μπορεί να ανακαλύψει ο αντίπαλος, λέγονται ευαίσθητες (sensitive attributes).

Έτσι στόχος του privacy preservation είναι η διασφάλιση της ανωνυμίας των ευαίσθητων δεδομένων ενός ατόμου στο δημοσιευμένο πίνακα. Η μεθοδολογία η οποία έχει προταθεί για την επίλυση αυτού του προβλήματος είναι το l-diversity, η οποία επιτρέπει σε ένα εξωτερικό παράγοντα να ανακαλύψει με πιθανότητα περίπου $1/l$ τα ευαίσθητα δεδομένα ενός ατόμου, ανεξάρτητα σε ποια εγγραφή ανήκει αυτό το άτομο. Ας δούμε για παράδειγμα το πίνακα του Σχήματος 1.3. Προσέξτε ότι σε αυτή την περίπτωση ο αντίπαλος ακόμα και αν κάνει join με ένα εξωτερικό πίνακα, είναι σε θέση να ανακαλύψει την ασθένεια ενός ατόμου με πιθανότητα 1/2, δηλαδή ικανοποιείται το 2-diversity. Με άλλα λόγια το l-diverse συσχετίζει κάθε εγγραφή με l τιμές για την sensitive attribute.

Patient Data			
Age	Sex	Zipcode	Disease
25	Male	53771	Prostate Cancer
28	Male	53711	Lung Cancer
26	Male	53771	Prostate Cancer
27	Male	53710	Broken Arm
27	Female	53712	AIDS
25	Female	53772	Hepatitis

Σχήμα 1.3: Ένας 2-diverse πίνακας.

Είναι προφανές ότι σε κάθε νέα δημοσιευμένη βάση, δεν είναι απαραίτητο να ικανοποιούνται και οι δύο στόχοι. Εξαρτάται από την εκάστοτε εφαρμογή και τις ανάγκες τις επιχείρησης ποια μεθοδολογία (ή συνδυασμός αυτών) θα χρησιμοποιηθεί. Πάντως είναι πιο πιθανό επιθυμούμε να ικανοποιείται το l- diversity το οποίο είναι μία πιο ισχυρή μεθοδολογία διασφάλισης της ανωνυμίας, αφού μας ενδιαφέρει συνήθως η διασφάλιση των ευαίσθητων δεδομένων.

1.3 Dynamic Data, m-Invariance

Ένα πρόβλημα και των δύο μεθοδολογιών είναι ότι δεν λαμβάνουν καμία μέριμνα για τα δυναμικά δεδομένα. Με άλλα λόγια τι συμβαίνει στην περίπτωση που εισάγονται εγγραφές στην βάση ή στην περίπτωση που διαγράφονται. Για το l-diversity έχει προταθεί μία μεθοδολογία, το m-invariance, η οποία είναι επέκταση αυτού, ώστε να μπορούμε να χειριστούμε και δυναμικά δεδομένα.

Ας δούμε σε πρώτη φάση τον πίνακα του Σχήματος 1.4. Εν συγκρίσει με τον αρχικό μας πίνακα έχουν διαγραφεί τρεις εγγραφές η δεύτερη, η τρίτη και η τελευταία και έχουν εισαχθεί τρεις καινούργιες στις θέσεις τους. Μπορεί να παρατηρήσει κανείς, ότι ενώ και στο Σχήμα a και στο Σχήμα b ικανοποιούμε το l-diversity, ο συνδυασμός των δύο δίνει την δυνατότητα στον αντίπαλο να εξάγει σημαντικά συμπεράσματα. Πιο συγκεκριμένα, ο Ahmed στην πρώτη περίπτωση έχει είτε prostate cancer είτε colorectal cancer και στην δεύτερη είτε prostate cancer είτε hang nail. Ο Bob από αυτά τα στοιχεία είναι σε θέση να γνωρίζει με σιγουριά την ασθένεια του Ahmed. Αντίθετα ας δούμε τον πίνακα του Σχήματος 1.5. Μπορούμε να προσέξουμε ότι δύο από τις εγγραφές οι οποίες διαγράφτηκαν παραμένουν στην βάση μας, ενώ η άλλη δεν υπάρχει πια. Μπορεί κάποιος να παρατηρήσει ότι η διαγραφή της δεύτερης εγγραφής ήταν εφικτή γιατί υπήρχε μία εισαγωγή με την ίδια τιμή στην ευαίσθητη ιδιότητα (prostate cancer). Αντίθετα για τις άλλες δύο αυτό δεν ήταν εφικτό. Για την ακρίβεια το m-invariance απαιτεί την ικανοποίηση του m-diversity και παράλληλα μία εγγραφή να ανήκει πάντα σε ένα group, το οποίο να μην έχει απαραίτητα τις ίδιες εγγραφές, αλλά το ίδιο μοναδικό σύνολο τιμών της ευαίσθητης ιδιότητας.

Patient Data, First Instance			
Age	Sex	Zipcode	Disease
25	Male	53771	Prostate Cancer
25	**Female**	**53772**	**Hepatitis**
26	**Male**	53771	**Prostate Cancer**
27	Male	53710	Broken Arm
27	Female	53712	AIDS
28	**Male**	**53711**	**Lung Cancer**

(α′)

Anonymized Patient Data of first instance			
Age	Sex	Zipcode	Disease
25	Male	53771	Prostate Cancer
28	**Male**	**53711**	Lung Cancer
26	**Male**	**53771**	Prostate Cancer
27	Male	53710	Broken Arm
27	Female	53712	AIDS
25	**Female**	**53772**	Hepatitis

(β′)

Patient Data, Second Instance			
Age	Sex	Zipcode	Disease
25	Male	53771	Prostate Cancer
16	**Female**	**43772**	Melanoma
40	**Male**	**63771**	Prostate Cancer
27	Male	53710	Broken Arm
27	Female	53712	AIDS
31	**Male**	**63711**	Hang Nail

(γ′)

Anonymized Patient Data of Second instance			
Age	Sex	Zipcode	Disease
25	Male	53771	Prostate Cancer
31	**Male**	**63711**	Hang Nail
40	**Male**	**63771**	Prostate Cancer
27	Male	53710	Broken Arm
27	Female	53712	AIDS
16	**Female**	**43772**	Melanoma

(δ′)

Σχήμα 1.4: Μη διασφάλιση της ανωνυμίας σε δυναμικά δεδομένα.

Anonymized Patient Data of second instance			
Age	Sex	Zipcode	Disease
25	Male	53771	Prostate Cancer
28	**Male**	**53711**	Lung Cancer
40	**Male**	**63771**	Prostate Cancer
27	Male	53710	Broken Arm
27	Female	53712	AIDS
25	**Female**	**53772**	Hepatitis
31	**Male**	**6371**1	Hang Nail
16	**Female**	**43772**	Melanoma

Σχήμα 1.5: Η δεύτερη έκδοση ικανοποιεί το 2-invariance.

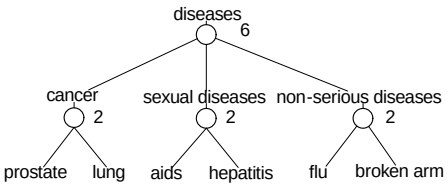
1.4 Correlation Based Anonymity

Το μοντέλο του m-invariance, όπως και του l-diversity έχουν ένα σημαντικό μειονέκτημα, δεν είναι σε θέση να λάβουν υπόψη τους τον βαθμό συσχετισμού των sensitive attributes. Η συγκεκριμένη διπλωματική έχει ως στόχο να προτείνει ένα νέο μοντέλο και σύνολο αλγορίθμων ώστε να ληφθεί υπόψη ο βαθμός συσχετισμού. Πιο συγκεκριμένα, ας δούμε ξανά το Σχήμα 1.3. Ο Bob μπορεί ναι μεν να μην γνωρίζει εάν ο Ahmed πάσχει από prostate cancer ή lung cancer, γνωρίζει όμως ότι πάσχει από καρκίνο. Το μοντέλο του correlation based anonymity προσπαθεί να αποτρέψει τέτοια φαινόμενα. Πιο συγκεκριμένα, εάν έχουμε μία μετρική με βάση την οποία να υπολογίζουμε το συσχετισμό μεταξύ των τιμών της ευαίσθητης ιδιότητας, θα απαιτούμε την ικανοποίηση ενός upper bound. Στο δικό μας μοντέλο, για να μετρήσουμε τον συσχετισμό θεωρήσαμε ότι η ευαίσθητη ιδιότητα είναι κατηγορική (όπως εξάλλου η πλειοψηφία των μοντέλων του privacy) και ταυτόχρονα θεωρήσαμε ότι έχουμε ιεραρχία γενίκευσης για αυτήν την ιδιότητα. Μία τέτοια ιεραρχία μπορούμε να δούμε στο Σχήμα 1.6. Εν συνεχεία, ορίσαμε ως βαθμό συσχετισμού, δύο τιμών-κόμβων $u1, u2$ το πλήθος των φύλλων όπου αντιστοιχούν στον κόμβο v, όπου v ο κοντινότερος πρόγονος των $u1, u2$. Αυτή η μετρική, γενικότερα, έχει προταθεί ξανά στην βιβλιογραφία αλλά δεν είχε χρησιμοποιηθεί για την ευαίσθητη ιδιότητα. Έτσι αν θέσουμε ως upper bound μία τιμή e, τότε μία εγγραφή θα πρέπει

να ανήκει σε ένα group όπου οι τιμές της ευαίσθητης ιδιότητας έχουν το πολύ συσχετισμό e. Προσέξτε ότι πάλι απαιτούμε να ισχύει το $m-$invariance. Ένα τέτοιο παράδειγμα δίνεται στο Σχήμα 1.6. Ο Bob δεν είναι πια σε θέση να αποφασίσει αν ο Ahmed πάσχει από καρκίνο ή όχι και παράλληλα να βρει την οικογένεια των ασθενειών του με συσχετισμό μεγαλύτερο του 2. Με άλλα λόγια η συγκεκριμένη μεθοδολογία εμποδίζει τον αντίπαλο να ανακαλύψει την οικογένεια των ασθενειών (ή γενικότερα της ευαίσθητης ιδιότητας) του ατόμου. Τέλος για την επίλυση του προβλήματος προτάθηκε ένας αλγόριθμος. Ο αλγόριθμος παίρνει την βάση μας, προσθέτει μία νέα ιδιότητα gsa και εν συνεχεία θεωρεί ως ευαίσθητη ιδιότητα την gsa. Το πλεονέκτημα του αλγορίθμου είναι ότι μετά την δημιουργία της gsa, αν εκτελέσουμε οποιοδήποτε γνωστό αλγόριθμο του m-invariance για την gsa τότε θα ικανοποιείται και το correlation. Η δημιουργία αυτής της ευαίσθητης ιδιότητας έχει πολύ μικρό κόστος, για την ακρίβεια κατά την διάρκεια αρχικοποίησης του συστήματος σκανάρουμε μία φορά την ιεραρχία (είναι πιθανόν να αναγκαστούμε να την σκανάρουμε όλη, αλλά στην πραγματικότητα χρειάζεται πολύ μικρό κομμάτι αυτής) και εν συνέχεια κατά την διάρκεια δημιουργίας μίας ανώνυμης όψης σκανάρουμε μία φορά μόνο γραμμικά την βάση μας.

Patient Data			
Age	Sex	Zipcode	Disease
25	Male	53771	Prostate Cancer
25	Female	53772	Hepatitis
26	Male	53771	Prostate Cancer
27	Male	53710	Broken Arm
27	Female	53712	AIDS
28	Male	53711	Lung Cancer

(α′) Cba-Table



(β′) Η ιεραρχία

Σχήμα 1.6: Correlation Based Anonymity.

1.5 Utility

Επιπλέον βασικό πρόβλημα όλων των παραπάνω μεθοδολογιών, είναι η διατήρηση όσο το δυνατόν περισσότερης πληροφορίας. Εξάλλου αν δεν μας ενδιαφέρει η σημασία της πληροφορίας την οποία δημοσιεύουμε θα μπορούσαμε να αφαιρέσουμε όποιες εγγραφές ή ιδιότητες, μας δημιουργούν προβλήματα. Αυτή η λύση είναι προφανές ότι δεν προσφέρεται, αφού ο δημοσιευμένος πίνακας μπορεί να χρησιμοποιηθεί για διάφορους σκοπούς όπως στατιστική μελέτη και άλλα. Στόχος όλων των αλγορίθμων εκτός από την ικανοποίηση κάποιας μεθοδολογίας, είναι να διατηρήσουν και το utility μέγιστο. Για αυτό το σκοπό έχουν προταθεί διάφορες μεθοδολογίες, όπως γενίκευση δεδομένων με βάση κάποια ιεραρχία.

Ένα πρόβλημα το οποίο δεν έχει μελετηθεί προηγουμένως είναι πόσο επηρεάζει το utility ο αποσυσχετισμός της εγγραφής από την πραγματική της τιμή στην ευαίσθητη ιδιότητα. Ας θυμηθούμε ξανά το Σχήμα 1.6. Ο Ahmed ενώ στην πραγματικότητα έχει prostate cancer, στην εκδιδόμενη όψη ισχυριζόμαστε ότι έχει είτε prostate cancer είτε hepatitis. Η δημιουργία αυτού του group επηρεάζει σημαντικά το utility και κυρίως την ικανότητα μας να απαντήσουμε range queries. Όπως δείχθηκε και στην εργασία είναι προτιμότερο μία εγγραφή να συσχετίζεται με τιμές της ευαίσθητης ιδιότητας οι οποίες να έχουν μεγάλο βαθμό συσχετισμό μεταξύ τους (αρκεί βέβαια να μην ξεπεράσουμε το upper bound). Για αυτό το λόγο προτάθηκε ένας νέος αλγόριθμος ο οποίος χωρίζει τον αρχικό μας πίνακα σε υποπίνακες έτσι ώστε σε κάθε υποπίνακα οι τιμές της ευαίσθητης να έχουν μεγάλο βαθμό συσχετισμού. Το μέγιστο κόστος του αλγορίθμου είναι ίσο με το σκανάρισμα όλης της ιεραρχίας.

Επίσης τα προηγούμενα μοντέλα δεν εξέταζαν την απόσταση των εγγραφών σε ένα group. Πιο συγκεκριμένα ας δούμε ξανά το Σχήμα 1.6. Μπορεί να παρατηρήσει ότι κανείς ότι αν εκτελέσουμε το query: select disease from T where age = 25 and zipcode = 53771–53772, αυτό θα απαντηθεί με απόλυτή ακρίβεια. Αυτό είναι εφικτό γιατί οι δύο πρώτες εγγραφές είναι πολύ κοντά μεταξύ τους, δηλαδή η περίμετρος των δύο εγγραφών είναι πολύ μικρή. Έτσι είναι επιθυμητό οι εγγραφές σε ένα group να έχουν μικρή περίμετρο. Βέβαια επειδή συνήθως είναι πιο δύσκολο να ελέγξουμε με ακρίβεια την περίμετρο, μιας και το πρόβλημα είναι $NP-hard$, ελέγχουμε πρώτα κριτήριο που προτάθηκε στην προηγούμενη παράγραφο. Για την επίλυση χρησιμοποιήθηκε ένας υπάρχων αλγόριθμος που χρησιμοποιεί το κριτήριο της περιμέτρου και σε αντίθεση με τον προηγούμενο υπάρχει στην βιβλιογραφία.

Τέλος ένα βασικό πρόβλημα, είναι ο χειρισμός των διαγραφών και των εισαγωγών. Υπενθυμίζουμε ότι για να αντικατασταθεί μία διαγραφή στον πίνακά μας, τότε πρέπει να αναμένουμε μία εισαγωγή. Αυτό πολλές φορές είναι δύσκολο να επιτευχθεί. Μάλιστα αν σε κάποια χρονική στιγμή στην βάση, οι διαγραφές ως προς μία τιμή της sensitive attribute είναι πιο συχνές από τις εισαγωγές ως προς την ίδια τιμή, τότε θα οδηγηθούμε σε μεγάλη μείωση του utility. Αν όμως εμείς γενικεύσουμε τις τιμές με κάποιο μοντέλο γενίκευσης, τότε είναι πιο εύκολο να αντικαταστήσουμε τιμές. Πιο συγκεκριμένα λαμβάνοντας υπόψη μας την πιθανότητα διαγραφής μίας εγγραφής μπορούμε να γενικεύσουμε την ευαίσθητη ιδιότητα ώστε να έχουμε αύξηση της πιθανότητας αντικατάστασής της.

Συνοψίζοντας, το privacy preservation προσπαθεί από κάποια στοιχεία να εκδώσει ένα νέο πίνακα διαφυλάσσοντας την ανακάλυψη της ταυτότητας μίας εγγραφής (δηλαδή σε ποιο άτομο ανήκει) και την ανακάλυψη των ευαίσθητων δεδομένων διατηρώντας όσο το δυνατόν περισσότερη πληροφορία.

Κεφάλαιο 2

Το Πρόβλημα της Ιδιωτικότητας σε Δημοσιευμένα Δεδομένα

2.1 k-anonymity

Voter Registration Data			
Name	Age	Sex	Zipcode
Ahmed	25	Male	53771
Brooke	28	Female	55410
Claire	31	Female	90210
Dave	19	Male	02174
Evelyn	40	Female	02237

(α′)

Patient Data			
Age	Sex	Zipcode	Disease
25	Male	53771	Flu
25	Female	53772	Hepatitis
26	Male	53771	Bronchitis
27	Male	53710	Broken Arm
27	Female	53712	AIDS
28	Male	53711	Hang Nail

(β′)

Σχήμα 2.1: Ο πίνακας των ασθενειών κινδυνεύει από κάποιο join με εξωτερικές πληροφορίες.

Στην εισαγωγή αναφερθήκαμε στην δυνατότητα διαφόρων εξωτερικών παραγόντων να ανακαλύψουν την ταυτότητα μίας εγγραφής (δηλαδή σε ποιο άτομο ανήκει αυτή η εγγραφή). Όπως προαναφέρθηκε δεν αρκεί πάντα η αφαίρεση διάφορων αναγνωριστικών για να μας εξασφαλίσει την ανωνυμία. Για παράδειγμα το Σχήμα 2.1 δίνει ένα χαρακτηριστικό παράδειγμα, όπου ένας δημοσιευμένος πίνακας κινδυνεύει από κάποιο εξωτερικό join. Ένας εξωτερικός παράγοντας κάνοντας join αυτούς του δύο πίνακες θα μπορούσε να ανακαλύψει ότι ο Ahmed ανήκει στον πίνακα και μάλιστα πάσχει από Flu.

Η προφανής λύση στο πρόβλημα μας είναι να αφαιρέσουμε όσες ιδιότητες μπορούνε να γίνουνε join με κάποιο εξωτερικό πίνακα. Αυτό όμως άμεσα συνεπάγεται απώλεια πληροφορίας, γεγονός μη επιθυμητό. Έτσι το k-anonymity έχει ως στόχο να εκδώσουμε ένα πίνακα, με όσο το δυνατόν μικρότερη απώλεια πληροφορίας. Πως μπορούμε να το κάνουμε αυτό; Γενικεύοντας διάφορες ιδιότητες μπορούμε να δημιουργήσουμε k-ίδιες εγγραφές, μειώνοντας έτσι την ικανότητα του εξωτερικού παράγοντα να ανακαλύψει σε ποιον ανήκει μία εγγραφή.

2.1.1 Βασικοί ορισμοί

Ορισμός 1. *Quasi-Identifier Attribute Set Είναι το ελάχιστο σύνολο από ιδιότητες $Q = X_1, \ldots, X_d$ με το οποίο ένας πίνακας T μπορεί να γίνει join με κάποιες εξωτερικές πληροφορίες για να αναγνωριστούν ατομικές εγγραφές.*

Είναι προφανές, ότι δεν γίνεται πάντα να υπολογιστεί το Quasi-Identifier με ακρίβεια. Στην συνέχεια της εργασίας θα θεωρήσουμε ότι είναι γνωστό με βάση κάποια συγκεκριμένη πληροφορία του domain.

Ορισμός 2. *Equivalence Class. Ένας πίνακας T αποτελείται από ένα πολυσύνολο εγγραφών. Equivalence Class για το T με βάση τις ιδιότητες $Q = X_1, \ldots, X_d$ είναι το σύνολο όλων των εγγραφών στο T οι οποίες περιέχουν ίσες τιμές $(x_1, \ldots, x_d)$ για το $Q = X_1, \ldots, X_d$.*

Στην SQL θα μπορούσαμε να βρούμε το Equivalence Class εκτελώντας ένα group by query στα $X_1, \ldots, X_d$.

Ορισμός 3. *Frequency Set. Έστω ένας πίνακα T και ένα σύνολο ιδιοτήτων $Q = X_1, \ldots, X_d$. Το frequency set του T με βάση το Q είναι μία αντιστοίχηση με κάθε μοναδικό συνδυασμό των τιμών $(x_1, \ldots, x_d)$ του Q στο T με το συνολικό αριθμών των εγγραφών στο T με βάση τιμές του Q.*

Στην SQL θα μπορούσαμε να βρούμε το frequency set εκτελώντας ένα group by query στα $X_1, \ldots, X_d$ και μετά απλά την εντολή count. Είναι προφανές ότι το frequency set είναι ο πληθάριθμος όλων των συνόλων του equivalence class.

Με βάση τους δύο παραπάνω ορισμούς είναι δυνατόν να οριστεί το k-anonymity καθώς και ποτέ ένα δημοσιευμένος πίνακας αποτελεί k-anonymization του αρχικού πίνακα.

Ορισμός 4. *k-anonymity Property. Ένας πίνακας T θα ικανοποιεί την k-anonymity property ή θα λέμε ότι είναι k-anonymous με βάση ένα σύνολο ιδιοτήτων $Q = X_1, \ldots, X_d$ εάν κάθε τιμή πλήθους στο frequency set του T με βάση το Q είναι μεγαλύτερο ή ίσο του k. Ισοδύναμα θα μπορούσαμε να πούμε ότι ένας πίνακας T είναι k-anonymous με βάση ένα σύνολο ιδιοτήτων $Q = X_1, \ldots, X_d$ αν κάθε μοναδική εγγραφή $(x_1, \ldots, x_d)$ στην προβολή του T πάνω στο $X_1, \ldots, X_d$ εμφανίζεται τουλάχιστον k φορές. Δηλαδή το μέγεθος κάθε equivalence class στο T με βάση τα $X_1, \ldots, X_d$ έχει πληθάριθμο τουλάχιστον k.*

Ορισμός 5. *k-anonymization. Μία όψη V ενός πίνακα T θα είναι k-anonymization του T αν η όψη αλλάζει ή γενικεύει τα δεδομένα του T με βάση κάποιο μοντέλο έτσι ώστε το V να είναι k-anonymous με βάση το quasi-identifier.*

Με άλλα λόγια το k-anonymity, ως μεθοδολογία, εκδίδει ένα νέο πίνακα (με βάση ένα παλιό) έτσι ώστε στο νέο πίνακα κάθε k- εγγραφές να είναι ίδιες ως προς το αναγνωριστικό τους. Όντως τότε ένας εξωτερικός χρήστης θα είχε το πολύ 1/k πιθανότητα να ανακαλύψει την ταυτότητα μίας εγγραφής.

2.1.2 Μοντέλα του k-anonymization

Με βάση τον ορισμό του k-anonymization απαιτείται ένα μοντέλο με βάση το οποίο θα κατασκευάζουμε μία k-anonymous όψη ενός πίνακα T. Μέχρι τώρα έχουν προταθεί διάφορα μοντέλα, των οποίων τα τρία βασικά χαρακτηριστικά είναι:

- Γενίκευση έναντι Συμπίεσης. Κάποια μοντέλα επιλέγουν να συμπιέσουν όλες τις τιμές μαζί. Αντίθετα κάποια άλλα επιλέγουν να γενικεύσουν τις τιμές του πίνακα, με βάση κάποια ενδιάμεσα στάδια.

- Global έναντι Local Recoding. Κάποια μοντέλα επιλέγουν να πετύχουν το k-anonymity αντιστοιχίζοντας τις τιμές του domain των ιδιοτήτων του quasi-identifier σε κάποιες αλλαγμένες τιμές. Με βάση την ορολογία το τελευταίο καλείται global recoding. Από την άλλη πλευρά, κάποια μοντέλα επιλέγουν να αλλάξουν τις ατομικές τιμές των

δεδομένων τοπικά. Το τελευταίο αναφέρεται ως local recoding. Βασική διαφορά των δύο είναι ότι στο global recoding όλες οι εγγραφές που έχουν τις ίδιες τιμές για το quasi-identifier θα αντιστοιχηθούν στην ίδια τιμή. Αντίθετα στο local recoding αυτό δεν συμβαίνει πάντα και εξαρτάται από την ΄θέση΄ των δεδομένων.

- Hierarchy-based έναντι Partition-based. Τα μοντέλα, τα οποία γενικεύουν τις τιμές, μπορούνε να χωριστούν σε δύο βασικές κατηγορίες, αυτά, τα οποία χρησιμοποιούν κάποια προκαθορισμένη ιεραρχία γενίκευσης τιμών [θα αναφερθούμε σε αυτό εκτενώς στο επόμενο κεφάλαιο] και αυτά, τα οποία θεωρούν το domain των attributes ταξινομημένο και ορίζουν γενικεύσεις χωρίζοντας τον χώρο σε μοναδικά ξένα υποσύνολα. Τα τελευταία μοντέλα, είναι προτιμότερα για αριθμητικά δεδομένα, ενώ τα πρώτα για κατηγορικά δεδομένα.

2.1.3 Σύνοψη-Αλγόριθμοι για το k-anonymization

Το k-anonymity, ως μεθοδολογία, ομαδοποιεί k εγγραφές με κοινό γενικευμένο quasi-identifier. Με αυτόν τον τρόπο ο εξωτερικός χρήστης δεν μπορεί να ανακαλύψει σε ποιόν ανήκει μία εγγραφή. άρα ο αντίπαλος για να σπάσει την ανωνυμία, πρέπει πρώτον να ανακαλύψει ότι το άτομο το οποίο ψάχνει είναι όντως σε μία από τις k εγγραφές και δεύτερον ακόμα και αν γνωρίζει ότι το άτομο τελικά ανήκει σε μία από αυτές, να μπορέσει να κάνει αντιστοίχηση με τα ευαίσθητα δεδομένα του.

Στην συνέχεια της εργασίας θα αναφερθούν τρεις βασικοί αλγόριθμοι για την επίτευξη του k-anonymization. Κάθε ένας από αυτός έχει τα πλεονεκτήματα του και ένα συγκεκριμένο σκοπό. Οι αλγόριθμοι οι οποίοι θα εξεταστούν είναι:

1. Incognito K-Anonymity
2. Mondrian Multidimensional K-Anonymity
3. Utility-Based Anonymization

2.2 Incognito

Οι K. LeFerve, D. J. DeWitt και R. Ramakrishnan του University of Wisconsin πρότειναν τον αλγόριθμο Incognito για να εκδίδουν ένα k-anonymization πίνακα. Ο αλγόριθμος αυτός επιστρέφει όλες τις δυνατές ελάχιστες περιπτώσεις του k-anonymization. Πιο συγκεκριμένα αναζητούν δυνατές γενικεύσεις του πίνακα, ώστε αυτές να είναι k-anonymization του αρχικού πίνακα και ταυτόχρονα απαιτούνε να μην υπάρχει μία λιγότερη γενική λύση εν συγκρίσει με την προηγούμενη. Ο αλγόριθμος χρησιμοποιεί global recoding και είναι hierarchy based.

2.2.1 Βασικοί Ορισμοί

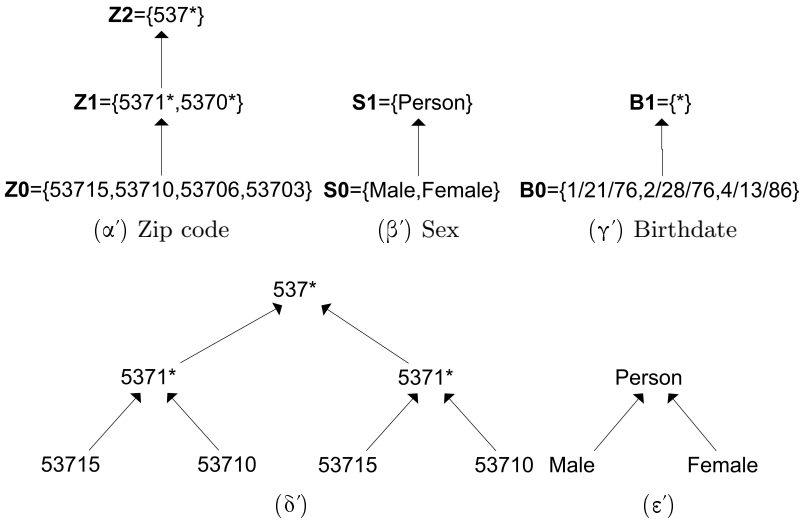
Domain generalization hierarchy

Πριν προχωρήσουμε στην διατύπωση και επεξήγηση του αλγορίθμου, για να είναι δυνατή η γενίκευση των δεδομένων, πρέπει πρώτα να οριστεί τι αποτελεί τελικά ‘γενίκευση κάτι άλλου’. Στον ορισμό του k-anonymization δεν προσδιορίστηκε το μοντέλο γενίκευσης (ή τροποποίησης των δεδομένων). Ο συγκεκριμένος αλγόριθμος έχει επιλέξει να γενικεύει δεδομένα. Για αυτό το λόγο για κάθε attribute (ή σύνολο από attributes) ορίζουμε μία ιεραρχία επιπέδων [ειδικό προς γενικό]. Για παράδειγμα, το φύλο ενός προσώπου είναι μία ειδική περίπτωση του

Patient Data			
Birthdate	Sex	Zip code	Disease
1/21/76	Male	53715	Flu
4/13/86	Female	53715	Hepatitis
2/28/76	Male	53703	Bronchitis
1/21/76	Male	53703	Broken Arm
4/13/86	Female	53706	Sprained Ankle
2/28/76	Female	53706	Hang Nail

Σχήμα 2.2: Ο πίνακας προς μετατροπή.

ατόμου. Κάποιες δυνατές γενικεύσεις για τον πίνακα του Σχήματος 2.2 φαίνονται στο Σχήμα 2.3. Η γενίκευση με βάση κάποιο ιεραρχικό επίπεδο μπορεί να γίνει τόσο για κατηγορικά όσο και για αριθμητικά δεδομένα. Άρα αν επιθυμούμε να γενικεύσουμε μία τιμή, δεν έχουμε παρά να επιλέξουμε κάποιο πρόγονό του.



Σχήμα 2.3: Domain γενίκευση και γενίκευση τιμή για Zip code(a,b), Birthdate(c,d) και Sex(e,f)

Αρχικά θα προσπαθήσουμε να ορίσουμε τι αποτελεί γενίκευση μίας ατομικής τιμής. Για παράδειγμα στο Σχήμα 2.3 η τιμή Male μπορεί να 'γενικευτεί' στην τιμή person. Δίνουμε λοιπόν παρακάτω μερικές βασικούς ορισμούς πράξεων και τον ορισμό της Domain generalization hierarchy:

Ορισμός 1. *Domain generalization : Το domain D_j είναι γενίκευση του domain D_i (με $D_i \neq D_j$) αν για όλες τις δυνατές τιμές του D_i υπάρχει μία ή περισσότερες τιμές στο D_j ώστε αυτή ή αυτές να αποτελεί γενίκευση των τιμών του D_i.*

Για παράδειγμα, στο Σχήμα 2.3 το Domain S_1 αποτελεί γενίκευση του S_0.
Ορίζουμε κάποιες βασικές πράξεις μεταξύ των domains:

Ορισμός 2. *Domain generalization σχέση $=$. Αν τα domain D_i και D_j είναι τα ίδια σύνολα τότε γράφουμε $D_i = D_j$.*

Ορισμός 3. *Domain generalization σχέση $<_D$. Αν το domain D_j αποτελεί γενίκευση του domain D_i τότε το συμβολίζουμε με $D_i <_D D_j$.*

Για παράδειγμα στο Σχήμα 2.3 ισχύει $Z1 <_D Z2$. Σημείωση: Η σχέση $<_D$ είναι μεταβατική, δηλαδή αν $D_i <_D D_k$ και $D_k <_D D_j$ τότε $D_i <_D D_j$.

Ορισμός 4. *Domain generalization σχέση $\leq$. Αν το Domain $D_i <_D D_j$ ή τα δύο domain είναι ίδια τότε το συμβολίζουμε με $D_i \leq_D D_k$.*

Ορισμός 5. *Value generalization function. Η συνάρτηση $\gamma : D_i \rightarrow D_j$ η οποία συσχετίζει κάθε γενίκευση domain $D_i <_D D_j$ ονομάζεται value generalization function.*

Για παράδειγμα στο Σχήμα 2.3 μία συνάρτηση γ θα συσχέτιζε τις τιμές του $Z0$ με αυτές του $Z1$. Θα μπορούσε βέβαια να συσχετίσει τις τιμές του ZO με του $Z2$.

Με βάση όλους τους παραπάνω ορισμούς είμαστε τώρα σε θέση να ορίσουμε τι είναι μία ιεραρχία domain η οποία μας επιτρέπει να γενικεύσουμε ατομικές τιμές.

Ορισμός 6. *Domain generalization hierarchy: ορίζεται ως ένα σύνολο από domains τα οποία είναι ταξινομημένα με βάση την σχέση $<_D$*

Μπορεί κάποιος να φανταστεί μία τέτοια ιεραρχία σαν ένα σύνολο από δέντρα, στο οποίο δεν έχουμε απαραίτητα μία κορυφή. Ένα τέτοιο παράδειγμα είναι αυτό του Σχήματος 2.3 (α,β και γ).

Δίνουμε τέλος μερικούς ακόμα ορισμούς οι οποίοι θα μας χρειαστούν παρακάτω.

Ορισμός 7. *Direct generalization: Το D_j λέγεται direct generalization του D_i αν $D_i <_D D_j$ και δεν υπάρχει D_k τέτοιο ώστε $D_i <_D D_k <_D D_j$.*

Ορισμός 8. *Implied generalization: Το D_j λέγεται implied generalization του D_i αν $D_i <_D D_j$ και υπάρχει D_k τέτοιο ώστε $D_i <_D D_k <_D D_j$.*

Αν φανταστούμε λοιπόν την domain generalization hierarchy σαν ένα (κατευθυνόμενο) δέντρο, αν υπάρχει μία ακμή που ενώνει το D_i με το D_j τότε το D_j είναι direct generalization του D_i. Αντίθετα αν υπάρχει μονοπάτι στον κατευθυνόμενο γράφο που να ενώνει το D_i με το D_j αλλά όχι άμεση ακμή που να ενώνει το D_i με το D_j τότε το D_j είναι implied generalization του D_i. Ένας τέτοιος γράφος είναι αυτός του Σχήματος 2.3.

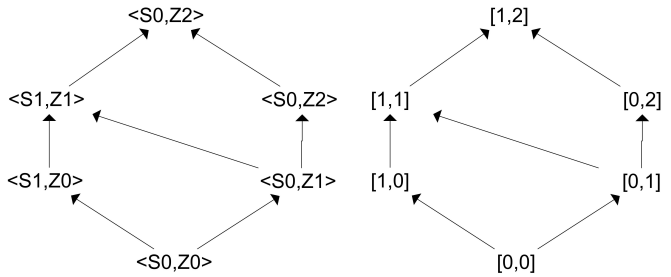
Ορισμός 9. *γ_+: Τέλος χρησιμοποιούμε τον συμβολισμό $\gamma+$ σαν συντόμευση γιατί την σύνθεση μίας ή περισσότερων value generalization functions που παράγουν είτε direct είτε implied generalizations.*

Multi-attribute generalization lattice

Μέχρι τώρα έχουμε αναφερθεί όμως μόνο σε ιεραρχία ατομικών τιμών. Για παράδειγμα σε ένα quasi-identifier κάθε attribute έχει την δικιά του ιεραρχία. Εμείς όμως δεν θέλουμε απλά να γενικεύσουμε μία μία τιμή αλλά το σύνολο τιμών όλου του quasi-identifier. Έτσι τα domain generalization hierarchies των ατομικών attributes μπορούνε να συνδυαστούν για να παράγουν ένα multi-attribute generalization lattice (δηλαδή ένα δίκτυο γενίκευσης πολλών ιδιοτήτων). Για παράδειγμα ένα multi-attribute generalization lattice δίνεται στο Σχήμα 2.4.

Παρακάτω δίνονται οι δύο βασικοί ορισμοί για ορίσουμε ένα τέτοιο δίκτυο.

Ορισμός 10. *Direct multi-attribute domain generalization. Έστω ένα διάνυσμα από n domains hierarchies $< H_1, \ldots, H_n >$. Ένα διάνυσμα από n domains $< D_{b_1}, \ldots, D_{b_n} >$ λέμε ότι είναι direct multi-attribute domain generalization ενός διανύσματος $< D_{a_1}, \ldots, D_{a_n} >$ εάν ισχύουν και οι δύο παρακάτω συνθήκες:*



Σχήμα 2.4: Ένα multi-attribute lattice για το Sex και Zip code και το αντίστοιχο Distance Vector

1. *Υπάρχει μία μοναδική τιμή j από $1 \ldots n$ τέτοια ώστε το domain hierarchy H_j να περιέχει την ακμή $D_{a_j} \rightarrow D_{b_j}$ (δηλαδή το D_{b_j} να είναι είτε direct είτε implied generalization του D_{a_j} με βάση την domain generalization H_j).*

2. *Για κάθε άλλη τιμή του i από $1 \ldots n$ τέτοια ώστε $i \neq j$ ισχύει $D_{a_i} = D_{b_i}$.*

Ορισμός 11. *Implied multi-attribute domain generalization. Έστω ένα διάνυσμα από n domains hierarchies $< H_1, \ldots, H_n >$. Ένα διάνυσμα από n domains $b =< D_{b_1}, \ldots, D_{b_n} >$ λέμε ότι είναι implied multi-attribute domain generalization ενός διανύσματος $a =< D_{a_1}, \ldots, D_{a_n} >$, αν υπάρχουν l διανύσματα από n domains $k_i =< D_{k_{i_1}}, \ldots, D_{k_{i_n}} >$ τέτοιο ώστε για κάθε $2 < i <= l$ το $< D_{k_{i_1}}, \ldots, D_{k_{i_n}} >$ να είναι direct multi-attribute domain generalization του $< D_{k_{i-1_1}}, \ldots, D_{k_{i-1_n}} >$, το $< D_{k_{1_1}}, \ldots, D_{k_{1_n}} >$ να είναι direct multi-attribute domain generalization του $< D_{a_1}, \ldots, D_{a_n} >$ και το $< D_{b_1}, \ldots, D_{b_n} >$ να είναι direct multi-attribute generalization του $< D_{k_{l_1}}, \ldots, D_{k_{l_n}} >$.*

Ορίζουμε λοιπόν τώρα μία ιεραρχία πλειότιμων τιμών, με βάση ένα δίκτυο.

Ορισμός 12. *Multi-attribute generalization lattice. Ένα multi-attribute generalization lattice σε ένα n μόνο-ιδιοτήτων domain generalization hierarchies είναι ένα ολοκληρωμένο δίκτυο στο οποίο :*

1. *Κάθε ακμή είναι μία direct multi attribute domain generalization σχέση*

2. *Το στοιχείο της βάσης είναι ένα n-διάνυσμα $< D_{a_1}, \ldots, D_{a_n}{}' >$ για τα οποία για κάθε i, το D_{a_i} είναι η βάση της ιεραρχικής αλυσίδας.*

3. *Το στοιχείο της κορυφής είναι ένα n-διάνυσμα $< D_{a_1}, \ldots, D_{a_n} >$, για το οποίο για κάθε i, το D_{a_i} είναι το τέλος της ιεραρχικής αλυσίδας*

Ένα τέτοιο lattice δίνεται στο Σχήμα 2.4, εύκολα μπορεί να δει κανείς ότι έχουμε μία βάση και μία κορυφή και ότι όταν υπάρχει μία ακμή που να ενώνει δύο κόμβους τότε έχουμε direct αλλιώς implied.

Ορισμός 13. *Distance Vector. Το distance vector μεταξύ δύο διανυσμάτων $< D_{a_1}, ..., D_{a_n} >$ και $< D_{b_1}, \ldots, D_{b_n} >$ είναι ένα διάνυσμα $DV = [d_1, \ldots, d_n]$ όπου κάθε τιμή d_i δηλώνει το μήκος του μονοπατιού μεταξύ του D_{a_i} και D_{b_i} με βάση κάποιο domain generalization hierarchy H_i.*

Σημείωση: Ένα δίκτυο (lattice) από distance vectors μπορεί να έχει ως αρχή την αρχή του generalization domain hierarchy. Ένα τέτοιο lattice δίνεται στο Σχήμα 2.4.

Full-Domain Generalization

Είμαστε λοιπόν τώρα έτοιμοι να ορίσουμε τι αποτελεί ¨γενίκευση' και να δώσουμε μερικές ιδιότητες της.

Ορισμός 14. *Full-Domain generalization. Έστω ένας πίνακας T και ένα σύνολο quasi-identifier ιδιοτήτων $Q_1, \ldots, Q_n$. Ένα full-domain generalization ορίζεται ως ένα σύνολο συναρτήσεων $\phi_1, \ldots, \phi_n$ τέτοια ώστε κάθε μία να είναι της μορφής: $D_{q_i} \rightarrow D_{a_i}$ όπου $D_{q_i} <_D D_{a_i}$. Το ϕ_i δηλαδή αντιστοιχεί κάθε τιμή $q \in D_{q_i}$ σε κάποιο $a \in D_{a_i}$ τέτοιο ώστε $a = q$ ή $a \in \gamma^+(q)$. Ένα full-domain generalization V του T μπορεί να ¨αποκτηθεί' αντικαθιστώντας κάθε τιμή q της ιδιότητας Q σε κάθε εγγραφή του T με την τιμή $\phi_i(q)$.*

Με άλλα λόγια αν θέλουμε να γενικεύσουμε κάποιες τιμές του quasi-identifier ενός πίνακα, δεν έχουμε παρά να ανέβουμε σε κάποιο υψηλότερο ιεραρχικό επίπεδο με βάση το δίκτυο που πριν ορίσαμε.

Δίνονται μερικές ιδιότητες:

Θεώρημα 1. *Generalization property. Έστω ένας πίνακας T και έστω P και Q δύο σύνολα από ιδιότητες στο T τέτοια ώστε $D_p <_D D_q$. Αν το T είναι k-anonymous με βάση το P τότε το T είναι k-anonymous με βάση το Q.*

Η παραπάνω ιδιότητα, λέει ότι εάν μία γενίκευση του πίνακα T είναι k-anonymous τότε οποιαδήποτε γενίκευση αυτής της γενίκευσης θα είναι οπωσδήποτε k-anonymous. Επί της ουσίας δηλαδή αν καταφέρουμε να βρούμε μία γενίκευση, δεν έχουμε λόγο να εξετάσουμε τις γενικεύσεις αυτής.

Θεώρημα 2. *Rollup property. Έστω ένας πίνακας T και έστω P και Q δύο σύνολο ιδιοτήτων τέτοια ώστε $D_p \leq D_q$. Εάν έχουμε το frequency set f_1 του T με βάση το P, τότε μπορούμε να βρούμε κάθε τιμή στο f_2, το οποίο είναι το frequency του T με βάση το Q αθροίζοντας κάθε σύνολο τιμών στο f_1 που συσχετίζονται με την συνάρτηση γ με κάθε τιμή στο f_2.*

Η παραπάνω ιδιότητα, μας επιτρέπει επί της ουσίας να γλυτώσουμε αρκετό υπολογιστικό κόστος, για να δούμε αν ικανοποιείται το k-anonymity. Η αξία της ιδιότητας θα γίνει πιο σαφής στον αλγόριθμο.

Θεώρημα 3. *Subset property. Έστω ένα πίνακας και ένα σύνολο Q από ιδιότητες στο T. Αν το T είναι k-anonymous με βάση το Q, τότε το T είναι k-anonymous με βάση κάθε σύνολο P από ιδιότητες έτσι ώστε $P \subseteq Q$.*

Η παραπάνω ιδιότητα, μας επιτρέπει επί της ουσίας να απορρίπτουμε λύσεις, χωρίς να αναγκαστούμε να εξετάζουμε όλες τις ιδιότητες και όλους τους κόμβους στο δίκτυο γενίκευσης.

Είναι προφανές λοιπόν ότι για να επιτύχουμε το k-anonymity αναζητάμε μία Full-Domain generalization ενός πίνακα T, η οποία να μας δίνει μία όψη η οποία να είναι k-anonymous.

2.2.2 Ο αλγόριθμος Incognito

Ο αλγόριθμος incognito λοιπόν, με βάση ένα full-domain generalization παίρνει ως είσοδο ένα πίνακα T και εξάγει μία όψη του, τέτοια ώστε να είναι k-anonymization αυτού. Το k-anonymity επιτυγχάνεται γενικεύοντας τα δεδομένα με βάση το full domain generalization. Τέλος ο αλγόριθμος θα αναζητήσει τις ελάχιστες δυνατές γενικεύσεις, αφού με βάση το generalization property κάθε άλλη γενίκευση ικανοποιεί το k-anonymity. (Ο λόγος για τον οποίο ο αλγόριθμος αναζητά τις ελάχιστες γενικεύσεις είναι προφανής, η ελάχιστη απώλεια πληροφορίας).

Algorithm 1 Incognito Algorithm

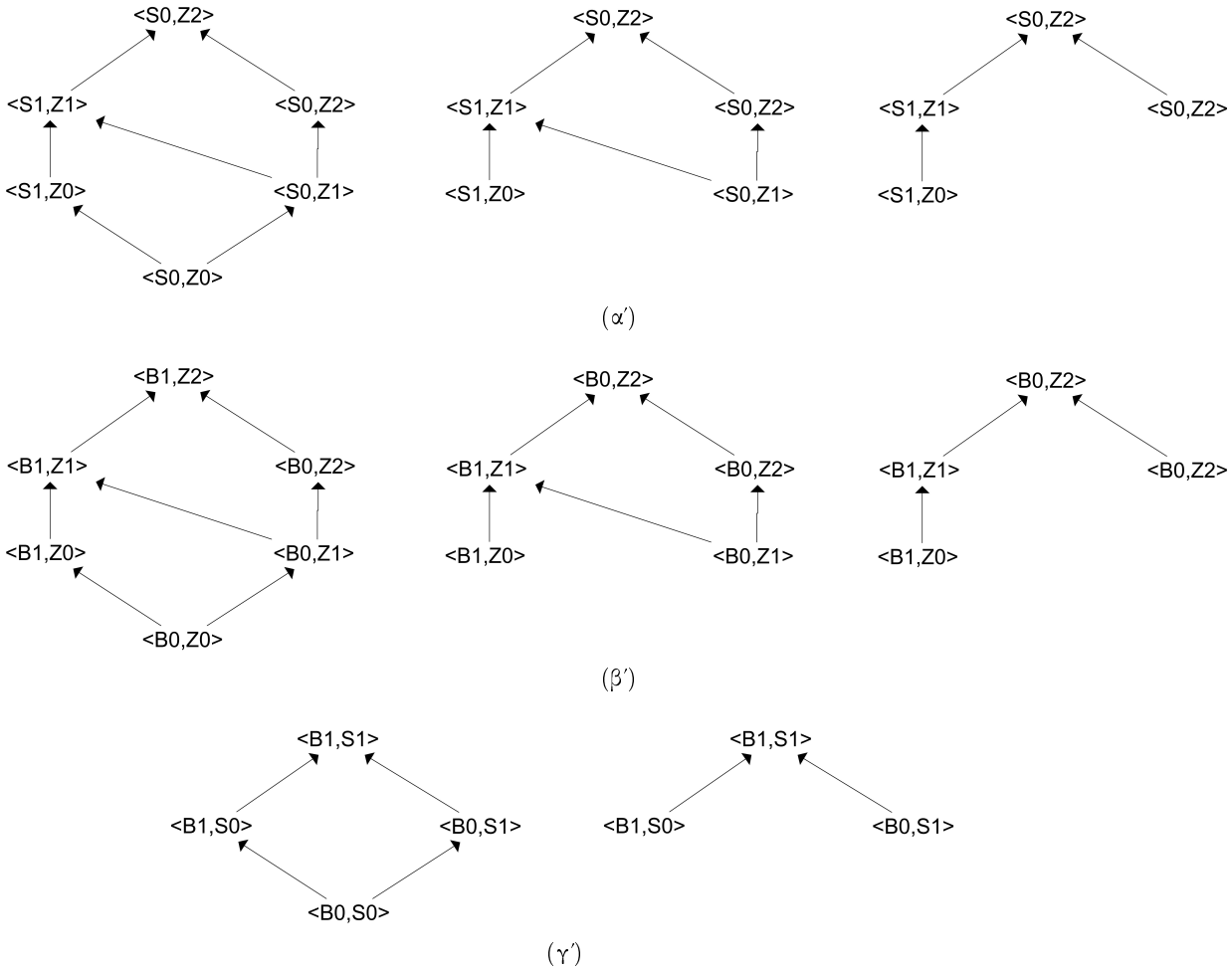
```
Input: A table T to be k-anonymized, a set Q of n quasi-identifier attributes, and a set of dimension
tables (one for each
quasi-identifier in Q)
Output: The set of k-anonymous full-domain generalizations of T
C1 = Nodes in the domain generalization hierarchies of attributes in Q
E1 = Edges in the domain generalization hierarchies of attributes in Q
queue = an empty queue
for i = 1 to n do
  //Ci and Ei define a graph of generalizations
  Si = copy of Ci
  roots = all nodes E Ci with no edge E Ei directed to them
  Insert roots into queue, keeping queue sorted by height
  while queue is not empty do do
    node = Remove first item from queue
    if node is not marked then
      if node is a root then
        frequencySet = Compute frequency set of T with respect to attributes of node using T.
      else
        frequencySet = Compute frequency set of T with respect to attributes of node using parent's
        frequency set.
      end if
      Use frequencySet to check k-anonymity with respect to attributes of node
      if T is k-anonymous with respect to attributes of node then
        Mark all direct generalizations of node
      else
        Delete node from Si
        Insert direct generalizations of node into queue, keeping queue ordered by height
      end if
    end if
  end while
  Ci+1,Ei+1 = GraphGeneration(Si, Ei)
end for
return Projection of attributes of Sn onto T and dimension tables
```

Δίνεται λοιπόν ο αλγόριθμος σε ψευδοκώδικα (αλγόριθμος 1). Ο αλγόριθμος σε κάθε iteration με βάση το subset property ξεκινάει ελέγχοντας μονοδιάστατες ιδιότητες του quasi-identifier και εν συνεχεία σε κάθε επανάληψη το k-anonymity με βάση μεγαλύτερα subsets, τα οποία αυξάνουν σε κάθε επανάληψη. Σε κάθε επανάληψη i κατασκευάζεται ένας γράφος από πιθανές multi-attribute generalizations κόμβους με βάση όλα τα υποσύνολα του quasi-identifier μεγέθους i. Για την ακρίβεια κάθε επανάληψη αποτελείται από δύο μέρη:

1. Σε κάθε επανάληψη i κατασκευάζεται ένας γράφος από πιθανές multi-attribute generalizations κόμβους με βάση όλα τα υποσύνολα του quasi-identifier μεγέθους i. Τα σύνολα των υποψήφιων κόμβων που θα εξεταστούν τα συμβολίζουμε με C_i. Το σύνολο των direct multi-attribute generalization ακμών οι οποίες συνδέουν αυτούς τους κόμβους το συμβολίζουμε με E_i. Έτσι μία παραλλαγή του breadth-first αναζήτησης κατασκευάζει ένα σύνολο S_i στο οποίο ανήκουν τα multi-attribute generalizations με βάση τα οποία το T είναι k-anonymous.

2. Με την κατασκευή του S_i ο αλγόριθμος κατασκευάζει το σύνολο των υποψήφιων κόμβων μεγέθους $i+1$ (C_{i+1}) και το σύνολο των ακμών που τους ενώνουν (E_{i+1}) με βάση το subset property.

Πιο συγκεκριμένα στο πρώτο μέρος της επανάληψης i ο Incognito με βάση μία αναζήτηση αποφαίνεται αν ο πίνακας T είναι k-anonymous με βάση όλες τις υποψήφιες γενικεύσεις του



Σχήμα 2.5: Τα δίκτυα που κατασκευάζει ο αλγόριθμος στην δεύτερη επανάληψη.

ID	dim_1	$index_1$	dim_2	$index_2$
1	SEX	0	Zip code	0
2	SEX	1	Zip code	0
3	SEX	0	Zip code	1
4	SEX	1	Zip code	1
5	SEX	0	Zip code	2
6	SEX	1	Zip code	2

(α′) Nodes

Start	End
1	2
1	3
2	4
3	4
3	5
4	6
5	6

(β′) Edges

Σχήμα 2.6: Το δίκτυο, ορισμένο από τους κόμβους και τις ακμές

C_i, όπου κάθε σύνολο είναι μεγέθους i. Αυτό επιτυγχάνεται με μία παραλλαγμένη bottom-up breadth first αναζήτηση, ξεκινώντας από κάθε κόμβο του γράφου ο οποίος δεν είναι direct generalization κάποιου άλλου κόμβου και ανεβαίνει σταδιακά 'ιεραρχικά' επίπεδα. Ο αλγόριθμος εκμεταλλεύεται δύο ιδιότητες που είχαμε ορίσει πιο πριν. Η πρώτη είναι αυτή της rollup, σύμφωνα με την οποία ο αλγόριθμος δεν είναι πάντα αναγκαίο να περάσει ξανά τον πίνακα για να ανακαλύψει το frequency set ενός κόμβου και άρα να ελέγξει το k-anonymity. Η δεύτερη είναι η generalization property. Αν με βάση ένα κόμβο ικανοποιείται το k-anonymity τότε δεν έχουμε λόγο να ελέγξουμε όλα τα ανώτερα επίπεδα και άρα ο αλγόριθμος μπορεί να τα αγνοήσει. Σημείωση: Ο αλγόριθμος θα ελέγξει όλες τις δυνατές i-ιδιότητες γενικεύσεις του C_i. Για παράδειγμα, έστω ξανά ο πίνακας του Σχήματος 2.2, με quasi identifier $< Birthdate, Sex, Zipcode >$. Στην πρώτη επανάληψη του Incognito ο αλγόριθμος βρίσκει ότι ο πίνακας T είναι k-anonymous με βάση τα $< B0 >$, $< S0 >$ και $< ZO >$ και άρα με όλες τις δυνατές γενικευμένες τιμές που ορίζεται από τα domain αυτών. Στην δεύτερη επανάληψη θα ελέγξει εάν είναι k-anonymous για τα multi-attribute generalizations των $< Birthdate, Sex >$, $< Birthdate, Zipcode >$ και $< Sex, Zipcode >$. Το Σχήμα 2.5 δείχνει ποιοι κόμβοι απορρίπτονται και ποιοι παραμένουν, ώστε να έχουμε το τελικό lattice. Για παράδειγμα, σε πρώτη φάση ο αλγόριθμος κατασκευάζει το frequency set του $< SO, ZO >$ και ανακαλύπτει ότι δεν ικανοποιεί το k-anonymity. Μετά με βάση την rollup ιδιότητα βρίσκει το frequency set των $< S1, Z0 >$ και $< S0, Z1 >$. Με βάση το $< S1, Z0 >$ ο πίνακας ικανοποιεί το k-anonymity και άρα όλες οι γενικεύσεις του $< S1, Z0 >$ το επιτυγχάνουν αυτό. Αντίθετα με βάση το $< S0, Z1 >$ δεν ικανοποιείται το k-anonymity. Ο επόμενος κόμβος που θα ελεγχθεί είναι ο $< S0, Z2 >$, αφού ο $< S1, Z1 >$ γνωρίζουμε ότι ικανοποιεί του k-anonymity ως direct generalization του $< S1, Z0 >$. Με βάση λοιπόν το frequency set του $< S0, Z2 >$ ικανοποιείται το k-anonymity και άρα δεν έχουμε άλλους έλεγχους να κάνουμε και σταματάει η αναζήτηση.

Το δεύτερο μέρος της επανάληψης έχει ως στόχο να κατασκευάσει τα σύνολο C_i και E_i στην επανάληψη $i-1$. Με άλλα λόγια σε κάθε επανάληψη προσπαθούμε να υλοποιήσουμε ένα multi-attribute generalization γράφο. Για αυτό χρησιμοποιούνται δύο σχεσιακοί πίνακες, ένας για τους κόμβους και ένας για τις ακμές. Έστω για παράδειγμα το Σχήμα 2.6. Σε κάθε κόμβο έχει ανατεθεί ένα μοναδικό αναγνωριστικό (ID - unique identifier).

Το πρόβλημα είναι λοιπόν είναι η υλοποίηση του γράφου, η οποία γίνεται σε τρεις φάσεις. Στη αρχή έχουμε μία φάση join μετά μία prune φάση ώστε να υλοποιηθεί το σύνολο των υποψήφιων κόμβων C_i τα οποία θα μπορούσαν (με βάση προηγούμενες επαναλήψεις) να οδηγήσουν σε k-anonymity. Στην τρίτη και τελευταία φάση κατασκευάζεται ο πίνακας με τις ακμές, δηλαδή όλες οι direct multi-attribute generalization σχέσεις.

Η φάση join: Κατασκευάζεται ένα υπερσύνολο του C_i με βάση το S_i. Να σημειωθεί ότι

απαιτείται για να υλοποιηθεί το join τα στοιχεία να έχουν κάποια μορφή ταξινόμησης. Το join είναι λοιπόν της παρακάτω μορφής :

Algorithm 2 Join Phase

INSERT INTO $C_i(dim_1, index_1, ..., dim_i, index_i, parent_1, parent_2)$
SELECT $p.dim_1, p.index_1, ..., p.dim_{i-1}, p.index_{i-1}, q.dim_{i-1}, q.index_{i-1}, p.ID, q.ID$
FROM $S_{i-1}p, S_{i-1}q$
WHERE $p.dim_1 = q.dim_1 \wedge p.index_1 = q.index_1 \wedge ... \wedge p.dim_{i-2} = q.dim_{i-2} \wedge p.index_{i-2} = q.index_{i-2} \wedge p.dim_{i-1} < q.dim_{i-1}$

Εν συνεχεία εφαρμόζουμε την prune φάση, στην οποία αφαιρούμε κόμβους οι οποίοι δεν χρειάζονται. Η διαδικασία είναι απλή. Ένα σύνολο ιδιοτήτων s με πληθάριθμο n θα περιέχεται στο S_i αν και μόνο στο S_{i-1} περιέχονται όλα τα δυνατά υποσύνολα του s με πληθάριθμο $n-1$. Βασισμένοι σε αυτή την ιδιότητα μπορούμε να αφαιρέσουμε όλα τα επιπλέον σύνολα που παράχθηκαν στην φάση του join. Μάλιστα με τη χρήση ενός hash tree μπορούμε αυτήν την διαδικασία να την κάνουμε αρκετά γρήγορα.

Ο λόγος που αυτά τα δύο παραπάνω μας παράγουν το C_i είναι προφανής. Επί της ουσίας είναι ο apriori αλγόριθμος, και έχουμε εκμεταλλευτεί το subset και generalization property. Γνωρίζουμε ότι αν ο πίνακας δεν είναι k-anonymous με βάση ένα σύνολο P τότε δεν είναι μπορεί να είναι k-anonymous με βάση ένα σύνολο Q αν $Q \leq_D P$. Πως εκμεταλλευόμαστε αυτό όμως στην υλοποίηση του C_i; Στην επανάληψη $i-1$ έχουμε Q μεγέθους $i-1$, έστω το $< q_1, ..., q_{i-1} >$ το οποίο δεν υπάρχει στο S_{i-1} (δηλαδή με βάση αυτό το T δεν είναι k-anonymous). Τότε κάθε σύνολο της μορφής $< q_1, ..., q_{i-1}, t >$ όπου t η αξία μίας νέας διάστασης δεν μπορεί να μας οδηγήσει σε k-anonymity. άρα αν έχουμε τα στοιχεία ταξινομημένα με βάση το ύψος τότε μπορούμε με το join σε πρώτη φάση και σε δεύτερη το prune να κρατήσουμε μόνο όσους κόμβους έχουν όντως πιθανότητα να μας οδηγήσουν σε k-anonymity.

Στην τρίτη φάση πρέπει να υλοποιηθούν όλες οι ακμές. Αυτό στηρίζεται στην εξής παρατήρηση, ότι ένας κόμβος A είναι generalization ενός άλλου B όταν συμβαίνει ένα από τα δύο:

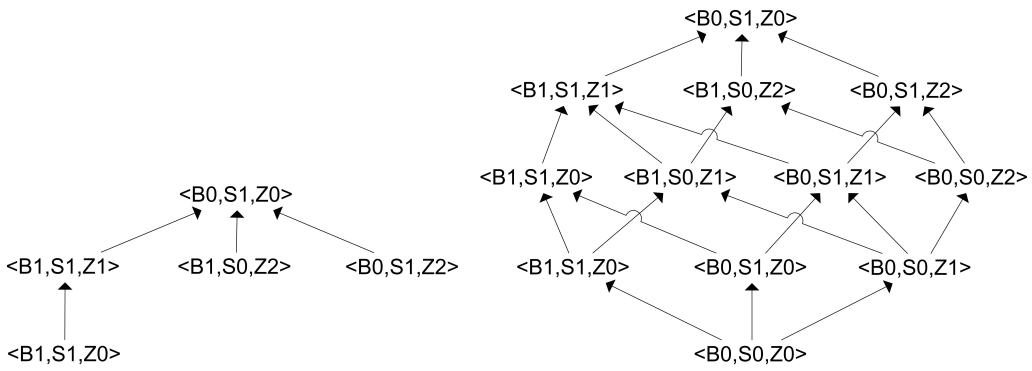
1. Οι γονείς του B είναι γενικεύσεις του γονείς του A.

2. Ένας από τους δύο γονείς του B είναι γενίκευση του αντίστοιχου γονέα του A και οι άλλοι δύο είναι ίσοι. Όλες οι implied generalization σχέσεις αφαιρούνται στο τέλος.

Δίνεται η διαδικασία εκφρασμένη σε sql.

Algorithm 3 Prune phase

INSERT INTO E_i(start, end)
WITH CandidateEdges (start, end) AS (
SELECT p.ID, q.ID
FROM C_i p, C_i q, E_{i-1} e, E_{i-1} f
WHERE $(e.start = p.parent_1 \wedge e.end = q.parent1 \wedge f.start = p.parent_2 \wedge f.end = q.parent2) \vee (e.start = p.parent_1 \wedge e.end = q.parent_1 \wedge p.parent_2 = q.parent_2) \vee (e.start = p.parent_2 \wedge e.end = q.parent_2 \wedge p.parent_1 = q.parent_1)$
)
SELECT D.start, D.end
FROM CandidateEdges D
EXCEPT
SELECT $D_1.start, D_2.end$
FROM CandidateEdges D_1, CandidateEdges D_2
WHERE $D_1.end = D_2.start$



(α΄) Ο γράφος που κατασκευάζει ο αλγόριθμος

(β΄) Όλος ο γράφος

Σχήμα 2.7: Αφαίρεση κόμβων μέσω των join και prune φάσεων

Χάρη στο apriori αλγόριθμο γλιτώνουμε ένα σημαντικό πλήθος ελέγχων. Για παράδειγμα στο Σχήμα 2.7, στην μία περίπτωση δίνεται το δίκτυο το οποίο θα εξεταστεί στην επόμενη επανάληψη με βάση την συγκεκριμένη υλοποίηση και το δίκτυο το οποίο θα εξεταζόταν αν δεν είχαμε αυτές τις τρεις φάσης.

2.2.3 Βελτιστοποιήσεις του αλγορίθμου

Στον αλγόριθμο μπορούνε να γίνουνε και κάποιες βελτιστοποιήσεις.

- Super -roots: Ένας κόμβος v στο C_i είναι ρίζα όταν δεν υπάρχει καμία generalization ακμή στο E_i από κάποιο άλλο κόμβο στο C_i στο v. Σε κάθε επανάληψη του αλγορίθμου η βάση σκανάρετε για κάθε ρίζα ώστε να βρούμε το frequency set, το frequency set όλων των άλλων κόμβων υπολογίζεται με βάση την ρίζα. Όμως λόγω του αλγορίθμου δεν είναι σίγουρο ότι όλοι οι κόμβοι θα συμμετέχουν στο γράφο, άρα υπάρχει το ενδεχόμενο κάποιες ρίζες να ανήκουν στην ίδια οικογένεια (γενικεύσεις του ίδιου quasi-identifier υποσυνόλου). Σε αυτή την περίπτωση είναι πιο αποδοτικό να υπολογιστεί το frequency set όλης της οικογένειας με βάση το super-root και μετά με βάση αυτό των roots. Για παράδειγμα στο Σχήμα 2.7 (α), τα $< B1, S1, Z0 >, < B1, S0, Z2 >$ και $< B0, S1, Z2 >$ είναι όλα ρίζες σε αυτόν των 3-ιδιοτήτων γράφο. Παρ΄ όλα αυτά όλα προέρχονται από την ίδια οικογένεια, η οποία όμως ΄διαγράφτηκε΄ σε κάποιο βήμα του αλγορίθμου. Η οικογένεια αυτή είναι η $< B0, S0, Z0 >$. Έτσι ο αλγόριθμος υπολογίζει πρώτα το frequency set αυτού του κόμβου σκανάροντας την βάση και μετά όλων των υπολοίπων, χωρίς όμως πια να ξαναχρειαστεί να σκανάρει την βάση και μάλιστα τρεις φορές. Αντίθετα θα χρησιμοποιήσει το frequency set του super root

- Bottom-Up Pre-computation.:

  Το πρόβλημα με τον apriori αλγόριθμο είναι ότι για να βρεθεί το frequency set πρέπει να σκανάρουμε όλη την βάση ξεκινώντας από τις ιδιότητες μία μία και ανεβαίνοντας σταδιακά προς τα πάνω σε σύνολα αυτών. Δεν γίνεται όμως για παράδειγμα να εκμεταλλευτούμε το frequency set του $< Zipcode >$ για να υπολογίσουμε το frequency του $< Sex, Zipcode >$. Μπορούμε όμως να κάνουμε όμως το αντίστροφο με την roll up ιδιότητα. Βασισμένοι λοιπόν σε αυτή την παρατήρηση είναι δυνατόν να υπολογίσουμε πρώτα όλα τα frequency sets του T για όλα τα δυνατά υποσύνολα του quasi-identifier

στο χαμηλότερο επίπεδο της γενίκευσης. Αυτά τα υποσύνολα μπορούνε να υπολογιστούν με μία bottom-up διαδικασία (π.χ. πρώτα του $< Sex, Zipcode >$ και μετά του $< Zipcode >$). Όταν υπολογίσουμε λοιπόν τα μικρότερα υποσύνολα, εν συνεχεία θα υπολογίσουν όλα τα υπόλοιπα με βάση αυτά χωρίς να χρειαστεί να ξανά σκανάρουμε την βάση.

2.2.4 Σχόλια

- Ο incognito έχει το πλεονέκτημα ότι υπολογίζει όλα τις δυνατές k-anonymous όψεις ενός πίνακα βασισμένος σε μία ιεραρχία γενικεύσεων σε αρκετά ικανοποιητικό χρόνο.

- Το μειονέκτημα είναι ότι στην πραγματικότητα απαιτείται μία k-anonymous όψη και όχι όλες, άρα ο υπολογισμός όλων των δυνατών k-anonymous όψεων είναι χρονοβόρος.

- Το άλλο μειονέκτημα του incognito είναι ότι δεν μας δίνει καμία ένδειξη για την απώλεια πληροφορίας. Όπως θα δούμε και στους επόμενους αλγορίθμους δεν μας ενδιαφέρει απλά να βρούμε μία k-anonymous όψη αλλά να βρούμε μία k-anonymous όψη η οποία να έχει την όσο δυνατόν λιγότερη απώλεια πληροφορίας. Έτσι για να επιλέξουμε κάποια από τις δυνατές λύσεις που μας παρέχει ο incognito πρέπει να ελέγξουμε όλες τις δυνατές λύσεις για να βρούμε την βέλτιστη, εισάγοντας μας έτσι ακόμα μεγαλύτερη χρονική πολυπλοκότητα.

- Ο incognito προϋποθέτει ότι υπάρχει ήδη μία ιεραρχία που ορίζει τις γενικεύσεις. Δεν τον ενδιαφέρει πως έχει προκύψει αυτή και με ποιο υπολογιστικό κόστος. Αυτό μπορεί να είναι ιδιαίτερα αρνητικό για αριθμητικά δεδομένα. Στα αριθμητικά δεδομένα συνήθως επιτυγχάνουμε την γενίκευση κάνοντας μία τιμή range και γενικότερα ένα range γενικεύεται επεκτείνοντας τα άκρα του. Είναι προφανές ότι το άκρα του range δεν είναι απαραίτητο να είναι μεγαλύτερα και μικρότερα αντίστοιχα από ότι την μέγιστη και ελάχιστη τιμή αντίστοιχα. Μία ιεραρχία δεδομένων η οποία έχει οριστεί πριν αναζητήσουμε γενικεύσεις δεν μπορεί να το επιτύχει αυτό.

2.3 Mondrian Multidimensional K-Anonymity

Όπως είδαμε στην προηγούμενη ενότητα ο αλγόριθμος incognito κατασκευάζει όλες τις k-anonymous όψεις ενός πίνακα, βασισμένος σε ένα δίκτυο ιεραρχίας τιμών. Ο αλγόριθμος αυτός έχει το μειονέκτημα όμως ότι ήταν αρκετά αργός καθώς και ότι δεν μας έδινε κάποιο μέτρο για την απώλεια πληροφορίας για την κάθε γενίκευση. Μάλιστα το πρώτο, τον καθιστά απαγορευτικό προς εκτέλεση. Για αυτό το λόγο προτάθηκε ένα νέο πολυδιάστατο μοντέλο γενίκευσης καθώς και ένας greedy αλγόριθμος για την εν μέρει επίλυση του προβλήματος. Με βάση αυτό το μοντέλο, θεωρούμε ότι έχουμε ένα χώρο μ-διαστάσεων (όπου μ ο πληθάριθμος του quasi identifier). Χωρίζοντας αυτόν τον χώρο σε partitions, αναζητούμε μία k-anonymous όψη. Το πρόβλημα (το οποίο θα αναλυθεί μετά) είναι NP-hard και για αυτό έχει προταθεί ο αλγόριθμος Mondrian ο οποίος μπορεί να μην δίνει την βέλτιστη λύση, δίνει όμως μία αρκετά ικανοποιητική εν συγκρίσει με τα άλλα μοντέλα που έχουν προταθεί και σε αρκετά ικανοποιητικό χρόνο ($O(nlogn)$). Πρόβλημα αυτού του αλγορίθμου και μοντέλου, παραμένει ότι δεν εξετάζει την απώλεια της πληροφορίας.

2.3.1 Βασικοί ορισμοί

Μετρικές Ποιότητας Γενίκευσης

Όπως θα δούμε αργότερα, ο greedy αλγόριθμος δεν αναζητάει partitions ακριβώς k-μεγέθους, αλλά μεγέθους μεγαλύτερου ή ίσου του k. Για την ακρίβεια οι αλγόριθμοι, οι οποίοι επιλύουν το πρόβλημα σε ικανοποιητικά γρήγορα χρόνο, δεν αναζητούν την βέλτιστη λύση, ούτε όλες οι Equivalence class στη νέα k-anonymous όψη να είναι ακριβώς μεγέθους k. Για αυτό το λόγο έχουν προταθεί δύο μετρικές, οι οποίες μετράνε την απώλεια της πληροφορίας εν συγκρίσει με το ιδανικό μοντέλο. Αυτές είναι:

- Το μέτρο διαφοροποίησης, το οποίο εισάγει σε κάθε εγγραφή t της k-anonymous όψης V μία ποινή, η οποία καθορίζεται από το μέγεθος του Equivalence class το οποίο περιέχει το t. Με βάση αυτό, η συνολική ποινή για όλον τον πίνακα είναι

$$C_{DM} = \sum_{EquivClass_E} |E|^2.$$

- το μέτρο κανονικοποιημένου μέσου μεγέθους του Equivalence class

$$G_{avg} = (totalRecords/totalEquivClasses)/(k).$$

Multidimensional Global Recoding

Σε μία σχεσιακή βάση, κάθε ιδιότητα έχει κάποιο domain από τιμές. Χρησιμοποιούμε τον συμβολισμό D_X για να συμβολίζουμε το domain της ιδιότητας X. Όπως έχει ήδη ειπωθεί η μεθοδολογία του global recoding πετυχαίνει το anonymity αντιστοιχίζοντας τα domains του quasi-identifier σε κάποιες τιμές γενίκευσης. Επιπλέον το global recoding μπορεί να διαιρεθεί σε δύο υπό-περιπτώσεις. Η πρώτη είναι το single-dimensional global recoding όπου ορίζουμε μία συνάρτηση $\phi_i : D_{X_i} \rightarrow D'$ για κάθε ιδιότητα X_i του quasi-identifier. Η ανωνυμία της όψης V επιτυγχάνεται με την αντικατάσταση κάθε τιμής του X_i με την τιμή που αντιστοιχίζει η συνάρτηση ϕ_i, για κάθε εγγραφή στο T. Η δεύτερη περίπτωση είναι το multi-dimensional global recoding όπου ορίζεται μία μοναδική συνάρτηση $\phi : D_{X_1} \times \ldots \times D_{X_n} \rightarrow D'$ η οποία χρησιμοποιείται για τη γενίκευση του διανύσματος τιμών ενός συνόλου από domains του quasi-identifier. Σύμφωνα με αυτό το μοντέλο, η ανωνυμία επιτυγχάνεται με την εφαρμογή της ϕ για κάθε διάνυσμα τιμών του quasi-identifier. Και οι δύο μεθοδολογίες μπορούνε να χρησιμοποιηθούν τόσο για κατηγορικά όσο και για αριθμητικά δεδομένα. Για τα αριθμητικά δεδομένα ή για κάποια απόλυτα ταξινομημένα domains μπορεί να χρησιμοποιηθεί κάποιο 'partitioning' μοντέλο.

Single and Multi dimensional partitioning

Σε πρώτη φάση θα ορίσουμε το μονοδιάστατο μοντέλο και εν συνεχεία το πολυδιάστατο partition.

Ορισμός. 1. *Το single-dimensional interval ορίζεται ως ένα ζευγάρι από άκρα p,u τα οποία ανήκουν στο domain D_X έτσι ώστε $p \leq u$ (τα άκρα ενός interval μπορούνε να είναι είτε ανοιχτά είτε κλειστά, για να χειριστούν συνεχή domain).*

Επί της ουσίας εάν αντιστοιχήσουμε το σύνολο τιμών ενός domain σε ένα σύνολο από intervals τότε έχουμε ορίσει ένα partition.

Patient Data			
Age	Sex	Zip code	Disease
25	Male	53771	Flu
25	Female	53772	Hepatitis
26	Male	53771	Bronchitis
27	Male	53710	Broken Arm
27	Female	53712	AIDS
28	Male	53711	Hang Nail

(α′)

SingleDimensional Anonymization			
Age	Sex	Zip code	Disease
[25 – 28]	Male	[53710 – 53771]	Flu
[25 – 28]	Female	53772	Hepatitis
[25 – 28]	Male	[53710 – 53771]	Bronchitis
[25 – 28]	Male	[53710 – 53771]	Broken Arm
[25 – 28]	Female	53712	AIDS
[25 – 28]	Male	[53710 – 53771]	Hang Nail

(β′)

MultiDimensional Anonymization			
Age	Sex	Zip code	Disease
[25 – 26]	Male	53771	Flu
[25 – 27]	Female	53772	Hepatitis
[25 – 26]	Male	53771	Bronchitis
[27 – 28]	Male	[53710 – 53771]	Broken Arm
[25 – 27]	Female	53712	AIDS
[27 – 28]	Male	[53710 – 53771]	Hang Nail

(γ′)

Σχήμα 2.8: Η anonymization του αρχικού πίνακα με τα δύο μοντέλα.

Ορισμός. 2. *Single-dimensional. Έστω ότι υπάρχει μία ταξινόμηση η οποία συσχετίζεται με κάθε domain κάθε ιδιότητας X_i του quasi-identifier. Ένα single-dimensional partitioning ορίζει για κάθε X_i ένα σύνολο από μη επικαλυπτόμενα single-dimensional intervals τα οποία καλύπτουν το D_{X_i}. Η ϕ_i αντιστοιχίζει κάθε x το οποίο ανήκει στο D_x σε κάποιο 'περιληπτικό' στατιστικό μέγεθος με βάση το interval το οποίο περιέχεται.*

Τα δεδομένα, λοιπόν τα οποία θα εκδοθούν θα είναι απλά στατιστικά τα οποία παρουσιάζουν περιληπτικά τα intervals τα οποία περιέχουν. Για την υπόλοιπη παρουσίαση του Mondrian θα θεωρήσουμε ότι αυτά είναι min-max ranges. Το μονοδιάστατο μοντέλο εύκολα επεκτείνεται στο πολυδιάστατο.

Ορισμός. 3. *Θεωρούμε πάλι μία ταξινόμηση για κάθε D_{X_i}. Μία πολυδιάστατη περιοχή ορίζεται ως ένα ζευγάρι από d-tuples $(p_1, ..., p_d), (u_1, ..., u_d) \in D_{X_1} \times \ldots \times D_{X_d}$ έτσι ώστε για κάθε $i, p_i \leq u_i$.*

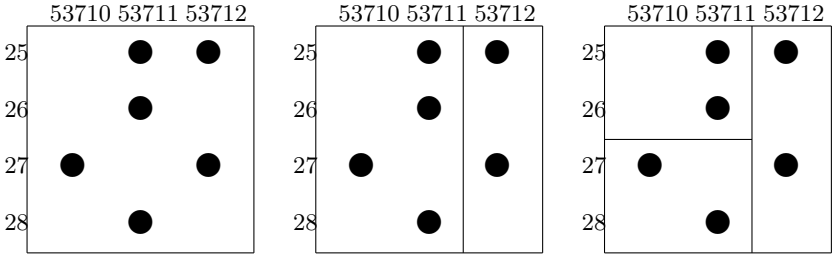
Τελικά κάθε περιοχή καλύπτεται από ένα πολυγωνικό κουτί, έτσι σε κάθε ακμή (ή κορυφή) του κουτιού να είναι είτε ανοιχτή είτε κλειστή.

Ορισμός. 4. *Strict Multidimensional Partitioning. Ένα strict multidimensional partitioning ορίζει ένα σύνολο από μη επικαλυπτόμενες πολυδιάστατες περιοχές οι οποίες καλύπτουν το $D_{X_1} \times \ldots \times D_{X_d}$. Η συνάρτηση ϕ αντιστοιχεί κάθε εγγραφή $(x_1, ..., x_d) \in D_{X_1} \times \ldots \times D_{X_d}$ σε κάποιο 'περιληπτικό' στατιστικό μέγεθος με βάση το region στο οποίο ανήκει.*

Όταν η συνάρτηση ϕ εφαρμόζεται στον πίνακα T, το σύνολο των εγγραφών οι οποίες ανήκουν σε κάθε μία κενή περιοχή ορίζουν το equivalence class της νέας όψης V, έχοντας έτσι την δυνατότητα να εξετάσουμε το k-anonymity. Πάλι για απλότητα, θεωρούμε ότι χρησιμοποιούμε min-max ranges. Στο Σχήμα 2.8 δίνεται ένα παράδειγμα και των δύο μοντέλων. Μπορεί κανείς μάλιστα να παρατηρήσει ότι το single-dimensional είναι πάντα μία υποπερίπτωση του multi-dimensional.

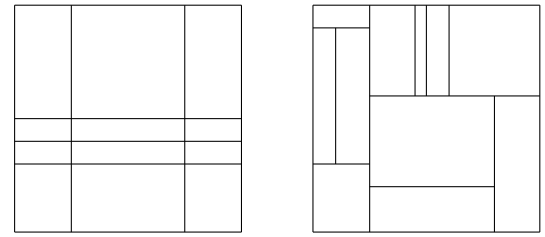
Πρόταση. 1. *Κάθε single-dimensional partitioning για ένα quasi-identifier με ιδιότητες $X_1, ..., X_d$ μπορεί να εκφραστεί ως ένα strict multi-dimensional partitioning. Αν όμως το μέ-*

γενος των διαστάσεων $d \geq 2$ τότε για κάθε $i, |D_{X_i}| \geq 2$ υπάρχει ένα strict multi-dimensional partitioning το οποίο δεν μπορεί να προκύψει από single-dimensional partitioning.



(α′) Patients (β′) Single-dimensional (γ′) Multidimensional

Σχήμα 2.9: Χωρική αναπαράσταση των Ασθενών και των partitions για Zip code-Age



(α′) Single-Dimensional (β′) Multidimensional

Σχήμα 2.10: Ένα παράδειγμα διαχωρισμού του χώρου από τα δύο μοντέλα.

Θα μπορούσε κανείς να φανταστεί κανείς και τα δύο μοντέλα στον χώρο. Κάθε ιδιότητα παίρνει τιμές από σύνολο το οποίο περιέχει ταξινομημένα σημεία. Έτσι κάθε ένα από αυτά τα σύνολο μπορεί να θεωρηθεί ως μία διάσταση-άξονα στο χώρο. Επί της ουσίας στο μονοδιάστατο μοντέλο για να ορίσουμε partitions χωρίζουμε τον χώρο τραβώντας παράλληλες ευθείες ως προς τους άξονες και αυτές οι ευθείες διασχίζουν όλο τον χώρο. Αντίθετα στον πολυδιάστατο μοντέλο, τραβάμε μία ευθεία παράλληλη ως προς τον ένα άξονα και έτσι ορίζουμε δύο νέους υποχώρους. Εν συνεχεία σε αυτούς τους υποχώρους μπορούμε αναδρομικά να τραβήξουμε μία ευθεία ως προς όποιον άξονα θέλουμε, μόνο που αυτή η ευθεία δεν θα τέμνει άλλους υποχώρους. Στα partitions κάθε εγγραφή μπορεί να παρασταθεί ως ένα σημείο στο χώρο. Τα σχήματα 2.9 και 2.10 αποτελούν δύο τέτοια παραδείγματα.

Για να επιλύσουμε το πρόβλημα του k-anonymity βέλτιστα λοιπόν αρκεί στο πολυδιάστατο μοντέλο, να χωρίσουμε το χώρο σε partitions έτσι ώστε να έχουμε το ελάχιστο C_{DM} ή C_{AVG} και τουλάχιστον k σημεία-εγγραφές. Προσοχή, τα partitions αυτά δεν είναι απαραίτητο να έχουν ακριβώς k στοιχεία, μπορούνε να έχουν και περισσότερα στην βέλτιστη λύση. Αυτό μπορεί να συμβεί στις περιπτώσεις όπου μία πανομοιότυπη εγγραφή (ως προς το quasi-identifier) εμφανίζεται αρκετές φορές στην βάση. Πιο απλά, μία εγγραφή η οποία επαναλαμβάνεται m φορές στην βάση, τότε ορίζει m φορές το ίδιο σημείο στον χώρο. Άρα το partition που περιέχει αυτή την εγγραφή θα έχει τουλάχιστον μέγεθος ίσο m. Αυτό δεν ισχύει μόνο για αυτό το μοντέλο αλλά για οποιοδήποτε άλλο global recoding μοντέλο.

Δυστυχώς όμως αποδεικνύεται ότι η εύρεση της βέλτιστης λύσης με αυτό το μοντέλο είναι NP-hard. Η απόδειξη παραλείπετε για οικονομία χώρου, απλά αναφέρουμε ότι γίνεται

αναγωγή στο πρόβλημα partition (έστω ένα σύνολο A, αναζητάμε δύο υποσύνολα του A που το άθροισμα των στοιχειών τους να είναι ίσο).

2.3.2 Μέγεθος του partition

Όπως είδαμε στο προηγούμενο section, το μέγεθος ενός partition δεν μπορεί να είναι πάντα ίσο με k, λόγω της ύπαρξης ενός σημείο πολλές φορές. Ακόμα παρατηρήσαμε ότι η αναζήτηση για partitions μεγέθους k είναι NP-Hard. Σε αυτή την ενότητα θα ορίσουμε πότε ένα partition ικανοποιεί το k-anonymity και επιπλέον θα δείξουμε ότι το μέγιστο δυνατό μέγεθος ενός partition εξαρτάται μόνο από το k και από τα 'αντίγραφα' ενός σημείου.

Αρχικά θα ορίσουμε πότε ένα partition μπορεί να χωριστεί σε μικρότερα, χωρίς να παραβιάζει το k-anonymity. Αρχικά θα ορίσουμε την πολυδιάστατη τομή. Επί της ουσίας, στην πολυδιάστατη τομή απλά παίρνουμε ένα υπάρχον σύνολο (το οποίο δεν έχει ήδη τμηθεί) και με βάση μίας ευθείας παράλληλης σε κάποια άξονα του χώρου το χωρίζουμε σε δύο άλλα σύνολο, τα οποία ικανοποιούν το k-anonymity.

Ορισμός. 5. *Allowable Multidimensional Cut.Έστω ένα πολυσύνολο P από σημεία (τα οποία επιτρέπεται να υπάρχουν περισσότερο από μία φορά) στον d-διάστατο χώρο. Μία τομή παράλληλη ως προς τον άξονα X_i με βάση την τιμή x_i είναι επιτρεπτή όταν και μόνο όταν $Count(P.X_i > x_i) \geq k$ και $Count(P.X_i \leq x_i) \geq k$.*

Όμοια μπορούμε να ορίσουμε και την επιτρεπτή μονοδιάστατη τομή. Η διαφορά τώρα είναι ότι σε κάθε παράλληλη γραμμή την οποία τραβάμε ως προς έναν άξονα χωρίζουμε όλα τα υπάρχοντα partitions τα οποία μπορεί να τμήσει αυτή η ευθεία.

Ορισμός. 6. *Allowable Single-Dimensional Cut. Έστω ένα πολυσύνολο P από σημεία (τα οποία επιτρέπεται να υπάρχουν περισσότερο από μία φορά) στον d-διάστατο χώρο. Θεωρούμε χωρίς βλάβη της γενικότητας ότι ήδη έχουμε τμήσει τον χώρο σε S single dimensional cuts με βάση κάποια επιτρεπτή μονοδιάστατη τομή και άρα έχουμε χωρίσει τον χώρο στις ξένες μεταξύ τους περιοχές $R_1, \ldots, R_n$. Ένα single-dimensional cut είναι επιτρεπτό με βάση κάποιον άξονα X_i σύμφωνα με την τιμή x_i αν $Count(R_j.X_i > x_i) \geq k$ και $Count(R_j.X_i \leq x_i) \geq k$ για κάθε R_j.*

Συνολικά, αν εφαρμόσουμε αναδρομικά το πρώτο ορισμό για όλα τα partitions θα καταλήξουμε σε ένα σύνολο από partitions τα οποία ικανοποιούν το k-anonymity και παράλληλα δεν μπορούνε να τμηθούν άλλο. Το ίδιο ισχύει και για το δεύτερο ορισμό. Η διαφορά των δύο είναι ότι θα τμήσουν το χώρο με διαφορετική προσέγγιση. Ένα παράδειγμα δίνεται στα σχήματα 2.9 και 2.10. Μπορεί κανείς να παρατηρήσει ότι στην δεύτερη περίπτωση (μονοδιάστατη τομή), όλες οι ευθείες τέμνουν όλο τις περιοχές. Μάλιστα ότι προκύπτει από κάποια μονοδιάστατη τομή, μπορεί να προκύψει από κάποια πολυδιάστατη.

Γενικότερα στο k-anonymity προσπαθούμε να ομαδοποιήσουμε-γενικεύσουμε όσο μικρότερα groups από εγγραφές μπορούμε. Για αυτό το λόγο αναζητείται πάντα το μικρότερο δυνατόν partition (ή έστω με κάποιο greedy αλγόριθμο μία ικανοποιητική λύση). Δίνονται οι παρακάτω δύο ορισμοί:

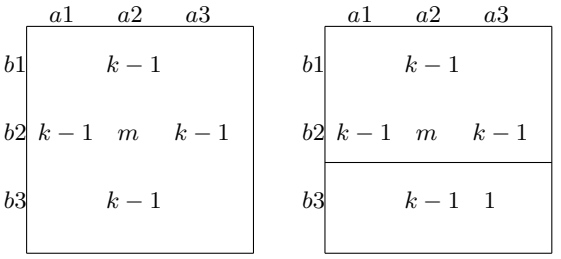
Ορισμός. 7. *Minimal Strict Multi-Dimensional Partitioning. Έστω $R_1, ..., R_n$ ένα σύνολο από περιοχές που έχουνε προκύψει από strict multi-dimensional partitioning και έστω ότι η περιοχή R_i περιέχει ένα πολυσύνολο P_i από σημεία. Αυτό το multi-dimensional partitioning είναι ελάχιστο (minimal) αν για κάθε i, $|P_i| \geq k$ και δεν υπάρχει κάποιο allowable multi-dimensional cut για το P_i.*

Ορισμός. 8. *Minimal Single-Dimensional Partitioning. Το σύνολο S όλων των allowable single-dimensional cuts είναι ένα minimal single-dimensional partitioning για το πολυσύνολο P από σημεία εάν δεν υπάρχει κάποιο allowable single-dimensional cut για το P δοθέντος του S.*

Τα δύο πάνω θεωρήματα μας επιτρέπουν να ορίσουμε το άνω μέγιστο μέγεθος του partition. Αποδεικνύονται οι παρακάτω δύο προτάσεις:

Πρόταση. 2. *Αν $R_1, ..., R_n$ συμβολίζουν το σύνολο των περιοχών τα οποία έχουν προκύψει από ένα minimal strict multidimensional partitioning για το πολυσύνολο σημείων P, τότε ο μέγιστος αριθμός σημείων ο οποίος περιλαμβάνεται σε κάθε R_i είναι $2d(k-1)+m$, όπου d το πλήθος των διαστάσεων, ο αριθμός k εκφράζει πόσες εγγραφές πρέπει να είναι ίδιες ως προς το quasi-identifier (να ικανοποιείται δηλαδή το k-anonymity και m ο μέγιστος αριθμός αντιγράφων ενός σημείο στο P.*

Πρόταση. 3. *Ο μέγιστος αριθμός σημείων ο οποίος περιλαμβάνεται σε μία περιοχή R η οποία έχει προκύψει από ένα minimal single-dimensional partitioning από ένα πολυσύνολο σημείων P σε ένα d-διάστατο χώρο ($d \geq 2$) είναι $O(P)$.*



(α′) No allowable cut (β′) allowable cut με πρόσθεση ενός σημείου

Σχήμα 2.11: Παράδειγμα για το ελάχιστο όριο του partition.

Είναι προφανές από τις δύο παραπάνω προτάσεις, ότι δεν είναι εφικτό πάντα να πετύχουμε partitions μεγέθους k. Το Σχήμα 2.11 δείχνει μία τέτοια περίπτωση. Ενώ στο πρώτο δεν υπάρχει η δυνατότητα να χωρίσουμε το χώρο, στην δεύτερη απλά με την πρόσθεση ενός σημείου αυτό είναι εφικτό.

Multidimensional Local Recoding

Σε αντίθεση με το global recoding, το local recoding αντιστοιχεί κάθε ατομική εγγραφή (ή δεδομένα) σε κάποια γενικευμένη τιμή τοπικά. Πιο συγκεκριμένα, μία local recoding function αντιστοιχεί κάθε μεμονωμένη εγγραφή σε μία νέα εγγραφή και έτσι προκύπτει η νέα k-anonymous όψη.

Όμοια με πριν μπορούμε να ορίσουμε και το Multidimensional local recoding. Δίνεται ο παρακάτω ορισμός:

Ορισμός. 9. *Relaxed Multidimensional Partitioning. Ένα relaxed multidimensional partitioning για την σχέση T ορίζει ένα σύνολο (από ίσως και επικαλυπτόμενες) μοναδικές πολυδιάστατες περιοχές οι οποίες καλύπτουν το $D_{X_1} \times \ldots \times D_{X_n}$. Μία local recoding συνάρτηση ϕ^* αντιστοιχεί κάθε εγγραφή σε κάποιο 'περιληπτικό' στατιστικό μέγεθος για κάθε περιοχή στην οποία περιλαμβάνεται.*

Μπορεί κανείς να παρατηρήσει κανείς τις διαφορές με το global recoding. Στο τελευταίο δεν είχαμε επικαλυπτόμενες περιοχές, ο χώρος χωρίζεται σε ξένες μεταξύ τους περιοχές. Μάλιστα κάθε σημείο του χώρου αντιστοιχίζεται πάντα στο ίδιο περιληπτικό μέγεθος, ανεξάρτητα το πόσες φορές υπάρχει. Αντίθετα στο local recoding τα διάφορα αντίγραφα ενός σημείου αντιστοιχίζονται σε μία νέα τιμή, με βάση σε ποια περιοχή ανήκουν, δηλαδή από την τοπική τους 'ιδιότητα'. Αυτός μας προσφέρει μία παραπάνω ευελιξία. Στον πίνακα ήμασταν αναγκασμένοι να αντιστοιχούμε το Zip code με την τιμή 53711 στην ίδια τιμή πάντα. Στο local recoding θα μπορούσαμε να αντιστοιχήσουμε το ένα στην τιμή [53710-53711] και το άλλο στο [53711-53712]. Επίσης παρατηρήστε το Σχήμα 2.11, ενώ για το global recoding στην πρώτη περίπτωση πάντα δεν υπάρχει δυνατόν partition για το local recoding υπάρχει.

Πρόταση. 4. *Κάθε strict multidimensional partitioning μπορεί να εκφραστεί ως ένα relaxed multidimensional partitioning. Όμως αν υπάρχουν τουλάχιστον δύο εγγραφές στον πίνακα οι οποίες να έχουν το ίδιο διάνυσμα για το quasi-identifier τότε υπάρχει ένα relaxed multidimensional partitioning το οποίο δεν μπορεί να εκφραστεί με κάποιο strict multidimensional partitioning.*

Θα ορίσουμε λοιπόν τώρα πότε μία τομή είναι επιτρεπτή:

Ορισμός. 10. *Allowable Relaxed Multidimensional Cut.Έστω ένα πολυσύνολο P από σημεία (τα οποία επιτρέπεται να υπάρχουν περισσότερο από μία φορά) στον d-διάστατο χώρο. Μία τομή παράλληλη ως προς τον άξονα X_i με βάση την τιμή x_i είναι επιτρεπτή όταν και μόνο όταν το P περιέχει τουλάχιστον $2k$ στοιχεία. Ισοδύναμα θα μπορούσαμε να πούμε ότι είναι επιτρεπτή όταν και μόνο όταν:*

- *για $Count(P.X_i > x_i) = b_1$, $Count(P.X_i < x_i) = b_2$ και $Count(P.X_i = x_i) = a$ υπάρχουν $a1, a2$ τέτοια ώστε $a1 + a2 = a$ και $b_1 + a_1 \geq k$ και $b_2 + a_2 \geq k$.*

Σε αντίθεση λοιπόν με πριν τα σημεία τα οποία αντιστοιχούν πάνω στην ευθεία που τέμνουν τον χώρο μπορούνε να πάνε σε οποιοδήποτε από τα δύο partitions αρκεί να ικανοποιείται το k-anonymity. Ξαναδείτε το Σχήμα 2.11, μπορεί κανείς να παρατηρήσει ότι για το local recoding για την πρώτη περίπτωση ο χώρος μπορεί πάλι να χωριστεί. Τέλος ορίζουμε όμοια με πριν το minimal relaxed multidimensional partitioning.

Ορισμός. 11. *Minimal Relaxed Multi-Dimensional Partitioning. Έστω $R_1, ..., R_n$ ένα σύνολο από περιοχές που έχουν προκύψει από relaxed multi-dimensional partitioning και έστω ότι η περιοχή R_i περιέχει ένα πολυσύνολο P_i από σημεία. Αυτό το multi-dimensional partitioning είναι ελάχιστο (minimal) αν για κάθε i, $|P_i| \geq k$ και δεν υπάρχει κάποιο allowable relaxed multi-dimensional cut για το P_i.*

Είναι πολύ εύκολο να παρατηρήσει κανείς ότι ισχύει:

Πρόταση. 5. *Ο μέγιστος αριθμός σημείων ο οποίος περιλαμβάνεται σε μία περιοχή R η οποία έχει προκύψει από ένα minimal relaxed multidimensional partitioning από ένα πολυσύνολο σημείων P σε ένα d-διάστατο χώρο ($d \geq 2$) είναι το πολύ $2k - 1$.*

2.3.3 Ο αλγόριθμος Mondrian

Με βάση τα παραπάνω, προτάθηκε ο αλγόριθμος mondrian ο οποίος δουλεύει τόσο για global recoding όσο και για local recoding. Ο αλγόριθμος αυτός είναι ένας top-down ο οποίος χωρίζει τον χώρο αναδρομικά σε partitions με την προϋπόθεση να είναι αυτό δυνατόν. Δίνεται ο αλγόριθμος :

Algorithm 4 Multi-dim

Anonymize(partition)
if no allowable multidimensional cut for partition **then**
  return $\phi : partition \rightarrow summary$
else
  $dim \leftarrow choose_dimension()$
  $fs \leftarrow frequency_set(partition, dim)$
  $splitVal \leftarrow find_median(fs)$
  $lhs \leftarrow tEpartition : t.dim \leq splitVal$
  $rhs \rightarrow tEpartition : t.dim > splitVal$
  return $Anonymize(rhs)UAnonymize(lhs)$
end if

Σε πρώτο στάδιο ελέγχουμε αν υπάρχει κάποια επιτρεπτή τομή, τότε απλά επιστρέφουμε το partition αφού πρώτα το έχουμε γενικεύσει. Αν έχουμε την δυνατότητα να χωρίσουμε το partition τότε αρχικά επιλέγουμε την διάσταση ως προς το οποίο θα το χωρίσουμε. Η επιλογή μπορεί να γίνει με κάποιο ευριστικό τρόπο (αφού δεν επηρεάζεται το μέγιστο μέγεθος του partition. Μία ευριστική μέθοδος είναι να επιλεγεί εκείνη με το μέγιστο εύρος κανονικοποιημένων τιμών (αρκεί βέβαια ως προς αυτή την διάσταση να έχουμε allowable cut).

Εν συνεχεία πρέπει να επιλέξουμε και την τιμή ως προς την οποία θα χωρίσουμε τον χώρο. Μία μέθοδος είναι να επιλέξουμε την μέθοδο του median-partitioning η οποία έχει χρησιμοποιηθεί για την κατασκευή KD-trees. Επί της ουσίας επιλέγουμε το μέσο του partition αν αυτό προβληθεί ως προς την συγκεκριμένη διάσταση που έχει ήδη επιλεχθεί. Το frequency set χρησιμοποιείται για λόγους απόδοσης και θα εξηγηθεί παρακάτω. Προσέξτε ότι αν η τομή ως προς ένα άξονα είναι επιτρεπτή, τότε σίγουρα είναι και ως προς το μέσο αυτού, αφού αυτό βρίσκεται στην μέση. (Όταν λέμε μέσο, έχουμε λάβει υπόψη μας και τα αντίγραφα, μέσο δηλαδή για ένα ταξινομημένο χώρο, είναι το σημείο με αθροιστική συχνότητα μεγαλύτερη του 50 τοις εκατό και έτσι ώστε να μην υπάρχει άλλο σημείο με αθροιστική συχνότητα μικρότερη αυτού του σημείου και ταυτόχρονα μεγαλύτερη του 50 τοις εκατό).

Έχοντας ποια επιλέξει το μέσο, μετά απλά χωρίζουμε τον χώρο σε δύο partitions και εφαρμόζουμε αναδρομικά τον αλγόριθμο.

Ο αλγόριθμος επεκτείνεται και για το local recoding. Αρχικά κάνουμε έλεγχο για allowable relaxed multidimensional cut. Προσέξτε ότι ο έλεγχος τώρα είναι πιο απλός αφού, πολύ απλά ελέγχουμε να έχουμε $2k$ σημεία έστω και ίδια. Εν συνεχεία, επιλέγουμε με ίδιο τρόπο το σημείο χωρισμού και χωρίζουμε τον χώρο έτσι ώστε όλα τα σημεία για τα οποία ισχύει $t.dim < splitVal$ να ανήκουν στο lhs και όλα τα σημεία για το οποία $t.dim > splitVal$ να ανήκουν στο rhs. Τα σημεία για τα οποία ισχύει $t.dim = splitVal$ θα διανεμηθούν στο lhs και στο rhs έτσι ώστε να ισχύει πάντα $|rhs| \leq |lhs| \leq |rhs| + 1$.

Η ποιότητα των partitions

Με βάση τα μέτρα τα οποία είχανε οριστεί στην αρχή, έχουμε την δυνατότητα να εξετάσουμε την τιμή τους για τον άνω αλγόριθμο. Εξ' ορισμού για να ικανοποιείται το k-anonymity το ελάχιστο C_{DM} το οποίο από εδώ και πέρα θα το συμβολίζουμε C_{DM^*} είναι

$$C_{DM^*} = k \times total - records.$$

. Όμοια

$$C_{AVG^*} = 1.$$

Ο global recoding αλγόριθμος με βάση τα άνω όρια που ορίσαμε έχει

$$G_{DM} \leq 2d(k-1) + m$$

και

$$C_{AVG} \leq (2d(k-1) + m)/k.$$

Αν πάρουμε τον λόγο C_{DM}/C_{DM^*} και C_{AVG}/C_{AVG^*} και θεωρήσουμε τον παράγοντα m/k σταθερό, τότε το παραπάνω σφάλμα που εισάγει ο αλγόριθμος επιτυγχάνει προσέγγιση $O(d)$. Όμοια για το local recoding επιτυγχάνουμε 2-προσέγγιση.

Απόδοση

Το βασικό πρόβλημα του αλγορίθμου είναι η επιλογή του μέσου. Η βασική ιδέα είναι η υλοποίηση ενός frequency set για κάθε ιδιότητα. Αν για κάθε τιμή μίας ιδιότητας έχουμε τον αριθμό των εμφανίσεων της τότε μπορούμε με κάποιο γνωστό median-finding αλγόριθμο να υπολογίσουμε το μέσο. Το πλεονέκτημα του frequency set είναι ότι είναι αρκετό μικρότερο από όλο τον πίνακα και είναι πολύ πιθανότερο να χωράει στην μνήμη. Έτσι ο αλγόριθμος αρχικά κατασκευάζει το frequency set και εν συνεχεία σε κάθε αναδρομική επανάληψη σκανάρουμε μία φορά τα δεδομένα για να βρούμε τον μέσο και εν συνεχεία άλλη μία φορά για να εγγράψουμε τα νέα partitions. Μπορεί εδώ κανείς να παρατηρήσει το πρόβλημα του local recoding. Στο global recoding αρκεί μία φορά να υπολογίσουμε το frequency set και αυτό έπειτα θα είναι κοινό για όλα τα partitions (μία τιμή μπορεί να ανήκει μόνο σε ένα από όλα). Αντίθετα για το local recoding θα πρέπει να ληφθεί ειδική μέριμνα για σημεία τα οποία πιθανόν να ανήκουν σε περισσότερα από ένα partitions. Τέλος όπως και στην κατασκευή των KD-trees με βάση το median-partitioning η χρονική πολυπλοκότητα του αλγορίθμου είναι $O(nlogn)$, όπου $n = |T|$.

Summary Statistics

Σε όλο το προηγούμενο κομμάτι θεωρήσαμε ότι γενικεύουμε τις τιμές στο min-max range. Αυτό όμως δεν είναι πάντα απαραίτητο. Θα μπορούσαμε για παράδειγμα να γενικεύσουμε τις τιμές, στην μέση τιμή του partition. Γενικότερα σε ποια τιμή θα γενικεύσουμε μία εγγραφή εξαρτάται από διάφορους παράγοντες και κυρίως από τι ανάλυση δεδομένων επιθυμεί κάποιος να κάνει από την νέα k-anonymous όψη. Για παράδειγμα, αν κάποιος επιθυμεί να υπολογίσει το άθροισμα ή μέσους όρους η μέση τιμή θα ήταν ένα προτιμότερη έναντι του εύρους τιμών. Αντίθετα αν κάποιος ήθελε να υπολογίσει ερωτήματα όπου υπάρχουν ανισότητες τότε το εύρος τιμών είναι προτιμότερο.

Βέβαια είναι δυνατόν να εκδώσουμε και τις δύο γενικευμένες τιμές (τόσο δηλαδή την μέση τιμή όσο και το εύρος τιμών). Παράλληλα θα μπορούσαμε να εκδώσουμε και κάποια γενικότερα στατιστικά μεγέθη για τον πίνακα, όπως τύπο κατανομής, τυπική απόκλιση και άλλα. Γενικότερα, δεδομένου ότι αναφερόμαστε σε αριθμητικά δεδομένα θα μπορούσαμε να δημοσιεύσουμε διάφορα στατιστικά μεγέθη τα οποία να βοηθήσουν για περαιτέρω ανάλυση τον χρήστη.

Σχόλια

Ο αλγόριθμος Mondrian είναι ένας ικανοποιητικά γρήγορος αλγόριθμος ο οποίος δουλεύει 'καλά' για αριθμητικά δεδομένα. Το βασικό μειονέκτημα του είναι ότι δεν μπορεί να επεξεργαστεί κατηγορικά δεδομένα, εκτός αν υπάρχει κάποια διάταξη για αυτά. Ακόμα όμως και να υπάρχει διάταξη δεν μπορεί να ικανοποιήσει το μέγιστο όριο των partitions.

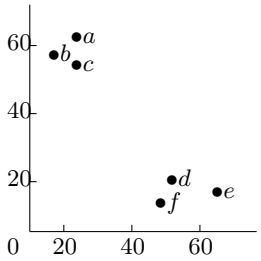
Παράλληλα ο αλγόριθμος δεν μας παρέχει κανένα μέτρο για την απώλεια της πληροφορίας. Το μέγεθος του partition ναι μεν αποτελεί κάποια μορφή ελέγχου αλλά δεν είναι αρκετό. Είναι δυνατόν ο χώρος να χωριστεί σε διάφορα partitions, το θέμα είναι ποιος από όλους αυτούς τους χωρισμούς προσφέρει την καλύτερη πληροφορία.

Τέλος θα μπορούσε κανείς να αναρωτηθεί γιατί δεν εφαρμόζουμε πάντα local recoding έναντι του global recoding αφού το πρώτο χωρίζει σίγουρα το χώρο σε μικρότερα partitions. Ο βασικός λόγος αναφέρθηκε παραπάνω και είναι η απώλεια πληροφορίας. Δεν μας εγγυάται κανείς ότι το local recoding είναι βέλτιστο ως προς αυτό. Μην ξεχνάμε ότι το k-anonymity απαιτεί απλά να υπάρχουν τουλάχιστον k ίδιες εγγραφές, εν συνεχεία αυτό που μας νοιάζει δεν είναι κατά πόσο ξεπερνάμε το k αλλά κατά πόσο διατηρούμε την πληροφορία. Θα μπορούσε εύκολα κανείς να φανταστεί περιπτώσεις όπου αν επιλέξουμε επικαλυπτόμενες περιοχές και τις γενικεύσουμε τότε είναι πιθανόν να έχουμε μεγαλύτερη απώλεια πληροφορίας. (Ισχύει βέβαια και το αντίστροφο).

2.4 Utility-Based Anonymization

Όλοι οι προηγούμενοι αλγόριθμοι που είδαμε είχανε το μειονέκτημα ότι χώριζαν το χώρο προσπαθώντας να ομαδοποιήσουν όσο το δυνατόν λιγότερες εγγραφές. Το βασικό πλεονέκτημα αυτού του αλγορίθμου είναι ότι αναζητά τρόπους ώστε

- να έχουμε όσον το δυνατόν μικρότερη απώλεια πληροφορίας και
- να μπορούμε αυτή την απώλεια πληροφορίας να την μετράμε δίνοντας διαφορετική βαρύτητα σε κάθε εγγραφή.



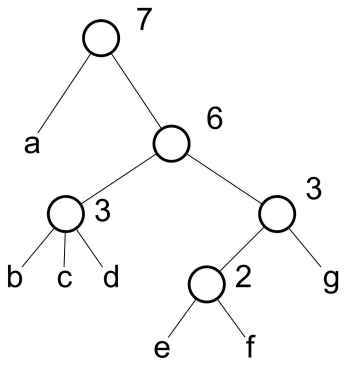
(α′) No allowable cut

Σχήμα 2.12: Παράδειγμα για το ελάχιστο όριο του partition.

Για παράδειγμα ας δούμε το παραπάνω Σχήμα 2.12. Εύκολα παρατηρεί κανείς ότι οι εγγραφές μπορούνε να χωριστούν στα groups $(a, b), (c, d), (e, f)$ ικανοποιώντας το 2-anonymity και με τα C_{DM} και C_{AVG} να είναι ελάχιστα. Εν συνεχεία επιλέγουμε να τις γενικεύσουμε με βάση το min-max range. Δηλαδή τα a, b στο ([10,20],[60,70]), το c, d στο ([20,50],[20,50]) και το e, f στο ([50,60],[10,15]).

Μπορούμε να υπολογίσουμε με βάση τα άνω το σφάλμα αβεβαιότητας κοιτάζοντας πόσο απέχει κάθε σημείο από την γενικευμένη τιμή. $U(a) = U(b) = 10 + 10 = 20$, $U(c) = U(d) = 60$ και $U(e) = U(f) = 15$. Συνολικά λοιπόν το σφάλμα είναι $U(T) = 190$.

Αντίθετα αν επιλέξουμε να φτιάξουμε τα groups a,b,c και e,d,f μπορεί εύκολα να παρατηρήσει κανείς ότι το σφάλμα είναι μικρότερο, ίσο με 150.



Σχήμα 2.13: Μία ιεραρχία κατηγορικών δεδομένων

2.4.1 Βασικοί ορισμοί

Στον πίνακα έχουμε δύο είδη δεδομένων, τα αριθμητικά και τα κατηγορικά. Θα δούμε και για τις δύο περιπτώσεις ένα τρόπο να μετράμε την απώλεια πληροφορίας ή καλύτερα πόσο κοντά ή μακριά είναι δύο εγγραφές.

Αριθμητικά δεδομένα

Θα θεωρήσουμε ότι τα αριθμητικά δεδομένα γενικεύονται με βάση το min-max range. Δίνεται αρχικά η ποινή για την απώλεια πληροφορίας σε μία μόνο μία ιδιότητα.

Ορισμός 1. *Έστω ένας πίνακας T με quasi-identifier $(A_1, \ldots, A_n)$, όπου κάποιες από τις ιδιότητες πιθανόν να είναι αριθμητικές. Έστω ότι η ιδιότητα A_i είναι αριθμητική, θεωρούμε μία εγγραφή $t = (x_1, \ldots, x_i, \ldots, x_n)$ η οποία γενικεύεται στην εγγραφή $t' = (x'_1, \ldots, [z_i, y_i], \ldots, x'_n)$ έτσι ώστε $y_i \leq x_i \leq z_i$. Για την ιδιότητα A_i η ποινή normalized certainty penalty ορίζεται ως*

$$NCP_{A_i}(t) = (z_i - y_i)/|A_i|$$

όπου $|A_i|$ είναι η διαφορά της μέγιστης και της ελάχιστης τιμής της ιδιότητας στον πίνακα T.

Είναι προφανές ότι σε όσο μικρότερο range γενικεύουμε μία ιδιότητα μίας εγγραφής τόσο μικρότερη είναι η ποινή, αφού ο παρανομαστής είναι πάντα σταθερός. Η ποινή αυτή αφορά την εγγραφή και όχι την ιδιότητα.

Κατηγορικά δεδομένα

Εκτός από αριθμητικά δεδομένα είναι πιθανόν να έχουμε και κατηγορικά δεδομένα. Συνήθως για να γενικεύσουμε τα κατηγορικά δεδομένα έχουμε κάποια ιεραρχία, όπως αυτή περιγράφηκε στον αλγόριθμο incognito. Το πρόβλημα είναι πως ακριβώς μπορούμε να χρησιμοποιήσουμε αυτή την ιεραρχία για να μετρήσουμε την απώλεια πληροφορίας σε κατηγορικά δεδομένα.

Η πρώτη σκέψη είναι σε κάποια ιεραρχία να μετρήσουμε το ελάχιστο μονοπάτι. Δηλαδή έστω ότι έχω ένα Domain $D = d_1, ..., d_n$ και θέλω να γενικεύσω τις τιμές d_i και d_j και αποφασίζω να τις αντιστοιχίσω στον ίδιο πατέρα d_k. Τότε θα μπορούσα να θεωρήσω ως μέτρος της ποινής το μήκος του ελάχιστου μονοπατιού το οποίο περνάει από αυτούς τους τρεις κόμβους.

Αυτό το μέτρο όμως δεν είναι ικανοποιητικό. Δείτε το Σχήμα 2.13, με βάση αυτό το μέτρο είναι καλύτερο να ομαδοποιήσουμε το b με το a παρά με το e αφού το ελάχιστο μονοπάτι που ενώνει τα δύο είναι μικρότερο. Στην πραγματικότητα όμως είναι καλύτερο το αντίθετο αφού το b και το e έχουν πιο κοντινό πρόγονο. Ένα καλύτερο μέτρο λοιπόν είναι για να μετρήσουμε την ποινή, είναι να επιλέξουμε το κοντινότερο πρόγονο. Ένας τρόπος να το κάνουμε αυτό είναι να μετρήσουμε ποιος πρόγονος έχει τα λιγότερα παιδιά, όσο τα λιγότερα τόσο μικρότερη ποινή. Δίνεται ο ακριβής ορισμός:

Ορισμός 2. *Έστω ένας πίνακας T με quasi-identifier $(A_1, \ldots, A_n)$, όπου κάποιες από τις ιδιότητες πιθανόν να είναι κατηγορικές. Έστω ότι η ιδιότητα A_i είναι κατηγορική, θεωρούμε μία εγγραφή $t = (x_1, \ldots, v, \ldots, x_n)$ η οποία γενικεύεται στην εγγραφή $t' = (x'_1, \ldots, u, \ldots, x'_n)$ όπου u είναι ένας πρόγονος του v με κάποιο domain hierarchy. Για την ιδιότητα A_i η ποινή normalized certainty penalty ορίζεται ως*

$$NCP_{A_i}(t) = size(u)/|A_i|$$

όπου $|A_i|$ είναι όλες οι τιμές τις οποίες μπορεί να πάρει η ιδιότητα T στον πίνακα A και $size(u)$ είναι ίσο με το πλήθος των φύλλων που έχουν πρόγονο το u.

Παρατηρούμε ότι πάλι ο παρανομαστής παραμένει πάντα σταθερός. Άρα όταν επιθυμούμε να γενικεύσουμε μία ιδιότητα μίας εγγραφής επιλέγουμε εκείνο τον πρόγονο-κόμβο στο δέντρο με τα λιγότερα παιδιά-φύλλα. Παρατηρήστε ότι με βάση με αυτό τον ορισμό είναι όντως προτιμότερο αυτή την φορά να ομαδοποιήσουμε το b με το e. Προσέξτε ότι σε κάθε τιμή μπορούμε να δώσουμε και διαφορετική αξία αν βάζαμε βάρη στις ακμές του δέντρου και έτσι για κάθε φύλλο μετά είχαμε 'άλλη αξία'.

Συνολική ποινή

Μέχρι τώρα έχουμε υπολογίσει την συνολική ποινή για την ιδιότητα μίας εγγραφής, είτε αυτή είναι κατηγορική είτε αριθμητική. Πως όμως υπολογίζουμε την συνολική ποινή για μία εγγραφή; Η πιο απλή λύση είναι προσθέσουμε την ποινή που προκύπτει από την γενίκευση για κάθε ιδιότητα ξεχωριστά στην συγκεκριμένη εγγραφή. Μπορούμε όμως να θεωρήσουμε ότι κάθε ιδιότητα σε μία εγγραφή έχει διαφορετικό βάρος. Έτσι μπορούμε να αντιστοιχήσουμε σε κάθε ιδιότητα ένα βάρος. Δίνεται ο ακριβής ορισμός.

Ορισμός 3. *Έστω ένας πίνακας T με quasi-identifier $(A_1, ..., A_n)$, και μία εγγραφή $t = (x_1, ..., x_n)$ η οποία γενικεύεται στην εγγραφή $t' = (x'_1, ..., x'_n)$. Για την εγγραφή t η ποινή normalized certainty penalty θα είναι:*

$$NCP(t) = \sum_{i=1}^{n}(w_i NCP_{A_i}(t)).$$

Τέλος πρέπει αν θέλουμε να υπολογίσουμε την συνολική για ένα block του πίνακα ή για όλο τον πίνακα, δεν έχουμε παρά να αθροίσουμε όλες τις ποινές για κάθε εγγραφή και το συμβολίζουμε με: $NCP(Q) = \sum_{\forall t \in T} NCP(t)$.

2.4.2 Οι αλγόριθμοι

Η εύρεση της βέλτιστης λύσης έτσι ώστε να έχουμε το optimal-utility είναι NP-hard ακόμα και αν έχουμε μόνο κατηγορικά δεδομένα. Για αυτό το λόγο έχουν προταθεί δύο greedy αλγόριθμοι ο bottom-up και ο top-down

Η bottom-up μέθοδος

Δίνεται ο αλγόριθμος:

Algorithm 5 bottom-up method

INPUT : a table t, parameter k, weights of attributes and hierarchies on categorical attributes
OUTPUT : a k-anonymous table T'
Initialization: create a group for each tuple.
while there exists some group G such that $|G| < k$ DO **do**
  for each group G such that $|G| < k$ DO **do**
    scan all other groups once to find group G' such that $NCP(G \cup G')$ us minimized
    merge G and G'
  end for
  for each group G such that $|G| \geq 2k$ DO **do**
    split the group into $|G|/k$ groups such that each group has at least k tuples
  end for
end while
return generalize and output the surviving groups

Η λογική είναι η εξής: για να πετύχουμε το μέγιστο utility αρκεί να ομαδοποιούμε groups με την μικρότερη ποινή. Στην αρχή κάθε εγγραφή είναι ακριβώς ένα group και σταδιακά συγχωνεύουμε τα groups με την μικρότερη ποινή, ώστε κάποια στιγμή κάθε group να έχει τουλάχιστον k εγγραφές. Ο αλγόριθμος είναι greedy αφού επιλέγουμε τοπικά την βέλτιστη λύση. Είναι πιθανόν κατά το στάδιο της συγχώνευσης να προκύψουν groups μεγαλύτερο του 2k για αυτό το λόγο αυτά τα groups τα σπάμε στην μέση.

Ο αλγόριθμος σε αντίθεση με όλους τους άλλους δεν προσπαθεί να πετύχει το μικρότερο δυνατόν μέγεθος για τα groups αλλά να ελαχιστοποιήσει την ποινή. Βασικό μειονέκτημα του αλγορίθμου είναι η μεγάλη πολυπλοκότητα του, η οποία είναι $O(log(k)|T|^2)$. Ο αλγόριθμος είναι τύπου local recoding, αφού τακτοποιεί τις εγγραφές τοπικά.

Η top-down μέθοδος

Βασικό μειονέκτημα του προηγούμενου αλγορίθμου ήταν η ταχύτητά του. Ο επόμενος αλγόριθμος, αν και δίνει χειρότερο αποτέλεσμα ως προς το utility είναι πιο γρήγορος. Δίνεται ο αλγόριθμος:

Algorithm 6 Top-down method

INPUT : a table t, parameter k, weights of attributes and hierarchies on categorical attributes
OUTPUT : a k-anonymous table T'
if $|T| \leq k$ THEN **then**
  return
else
  partition T into two exclusive subsets T_1 and T_2 such that T_1 and T_2 are more local than T and either T_1 or T_2 has at least k tuples.
  if $|T_1| > k$ **then**
    recursively partition T_1
  end if
  if $|T_2| > k$ **then**
    recursively partition T_2
  end if
end if
adjust the groups so that each group has at least k tuples

Η βασική ιδέα είναι η εξής: παίρνουμε τον αρχικό πίνακα και τον χωρίζουμε σε υποπίνακες ώστε ο καθένας τοπικά να διατηρεί μεγαλύτερη πληροφορία. Είναι πιο πιθανόν εξάλλου όταν σπάμε ένα partition να πετύχουμε μικρότερη ποινή. Στο τέλος απλά συγχωνεύουμε τα groups τα οποία είναι μικρότερα από το k έτσι ώστε να ικανοποιούμε το k-anonymity. Πιο συγκεκριμένα, το βασικό μας πρόβλημα είναι πως θα χωρίσουμε το partition έτσι ώστε να μειώσουμε την ποινή. Έχει επιλεχθεί η παρακάτω ευριστική μέθοδος:

- επιλέγουμε δύο αρχικές εγγραφές ώστε το NCP να είναι το μέγιστο δυνατόν, και εν συνεχεία οι δύο εγγραφές ορίζουν δύο διαφορετικά partitions.
- Για όλες τις υπόλοιπες εγγραφές τις βάζουμε στο group στο οποίο εισάγουν την μικρότερη δυνατή ποινή δηλαδή αν έχω το group G_1 και το group G_2 η εγγραφή w θα προστεθεί σε κάποιο από τα δύο groups με βάση τις τιμές $NCP(G_1, w)$ και $NCP(G_2, w)$. Για την ακρίβεια θα μπει στο group για το οποίο το NCP είναι το μικρότερο.

Άρα το μόνο πρόβλημα πια είναι ποιες θα είναι οι αρχικές μας εγγραφές. Το κόστος για να βρούμε δύο εγγραφές u,v από ένα partition G έτσι ώστε το $NCP(u, v)$ να είναι μέγιστο είναι $O(T^2)$. Για να περιοριστεί αυτό το κόστος, χρησιμοποιείται μία ευριστική μέθοδος. Σε πρώτη φάση επιλέγουμε τυχαία μία εγγραφή u_1. Εν συνεχεία σκανάρουμε μία φορά το partition και βρίσκουμε την εγγραφή v_1 έτσι ώστε $NCP(u_1, v_1)$ να είναι μέγιστο. Μετά επιλέγουμε μία άλλη εγγραφή u_2 έτσι ώστε πάλι το $NCP(v_1, u_2)$ να είναι μέγιστο. Η διαδικασία επαναλαμβάνεται όταν το NCP σταματάει να αυξάνεται σημαντικά. Επί της ουσίας αν επιλέξουμε ένα μικρό αριθμό επαναλήψεων (για παράδειγμα 6) μπορούμε να επιτύχουμε σχετικά ικανοποιητική προσέγγιση για το NCP.

Τέλος ο αλγόριθμος έχει ακόμα ένα βήμα, στο οποίο φιξάρονται τα groups με μέγεθος μικρότερο του k. Για να το επιτύχουμε αυτό αρκεί να εφαρμόσουμε μία παραλλαγή του bottom-up αλγορίθμου. Έχουμε δύο επιλογές. Έστω ότι έχουμε ένα group G, τότε από αναζητάμε κάποιο set G' με εγγραφές $(k-|G|)$ το οποίο περιέχεται σε ένα group με τουλάχιστον $2k-|G|$ εγγραφές έτσι ώστε $NCP(G \cup G')$ να είναι ελάχιστο. Δηλαδή το group G προσπαθεί να κλέψει εγγραφές από κάποιο άλλο group έτσι ώστε να ικανοποιεί το k-anonymity και παράλληλα να έχει την μικρότερη ποινή στην απώλεια πληροφορίας. Η άλλη μας επιλογή είναι το group G να συγχωνευθεί άμεσα με το πιο κοντινό του group, δηλαδή να το συγχωνεύσουμε με το group

το οποίο μας δίνει την μικρότερη ποινή. Ο δρόμος που θα επιλεγεί, είναι προφανές αυτός ο οποίος τοπικά μας δίνει την βέλτιστη λύση.

Ο αλγόριθμος έχει πολυπλοκότητα $O(T^2)$. Τόσα το στάδιο του partitioning όσο και το στάδιο του adjustment έχουν πολυπλοκότητα $O(T^2)$. Ο αλγόριθμος είναι local recoding αφού είναι πιθανόν εγγραφές με το ίδιο quasi-identifier να αντιστοιχούν σε διαφορετικά group.

2.4.3 Σχόλια

Το πλεονέκτημα των δύο συγκεκριμένων μεθόδων έναντι των προηγούμενων αλγορίθμων είναι προφανές. Όχι μόνο μας δίνουνε ένα σαφή μέτρο για την απώλεια της πληροφορίας, αλλά παράλληλα επιλέγουν να ομαδοποιήσουν τις εγγραφές με βάση την ελαχιστοποίηση της απώλειας της πληροφορίας. Αντίθετα όλοι οι προηγούμενοι αλγόριθμοι προσπαθούσαν να διατηρήσουν τα partitions όσο το δυνατόν μικρότερα, περιορίζοντας έτσι την απώλεια πληροφορίας. Αυτό όμως δεν αποτελούσε ικανοποιητική λύση. Έτσι οι δύο αυτές μέθοδοι είναι σε θέση να απαντήσουν με μικρότερα σφάλματα διάφορα ερωτήματα και αναλύσεις που πιθανόν να επιθυμεί να κάνει ο χρήστης. Το μειονέκτημα αυτών των δύο αλγορίθμων είναι επίσης προφανές. Είναι πιο αργοί από τον αλγόριθμο multi-dim (mondrian). Ο τελευταίος έχει καλύτερη χρονική και χωρική απόδοση και από τους δύο. Παρ΄ όλα αυτά, αυτό δεν είναι ιδιαίτερα σημαντικό. Η διαδικασία του privacy είναι συνήθως offline και πραγματοποιείται μία φορά. Άρα είναι ίσως καλύτερο να προτιμήσουμε καλύτερη διατήρηση της πληροφορίας από ένα γρήγορο αποτέλεσμα.

Επίσης αξίζει να σχολιάσουμε το γεγονός, ότι και οι δύο αλγόριθμοι χρησιμοποιούν local recoding. Οι αλγόριθμοι εκμεταλλεύονται το γεγονός ότι η μεθοδολογία του τελευταίου αποφασίζει τοπικά, που θα αντιστοιχηθεί μία εγγραφή. Με αυτό τον τρόπο, οι αλγόριθμοι μπορούνε τοπικά να αποφασίσουν πως θα χωρίσουν τον χώρο, με βάση όποιο κριτήριο επιθυμούν αυτοί. Αντίθετα το global recoding δεν θα τους έδινε αυτή την δυνατότητα πάντα, εξαναγκάζοντας κάποιες φορές να μην επιλέξουν τοπικά την βέλτιστη λύση, ως προς την απώλεια πληροφορίας. Προσέξτε όμως ότι αυτό δεν ισχύει και για τον αλγόριθμο mondrian, γιατί ο τελευταίος δεν έχει ως τοπικό κριτήριο την απώλεια της πληροφορίας, αλλά το μέγεθος του partition.

Τέλος ο αλγόριθμος από την μία είναι partition-based για αριθμητικά δεδομένα, αλλά hierarchy-based για κατηγορικά. Για τα αριθμητικά δεδομένα είναι προτιμότερο να χωρίσουμε τον χώρο σε partitions παρά να ορίσουμε μία ιεραρχία. Εξάλλου, εξ΄ ορισμού ένα range a_1 είναι πιο γενικό από ένα άλλο a_2 όταν τελευταίο περιέχεται μέσα στο πρώτο. Αυτό μας δυσκολεύει και παράλληλα μας περιορίζει στον ορισμό μίας ιεραρχίας, όπου δεν θα μπορούσαμε να έχουμε όποιο range θέλαμε. Για αυτό γενικότερα είναι προτιμότερο να χωρίζουμε το χώρο σε partitions για αριθμητικές ιδιότητες. Δεν ισχύει το ίδιο για τα κατηγορικά δεδομένα. Για τα τελευταία, αν δεν έχουμε κάποια ταξινόμηση, τότε δεν μπορούμε να χωρίσουμε το χώρο όπως πριν. Αλλά ακόμα και αν έχουμε κάποια μορφή ταξινόμησης, είναι δύσκολο να μετρήσουμε την απώλεια πληροφορίας, εν συγκρίσει με ένα δίκτυο γενίκευσης. Η επιλογή αυτή του αλγορίθμου, να χρησιμοποιήσει διαφορετική προσέγγιση για τα αριθμητικά δεδομένα και διαφορετική για τα κατηγορικά το προσφέρει μεγαλύτερη ευελιξία και μικρότερη απώλεια πληροφορίας.

2.5 l-diversity

Το k-anonymity μας επιτρέπει να διασφαλίσουμε ότι η πιθανότητα να αναγνωρίσουμε την ταυτότητα μίας εγγραφής είναι 1/k. Είναι αυτό όμως αρκετό για την διασφάλιση της ανωνυμίας του ατόμου; Το πρόβλημα πολλές φορές δεν είναι μόνο να βρούμε σε ποιον ανήκει μία εγγραφή,

Patient Data				
	Non-sensitive			Sensitive
ID	Zip code	Age	Nationality	Condition
1	13053	28	Russian	Heart Disease
2	13068	29	American	Heart Disease
3	13068	21	Japanese	Viral Infection
4	13053	23	American	Viral Infection
5	14853	50	Indian	Cancer
6	14853	55	Russian	Heart Disease
7	14850	47	American	Viral Infection
8	14850	49	American	Viral Infection
9	13053	31	American	Cancer
10	13053	37	Indian	Cancer
11	13068	36	Japanese	Cancer
12	13068	35	American	Cancer

(α′) Ακατέργαστα Δεδομένα

Patient Data				
	Non-sensitive			Sensitive
ID	Zip code	Age	Nationality	Condition
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

(β′) 4-anonymous

Σχήμα 2.14: Ο πίνακας που πρέπει να επαναδημοσιευτεί

αλλά να διαφυλάξουμε και τα προσωπικά δεδομένα του ατόμου. Για παράδειγμα, έστω ο πίνακας του Σχήματος 2.14 που είναι όντως 4-anonymous. Μπορεί κανείς να παρατηρήσει ότι για το τελευταίο block όλοι έχουν την ίδια ασθένεια. Ακόμα και αν κάποιος δεν γνωρίζει σε ποιον ανήκει ποια εγγραφή, δεν τον ενδιαφέρει, αφού έχει καταφέρει να ανακτήσει απόλυτα τα προσωπικά του δεδομένα. Το φαινόμενο αυτό μάλιστα δεν είναι ιδιαίτερα σπάνιο, αφού είναι αρκετά πιθανόν σε ένα πίνακα να υπάρχουν τιμές με αρκετά συχνή εμφάνιση.

Στόχος του l-diversity είναι να διασφαλίσει όχι την ταυτότητα μίας εγγραφής, αλλά την ανωνυμία των ευαίσθητων δεδομένων.

2.5.1 Επιθέσεις στο k-anonymity

Ας δούμε με την σειρά ποια είναι τα βασικά μειονεκτήματα του k-anonymity:

1. Homogeneity Attack: Είναι πιθανόν κάποιος εξωτερικός χρήστης να γνωρίζει ολόκληρο (ή μέρος) του quasi-identifier ενός ατόμου. Με αυτό τον τρόπο μπορεί να υπολογίσει τα προσωπικά δεδομένα του άτομο, από τον νέο πίνακα, αν αυτά έχουν μεγάλη συχνότητα εμφάνισης. Για παράδειγμα, έστω ξανά ο πίνακας του Σχήματος 2.14, μπορούμε να υπολογίσουμε με πιθανότητα 100 τοις εκατό τα ευαίσθητα δεδομένα των ατόμων 9-12 και με πιθανότητα 50 τοις εκατό των ατόμων 1-4. Πιο συγκεκριμένα ας φανταστούμε την Alice η οποία είναι σε έχθρα με τον Bob. Ο Bob κάποια μέρα αρρωσταίνει και πηγαίνει μέσο ασθενοφόρου στο νοσοκομείο. Η Alice αποκτά τον 4-anonymous πίνακα και γνωρίζει ότι σε μία από όλες τις εγγραφές ανήκει ο Bob. Επίσης η Alice όντας γείτονας του Bob γνωρίζει ότι είναι 31 χρονών, έχει ταχυδρομικό κώδικα 13053 και άρα ξέρει τελικά ότι ανήκει σε μία από τις εγγραφές 9-12. Δεδομένου ότι όλοι οι ασθενείς παρουσιάζουν την ίδια ασθένεια, η Alice μπορεί να γνωρίζει την ασθένεια του Bob. Το φαινόμενο αυτό μάλιστα δεν είναι σπάνιο, σε ένα μεγάλο πίνακα, όπου οι ευαίσθητες τιμές παίρνουνε λίγες τιμές (για παράδειγμα τρεις), είναι πολύ πιθανόν να προκύψει ένα τέτοιο πρόβλημα.

2. Background knowledge attack: Εκτός από την γνώση του quasi-identifier, ο χρήστης μπορεί να έχει κάποιες πληροφορίες που αφορούν την συσχέτιση του ατόμου με κάποιο ευαίσθητο δεδομένο ή γενικότερα του συνόλου με τα ευαίσθητα δεδομένα. Πιο συγκεκριμένα έχουμε δύο είδη :

- Instance-level background knowledge: Είναι πιθανόν ο χρήστης να γνωρίζει ποιες τιμές μπορούνε να πάρουνε τα ευαίσθητα δεδομένα ενός ή περισσοτέρων ατόμων στο πίνακα. Για παράδειγμα η Alice γνωρίζει ότι ο φίλος της ο John νοσηλεύτηκε στο νοσοκομείο τελευταία. Ο φίλος της είναι περίπου 50 ετών. Γνωρίζει με σιγουριά ότι δεν είχε κάποια γρίπη, αφού δεν παρουσίασε τέτοια συμπτώματα και παράλληλα γνωρίζει ότι δεν έχει καρκίνο. Έτσι μπορεί να εξάγει από το πίνακα το συμπέρασμα ότι ο John παρουσίασε Heart Disease.
- Demographic background knowledge: Είναι πιθανόν ο χρήστης να έχει μερική γνώση της κατανομής των ευαίσθητων δεδομένων και των μη ευαίσθητων στο πληθυσμό. Για παράδειγμα μπορεί να γνωρίζει την πιθανότητα $P(t[Condition] =' cancer'|t[Age] \geq 40)$ και με βάση αυτή να ανακαλύψει τα ευαίσθητα δεδομένα ενός ατόμου. Πιο συγκεκριμένα, ας φανταστούμε ότι η Alice έχει έναν συνάδελφο, τον Umeko ο οποίος πρόσφατα επισκέφτηκε το νοσοκομείο. Η Alice γνωρίζει ότι είναι 21 ετών, ότι έχει ταχυδρομικό κώδικα 13068 και ότι είναι γιαπωνέζος. Παράλληλα η Alice γνωρίζει ότι αποκλείεται να εμφάνισε heart disease αφού οι γιαπωνέζοι έχουν πολύ σπάνια ως σχεδόν ποτέ έχουν προβλήματα με την καρδιά τους. Έτσι γνωρίζει ότι ο Umeko έπασχε από γρίπη.

Για όλα τα παραπάνω το k-anonymity δεν μας διασφαλίζει πάντα, αφού δεν ασχολείται με την κατανομή των ευαίσθητων δεδομένων. Αναζητείται λοιπόν μία μεθοδολογία, η οποία να μας διασφαλίζει έναντι της γνώσης της οποίας μπορεί να έχεις ένα εξωτερικός παράγοντας. Το πρόβλημα είναι ότι σπανίως μπορούμε να υπολογίσουμε την γνώση την οποία πιθανώς να έχει ένας ή περισσότεροι εξωτερικοί παράγοντες. Στην συνέχεια αυτή της ενότητας θα παρουσιαστεί ένα ιδανικό μοντέλο, στο οποίο θεωρούμε ότι γνωρίζουμε τι γνωρίζει ο εξωτερικός χρήστης και μετά θα γενικευτεί στην περίπτωση στην οποία προστατεύουμε τον νέο μας πίνακα από επιθέσεις χωρίς να είμαστε σε θέση να ξέρουμε την γνώση των εξωτερικών παραγόντων.

Βασικοί Ορισμοί και Συμβολισμοί

Ορισμός 1. *Sensitive Set. Έστω ένα πίνακας T. Το σύνολο των ιδιοτήτων $S = A_1, ..., A_m$ τις οποίες ένα εξωτερικός παράγοντας δεν θέλουμε να γνωρίζει (ή να μπορεί να τις βρει με αρκετά μικρή πιθανότητα) ονομάζεται sensitive set και οι ιδιότητες αυτού sensitive attributes.*

Βασικοί συμβολισμοί:

- Συμβολίζουμε με $T = t_1, .., t_n$ τον αρχικό μας πίνακα, με ιδιότητες $A = A_1, ..., A_m$.
- Υποθέτουμε ότι ο πίνακας είναι υποσύνολο ενός μεγαλύτερου πληθυσμού Ω.
- Συμβολίζουμε με $t[A_i]$ την τιμή της ιδιότητας A_i στην εγγραφή t.
- Αν $C = C_1, ..., C_p$ υποσύνολο του A τότε το $t[C]$ συμβολίζει την εγγραφή $(t[C_1], ..., t[C_p])$ η οποία είναι η προβολή του t πάνω στις ιδιότητες του C.
- Συμβολίζουμε με S το σύνολο των sensitive attributes, με Q το σύνολο των non-sensitive attributes και με QI το quasi-identifier.
- Συμβολίζουμε με T^* τον νέο δημοσιευμένο πίνακα, στον οποίο με κάποια συνάρτηση αντιστοιχούμε μία εγγραφή $t \rightarrow t^*$.

Bayes-Optimal Privacy

Θα θεωρήσουμε ότι ο πίνακας T τον οποίο επιθυμούμε να επαναδημοσιεύσουμε, ως τον πίνακα T^*, είναι υποσύνολο του πληθυσμού Ω. Επιπλέον υποθέτουμε ότι υπάρχει μόνο μία ιδιότητα που αποτελεί sensitive attribute. Ακόμα υποθέτουμε ότι ο χρήστης ο οποίος θα επιτεθεί (για παράδειγμα η Alice) γνωρίζει την συνάρτηση κατανομής f του πληθυσμού, του Q και του S. Επίσης γνωρίζει ότι υπάρχει η εγγραφή t του ατόμου που αναζητάει στον πίνακα, και γνωρίζει ότι θα γενικευτεί σε κάποια άλλη εγγραφή t^*. Στόχος του εξωτερικού χρήστη είναι δεδομένου την γνώση του για μία εγγραφή $t[Q] = q$ να ανακαλύψει επιπλέον πληροφορία για το $t[S] = s$.

Ο εξωτερικός χρηστής, όπως η Alice, σε πρώτη φάση έχει μία συγκεκριμένη γνώση για την sensitive attribute. Αυτή έχει προκύψει από την background knowledge. Για παράδειγμα η Alice μπορεί από πριν να ήξερε ότι ο Bob έχει καρκίνο με πιθανότητα 50 τοις εκατό, ανεξάρτητα από τον δημοσιευμένο πίνακα. Έχουμε λοιπόν τον παρακάτω ορισμό, για την γνώση του χρήστη πριν τη δημοσίευση ενός πίνακα για μία εγγραφή:

Ορισμός 2. *Prior belief. Η γνώση ενός εξωτερικού χρήστη ότι η sensitive ιδιότητα ενός ατόμου (δηλαδή εγγραφής) έχει τιμή s δεδομένου ότι η τιμή των non-sensitive ιδιοτήτων είναι q και χωρίς να γνωρίζει τον νέο δημοσιευμένο πίνακα ονομάζεται prior belief και δίνεται από τον τύπο:*

$$\alpha_{(q,s)} = P_f(t[S] = s | t[Q] = q).$$

Όταν ο εξωτερικός χρήστης δει τον νέο πίνακα T^*, η γνώση για την τιμή του sensitive attribute του ατόμου θα αλλάξει, για παράδειγμα όταν η Alice είδε τον νέο πίνακα ήξερε σίγουρα ότι ο Bob είχε καρκίνο. Δηλαδή:

Ορισμός 3. *Posterior belief. Η γνώση ενός εξωτερικού χρήστη ότι η sensitive ιδιότητα ενός ατόμου (δηλαδή εγγραφής) έχει τιμή s δεδομένου ότι η τιμή των non-sensitive ιδιοτήτων είναι q και γνωρίζοντας τον νέο δημοσιευμένο πίνακα ονομάζεται posterior belief και δίνεται από τον τύπο: $\beta_{(q,s,T^*)} = P_f(t[S] = s | t[Q] = q \wedge \exists t^* \in T^*, t \to t^*)$.*

Ο παραπάνω τύπος δεν μας δίνει ιδιαίτερες πληροφορίες για την τιμή του β. Αποδεικνύεται το παρακάτω θεώρημα:

Θεώρημα 1. *Έστω :*

- *q η τιμή των nonsensitive ιδιοτήτων του S στον πίνακα T,*
- *q^* η γενικευμένη τιμή του q στον δημοσιευμένο πίνακα T^*,*
- *s μία πιθανή τιμή της sensitive ιδιότητας,*
- *$n_{q^*,s'}$ το πλήθος των εγγραφών $t^* \in T^*$ για τις οποίες $t^*[Q] = q^* \wedge t^*[S] = s'$ και*
- *$f(s'|q^*)$ να είναι η δεσμευμένη πιθανότητα της sensitive ιδιότητα δεδομένου του γεγονότος ότι η nonsensitive ιδιότητα Q μπορεί να γενικευτεί στην q^*.*

Ισχύει η παρακάτω σχέση :

$$\beta_{q,s,T^*} = \frac{n_{(q^*,s)} \frac{f(s|q)}{f(s|q^*)}}{\sum_{s' \in S} n_{(q^*,s')} \frac{f(s'|q)}{f(s'|q^*)}}.$$

Η τιμή του β λοιπόν εκφράζει κατά πόσο ο εξωτερικός χρήστης μπορεί να αποκλείσει ή να σιγουρεύσει κάποιες τιμές για την sensitive ιδιότητα. Δίνονται η παρακάτω δύο ορισμοί, οι οποίοι εκφράζουν την ιδιότητα του β.

Ορισμός 4. *Positive Disclosure. Ο δημοσιευμένος πίνακας T^* ο οποίος προέκυψε από τον πίνακα T έχει ως αποτέλεσμα positive disclosure αν ο αντίπαλος μπορεί με επιτυχία να αναγνωρίσει την τιμή μίας sensitive ιδιότητας με υψηλή πιθανότητα, με άλλα λόγια δοθέντος ενός $\delta > 0$ υπάρχει positive disclosure αν $\beta_{q,s,T^*} > 1-\delta$ και υπάρχει $t \in T$ έτσι ώστε $t[Q] = q$ και $t[S] = s$.*

Ορισμός 5. *Negative Disclosure. Ο δημοσιευμένος πίνακας T^* ο οποίος προέκυψε από τον πίνακα T έχει ως αποτέλεσμα negative disclosure αν ο αντίπαλος μπορεί με επιτυχία να αποκλείσει την τιμή μίας sensitive ιδιότητας με υψηλή πιθανότητα, με άλλα λόγια δοθέντος ενός $\epsilon > 0$ υπάρχει negative disclosure αν $\beta_{q,s,T^*} < \epsilon$ και υπάρχει $t \in T$ έτσι ώστε $t[Q] = q$ και $t[S] \neq s$.*

Για παράδειγμα, στην homogeneity επίθεση που περιγράφηκε, η Alice μπόρεσε να ανακαλύψει ότι ο Bob έχει καρκίνο χάρη στην ύπαρξη positive disclosure. Όμοια εξαιτίας της background knowledge η Alice κατάφερε να αποφασίσει ότι ο Umeko δεν έχει καρκίνο, αυτό αποτελεί παράδειγμα negative disclosure.

Προσέξτε ότι η ύπαρξη ενός από τα δύο στον πίνακα δεν πάντα κακό. Αν για παράδειγμα $a_{(q,s)} > 1 - \delta$ τότε ο αντίπαλος δεν θα μάθει κάτι επιπλέον, ακόμα και αν υπάρχει positive disclosure. Ισχύει και ακριβώς το ίδιο για το negative disclosure.

Με βάση τα δύο παραπάνω μπορούμε να ορίσουμε πότε διατηρείται το privacy ως προς τα sensitive attributes. Πιο συγκεκριμένα:

Ορισμός 6. *Uninformative Principle. Ο δημοσιευμένος πίνακας θα πρέπει να παρέχει στον αντίπαλο με λίγη περισσότερη πληροφορία εν συγκρίσει με την υπάρχουσα background knowledge που ήδη έχει. Δηλαδή θα πρέπει να υπάρχει μικρή διαφορά μεταξύ της prior και posterior beliefs.*

Η διαφορά μπορεί να μετρηθεί με διάφορους τρόπους. Οι μεθοδολογίες οι οποίες στηρίζονται στον παραπάνω ορισμό αναφέρονται και ως Bayes-optimal privacy definition. Ένας τρόπος να μετρήσουμε την διαφορά είναι ο ορισμός του (p_1, p_2)-privacy breach. Σύμφωνα με αυτόν τον ορισμό το privacy δεν ικανοποιείται όταν ισχύει ή $\alpha_{(q,s)} < p_1 \wedge \beta_{(q,S,T^*)} > p_2$ ή $\alpha_{(q,s)} > 1 - p_1 \wedge \beta_{(q,S,T^*)} < 1 - p_2$. Μία άλλη παραλλαγή του παραπάνω ορισμού είναι να απαιτούμε η διαφορά του α και του β να μην ξεπερνάει πότε ένα όριο.

Οι μεθοδολογίες όμως του Bayes-optimal privacy έχουν κάποια σημαντικά μειονεκτήματα, καθιστώντας τες στην πράξη μη εφαρμόσιμες.

- Έλλειψη γνώσης. Αυτός ο οποίος δημοσιεύει τον πίνακα δεν γνωρίζει πάντα την πλήρη συνάρτηση κατανομής f τόσα για τις sensitive όσο και για τις nonsensitive ιδιότητες στον πληθυσμό Ω του οποίο δείγμα είναι ο T.

- Η γνώση του αντιπάλου-εξωτερικού παράγοντα. Επιπλέον είναι πολύ πιθανόν ο αντίπαλος να μην έχει και αυτός πλήρη γνώση της συνάρτησης κατανομής. Όμως αυτός που εκδίδει τον πίνακα δεν μπορεί να γνωρίζει την ακριβή γνώση του αντιπάλου.

- Instance-Level Knowledge Το συγκεκριμένο μοντέλο δυστυχώς δεν μας καλύπτει έναντι αυτής της περίπτωσης, αφού δεν μπορεί να προσεγγιστεί από κάποιο πιθανοτικό μοντέλο.

- Το πλήθος των αντιπάλων-εξωτερικών παραγόντων. Μπορεί να υπάρχουν διάφοροι αντίπαλοι, οι οποίοι να έχουν διαφορετική γνώση. Είναι δύσκολο να προσομοιώσουμε και να προσεγγίσουμε τις γνώσεις του κάθε αντιπάλου.

2.5.2 l-diversity

Λύση στα παραπάνω προβλήματα έρχεται να δώσει η θεωρία του l-diversity. Υπενθυμίζουμε ότι μέχρι τώρα θεωρούμε ότι έχουμε μόνο μία sensitive ιδιότητα.

Ας δούμε όμως ξανά πως υπολογίζουμε ξανά την πίστη belief του αντιπάλου. Έστω ένα $q^*-block$ ένα σύνολο από εγγραφές στο T^* των οποίων οι nonsensitive ιδιότητες γενικεύονται στο q^*. Ας δούμε αρχικά την περίπτωση των positive disclosures. Ένας αντίπαλος, για παράδειγμα η Alice είναι σε θέση να γνωρίζει ότι ένα άτομο, ο Bob στην περίπτωση μας, έχει $t[S] = s$ με υψηλή πιθανότητα όταν ισχύει:

$$\exists s, \forall s' \neq s, n_{(q^*,s')} \frac{f(s'|q)}{f(s'|q^*)} \ll n_{(q^*,s)} \frac{f(s|q)}{f(s|q^*)}.$$

Αυτή η σχέση μπορεί να ικανοποιηθεί σε δύο περιπτώσεις:

- Lack of Diversity Αν υπάρχει έλλειψη ποικιλίας για την sensitive ιδιότητα. Αυτό συμβαίνει όταν:

  $$\exists s, \forall s' \neq s, n_{(q^*,s')} \ll n_{(q^*,s)}$$

  Σε αυτή την περίπτωση όλες σχεδόν οι εγγραφές έχουν την ίδια τιμή s για την sensitive ιδιότητα S και έτσι $\beta_{(q,s,T^*)} \approx 1$. Αυτή η περίπτωση μάλιστα είναι εύκολο να ελεγχθεί με ένα απλό count στο δημοσιευμένο πίνακα για κάθε group.

- Strong Background Knowledge Η δεύτερη περίπτωση η οποία μπορεί να οδηγήσει σε positive disclosure είναι η υπάρχουσα γνώση του αντιπάλου. Αυτό συμβαίνει όταν ισχύει η παρακάτω σχέση:

  $$\exists s', \frac{f(s'|q)}{f(s'|q^*)} \approx 0.$$

  Με άλλα λόγια το άτομο με quasi identifier $t[Q] = q$ έχει πολύ λίγες πιθανότητες να έχει sensitive value s' και άρα πιο πιθανόν να έχει s.

Πως μπορούμε να προστατευτούμε έναντι αυτών των δύο περιπτώσεων; Μπορούμε να εξασφαλίσουμε αρχικά το diversity αν απαιτήσουμε ότι όλες οι πιθανές τιμές $s' \in domain(S)$ εμφανίζονται εξίσου συχνά σε κάθε $q^*-block$. Αυτό όμως έχει το πρόβλημα ότι αν το domain είναι αρκετά μεγάλο τότε και τα $q^* - blocks$ θα πρέπει να είναι πολύ μεγάλα, έχοντας έτσι μεγάλη απώλεια πληροφορίας. Μία καλύτερη λύση για να εξασφαλίσουμε το diversity είναι να απαιτήσουμε κάθε $q^* - block$ να έχει τουλάχιστον $l \geq 2$ διαφορετικές sensitive τιμές έτσι ώστε οι l πιο συχνά εμφανιζόμενες τιμές στο $q^* - block$ να έχουν περίπου την ίδια συχνότητα εμφάνισης. Θα λέμε τότε ότι το $q^* - block$ είναι well-represented από l sensitive τιμές.

Με τον παραπάνω τρόπο όχι μόνο διασφαλίζουμε ότι δεν θα έχουμε lack of diversity αλλά παράλληλα διασφαλιζόμαστε έναντι της ύπαρξης background knowledge. Ο αντίπαλος πρέπει να έχει γνώση για $l-1$ τιμές ώστε να μπορέσει να βγάλει ασφαλή συμπεράσματα. Ακόμα και να γνωρίζει απόλυτα ότι μπορεί να απορρίψει μία τιμή, γεγονός το οποίο πιθανόν να οφείλονται σε instance-level knowledge πάλι θα πρέπει να βρει ένα τρόπο να απορρίψει άλλες $l-2$.

Με άλλα λόγια αν έχουμε l well represented sensitive τιμές σε κάποιο $q^* - block$ τότε έχουμε σε ένα βαθμό διασφαλίσει την ανωνυμία των δεδομένων μας. Η παράμετρος l καθορίζει το βαθμό, ακόμα και αν η background γνώση μας είναι πλήρως άγνωστη.

Έχουμε λοιπόν τον παρακάτω ορισμό:

Ορισμός 7. *l-diversity. Ένα $q^* - block$ είναι l-diverse αν περιέχει τουλάχιστον l well-represented τιμές για την sensitive ιδιότητα S. Ένας πίνακας είναι l-diverse αν κάθε $q^*-block$ είναι l-diverse.*

	Non-sensitive			Sensitive
ID	Zip code	Age	Nationality	Condition
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

Σχήμα 2.15: 3-diverse πίνακας

Ας δούμε λοιπόν τον πίνακα του Σχήματος 2.15. Μπορούμε να παρατηρήσουμε ότι στον νέο πίνακα όπου είναι 3-diverse η Alice δεν μπορεί να ανακαλύψει ούτε την ασθένεια του Bob ούτε του Umeko.

2.5.3 *l*-diversity instantiations

Ο παραπάνω ορισμός για το *l*-diversity δεν είναι σαφής, αφού δεν ορίζει τι ακριβώς θα πει well represented. Για την ακρίβεια δεν αποτελεί ακριβή ορισμό, αλλά μία εποπτική θεώρηση. Θα προσπαθήσουμε παρακάτω να δώσουμε διάφορους ορισμούς για το *l*-diversity

Ορισμός 8. *Entropy l-diversity. Ένας πίνακας είναι entropy l-diverse αν για κάθε $q^*-block$ ισχύει:*

$$-\sum_{s\in S} p_{(q*,s)} \log(p_{(q*,s)}) \geq \log(l)$$

όπου $p_{(q*,s)} = \frac{n_{(q*,s)}}{\sum_{s'\in S} n_{(q*,s')}}$

Με βάση αυτό τον ορισμό κάθε $q^* - block$ έχει τουλάχιστον l διαφορετικές τιμές για το sensitive value. Παρ' όλα αυτά αυτός ο ορισμός έχει ένα σημαντικό μειονέκτημα. Αφού η $-x\log x$ είναι φθίνουσα συνάρτηση τότε αν χωρίσουμε το $q^* - block$ σε δύο υπο-block q_a και q_b τότε ισχύει $entropy(q^*) \geq \min(entropy(q_a), entropy(a_b))$ Αυτό άμεσα συνεπάγεται ότι για να ικανοποιεί κάθε block το *l*-diversity τότε θα πρέπει και η εντροπία του αρχικού πίνακα να είναι τουλάχιστον $\log(l)$. Στην πραγματικότητα όμως αυτό δεν είναι πάντα εφικτό. Για παράδειγμα, είναι πολύ πιθανόν στον πίνακα μας με τις ασθένειες το 90 τοις εκατό των ασθενών να εμφανίσει viral infection. Τότε ο συγκεκριμένος ορισμός είναι αρκετά περιοριστικός.

Αναζητάμε λοιπόν έναν ορισμό ώστε κάποιες positive disclosures να είναι αποδεκτές. Για αυτό το λόγο θα ορίσουμε μία άλλη μορφή της *l*-diversity την recursive *l*-diversity. Έστω ότι έχουμε $s_1, \ldots, s_2$ πιθανές τιμές για την sensitive ιδιότητα S στο $q^* - block$. Ας υποθέσουμε ότι ταξινομούμε τα $n_{(q^*,s_1)}, \ldots, n_{(q^*,s_m)}$ σε φθίνουσα σειρά και την νέα σειρά την συμβολίζουμε ως : $r_1, \ldots, r_m$. Ένας τρόπος να σκεφτούμε το *l*-diversity, είναι να θεωρήσουμε ότι ο αντίπαλος πρέπει για να πετύχει positive disclosure για μία sensitive ιδιότητα πρέπει να μπορεί να εξαλείψει τουλάχιστον $l-2$ πιθανές τιμές. Για παράδειγμα αυτό θα σήμαινε ότι σε ένα 2-diverse πίνακα, καμία τιμή δεν θα έπρεπε να εμφανίζεται ιδιαίτερα συχνά. Ένας τρόπος λοιπόν να το πετύχουμε αυτό είναι ο παρακάτω ορισμός:

Ορισμός 9. *Positive Disclosure-Recursive(c,l)-Diversity. Έστω ότι έχουμε ένα $q^* - block$ και ότι r_i συμβολίζει το πλήθος των εμφανίσεων της i^{th} πιο συχνής sensitive τιμής στο*

συγκεκριμένο block και έστω ότι έχουμε m πιθανές τιμές για την sensitive ιδιότητα. Έστω Y το σύνολο των sensitive τιμών για τις οποίες επιτρέπεται positive disclosure και έστω ότι η y^{th} είναι η πιο συχνή sensitive τιμή η οποία δεν ανήκει στο Y (δηλαδή είναι το $r_y \in (r_1, \ldots, r_m)$ έτσι ώστε $r_y \notin Y$). Δοθέντος μίας σταθεράς c θα λέμε ότι το $q^* - block$ ικανοποιεί την pd-recursive (c,l)-diversity αν ισχύει ένα από τα παρακάτω:

- $y \leq l-1 \wedge r_y < c(r_l + r_{l+1} + \ldots + r_m)$ *ή*
- $y > l-1 \wedge r_y < c\sum_{j=l-1}^{y-1} r_j + c\sum_{j=y+1}^{m} r_j$

Θα λέμε ότι ο πίνακας T^* ικανοποιεί την positive disclosure-recursive (c,l)-diversity αν κάθε $q^* \in T^*$ ικανοποιεί την pd-recursive (c,l)-diversity.

Με άλλα λόγια, σύμφωνα με τον ορισμό, απαιτούμε η πιο συχνά εμφανιζόμενη τιμή για την οποία δεν επιτρέπεται positive disclosure να εμφανίζεται περίπου (εξαρτάται από το c) εξίσου συχνά με τουλάχιστον άλλες $l-1$ ή αρκετά πιο σπάνια εν συγκρίσει με τις positive disclosure τιμές. Προσέξτε ότι αν $r_y = 0$ τότε οι σχέσεις ισχύουν πάντα, αφού επιτρέπονται για όλα τις ιδιότητες positive disclosure. Επίσης αξίζει να σημειώσουμε ότι αν $c > 1$ τότε αν ισχύει $y > l-1$ η άλλη ανισότητα ισχύει άμεσα, αφού $r_y \leq r_{l-1}$.

Αυτός ο ορισμός όντως μας καλύπτει για περιπτώσεις που για κάποιες τιμές όπου έχουμε positive disclosure αλλά δεν μας καλύπτει για τις περιπτώσεις όπου έχουμε negative disclosure. Έτσι έχουμε τον επόμενο ορισμό:

Ορισμός 10. *Negative/Positive Disclosure-Recursive(c_1, c_2, l)-Diversity. Έστω W το σύνολο των sensitive τιμών για τις οποίες δεν επιτρέπεται negative disclosure. Θα λέμε ότι ένας πίνακας ικανοποιεί το npd-recursive(c_1, c_2, l)-diversity αν ικανοποιεί το pd-recursive(c_1, l)-diversity και αν κάθε $s \in W$ εμφανίζεται τουλάχιστον c_2 τοις εκατό στις εγγραφές κάθε $q^* - block$.*

Μέχρι τώρα είδαμε πως μπορούμε να ορίσουμε το l-diversity για μία μόνο ιδιότητα. Επεκτείνουμε τον ορισμό μας όταν έχουμε πολλές ιδιότητες οι οποίες να είναι sensitive:

Ορισμός 11. *Multi-Attribute l-Diversity. Έστω ένας πίνακας T με nonsensitive ιδιότητες $Q_1, \ldots, Q_m$ και sensitive ιδιότητες $S_1, \ldots, S_n$. Θα λέμε ότι ο T είναι l-diverse αν για κάθε $i = 1 \ldots m$, ο πίνακας T είναι l-diverse αν το S_i θεωρηθεί ότι είναι η sensitive ιδιότητα και το $\{Q_1, \ldots, Q_n, S_1, \ldots, S_i, \ldots, S_m\}$ θεωρηθεί το quasi-identifier.*

Έτσι ορίσαμε πως μπορούμε να ορίσουμε την έννοια well-represented values και το l-diversity για πολλές ιδιότητες. Μπορεί μάλιστα κανείς να παρατηρήσει ότι όσο μεγαλώνει το πλήθος των sensitive ιδιοτήτων είναι πολύ πιθανόν να χρειαστούμε αρκετά μεγαλύτερα block ώστε να ικανοποιούμε το l-diversity.

Αξίζει να σχολιάσουμε την μονοτονία του l-diversity. Αποδεικνύεται ότι αν ο πίνακας T^* ικανοποιεί ή την entropy l-diversity ή την npd recursive(c_1, c_2, l)-diversity τότε κάθε γενίκευση T^{**} του T^* θα την ικανοποιεί επίσης. Αντίθετα αυτό δεν ισχύει για την bayes optimal privacy.

2.5.4 Σχόλια

Ας θυμηθούμε ξανά το προβλήματα που είχανε ανακύψει και πως το l-diversity κατάφερε εν μέρει να τα λύσει. Είναι πιθανόν ο εξωτερικός χρήστης -αντίπαλος να μπορέσει να ανακτήσει επιπλέον πληροφορίες για ένα άτομο, ακόμα και αν ικανοποιείται το k-anonymity. Αυτό μπορούσε να συμβεί είτε επειδή κάποια εγγραφή εμφανιζόταν πολύ συχνά στον πίνακα, είτε

επειδή ο αντίπαλος είχε κάποια επιπλέον γνώση. Στην προσπάθεια να αποτραπούν τέτοιες επιθέσεις δυστυχώς δεν μπορούμε να γνωρίζουμε ούτε την κατανομή των τιμών στο πληθυσμό ούτε την γνώση του αντιπάλου καθώς και το πλήθος των αντιπάλων με πιθανόν διαφορετική γνώση. Ας δούμε λοιπόν τι πέτυχε το l-diversity:

- Δεν χρειάζεται να γνωρίζουμε πλήρως την συνάρτηση κατανομής τόσο των sensitive όσο και των nonsensitive ιδιοτήτων.
- Δεν είναι απαραίτητο αυτός που εκδίδει τον πίνακα να έχει όση γνώση ή περισσότερη από τον αντίπαλο που θέλει να αποτρέψει, η παράμετρος l είναι αυτή που μας προστατεύει, όσο μεγαλύτερη, τόσο πιο εύκολα αποτρέπουμε αντιπάλους με μεγάλη γνώση.
- Η instance level knowledge λαμβάνεται και καλύπτεται επίσης αυτόματα, αφού επί της ουσίας την χειριζόμαστε σαν ένα οποιοδήποτε τρόπο απόκλισης sensitive τιμών.
- Το πλήθος των αντιπάλων, καθώς και η γνώση αυτών (ακόμα και αν είναι διαφορετική) οδηγώντας σε διαφορετικές επιθέσεις, δεν μας επηρεάζει.

Συνολικά θα λέγαμε ότι το l-diversity συνήθως εξασφαλίζει περισσότερο την ανωνυμία από k-anonymity (αν μπορούσαμε να ισχυριστούμε κάτι τέτοιο). Το τελευταίο διασφαλίζει την ταυτότητα της εγγραφής, ενώ το πρώτο τα ευαίσθητα δεδομένα. Πιο συγκεκριμένα το l-diversity μας διασφαλίζει την ανωνυμία των ευαίσθητων δεδομένων. Μάλιστα θα μπορούσε κανείς να ισχυριστεί ότι είναι πολύ καλύτερος τρόπος διασφάλισης της ανωνυμίας από το k-anonymity αν είχαμε να επιλέξουμε μεταξύ των δύο. Για την ακρίβεια αν θεωρήσουμε ότι ο αντίπαλος γνωρίζει ότι η εγγραφή υπάρχει μέσα στον εκδιδόμενο πίνακα, τότε δεν έχουμε κανένα λόγο να προστατέψουμε την ταυτότητα αυτής. Αντίθετα μας αρκεί να προστατέψουμε τα ευαίσθητα δεδομένα. Μάλιστα ο αλγόριθμος ο οποίος θα δούμε στην συνέχεια δεν προστατεύει την ταυτότητα. Μάλιστα αυτό είναι κάποιες φορές προτιμότερο, αφού έτσι πιθανόν να διατηρούμε περισσότερη πληροφορία, μην εφαρμόζοντας κάποια γενίκευση, ή ελαχιστοποιώντας και άλλο την γενίκευση στο quasi-identifier. Παρ΄ όλα αυτά αν ξέρουμε ότι ο αντίπαλος δεν γνωρίζει ποιες εγγραφές υπάρχουν στο πίνακα και επιθυμούμε την προστασία της ταυτότητας το l-diversity δεν μας καλύπτει. Θα πρέπει να γίνει κάποια γενίκευση στο quasi-identifier (όχι απαραίτητα k-anonymity).

2.6 Anatomy

Όπως είδαμε και πριν, ένα τρόπος να εξασφαλίσουμε το l-diversity είναι να χρησιμοποιήσουμε κάποια παραλλαγή, κάποιου αλγόριθμου για το k-anonymity, μόνο που κάθε φορά αντί να ελέγχουμε το τελευταίο θα ελέγχουμε το πρώτο. Αυτό όμως μπορεί να οδηγήσει σε απώλεια πληροφορίας. Μάλιστα πολλές φορές δεν έχουμε λόγο να γενικεύσουμε τις sensitive ιδιότητες και πιθανόν και το quasi-identifier.

Η μεθοδολογία η οποία θα παρουσιαστεί στην συνέχεια, δεν γενικεύει τα δεδομένα όπως οι προηγούμενοι αλγόριθμοι. Αντίθετα κάνεις το εξής, δεδομένου ενός πίνακα T κατασκευάζει ένα νέο πίνακα έτσι ώστε κάθε εγγραφή να μπορεί να πάρει τουλάχιστον l sensitive τιμές, χωρίς να ξέρουμε ποια είναι αυτή. Για παράδειγμα ας δούμε το Σχήμα 2.16. Μπορεί κανείς να παρατηρήσει ότι κάθε εγγραφή έχει συσχετιστεί με ακριβώς l (ή παραπάνω) τιμές. Μπορεί μάλιστα να παρατηρήσει ότι έτσι διατηρείται και το l-diversity αλλά και περισσότερη πληροφορία. Πιο συγκεκριμένα, έχει αποσυσχετιστεί το quasi-identifier από τα sensitive δεδομένα. Επιλέγουμε εν συνεχεία να συσχετίσουμε κάθε εγγραφή με ένα group το οποίο περιέχει τουλάχιστον l τιμές. Επί της ουσίας, η διαδικασία, αυτή, αποτελεί μία μέθοδο γενίκευσης, αλλά όχι όπως την παρουσιάσαμε τις προηγούμενες φορές. Προσέξτε μάλιστα ότι το k-anonymity

Patient Data				
	Non-sensitive			Sensitive
ID	Age	Sex	Zip code	Condition
1	23	M	11000	pneumonia
2	27	M	13000	dyspepsia
3	35	M	59000	dyspepsia
4	59	M	12000	pneumonia
5	61	F	54000	flu
6	65	F	25000	gastritis
7	65	F	25000	flu
8	70	F	30000	bronchitis

(α′) Ακατέργαστα Δεδομένα

Patient Data				
ID	Age	Sex	Zip code	Group-ID
1	23	M	11000	1
2	27	M	13000	1
3	35	M	59000	1
4	59	M	12000	1
5	61	F	54000	2
6	65	F	25000	2
7	65	F	25000	2
8	70	F	30000	2

(β′) QIT table

Patient Data		
Group-ID	Disease	Count
1	dyspepsia	2
1	pneumonia	2
2	bronchitis	1
2	flu	2
2	gastritis	1

(γ′) ST table

Σχήμα 2.16: anatomized tables

δεν ικανοποιείται όπως και η διασφάλιση της ταυτότητας της εγγραφής. Αυτό όπως θα δειχθεί αργότερα δεν αποτελεί απαραίτητα μειονέκτημα.

Εν συνεχεία θα παρουσιάσουμε πως η συγκεκριμένη μεθοδολογία προστατεύει τα δεδομένα, διατηρεί περισσότερη πληροφορία και τέλος έναν αλγόριθμο για να επιτύχουμε αυτό το αποτέλεσμα.

2.6.1 Βασικοί Ορισμοί

Όπως και στις προηγούμενες ενότητες, έχουμε ένα πίνακα T τον οποίο επιθυμούμε να δημοσιεύσουμε. Θεωρούμε ότι ο πίνακας έχει ένα d-διάστατο quasi-identifier (QI) με ιδιότητες $A_1, \ldots, A_d$ και μία μόνο sensitive ιδιότητα A^s. Κάθε ιδιότητα του QI μπορεί να είναι είτε αριθμητική είτε κατηγορική, αλλά η sensitive ιδιότητα μόνο κατηγορική!

Για κάθε εγγραφή $t \in T$ συμβολίζουμε με $t[i](1 \leq i \leq d)$ την τιμή της ιδιότητας A_i και με $t[d+1]$ την τιμή της A^s. Τέλος το $d+1$ χώρο τον συμβολίζουμε με DS (από το dimensional data space).

Όπως και τα προηγούμενα μοντέλα, έτσι και τώρα πρέπει να χωρίσουμε σε partitions τον χώρο. Δίνουμε τον παρακάτω ορισμό:

Ορισμός 1. *Partition/QI-group. Ένα partition αποτελείται από διάφορα υποσύνολα του T έτσι ώστε κάθε εγγραφή στο T να ανήκει ακριβώς σε ένα υποσύνολο. Αναφερόμαστε σε αυτά τα υποσύνολα σαν QI-groups και τα συμβολίζουμε $QI_1, \ldots, QI_m$. Με άλλα λόγια ένα partition του T είναι ένα σύνολο ξένων συνόλων των οποίων η ένωση μας δίνει το T.*

Εμείς επιθυμούμε ένα partition το οποίο να είναι l-diverse:

Ορισμός 2. *l-diverse partition. Ένα partition από m QI-groups είναι l-diverse, αν κάθε QI-group $QI_j(1 \leq j \leq m)$ ικανοποιεί την παρακάτω συνθήκη: Έστω u η πιο συχνή τιμή της*

ιδιότητας A^s στο QI_j και $c_j(u)$ ο αριθμός των εγγραφών $t \in QI_j$ έτσι ώστε $t[d+1] = u$, τότε πρέπει $c_j(u)/|QI_j| \leq 1/l$ όπου $|QI_j|$ ο πληθάριθμος του QI_j.

Προσέξτε ότι αυτός ο ορισμός δεν είναι τόσο αυστηρός όσο αυτοί που δώσαμε στην προηγούμενη ενότητα. Είναι αρκετά δύσκολο να βρούμε ακόμα ποιες εγγραφές ικανοποιούν το positive ή negative disclosure. Όμως η μεθοδολογία που θα περιγράψουμε παρακάτω, μπορεί να επεκταθεί για ικανοποιήσει και τους άλλους ορισμούς του l-diversity.

Όπως είδαμε στην εισαγωγή το anatomy κατασκευάζει δύο νέους πίνακες, τον QIT και τον ST ώστε να ικανοποιήσει το l-diversity. Πιο συγκεκριμένα:

Ορισμός 3. *Anatomy. Δοθέντος ενός l-diverse partition με m QI-groups, ο anatomy παράγει ένα quasi-identifier table (QIT) και ένα sensitive value (ST) με βάση τις παρακάτω ιδιότητες:*

- *Το QIT έχει το σχήμα $(A_1, \ldots, A_d, Group-ID)$.*
- *Για κάθε QI-group $QI_j (1 \leq j \leq m)$ και για κάθε εγγραφή $t \in QI_j$, το QIT έχει μία εγγραφή της μορφής: $(t[1], t[2], ..., t[d], j)$*
- *Το ST έχει σχήμα $(Group-ID, A^s, Count)$.*
- *Για κάθε QI-group $QI_j (1 \leq j \leq m)$ και για κάθε τιμή v της sensitive ιδιότητας A^s στο QI_j, το ST έχει μία εγγραφή της μορφής: $(j, v, c_j(v))$, όπου $c_j(u)$ ο αριθμός των εγγραφών $t \in QI_j$ έτσι ώστε $t[d+1] = u$.*
- *Εκτός από τις προηγούμενες εγγραφές δεν υπάρχουν άλλες στους δύο πίνακες.*

Αποδεικνύεται ότι σε ένα πίνακα που έχει εφαρμοστεί το anatomy τότε η πιθανότητα ο αντίπαλος να κατασκευάσει μία εγγραφή όπως ήταν εξαρχής είναι $1/l$.

Για παράδειγμα οι πίνακες που δώσαμε ικανοποιούν το 2-diverse. Το πρόβλημα μας λοιπόν είναι πως θα σπάσουμε τον πίνακα σε υποσύνολα που να ικανοποιούν το l-diversity και μάλιστα με όσο το δυνατόν μικρότερη απώλεια πληροφορίας.

Τέλος επειδή το νέο μοντέλο θα συγκριθεί με την γενίκευση θα δώσουμε τον ακριβή ορισμό της γενίκευσης:

Ορισμός 4. *Generalization. Δοθέντος ενός partition του T με m QI-groups, για κάθε εγγραφή $t \in T$, ο γενικευμένος πίνακας T^*, περιέχει μία εγγραφή της μορφής: $(QI_j[1], ..., QI_j[d], t[d+1])$ όπου $QI_j (1 \leq i \leq m)$ είναι το μοναδικό QI-group το οποίο περιλαμβάνει το t και $QI_j[i] (1 \leq i \leq d)$ είναι ένα interval το οποίο καλύπτει το $t[i]$. Επιπλέον το $QI_j[i]$ είναι ίδιο για όλες τις εγγραφές $t \in QI_j$.*

Αν παρατηρήσει κανείς θα δει ότι αν ο πληθάριθμος κάθε QI-group είναι μεγαλύτερος ή ίσος του k τότε ικανοποιείται το k-anonymity.

2.6.2 Διατήρηση της συσχέτισης

Το πρόβλημα κατά την προσπάθεια διατήρησης της ανωνυμίας είναι διατηρήσουμε την συσχέτιση των δεδομένων, ώστε να έχουμε όσο το δυνατόν περισσότερη πληροφορία. Στην συγκεκριμένη ενότητα θα συγκρίνουμε το anatomy με το generalization, ως προς κατά πόσο διατηρούν την συσχέτιση.

Θα ξεκινήσουμε με ένα παράδειγμα και εν συνεχεία θα γενικεύσουμε τον τρόπο μέτρησης του σφάλματος -απώλεια συσχέτισης. Ο πίνακας που έχουμε δώσει στην αρχή, με βάση το generalization μας δίνει τον πίνακα του Σχήματος 2.17 . Ας μετρήσουμε κατά πόσο

Patient Data				
	Non-sensitive			Sensitive
ID	Age	Sex	Zip code	Condition
1	[21 – 60]	M	[10001 – 60000]	pneumonia
2	[21 – 60]	M	[10001 – 60000]	dyspepsia
3	[21 – 60]	M	[10001 – 60000]	dyspepsia
4	[21 – 60]	M	[10001 – 60000]	pneumonia
5	[61 – 70]	F	[10001 – 60000]	flu
6	[61 – 70]	F	[10001 – 60000]	gastritis
7	[61 – 70]	F	[10001 – 60000]	flu
8	[61 – 70]	F	[10001 – 60000]	bronchitis

Σχήμα 2.17: generalized tables

διατηρείται η συσχέτιση μεταξύ της Age και της Disease. Οι δύο αυτές ιδιότητες ορίζουν έναν δυδιάστατο χώρο $DS_{a,d}$ και κάθε εγγραφή ορίζει ένα σημείο σε αυτόν. Θα μπορούσαμε να μοντελοποιήσουμε την συσχέτιση με συνάρτηση κατανομής (pdf) $G_{t_1} : DS_{A,D} \rightarrow [0, 1]$. Ο ακριβής ορισμός είναι:

$$G_{t_1}(x) = \begin{cases} 1 & \text{if } x = (t_1[A], t_1[D]) \\ 0 & otherwise \end{cases}$$

όπου x είναι μία δυδιάστατη τυχαία μεταβλητή στον δυδιάστατο χώρο.

Ο ερευνητής, επιθυμεί από έναν νέο γενικευμένο πίνακα να κατασκευάσει την συνάρτηση κατανομής με όσο τον δυνατόν μεγαλύτερη επιτυχία. Ας δούμε λοιπόν την πρώτη εγγραφή, με βάση το μοντέλο γενίκευσης, ένας ερευνητής ξέρει ότι η ηλικία είναι μεταξύ του [21,60] και έχει οπωσδήποτε την ασθένεια pneumonia. Άρα η πιθανότητα να βρει ακριβώς το σημείο στο χώρο είναι ίσο με το range[21,60] αφού η ηλικία μπορεί να πάρει κάποιες από αυτές τις τιμές. Έτσι η pdf είναι:

$$\bar{G}^{gen}_{t_1}(x) = \begin{cases} 1/40 & \text{if } x[A] \in [21, 60] \wedge x[D] = pneumonia \\ 0 & otherwise \end{cases}$$

Αντίθετα στο anatomy ο ερευνητής γνωρίζει με ακρίβεια την τιμή της ηλικίας και καλείται να επιλέξει μεταξύ δύο τιμών για τις ασθένειες της pneumonia και της dyspepsia. Άρα η πιθανότητα να ανακαλύψει το ακριβές σημείο στο χώρο είναι ακριβώς 0.5. Η συνάρτηση pdf είναι λοιπόν:

$$\bar{G}^{ana}_{t_1}(x) = \begin{cases} 1/2 & \text{if } x = (23, pneumonia) \text{ or } x = (23, dyspepsia) \\ 0 & otherwise \end{cases}$$

Είναι προφανές ότι η δεύτερη περίπτωση μας δίνει ποιο ακριβές αποτέλεσμα, αφού η ακριβής συνάρτηση κατανομής είναι:

$$G_{t_1}(x) = \begin{cases} 1 & \text{if } x = (23, pneumonia) \\ 0 & otherwise \end{cases}$$

Ένα καλύτερο μέτρο βέβαια για να μετρήσουμε το σφάλμα είναι το μέσο τετραγωνικό σφάλμα των συναρτήσεων κατανομής, δηλαδή:

$$\sum_{x \in DS_{A,D}} (\bar{G}_{t_1}(x) - G_{t_1}(x))^2$$

Στην περίπτωση του anatomy το μέσο τετραγωνικό σφάλμα βγαίνει 0.5 ενώ στην άλλη 22.5!

Ας δώσουμε τώρα τον ακριβή ορισμό:

Ορισμός 5. *pdf-συνάρτηση. Για κάθε εγγραφή t η τιμή της pdf $G_t(x) : DS \rightarrow [0, 1]$:*

$$G_t(x) = \begin{cases} 1 & if\ x = t \\ 0 & otherwise \end{cases}$$

όπου x η τυχαία μεταβλητή στο DS.

Ας δούμε την νέα τιμή αυτής της συνάρτησης για το μοντέλο γενίκευσης για μία εγγραφή t. Η εγγραφή t ανήκει σε κάποιο QI-group QI. Η γενικευμένη μορφή της εγγραφής θα είναι

$$(QI[1], QI[2], \ldots, QI[d], t[d+1]),$$

όπως αυτό ορίστηκε στην προηγούμενη υπο-ενότητα. Συμβολίζουμε το μήκος του $QI[i]$ ως $L(QI[i])$ (αν η τιμή είναι συνεχής τότε το μήκος ορίζεται ίσο με το range και αν είναι διακριτή, ορίζεται ίσο με το πλήθος των δυνατών τιμών του $QI[i]$ στο QI (για παράδειγμα σε ένα δέντρο γενίκευσης θα ήταν ίσο με το πλήθος των φύλλων που έχει ο συγκεκριμένος κόμβος». Τότε η νέα κατασκευασμένη συνάρτηση κατανομής θα έχει τιμή για την εγγραφή t:

$$\bar{G}_t^{gen}(x) = \begin{cases} \frac{1}{\prod_{i=1}^{d} L(QI[i])} & \text{if } x[i] \in QI[i] \forall i \in [1, d] \\ 0 & otherwise \end{cases}$$

Εν συνεχεία θα δούμε την τιμή της νέας συνάρτησης κατανομής η οποία προκύπτει από anatomized πίνακες. Έστω πάλι QI το QI-group το οποίο περιέχει την εγγραφή t. Έστω $u_1, ..., u_\lambda$ να είναι όλες οι τιμές της ιδιότητας A^s στο QI. Συμβολίζουμε με $c(u_h)(1 \leq h \leq \lambda)$ ως το πλήθος στο ST με βάση το u_h. Η νέα κατασκευασμένη συνάρτηση κατανομής είναι:

$$\bar{G}_t^{ana}(x) = \begin{cases} \frac{c(u_1)}{|QI|} & \text{if } x = (t[1], \ldots, t[d], u_1) \\ \ldots & \ldots \\ \frac{c(u_\lambda)}{|QI|} & \text{if } x = (t[1], \ldots, t[d], u_\lambda) \\ 0 & otherwise \end{cases}$$

Παρατηρείστε ότι ενώ στην περίπτωση του generalization είχαμε ένα συνεχή χώρο, ο οποίος είναι κάποιο ορθογώνιο, στην περίπτωση του anatomy έχουμε διακριτά σημεία (ή καλύτερα διακριτές περιοχές), που θυμίζουν καρφιά στο χώρο.

Υπολογίζουμε λοιπόν το μέσο τετραγωνικό σφάλμα για κάθε εγγραφή, το οποίο δίνεται από τον τύπο:

$$Err_t = \int_{x \in DS} \left(\bar{G}_t(x) - G_t(x)\right)^2 dx$$

Τέλος αυτός που μας ενδιαφέρει τελικά είναι το σφάλμα για ολόκληρο τον πίνακα.

Ορισμός 6. *RCE. Ορίζουμε ως σφάλμα επανακατασκευής : $RCE = \sum_{\forall t \in T} Err_t$.*

2.6.3 Ο αλγόριθμος

Στόχος μας λοιπόν είναι να κατασκευάσουμε ένα αλγόριθμο, ο οποίος θα ακολουθεί τις αρχές του anatomy και θα ελαχιστοποιεί το RCE. Αποδεικνύεται το παρακάτω θεώρημα:

Θεώρημα 1. *Το RCE είναι τουλάχιστον $n(1 - 1/l)$ για κάθε ζευγάρι QIT και ST, όπου n ο πληθάριθμος του T.*

Έτσι στόχος του αλγορίθμου θα είναι να μην ξεπεραστεί κατά πολύ αυτό το κάτω όριο. Δίνεται ο αλγόριθμος:

Algorithm 7 Anatomy

```
Algorithm Anatomize (T, l)
QIT = ∅; ST = ∅; gcnt = 0
hash the tuples in T by their As values (each bucket per As value)
/* Lines 3-8 are the group-creation step */
while there are at least l non-empty hash buckets do
   /* Lines 4-8 form a new QI-group */
   gcnt = gcnt + 1; QI_gnct = ∅ ;
   S = the set of l largest buckets
   for each bucket in S do
      remove an arbitrary tuple t from the bucket
      QI_gcnt = QI_gcnt ∪ t
      /* Lines 9-12 are the residue-assignment step */
   end for
end while
for each non-empty bucket do
   /* this bucket has only one tuple*/
   t = the only residue tuple of the bucket
   S' = the set of QI-groups that do not contain the A^s value t[d + 1]
   /* S' has at least one QI-group*/
   assign t to a random QI-group in S'
end for
/* Lines 13-18 populate QIT and ST *
for j = 1 to gcnt do
   for each tuple t ∈ QI_j do
      insert tuple (t[1], ..., t[d], j) into QIT
   end for
   for each distinct A^s value v in QI_j do
      c_j(v) = the number of tuples in QI_j with A_s value v
      insert record (j, v, cj(v)) into ST
   end for
end for
return QIT and ST
```

Ας δούμε σταδιακά τα βήματα του αλγορίθμου. Σε πρώτη φάση κατασκευάζουμε ένα hash table όπου κάθε τιμή της ιδιότητας A^s ορίζει ένα bucket. Σε δεύτερη φάση κατασκευάζουμε τα group. Η διαδικασία είναι απλή, όσο έχουμε τουλάχιστον l το πλήθος buckets τα οποία δεν είναι κενά, επιλέγουμε τα l μεγαλύτερα και από αυτά εξάγοντας μία εγγραφή κατασκευάζουμε ένα group, εν συνεχεία επαναλαμβάνουμε την διαδικασία κατασκευάζοντας νέα group μέχρι να έχουμε λιγότερα από l buckets μη κενά.

Όταν τελειώσει αυτό το στάδιο θα μας έχουν μείνει το πολύ $l - 1$ buckets μη κενά. Αποδεικνύεται ότι:

1. Κάθε ένα από αυτά τα buckets έχει ακριβώς μία εγγραφή.

2. Για κάθε ένα από αυτά τα buckets υπάρχει τουλάχιστον ένα group (το οποίο έχει φτιαχτεί στο προηγούμενο στάδιο) έτσι ώστε να μην περιέχει την sensitive τιμή η οποία αντιστοιχεί σε αυτό το bucket.

Άρα στο επόμενο στάδιο, για κάθε μη κενό bucket επιλέγουμε ένα τυχαίο group που δεν έχει αυτή την τιμή και αντιστοιχούμε αυτή την εγγραφή.

Στο τρίτο και τελευταίο στάδιο απλά κατασκευάζουμε το QIT και το ST.

Προσέξτε ότι ο αλγόριθμος είναι σωστός αν και μόνο αν ο πίνακας είναι ικανός να είναι l-diverse. Μπορούμε αυτό όμως να το ελέγξουμε αρκετά απλά. Για παράδειγμα αν στο τέλος του πρώτου σταδίου, υπάρχει bucket με παραπάνω από μία εγγραφή, τότε δεν μπορούμε να ικανοποιήσουμε το l-diverse. Επίσης αν στο στάδιο δύο δεν μπορούμε να ικανοποιήσουμε την

δεύτερη ιδιότητα (δηλαδή να βρούμε ένα group το οποίο να μην περιέχει αυτή την τιμή) πάλι ο πίνακας μας δεν μπορεί να ικανοποιήσει το l-diverse. Άρα ο αλγόριθμος αυτός ικανοποιεί και το correctness.

Απόδοση

Αποδεικνύονται οι παρακάτω δύο προτάσεις:

Θεώρημα 2. *Ο αλγόριθμος του anatomy απαιτεί $O(\lambda)$ μνήμη και $O(n/b)$ I/Os, όπου λ είναι το πλήθος των διαφορετικών τιμών του A^s στο T, n ο πληθάριθμος του T και b το μέγεθος της σελίδας του δίσκου.*

Θεώρημα 3. *Αν ο πληθάριθμος n του T είναι πολλαπλάσιο του l, τότε τα QIT και ST τα οποία παράγονται από τον αλγόριθμο ικανοποιούν το ελάχιστο όριο για το RCE, δηλαδή το RCE είναι $n(1-1/l)$. Αν ο πληθάριθμος n του T δεν είναι πολλαπλάσιο του l, τότε το RCE του νέου πίνακα είναι το πολύ ίσος με το ελάχιστο όριο αυξημένο κατά ένα παράγοντα $1+1/n$.*

Προσέξτε ότι αν ο πίνακας T είναι ιδιαίτερα μεγάλος τότε επί της ουσίας $RCE \approx n(1-1/l)+1$, δηλαδή ο παράγοντας αύξησης είναι περίπου ίσος με 1, και μάλιστα δεδομένου ότι το n είναι αρκετά μεγάλο, δεν επηρεάζει ιδιαίτερα.

2.6.4 Σχόλια

Εν συντομία λοιπόν, η μεθοδολογία του anatomy μας εξασφαλίζει το l-diversity και μάλιστα με αρκετά χαμηλή πολυπλοκότητα, τόσο χρονική όσο και χωρική. Επιπλέον διατηρεί καλύτερα την συσχέτιση από την γενίκευση. Παρ' όλα αυτά θα μπορούσε κανείς να παρατηρήσει δύο μειονεκτήματα:

Πρώτον η συσχέτιση δεν μετρήθηκε με απόλυτα ακριβή τρόπο. Στην παραπάνω ανάλυση θεωρήσαμε ότι απώλεια πληροφορίας μπορεί να μετρηθεί από την ικανότητα μας να προσεγγίσουμε την ακριβή θέση του σημείο στους χώρου. Στην πραγματικότητα η απώλεια πληροφορίας, πρέπει να μετρηθεί όχι ως προς την ικανότητα να βρούμε την ακριβή θέση του σημείου στον χώρο, αλλά την ικανότητα μας να ανακαλύψουμε ότι το σημείο ανήκει σε μία περιοχή. Ας θυμηθούμε ξανά το παράδειγμα με την ασθένεια και την ηλικία. Το πρόβλημα της γενίκευσης ήταν ότι δεν διατηρούσε την ακριβή ηλικία. Αποτελεί όμως αυτό όντως πρόβλημα; Για παράδειγμα ας σκεφτούμε κάποια εφαρμογή, όπου προσπαθούμε να κάνουμε ανάλυση και στο query υπάρχουν ερωτήσεις την μορφής: WHERE $a \leq age < a+10$. Δηλαδή προσπαθούμε να βγάλουμε συμπεράσματα για κάποιες ηλικιακές ομάδες, με range 10, θα μας πείραζε αν η γενίκευση οδηγούσε σε range απόστασης 5; Όχι, αντίθετα θα μας πείραζε ότι για κάθε ηλικιακή ομάδα να έχω κάποιες τιμές οι οποίες πιθανόν να είναι άσχετη με αυτή.

Παρόλα αυτά η αποσυσχέτιση του QI από την sensitive ιδιότητα, αποτελεί σημαντικό πρόβλημα και απώλεια πληροφορίας, ακόμα και αν θεωρήσουμε ότι η γενίκευση είναι πάντα κάτι αρνητικό. Φανταστείτε ότι επιθυμούμε να βρούμε τις ασθένειες όλων των ατόμων που μένουν στα Βόρεια Προάστια. Το γεγονός ότι σε κάθε εγγραφή αντιστοιχούμε και l οι οποίες δεν έχουν σχέση μπορεί να μας οδηγήσει σε μεγάλο σφάλμα. Ας γίνουμε όμως λίγο πιο συγκεκριμένοι, ας φανταστούμε ότι έχουμε τέσσερις πόλεις την Α,Β,Γ και Δ, οι δύο πρώτες είναι στα Βόρεια Προάστια της Αθήνας και άλλες δύο στα Νότια. Στην πόλη Α παρουσιάζεται η ασθένεια α, στην πόλη Β η β, στην πόλη Γ η γ και στην πόλη Δ η δ και καμία άλλη. Θεωρούμε ότι κάθε πόλη έχει το ίδιο πλήθος εγγραφών ακριβώς n και επιθυμούμε να πετύχουμε 2-diverse. Εάν εφαρμόσουμε τον προηγούμενο αλγόριθμο τότε είναι πολύ πιθανόν να πάρουμε την παρακάτω λύση:

- Κατασκευάζονται ακριβώς 2n groups, με δύο εγγραφές ακριβώς το καθένα.

- Κάθε group από τα πρώτα n έχει μία εγγραφή από την πόλη Α και μία από την πόλη Γ.
- Κάθε group από τα υπόλοιπα n έχει μία εγγραφή από την πόλη Β και μία από την πόλη Δ.

Έστω λοιπόν ότι τώρα επιθυμούμε να κάνουμε μία στατιστική μελέτη και προσπαθούμε να βρούμε κάποια στατιστικά στοιχεία για κάθε περιοχή. Με βάση τον παραπάνω δημοσιευμένο πίνακα, θα γνωρίζουμε ότι στα Βόρεια Προάστια και Νότια Προάστια εμφανίζονται οι ασθένειες α,β,γ,δ ισοπίθανα.

Αντίθετα αν είχαμε την παρακάτω λύση:

- Κατασκευάζουμε ακριβώς 2n groups, με δύο εγγραφές ακριβώς το καθένα.
- Κάθε group από τα πρώτα n έχει μία εγγραφή από την πόλη Α και μία από την πόλη Β.
- Κάθε group από τα υπόλοιπα n έχει μία εγγραφή από την πόλη Γ και μία από την πόλη Δ.

Θα μπορούσαμε με απόλυτη ακρίβεια να υπολογίσουμε τις ασθένειες για κάθε περιοχή.

Προσέξτε ότι αυτό το πρόβλημα είναι πιθανόν να εμφανιστεί και στην γενίκευση σε ακόμα μεγαλύτερο βαθμό, γιατί στην γενίκευση είναι πολύ πιθανόν να αναγκαστούμε να βάλουμε τις πόλεις Α, Β και Γ μαζί.

Ένας τρόπος να αποφύγουμε αυτό το πρόβλημα θα ήταν να επιλέξουμε να βάλουμε σε groups μόνο εγγραφές οι οποίες είναι local με βάση για παράδειγμα κάποια ιεραρχία γενίκευσης όπως είχαμε κάνει στο k-anonymity. Μία άλλη λύση θα ήταν να εκδίδουμε στατιστικά στοιχεία.

Τέλος, ο αλγόριθμος δεν μας εξασφαλίζει την ταυτότητα μίας εγγραφής. Αυτό αποτελεί επιλογή της υλοποίησης και όχι μειονέκτημα. Με αυτό το τρόπο η απώλεια της πληροφορίας είναι μικρότερη. Ο αλγόριθμος όμως μπορεί να επεκταθεί για να υποστηρίζει διασφάλιση της ταυτότητας μία εγγραφής (αυξάνοντας όμως σημαντικά την πολυπλοκότητα - το πρόβλημα είναι NP-hard).

2.7 m-invariance

Τα δύο προηγούμενα μοντέλα (του k-anonymity και του l-diversity), τα οποία παρουσιάστηκαν, είχανε ένα βασικό πρόβλημα. Δεν υπήρχε καμία πρόβλεψη για δυναμικά δεδομένα. Πιο συγκεκριμένα δεν είχε εξεταστεί πως επηρεάζει το πρόβλημα του privacy preservation η ύπαρξη εισαγωγών και διαγραφών στην βάση. Ας δούμε για παράδειγμα τα σχήματα 2.18 και 2.19. Ένα νοσοκομείο αποφασίζει να εκδίδει κατά χρονικά διαστήματα το ιστορικό των ασθενών του, προσπαθώντας όμως να διατηρήσει το privacy. Στο παράδειγμα μας έχουμε δύο διαδοχικές εκδόσεις. Το βασικό μας πρόβλημα είναι ότι οι εγγραφές των Alice, Andy, Helen, Ken και Paul διαγράφτηκαν μετά την έκδοση της πρώτης ανώνυμης όψης και προστέθηκαν πέντε νέες εγγραφές. Μπορεί να παρατηρήσει κανείς ότι ενώ και οι δύο ανώνυμες όψεις ικανοποιούν το 2-diversity, τα ευαίσθητα δεδομένα δεν είναι δυνατόν να διαφυλαχθούν Ας υποθέσουμε ότι ο αντίπαλος γνωρίζει ότι ο Bob υπάρχει στην βάση, τότε παίρνοντας τις δύο αυτές όψεις μπορεί να ανακαλύψει ότι στο QI-group της πρώτης όψης ο Bob είχε είτε dyspepsia είτε bronchitis ενώ στο QI-group της δεύτερης όψης είχε είτε dyspepsia είτε gastritis. Αυτό άμεσα σημαίνει ότι ο Bob πάσχει από dyspepsia. Μπορούμε να προσέξουμε μάλιστα ότι εξαιτίας της ανακάλυψης της ασθένειας του Bob τότε μπορούμε να ανακαλύψουμε και την ασθένεια των David και Alice.

Patient Data: 1st Instance			
Name	Age	Zipcode	Disease
Bob	21	12000	dyspepsia
Alice	22	14000	bronchitis
Andy	24	18000	flu
David	23	25000	gastritis
Gary	41	20000	flu
Helen	36	27000	gastritis
Jane	37	33000	dyspepsia
Ken	40	35000	flu
Linda	43	26000	gastritis
Paul	52	33000	dyspepsia
Steve	56	34000	gastritis

(α′) Microdata $T(1)$

Anonymized Data: 1st Instance			
Group-Id	Age	Zipcode	Disease
1	21	12000	dyspepsia
1	22	14000	bronchitis
2	24	18000	flu
2	23	25000	gastritis
3	41	20000	flu
3	36	27000	gastritis
4	37	33000	dyspepsia
4	40	35000	flu
4	43	26000	gastritis
5	52	33000	dyspepsia
5	56	34000	gastritis

(β′) Anatomized $T^a(1)$

Σχήμα 2.18: Τα δεδομένα και η anonymized view στην πρώτη έκδοση.

Patient Data: 2st Instance			
Name	Age	Zipcode	Disease
Bob	21	12000	dyspepsia
David	23	25000	gastritis
Emily	25	21000	gastritis
Jane	37	33000	dyspepsia
Linda	43	26000	gastritis
Gary	41	20000	flu
Mary	46	30000	gastritis
Ray	54	31000	dyspepsia
Steve	56	34000	gastritis
Tom	60	44000	gastritis
Vince	65	36000	flu

(α′) Microdata $T(2)$

Anonymized Data: 2st Instance			
Group-Id	Age	Zipcode	Disease
1	21	12000	dyspepsia
1	23	25000	gastritis
2	25	21000	gastritis
2	37	33000	dyspepsia
2	43	26000	gastritis
3	41	20000	flu
3	46	30000	gastritis
4	54	31000	dyspepsia
4	56	34000	gastritis
5	60	44000	gastritis
5	65	36000	flu

(β′) Anatomized $T^a(2)$

Σχήμα 2.19: Τα δεδομένα και η anonymized view στην δεύτερη έκδοση.

Συνολικά θα μπορούσε κάποιος να παρατηρήσει ότι όταν εκδίδουμε ένα σύνολο από ανώνυμες όψεις ενός πίνακα, στον οποίο μπορεί να υπήρξαν διαγραφές ή εισαγωγές, τότε πρέπει να λάβουμε υπόψη μας τι συνέβει σε όλες τις όψεις. Η πιο απλή λύση, την οποία θα μπορούσε να σκεφτεί κανείς είναι να μην υποστηρίζουμε διαγραφές. Με άλλα λόγια, ακόμα και αν μία εγγραφή διαγραφεί εμείς να συνεχίζουμε να την εκδίδουμε. Παρ΄ όλα αυτά το βασικό μας πρόβλημα είναι η μείωση του utility αφού οι διαγραφείσες εγγραφές οι οποίες εκδίδονται είναι σκουπίδια και εμποδίζουν τον ερευνητή να χρησιμοποιήσει με αποδοτικό τρόπο τα εκδιδόμενα δεδομένα.

2.7.1 Βασικοί Ορισμοί

Το πρόβλημα μας λοιπόν είναι να ορίσουμε το πρόβλημα για δυναμικά δεδομένα. Θα συμβολίζουμε λοιπόν με $T(j)$ την j-ιοστό snapshot της βάσης μας. Άρα μπορούμε να επεκτείνουμε τους ήδη υπάρχοντες ορισμούς.

Ορισμός 7. *Partition/QI-group. Ένα partition αποτελείται από διάφορα υποσύνολα του $T(j)$ έτσι ώστε κάθε εγγραφή στο $T(j)$ να ανήκει ακριβώς σε ένα υποσύνολο. Αναφερόμαστε σε αυτά τα υποσύνολα σαν QI-groups και τα συμβολίζουμε $QI(j)_1, \ldots, QI(j)_m$. Με άλλα*

λόγια ένα partition του $T(j)$ είναι ένα σύνολο ξένων συνόλων των οποίων η ένωση μας δίνει το $T(j)$.

υπάρχουν δύο τρόποι, όπως θα δούμε, για να εκδώσουμε την βάση. Ο πρώτος είναι αυτός της γενίκευσης και ο δεύτερος αυτός του anatomy:

Ορισμός 1. *Counterfeited Generalization. Ο νέος εκδιδόμενος πίνακας $T^*(j)$ προκύπτει από ένα partition του $T(j)$ με τα εξής χαρακτηριστικά:*

- *Έχει τις εξής attributes:*
  - *την A^g, η οποία ονομάζεται Group-ID.*
  - *όλες τις ιδιότητες του $T(j)$ με εξαίρεση την A^{id}*
- *Κάθε εγγραφή t κάθε QI-group του partition του $T(j)$ αντιστοιχίζεται σε μία γενικευμένη εγγραφή $t^* \in T^*(j)$ έτσι ώστε*
  - *$t^*[A^s] = t[A^s]$ (δηλαδή η sensitive ιδιότητα να διατηρεί την τιμή της),*
  - *$t^*[A^g]$ να έχει την τιμή του group-id στο οποίο ανήκει η εγγραφή (hosting group) και*
  - *$t^*[A^{qi}]$ να είναι ένα interval το οποίο καλύπτει το $t[A^{qi}]$ (αυτό έχει προκύψει από κάποια μέθοδο γενίκευσης).*
- *Είναι πιθανόν ο πίνακας $T^*(j)$ να περιέχει εγγραφές οι οποίες δεν υπάρχουν στον $T(j)$, αυτές οι εγγραφές ονομάζονται counterfeit tuples και βέβαια αντιστοιχίζονται σε κάποιο QI-group.*
- *Κάθε εγγραφή που ανήκει στο ίδιο QI-group του $T^*(j)$ έχει την ίδια τιμή για την QI ιδιότητα και την ίδια τιμή για το A^g αφού το τελευταίο δηλώνει σε ποιο group ανήκει μία εγγραφή (αυτή η ιδιότητα είναι ένα id του group).*

Ορισμός 2. *Counterfeited Anatomy.Δοθέντος ενός m-diverse partition με p QI-groups, ο anatomy παράγει ένα quasi-identifier table (QIT) και ένα sensitive value (ST) με βάση τις παρακάτω ιδιότητες:*

- *Το QIT έχει το σχήμα $(A_1, \ldots, A_d, Group-ID)$.*
- *Για κάθε QI-group $QI(j)_i (1 \leq i \leq m)$ και για κάθε εγγραφή $t \in QI(j)_i$, το QIT έχει μία εγγραφή της μορφής: $(t[1], t[2], ..., t[d], i)$*
- *Το QIT είναι πιθανόν να έχεις και άλλες εγγραφές της μορφής: $(t[1], t[2], ..., t[d], i)$, οι οποίες είναι counterfeits.*
- *Το ST έχει σχήμα $(Group-ID, A^s, Count)$.*
- *Για κάθε QI-group $QI(j)_i (1 \leq i \leq m)$ και για κάθε τιμή v της sensitive ιδιότητας A^s στο $QI(j)_i$, το ST έχει μία εγγραφή της μορφής: $(i, v, c_i(v))$, όπου $c_i(u)$ ο αριθμός των εγγραφών $t \in QI(j)_i$ έτσι ώστε $t[d+1] = u$.*
- *Εκτός από τις προηγούμενες εγγραφές δεν υπάρχουν άλλες στους δύο πίνακες.*

Μαζί με τις εκδιδόμενες ανώνυμες όψεις μπορούμε παράλληλα να εκδίδουμε ένα βοηθητικό πίνακα για τον ερευνητή, ο οποίος να τον ενημερώνει για το counterfeits τα οποία υπάρχουν σε κάθε QI-group.

Ορισμός 3. *Auxiliary Relation. Ο auxiliary relation $R(j)$, ο οποίος συνοδεύει την ανώνυμη όψη $T^*(j)$ έχει δύο στήλες Group-ID και Count. Για κάθε QI-group στο $T^*(j)$ το οποίο περιέχει τουλάχιστον ένα counterfeit υπάρχει μία γραμμή $< g, c >$ στο $R(j)$ έτσι ώστε g να είναι το ID του QI-group και c ο αριθμός των counterfeit εγγραφών σε αυτό.*

Counterfeited Anonymized Data: 2st Instance				
Name	Group-Id	Age	Zipcode	Disease
Bob	1	21	12000	dyspepsia
c_1	1	22	14000	bronchitis
David	2	23	25000	gastritis
Emily	2	25	21000	gastritis
Jane	3	37	33000	dyspepsia
c_2	3	40	35000	flu
Linda	3	43	26000	gastritis
Gary	4	41	20000	flu
Mary	4	46	30000	gastritis
Ray	5	54	31000	dyspepsia
Steve	5	56	34000	gastritis
Tom	6	60	44000	gastritis
Vince	6	65	36000	flu

(α′) Anatomized $T^a(2)$

Auxiliary Relation	
Group-Id	count
1	1
3	1

(β′) Published counterfeit statistics

Σχήμα 2.20: Τα δεδομενα και η anonymized view στην δεύτερη έκδοση με counterfeits.

Μπορούμε να δούμε το παράδειγμα του σχήματος 2.20. Μπορεί κανείς να παρατηρήσει ότι η νέα εκδιδόμενη όψη για το 2 instance της βάσης δεν μας παραβιάζει το privacy. Ο λόγος είναι ότι κρατήσαμε κάποιες από τις διαγραφέντες εγγραφές στην δεύτερη εκδιδόμενη όψη. Μπορεί μάλιστα κανείς να παρατηρήσει τα παρακάτω:

- Μία εγγραφή μπορεί να διαγραφεί από την βάση, μόνο αν βρεθεί μία εισαγωγή με την ίδια τιμή στην sensitive attribute με αυτή.
- Κάθε εγγραφή, κάθε όλη την διάρκεια εμφάνισης της στις εκδιδόμενες ανώνυμες όψεις ανήκει σε ένα QI-group το οποίο έχει το ίδιο σύνολο τιμών για την ευαίσθητη ιδιότητα.

2.7.2 Ο ορισμός

Με βάση τα προηγούμενα συμπεράσματα μπορούμε να παρατηρήσουμε τι χρειάζεται για να προστατέψουμε το privacy. Πρώτον θα υποθέσουμε ότι κάθε εγγραφή έχει μία διάρκεια ζωής (lifespan) την οποία θα την συμβολίζουμε με $[x, y]$. Γενικότερα θα υποθέτουμε ότι αντίπαλος έχει πλήρη γνώση, όχι μόνο του Quasi-Identifier μίας εγγραφής, αλλά και του lifespan. Αυτό σημαίνει ότι ο αντίπαλος γνωρίζει πότε μία εγγραφή διαγράφεται από την βάση ή πότε εισάγεται. Επίσης θα υποθέσουμε ότι ο αντίπαλος έχει στην διάθεση του όλες τις ανώνυμες όψεις τις οποίες είχαμε εκδώσει κατά το παρελθόν. Στόχος λοιπόν του m-invariance είναι επανέκδοση μίας νέας ανώνυμης όψης, έτσι ώστε ο αντίπαλος με βάση την γνώση την οποία ήδη έχει να μην μπορεί να εξάγει με πιθανότητα μεγαλύτερη του $1/m$ την ασθένεια μίας εγγραφής.

Είμαστε λοιπόν τώρα σε θέση να δώσουμε τον ακριβή ορισμό του m-invariance. Αρχικά όμως θα δώσουμε δύο βοηθητικούς ορισμούς.

Ορισμός 4. *QI-Group Signature. Έστω QI^* ένα QI-group στην $T^*(j)$ για κάθε $j \in [1, n]$. Θα ονομάζουμε signature του QI^* το σύνολο των μοναδικών ευαίσθητων τιμών στο QI^*.*

Ορισμός 5. *Tuple Signature. Έστω μία εγγραφή t η οποία ανήκει σε ένα QI-group. Η signature του QI-group θα λέγεται και signature της εγγραφής t.*

Ορισμός 6. *m-Unique. Ένας πίνακας T θα είναι m-unique αν σε κάθε QI-group στον πίνακα υπάρχουν τουλάχιστον m εγγραφές και όλες οι εγγραφές σε αυτό έχουν διαφορετικές τιμές στην ευαίσθητη ιδιότητα.*

Είμαστε λοιπόν τώρα σε θέση να δώσουμε τον ορισμό του m-invariance

Ορισμός 7. *m-invariance. Έστω μία αλληλουχία anonymous όψεων $T^*(1), ..., T^*(n)$ μίας βάσης T. Θα λέμε ότι ικανοποιούν το m-invariance αν*

- *Κάθε όψη είναι m-equal και*
- *αν κάθε εγγραφή t η οποία ανήκει στο T και υπάρχει σε κάποιες από αυτές τις όψεις ως generalized ή anatomized εγγραφή t^*, έχει πάντα την ίδια signature.*
- *Είναι πιθανόν σε αυτές τις όψεις να υπάρχουν counterfeits.*

Μπορεί κανείς να παρατηρήσει από τον ορισμό ότι κάθε εγγραφή ανήκει πάντα σε ένα QI-group με την ίδια signature. Αυτό το QI-group δεν είναι ανάγκη πάντα να είναι το ίδιο, αρκεί η υπογραφή να είναι πάντα η ίδια. Προσέξτε επίσης ότι δεν ικανοποιούμε τον αυστηρό ορισμό του l-diversity αλλά μία υποπερίπτωση αυτού, το m-unique. Αν δούμε λοιπόν ξανά τα αρχικά μας σχήματα 2.18.και 2.20.μπορούμε να παρατηρήσουμε ότι με την χρήση των counterfeits μπορούμε να ικανοποιήσουμε το 2-invariant. Πιο συγκεκριμένα αν δούμε ξανά τους δύο ανώνυμους πίνακες, παρατηρούμε το εξής, ότι ισχύει ένα από τα παρακάτω:

- Όσοι ανήκουν στην πρώτη όψη και στην δεύτερη όψη, τότε ανήκουν σε QI-group το οποίο παίρνει τις ίδιες τιμές στην ευαίσθητη ιδιότητα, δηλαδή οι εγγραφές έχουνε την ίδια υπογραφή και στις δύο όψεις.
- Οι υπόλοιπες εγγραφές δεν μας απασχολούν

Παράλληλα μπορούμε να κάνουμε μερικές ακόμα παρατηρήσεις οι οποίες δεν είναι εμφανείς:

- Μία εγγραφή η οποία έχει lifespan $[x, y]$ τότε στις ανώνυμες όψεις θα έχει lifespan $[x, y']$ με $y' \geq y$. Δηλαδή μία εγγραφή δεν θα διαγραφεί νωρίτερα στις ανώνυμες όψεις.
- Αν μία εγγραφή εμφανίζεται στις όψεις $T^*(k)$ και $T^*(l)$ με $1 \leq k \leq l \leq n$ τότε θα εμφανίζεται σε οποιαδήποτε άλλη όψη $T^*(j)$ με $k \leq j \leq l$.

Από τα παραπάνω μπορούμε να παρατηρήσουμε τα εξής, ότι για να γνωρίζουμε το signature το οποίο είχε μία εγγραφή στις προηγούμενες όψεις, μας αρκεί μόνο μία από τις προηγούμενες όψεις στις οποίες άνηκε αυτή η εγγραφή. Και για να είμαστε περισσότερο ακριβείς, για να γνωρίζουμε το signature όλων των εγγραφών σε προηγούμενες όψεις μας αρκεί μόνο η αμέσως προηγούμενη όψη, αφού μία εγγραφή για να υπάρχει στην επόμενη όψη ή θα υπάρχει στην αμέσως προηγούμενη ή σε καμία από τις προηγούμενες. Επίσης μπορούμε να παρατηρήσουμε ότι ακόμα και αν αντίπαλος γνωρίζει το lifespan μίας εγγραφής πάλι δεν γίνεται να βρει με ασφάλεια την ευαίσθητη τιμή της εγγραφής. Ο λόγος είναι ότι η εγγραφή θα έχει πάντα την ίδια υπογραφή, η οποία θα περιέχει τουλάχιστον m τιμές.

Δίνεται λοιπόν η παρακάτω πρόταση:

Πρόταση 1. *Δοθέντος ένα σύνολο διαδοχικών χρονικών πινάκων $T(1), ..., T(n)$ και των ανώνυμων όψεων αυτών $T^*(1), ..., T^*(n-1)$ οι οποίες είναι m-invariant, το σύνολο των όψεων $T^*(1), ..., T^*(n)$ είναι m-invariant αν και μόνο αν ισχύουν οι δύο παρακάτω προτάσεις:*

- *Το $T^*(n)$ είναι m-unique.*
- *για κάθε εγγραφή $t \in T(n-1) \cap T(n)$, να έχει την ίδια signature και στις δύο όψεις.*

Τέλος αξίζει να αναφέρουμε τον παρακάτω ορισμό και την παρακάτω πρόταση:

Ορισμός 8. *m-Eligible. Δοθέντος ένα σύνολο διαδοχικών χρονικών πινάκων $T(1), ..., T(n)$ και των ανώνυμων όψεων αυτών $T^*(1), ..., T^*(n-1)$, θα λέμε ότι ο $T(n)$ είναι m-eligible αν μπορούμε να κατασκευάσουμε μία ανώνυμη όψη $T^*(n)$ έτσι ώστε το σύνολο των όψεων $T^*(1), ..., T^*(n)$ να είναι m-invariant.*

Πρόταση 2. *Δοθέντος ένα σύνολο διαδοχικών χρονικών πινάκων $T(1), ..., T(n)$ και των ανώνυμων όψεων αυτών $T^*(1), ..., T^*(n-1)$ οι οποίες είναι m-invariant, ο $T(n)$ είναι m-eligible αν το πολύ $1/m$ εγγραφές στον $T(n) - T(n-1)$ έχουν την ίδια τιμή στην ευαίσθητη ιδιότητα.*

Προσέξτε ότι η παραπάνω πρόταση είναι επί της ουσίας η γνωστή μας ανίσωση από το l-diversity με την διαφορά ότι τώρα λαμβάνουμε υπόψη μας και την ύπαρξη προηγούμενων όψεων. Επί της ουσίας $T(n) - T(n-1)$ είναι μόνο οι νέες εγγραφές στην βάση. Η παραπάνω πρόταση λέει ότι για να μπορούμε να εκδώσουμε μία m-invariant όψη αρκεί για τις νέες μόνο εγγραφές να μπορούμε να εκδώσουμε μία m-diverse όψη.

2.7.3 Ο αλγόριθμος

Ο αλγόριθμος έχεις δύο στόχους:

- Να μειωθεί το πλήθος των counterfeit εγγραφών, οι οποίες αποτελούνε σκουπίδια και δεν θέλουμε να υπάρχουν στην βάση.
- Να μειωθεί η περίμετρος κάθε QI-group, πιο συγκεκριμένα στόχος είναι όλες οι εγγραφές οι οποίες υπάρχουν σε ένα QI-group να είναι κοντά μεταξύ τους με βάση τις τιμές του Quasi-Identifier. Αυτό στην περίπτωση της γενίκευσης μπορούμε να το σκεφτούμε και ως ελαχιστοποίηση του βαθμού γενίκευσης του Quasi-Identifier.

Ο αλγόριθμος για να πετύχει το m-invariance χρησιμοποιεί μόνο την $T(n)$, $T(n-1)$ και $T^*(n-1)$, αφού με βάση μόνο αυτά μπορούμε να βρούμε την υπογραφή κάθε εγγραφής και παράλληλα να εκδώσουμε την νέα ανώνυμη όψη ώστε αυτή με βάση της προηγούμενες να ικανοποιεί το m-invariance. Ορίζουμε ως $S_\cap = T(n) \cap T(n-1)$ και ως $S_- = T(n) - T(n-1)$. Ο αλγόριθμος για να πετύχει το m-invariance αρκεί να ικανοποιεί τις δύο παρακάτω ιδιότητες:

- Κάθε εγγραφή $t \in S_\cap$ έχει την ίδια signature στις όψεις $T^*(n-1)$ και $T^*(n)$.
- Κάθε εγγραφή $t \in S_-$ ανήκει στην $T^*(n)$ σε ένα QI-group με τουλάχιστον m εγγραφές, με διαφορετικές ευαίσθητες τιμές (m-diversity).

Ο αλγόριθμος αποτελείται από τέσσερις φάσεις: division, balancing, assignment και spit. Παράλληλα με την παρουσίαση αυτών των φάσεων θα τρέχουμε και ένα παράδειγμα για $m = 2$ και $n = 2$. Ο $T(1)$ και ο $T^*(1)$ υπάρχουν στο σχήμα 2.18, ο $T(2)$ στο σχήμα 2.19 και το αποτέλεσμα $T^*(2)$ στο σχήμα 2.20.

Division Στην φάση αυτή κατασκευάζουμε ένα σύνολο από buckets με βάση τις εγγραφές του $S_\cap$ έτσι ώστε σε κάθε bucket κάθε εγγραφή να έχει την ίδια υπογραφή.

Ας δούμε το παράδειγμα μας. Με βάση τα σχήματα 2.19 και 2.20 ισχύει:

$$S_\cap = \{Bob, David, Linda, Jane, Gary, Steve\}$$

Gary	David
flu	gast.

	Steve
dysp.	gast.

Bob	
dysp.	bron.

Jane		Linda
dysp.	flu	gast

(α′) Buckets Contents after Division Phase

Gary	David
flu	gast.

Ray	Steve
dysp.	gast.

Bob	c_1
dysp.	bron.

Jane	c_2	Linda
dysp.	flu	gast

(β′) Buckets Contents after Balancing Phase

Vince Emily Gary	Tom Mary David
flu	gast.

Ray	Steve
dysp.	gast.

Bob	c_1
dysp.	bron.

Jane	c_2	Linda
dysp.	flu	gast

(γ′) Buckets Contents after Assignment Phase

Σχήμα 2.21: Ο αλγόριθμος σε παράδειγμα.

Το σχήμα 2.21α′ δείχνει ποια buckets κατασκευάζονται μετά από αυτό το στάδιο. Ο Bob για παράδειγμα, ο οποίος έχει signature $\{dyspepsia, bronchitis\}$ ανήκει στο BUC_3. Αντίθετα ο Gary και ο David που έχουνε την ίδια υπογραφή $\{flu, gastritis\}$ ανήκουν στο ίδιο bucket BUC_1.

Balancing Σε αυτό το στάδιο δεν θα ασχοληθούμε με τις signatures των εγγραφών, αλλά με τις τιμές στην sensitive attribute. Θα λέμε ότι ένα bucket θα είναι balanced, αν σε κάθε bucket μεγέθους $k = a * b$, του οποίου η signature περιέχει b τιμές $s_1, ..., s_b$ της sensitive attribute, υπάρχουν ακριβώς a εγγραφές με την ίδια τιμή s_i για κάθε i. Επί της ουσίας θέλουμε το bucket να μπορεί να ικανοποιήσει το m-unique.

Στόχος λοιπόν αυτού του σταδίου είναι όλα τα buckets στο τέλος του να είναι balanced. Για να το πετύχουμε αυτό διατρέχουμε κάθε bucket στην σειρά. Ένα κάποιο bucket δεν είναι balanced τότε αφαιρούμε μία εγγραφή από το S_- με την προϋπόθεση αυτό να παραμένει m-eligible (αν ήταν).

Ας δούμε για παράδειγμα ξανά το σχήμα 2.21α′. Το BUC_2 είναι unbalanced, αφού υπάρχει εγγραφή με gastritis αλλά όχι με dyspepsia. Επίσης γνωρίζουμε ότι ισχύει:

$$S_- = \{Emily, Mary, Ray, Tom, Vince\}$$

Μπορούμε να αφαιρέσουμε από το S_- τον Ray και να τον βάλουμε στο BUC_2, αφού το S_- θα παραμείνει μετά $2 - eligible$. Το αποτέλεσμα δίνεται στο σχήμα 2.21β′.

Όπως προαναφέρθηκε αφαιρούμε μία εγγραφή μόνο αν το S_- παραμένει m-eligible. Τι θα γίνει όμως αν δεν μπορούμε να αφαιρέσουμε κάποια άλλη εγγραφή και τα buckets μας παραμένουν unbalanced; Τότε εισάγουμε counterfeits. Το ίδιο κάνουμε και σε περίπτωση που η τιμή της ευαίσθητης ιδιότητας που αναζητάμε δεν υπάρχει στο S_-.

Συνεχίζοντας λοιπόν το παράδειγμά μας, στο σχήμα 2.21α′ τόσο το bucket BUC_3 και το BUC_4 είναι unbalanced. Δεν μπορούμε να βρούμε εγγραφές στο S_- για να τα γεμίσουμε και τελικά επιλέγουμε την λύση των counterfeits. Στο σχήμα 2.21β′ δίνονται τα buckets μετά το πέρας αυτού του σταδίου.

Assignment Στο στάδιο αυτό, για όλες τις εγγραφές οι οποίες έχουνε παραμείνει στο S_- κατασκευάζουμε buckets, ικανοποιώντας δύο κριτήρια. Πρώτον κάθε εγγραφή πρέπει να εισαχθεί σε ένα bucket του οποίου η υπογραφή περιέχει την τιμή της ευαίσθητης ιδιότητας της εγγραφής και δεύτερον στο τέλος του σταδίου όλα τα buckets να είναι balanced. Τα buckets

μπορεί να είναι είτε αυτά του προηγούμενου σταδίου, είτε καινούργια. Το στάδιο αυτό δίνει πάντα λύση, αρκεί το S_- να είναι m-eligible.

Σε αυτό το στάδιο θα εκτελεστούν ένα πλήθος από επαναλήψεις. Σε κάθε επανάληψη αφαιρούμε ένα σύνολο από εγγραφές $S_{rmv} = a*b$ από το S_- και με βάση αυτές κατασκευάζουμε ένα νέο bucket. Η αφαίρεση των εγγραφών λοιπόν πρέπει να γίνει με τα εξής κριτήρια:

- Η signature του νέου bucket παίρνει $b \geq m$ τιμές.
- Για κάθε τιμή της signature, υπάρχουν ακριβώς a εγγραφές με αυτή την τιμή στην sensitive attribute στο bucket.
- Το S_- παραμένει m-eligible.

Έχουμε δύο βασικά προβλήματα. Το πρώτο είναι η επιλογή των κατάλληλων τιμών για τα a και b και το δεύτερο μετά την αφαίρεση των εγγραφών το S_- να παραμένει m-eligible. Ο βασικός μας στόχος είναι να μειώσουμε το b στο ελάχιστο, γιατί μικρότερα QI-groups μας δίνουνε καλύτερο utility και να μεγιστοποιήσουμε το b ώστε να μειώσουμε το πλήθος των επαναλήψεων.

Σε κάθε επανάληψη θα έχουμε την παρακάτω διαδικασία:

Έστω λοιπόν ότι η ευαίσθητη ιδιότητα μας παίρνει τις μοναδικές τιμές $v_1, ..., v_\lambda$ στο S_-, οι οποίες θεωρούμε ότι χωρίς βλάβη της γενικότητας είναι ταξινομημένες ως προς φθίνουσα σειρά με βάση την συχνότητα εμφάνισης..Ο πληθάριθμος αυτών των τιμών είναι αντίστοιχα $n_1, ..., n_\lambda$. Θα συμβολίζουμε ως $\gamma = \sum_{i=1}^{\lambda} n_i$.

Επιλέγουμε λοιπόν ένα b (το πως θα το δείξουμε παρακάτω). Επιλέγουμε τις b ποιο συχνά εμφανιζόμενες τιμές $v_1, ...v_b$ και ορίζουμε λοιπόν με αυτό το τρόπο την signature του νέου μας bucket. Εν συνεχεία για κάθε $i \in [1, b]$ επιλέγουμε τυχαία a εγγραφές από το S_- οι οποίες έχουν τη τιμή v_i. Αφού υπάρχουν το πολύ n_b εγγραφές με την τιμή v_b, τότε :

$$a \leq n_b$$

Αφού αφαιρέσουμε λοιπόν τις εγγραφές από το S_-, το S_- θα έχει πληθάριθμο $\gamma - a * b$. Στο νέο S_- η πιο συχνά εμφανιζόμενη τιμή θα είναι είτε η v_1 είτε η v_{b+1}. Άρα για να είναι το νέο S_- m-eligible αρκεί να ισχύει:

$$n_1 - a \leq (\gamma - a * b)/m$$
$$n_{b+1} - a \leq (\gamma - a * b)/m$$

Το a λοιπόν επιλέγεται ως ο μέγιστο ακέραιος που ικανοποιεί τις δύο παραπάνω ανισότητες.

Το πρόβλημα μας λοιπόν τώρα είναι πως θα επιλέξουμε το b. Ξεκινάμε με αρχική τιμή για το $b = m$. Και αναζητάμε a το οποίο να ικανοποιεί τις παραπάνω ανισότητες. Αν βρούμε κάποιο a, τότε κρατάμε αυτό το ζευγάρι, αλλιώς αυξάνουμε το b κατά 1 και συνεχίζουμε την διαδικασία.

Στο σχήμα 2.21γ′ δίνεται η τελική μορφή του παραδείγματος μας μετά την εκτέλεση αυτού του σταδίου. Να σημειωθεί ότι αν το S_- είναι m-eligible ο αλγόριθμος δίνει πάντα λύση.

Split Τέλος πρέπει να κατασκευάσουμε QI-groups, έτσι σπάμε τα buckets. Πιο συγκεκριμένα ένα bucket έχει μία signature η οποία παίρνει $s \geq m$ τιμές στην ευαίσθητη ιδιότητα: $v_1, ..., v_s$. Κατασκευάζουμε τα groups $L_1, ..., L_s$ έτσι ώστε στο group L_j οι εγγραφές να έχουν τιμή στην sensitive attribute v_j. Είναι προφανές ότι κάθε group έχει μέγεθος ακριβώς $|BUC|/s$.

Ταξινομούμε λοιπόν τις εγγραφές σε κάθε group ως προς μία από τις ιδιότητες του quasi-identifier (θα δούμε παρακάτω ποια είναι αυτή η ιδιότητα).Μία πιθανή λύση είναι να πάρουμε μόνο την πρώτη εγγραφή από κάθε group L_j και να κατασκευάσουμε το νέο bucket BUC_1

και με όλες τις εγγραφές που έχουνε απομείνει να κατασκευάσουμε το bucket BUC_2. Μία άλλη πιθανή λύση είναι να πάρουμε τις δύο πρώτες εγγραφές, ενώ μία άλλη τις πρώτες τρεις. Γενικότερα έχουμε $|BUC|/s-1$ πιθανές λύσεις. Για την ακρίβεια στην i-ιοστή λύση, από κάθε group αφαιρούμε τις πρώτες i εγγραφές και τις βάζουμε στο BUC_1 οι υπόλοιπες μπαίνουν στο BUC_2.

Το πρόβλημα μας είναι με βάση ποια ιδιότητα του quasi-identifier θα ταξινομήσουμε τις εγγραφές. Μπορούμε να δοκιμάσουμε ως προς όλες. Με άλλα λόγια αν έχουμε d ιδιότητες για το quasi-identifier τότε θα έχουμε d επιλογές. Άρα για να κατασκευάσουμε τα buckets BUC_1 και BUC_2 έχουμε $d(|BUC|/s-1)$ επιλογές. Θα επιλέξουμε εκείνη που μας δίνει το ελάχιστο άθροισμα των περιμέτρων των δύο bucket. Πιο συγκεκριμένα προσπαθούμε να ελαχιστοποιήσουμε το άθροισμα:

$$|BUC_1|\sum_{i=1}^{d} l_i + |BUC_2|\sum_{i=1}^{d} l_i$$

όπου l_i το ελάχιστο interval το οποίο καλύπτει όλες τις εγγραφές στο συγκεκριμένο bucket για την αντίστοιχη ιδιότητα. Για να το πετύχουμε αυτό έχουμε πρώτα κανονικοποιήσει τις τιμές μας στο $[0, 1]$.

2.7.4 Σχόλια

Το m-invariance λοιπόν είναι μία μεθοδολογία η οποία επιλύει το πρόβλημα των δυναμικών δεδομένων. Μέσω της τεχνικής των counterfeits μπορεί να λάβει υπόψη του και τις εισαγωγές και τις διαγραφές.

Patient Data: 1st Instance			
Name	Age	Zipcode	Disease
Bob	21	12000	dyspepsia
Alice	22	14000	flu
Andy	24	18000	dyspepsia
David	23	25000	flu
Gary	41	20000	flu
Helen	36	27000	dyspepsia

(α′) Microdata $T(1)$

Anonymized Data: 1st Instance			
Group-Id	Age	Zipcode	Disease
1	21	12000	dyspepsia
1	22	14000	flu
2	24	18000	dyspepsia
2	23	25000	flu
3	41	20000	flu
3	36	27000	dyspepsia

(β′) Anatomized $T^a(1)$

Σχήμα 2.22: Τα δεδομένα και η anonymized view στην πρώτη έκδοση.

Patient Data: 2st Instance			
Name	Age	Zipcode	Disease
Bob	21	12000	dyspepsia
John	25	15000	flu
Andy	24	18000	dyspepsia
Sam	30	28000	flu
Gary	41	20000	flu
George	37	27000	dyspepsia

(α′) Microdata $T(2)$

Anonymized Data: 2st Instance			
Group-Id	Age	Zipcode	Disease
1	21	12000	dyspepsia
1	25	15000	flu
2	24	18000	dyspepsia
2	30	28000	flu
3	41	20000	flu
3	37	27000	dyspepsia

(β′) Anatomized $T^a(2)$

Σχήμα 2.23: Τα δεδομένα και η anonymized view στην δεύτερη έκδοση.

Ο αλγόριθμος όμως ο οποίος προτάθηκε έχει ένα βασικό πρόβλημα. Ενώ είναι πιθανό να υπάρχει κάποια λύση μπορεί να μην την βρει. Αυτό το πρόβλημα μπορούμε να το παρατηρήσουμε στο στάδιο του balancing. Ο αλγόριθμος απαιτεί η αφαίρεση μίας εγγραφής από το

S_- να διατηρεί πάντα το S_- m-eligible. Αυτό όμως δεν είναι πάντα απαραίτητο. Η αιτία είναι απλή. Μπορεί η αφαίρεση μίας εγγραφής (οποιασδήποτε και να είναι) να οδηγεί σε μη m-eligible λύση άλλη η αφαίρεση περισσότερων να οδηγεί. Ας δούμε για παράδειγμα τα σχήματα 2.22 και 2.23 Ενώ υπάρχει λύση ο αλγόριθμος δεν μπορεί να μας δώσει κάποια λύση. Για την ακρίβεια ο μόνος τρόπος να μας δώσει λύση είναι να αφαιρεί πάντα δύο εγγραφές από το S_-. Πιο συγκεκριμένα μπορεί κανείς να παρατηρήσει κανείς ότι στην αρχή ισχύει:

$$S_- = \{Bob, Andy, Gary\}$$

Αν αφαιρέσουμε τον *Bob* στο πρώτο iteration τότε καταλήγουμε σε ένα m-eligible S_-. Εν συνεχεία όμως ο μόνος τρόπος να αφαιρέσουμε μία εγγραφή είναι να αφαιρέσουμε δύο ταυτόχρονα. Αυτό όμως ο αλγόριθμος μας δεν το λαμβάνει υπόψη. Το πρόβλημα μας είναι πιο σύνθετο και στο επόμενο κεφάλαιο θα παρουσιαστεί ένας αλγόριθμος ο οποίος λύνει αυτό το πρόβλημα.

Κεφάλαιο 3

Correlation-Frequency Anonymization

Τα προηγούμενα μοντέλα του anonymization υπέθεταν ότι δεν υπάρχει καμία σημασιολογική εξάρτηση μεταξύ των τιμών της sensitive attribute και παράλληλα δεν έδιναν βάρος στην συχνότητα σύμφωνα με την οποία αυτές αλλάζουν καθώς και πως αυτό επηρεάζει το utility. Στόχος του παρακάτω κειμένου, είναι να παρουσιάσει ένα νέο μοντέλο με βάση το οποίο οι τιμές της ευαίσθητης ιδιότητας θα έχουν ένα βαθμό συσχετισμού και με βάση αυτόν να επιτύχει μεγαλύτερο βαθμό διαφύλαξης της ανωνυμίας και ταυτόχρονα να προσπαθεί να αυξήσει το utility της εκδιδόμενης βάσης λαμβάνοντας υπόψη πόσο συχνά γίνονται οι διαγραφές και οι εισαγωγές. Το πρώτο επιτυγχάνεται κατασκευάζοντας ένα νέο μοντέλο, το οποίο δεν απαιτεί μόνο να μπορεί ο αντίπαλος να ανακαλύψει με μικρή πιθανότητα την τιμή της ευαίσθητης ιδιότητας μίας εγγραφής, αλλά να μην μπορεί να ανακαλύψει εύκολα και σε ποια οικογένεια τιμών ανήκει. Το δε δεύτερο επιτυγχάνεται, λαμβάνοντας υπόψη την πιθανότητα διαγραφής μίας εγγραφής και πως αυτό συσχετίζεται με τα counterfeits και γενικεύοντας την ευαίσθητη ιδιότητα ώστε να μειωθούν τα counterfeits και να αυξηθεί το utility.

3.1 Κινητήριο Παράδειγμα

Τα περισσότερα μοντέλα του anonymization μέχρι τώρα υποθέτουν ότι οι τιμές μίας ευαίσθητης ιδιότητας είναι ανεξάρτητες μεταξύ τους. Στην πραγματικότητα όμως αυτό δεν ισχύει πάντα. Είναι πολύ πιθανόν οι τιμές να έχουν κάποια εξάρτηση μεταξύ τους και να ορίζουν ένα σύνολο οικογενειών. Για παράδειγμα ας δούμε το Σχήμα 3.1. Ο αντίπαλος μπορεί να ανακαλύψει με πιθανότητα 50% την ασθένεια του Bob, ο οποίος αντιστοιχεί στην εγγραφή 7. Όμως γνωρίζει ότι η ασθένεια του Bob είναι καρκίνος, αλλά δεν γνωρίζει τι τύπου. Με άλλα λόγια ο αντίπαλος κατάφερε να ανακαλύψει την οικογένεια των ασθενειών, από την οποία πάσχει ο Bob. Αν αυτή η οικογένεια λοιπόν δεν είναι ιδιαίτερα γενική, τότε τα υπάρχοντα μοντέλα δεν είναι σε θέση να διαφυλάξουν την ανωνυμία μίας εγγραφής.

Επιπλέον μέχρι τώρα τα περισσότερα μοντέλα, εκτός ενός, έχουν αντιμετωπίσει sensitive attributes οι οποίες είναι μόνο κατηγορικές. Στόχος αυτού του μοντέλου, είναι να υπάρξει ένας ενιαίος ορισμός τόσο για κατηγορικά όσο και για αριθμητικά δεδομένα.

Τέλος, το μοντέλο του m-invariance το οποίο εξετάζει το privacy από την δυναμική του πλευρά, επιτυγχάνει την διαφύλαξη της ανωνυμίας με την δημιουργία των counterfeits. Μέχρι τώρα τα counterfeits δημιουργούνται στην βάση, ανάλογα με τις διαγραφές και τις εισαγωγές που υπάρχουν. Αυτό όμως είναι πιθανόν να οδηγήσει σε σημαντική μείωση του utility. Αν έχουμε μία σημαντική αύξηση των counterfeits ως προς την τιμή μίας εγγραφής τότε αυτό

Patient Data				
	Non-sensitive			Sensitive
ID	Age	Sex	Zip code	Condition
1	23	M	11000	pneumonia
2	27	M	13000	dyspepsia
3	35	M	59000	dyspepsia
4	59	M	12000	pneumonia
5	61	F	54000	prostate cancer
6	65	F	25000	melanoma cancer
7	65	F	25000	prostate cancer
8	70	F	30000	colorectal cancer

(α′) Ακατέργαστα Δεδομένα

Patient Data				
ID	Age	Sex	Zip code	Group-ID
1	23	M	11000	1
2	27	M	13000	1
3	35	M	59000	1
4	59	M	12000	1
5	61	F	54000	2
6	65	F	25000	2
7	65	F	25000	2
8	70	F	30000	2

(β′) QIT table

Patient Data		
Group-ID	Disease	Count
1	dyspepsia	2
1	pneumonia	2
2	colorectal cancer	1
2	melanoma cancer	2
2	prostate cancer	1

(γ′) ST table

Σχήμα 3.1: anatomized tables

θα οδηγήσει σε σημαντική αλλαγή της κατανομής και βέβαια θα δυσκολέψει τον ερευνητή να βγάλει ασφαλή συμπεράσματα από την εκδιδόμενη βάση. Αν όμως λάβουμε υπόψη μας την πιθανότητα εμφάνισης ενός counterfeit και έχουμε και την ικανότητα της γενίκευσης (ή τροποποίησης) της τιμής της sensitive attribute τότε μπορούμε όπως θα δειχθεί και παρακάτω να αυξήσουμε το utility.

3.2 Βασικοί Ορισμοί

Σε κάθε πίνακα T υπάρχει ένα σύνολο ιδιοτήτων βάση των οποίων είναι πιθανόν να αναγνωριστεί μία εγγραφή.

Ορισμός 4. *Quasi-Identifier Attribute Set Είναι το ελάχιστο σύνολο από ιδιότητες $Q = X_1, \ldots, X_d$ με το οποίο ένας πίνακας T μπορεί να γίνει join με κάποιες εξωτερικές πληροφορίες για να αναγνωριστούν ατομικές εγγραφές.*

Μία προσέγγιση του privacy είναι το k-anonymity το οποίο έχεις ως στόχο την διαφύλαξη της ταυτότητας μίας εγγραφής:

Ορισμός 5. *Frequency Set. Έστω ένας πίνακα T και ένα σύνολο ιδιοτήτων $Q = X_1, \ldots, X_d$ Το frequency set του T με βάση το Q είναι μία αντιστοίχιση με κάθε μοναδικό συνδυασμό των τιμών $(x_1, \ldots, x_d)$ του Q στο T με το συνολικό αριθμών των εγγραφών στο T με βάση τιμές του Q.*

Ορισμός 6. *k-anonymity Property. Ένας πίνακας T θα ικανοποιεί την k-anonymity property ή θα λέμε ότι είναι k-anonymous με βάση ένα σύνολο ιδιοτήτων $Q = X_1, \ldots, X_d$ εάν κάθε τιμή πλήθους στο frequency set του T με βάση το Q είναι μεγαλύτερο ή ίσο του k.*

Ορισμός 7. *k-anonymization. Μία όψη V ενός πίνακα T θα είναι k-anonymization του T αν η όψη αλλάζει ή γενικεύει τα δεδομένα του T με βάση κάποιο μοντέλο έτσι ώστε το V να είναι k-anonymous με βάση το quasi-identifier.*

Μία άλλη προσέγγιση είναι το m-invariant, όπου αποτελεί μία επέκταση του l-diversity σε δυναμικά δεδομένα:

Ορισμός 8. *Partition/QI-group. Ένα partition αποτελείται από διάφορα υποσύνολα του T έτσι ώστε κάθε εγγραφή στο T να ανήκει ακριβώς σε ένα υποσύνολο. Αναφερόμαστε σε αυτά τα υποσύνολα σαν QI-groups και τα συμβολίζουμε $QI_1, \ldots, QI_p$. Με άλλα λόγια ένα partition του T είναι ένα σύνολο ξένων συνόλων των οποίων η ένωση μας δίνει το T.*

Εμείς επιθυμούμε ένα partition το οποίο να είναι l-diverse:

Ορισμός 9. *l-diverse partition. Ένα partition από p QI-groups είναι l-diverse, αν κάθε QI-group $QI_j (1 \leq j \leq p)$ ικανοποιεί την παρακάτω συνθήκη: Έστω u η πιο συχνή τιμή της sensitive attribute S στο QI_j και $c_j(u)$ ο αριθμός των εγγραφών $t \in QI_j$ έτσι ώστε $t[S] = u$, τότε πρέπει $c_j(u)/|QI_j| \leq 1/l$ όπου $|QI_j|$ ο πληθάριθμος του QI_j.*

Εμείς επιθυμούμε λοιπόν να εκδώσουμε ένα l-diverse partition. Στην βιβλιογραφία υπάρχουν δύο τρόποι για να διατηρηθεί το privacy, ο ένας είναι ο anatomy και ο άλλος η generalization. Δίνονται οι δύο ορισμοί:

Ορισμός 10. *Anatomy. Δοθέντος ενός l-diverse partition με m QI-groups, ο anatomy παράγει ένα quasi-identifier table (QIT) και ένα sensitive table (ST) με βάση τις παρακάτω ιδιότητες:*

- *Το QIT έχει το σχήμα $(X_1, \ldots, X_d, Group-ID)$ όπου $X_1, ..., X_d$ όλες οι ιδιότητες της βάσης με εξαίρεση της sensitive.*
- *Για κάθε QI-group $QI_j (1 \leq j \leq p)$ και για κάθε εγγραφή $t = (t[1], t[2], ..., t[d], s) \in QI_j$, το QIT έχει μία εγγραφή της μορφής: $(t[1], t[2], ..., t[d], j)$*
- *Το ST έχει σχήμα $(Group - ID, A^s, Count)$.*
- *Για κάθε QI-group $QI_j (1 \leq j \leq p)$ και για κάθε τιμή v της sensitive ιδιότητας A^s στο QI_j, το ST έχει μία εγγραφή της μορφής: $(j, v, c_j(v))$, όπου $c_j(u)$ ο αριθμός των εγγραφών $t \in QI_j$ έτσι ώστε $t[d+1] = u$.*
- *Εκτός από τις προηγούμενες εγγραφές δεν υπάρχουν άλλες στους δύο πίνακες.*

Ορισμός 11. *Generalization. Δοθέντος ενός partition του T με m QI-groups, για κάθε εγγραφή $t \in T$, ο γενικευμένος πίνακας T^*, περιέχει μία εγγραφή της μορφής: $(QI_j[1], ..., QI_j[d], t[d+1])$ όπου $QI_j (1 \leq i \leq p)$ είναι το μοναδικό QI-group το οποίο περιλαμβάνει το t και $QI_j[i] (1 \leq i \leq d)$ είναι ένα interval το οποίο καλύπτει το $t[i]$. Επιπλέον το $QI_j[i]$ είναι ίδιο για όλες τις εγγραφές $t \in QI_j$.*

Τέλος δίνεται ο ορισμός του m-invariance:

Ορισμός 12. *Signature. Έστω ένας πίνακας T και μία anonymous όψη T^*. Ονομάζουμε signature μίας εγγραφής $t^* \in T^*$ το σύνολο των διαφορετικών τιμών που μπορεί να πάρει η sensitive attribute στο QI-group όπου ανήκει η εγγραφή.*

Ορισμός 13. *m-Equality. Έστω ένας πίνακας T και μία m-diverse όψη T^*. Έστω ότι η ευαίσθητη S παίρνει τις τιμές $s_1, ..., s_r$ σε κάποιο τυχαίο QI-group QI_j και ότι $|s_i|$ συμβολίζει τον πληθάριθμο της συγκεκριμένης τιμής στο QI_j. Θα λέμε ότι η όψη ικανοποιεί το m-equality αν $|s_i| = |s_k|$ για κάθε i, k σε κάθε QI-group.*

Με αλλά λόγια επί της ουσίας απαιτούμε σε κάθε QI-group μεγέθους $n * k$, με $k \geq m$, κάθε τιμή της ευαίσθητης να εμφανίζεται ή ακριβώς n φορές ή καμία.

Ορισμός 14. *Counterfeit. Έστω ένας πίνακας T και μία m-equal όψη T^*. Θα ονομάζουμε counterfeit μία εγγραφή η οποία υπάρχει στην όψη T^* αλλά όχι στον T. Το lifespan μίας counterfeit εγγραφής είναι ίσο με μηδέν.*

Ορισμός 15. *m-invariance. Έστω μία αλληλουχία anonymous όψεων $T^*(1), ..., T^*(n)$ μίας βάσης T. Θα λέμε ότι ικανοποιούν το m-invariance αν*

- *Κάθε όψη είναι m-equal και*
- *αν κάθε εγγραφή t η οποία ανήκει στο T και υπάρχει σε κάποιες από αυτές τις όψεις ως generalized ή anatomized εγγραφή t^*, έχει πάντα την ίδια signature.*
- *Είναι πιθανόν σε αυτές τις όψεις να υπάρχουν counterfeits.*

3.3 Anatomy και Γενίκευση

Όπως προαναφέρθηκε στην βιβλιογραφία υπάρχουν δύο βασικοί τρόποι για την διατήρηση του privacy, ο anatomy και ο generalization. Αν υποθέσουμε ότι ο αντίπαλος γνωρίζει ότι μία εγγραφή υπάρχει στην βάση, τότε όπως θα δειχθεί και παρακάτω ο anatomy υπερτερεί έναντι του generalization. Βέβαια ακόμα και σε περίπτωση, όπου ο αντίπαλος δεν γνώριζε την ύπαρξη μίας εγγραφής σε μία βάση τότε και πάλι δεν έχει νόημα η απλή γενίκευση, έχοντας ως κριτήριο την κατασκευή ενός QI-group. Το κριτήριο μας πρέπει να είναι το k-anonymity.

Πρώτον, είναι προφανές ότι εάν έχουμε χωρίσει τον πίνακα σε QI-groups ώστε να ικανοποιεί το l-diversity, τότε μπορούμε να εκδώσουμε τον πίνακα και με βάση το anatomy και με βάση το generalization. Υπάρχει μάλιστα μία προς μία σχέση μεταξύ των δύο. Δίνεται ο ορισμός την συνάρτησης όπου μετατρέπει ένα generalized table σε anatomized table:

Ορισμός 16. *Έστω ένας πίνακας T και μία l-diverse generalized view T^*, με QI-groups τα $QI_1, ..., QI_p$. Η συνάρτηση f η οποία έχει ως πεδίου ορισμού την T^* παράγει μία νέα όψη T^{**} η οποία έχει ένα quasi-identifier table (QIT) και ένα sensitive table (ST) με βάση τις παρακάτω ιδιότητες:*

- *Το QIT έχει το σχήμα $(X_1, \ldots, X_d, Group-ID)$ όπου $X_1, ..., X_d$ όλες οι ιδιότητες της βάσης με εξαίρεση της sensitive. Οι τιμές αυτές των ιδιοτήτων ξε-γενικεύονται στις κανονικές τιμές όπου αντιστοιχούν στην βάση T.*
- *Για κάθε QI-group $QI_j (1 \leq j \leq p)$ και για κάθε εγγραφή $t = (t[1], t[2], ..., t[d], s) \in QI_j$, το QIT έχει μία εγγραφή της μορφής: $(t[1], t[2], ..., t[d], j)$*
- *Το ST έχει σχήμα $(Group - ID, A^s, Count)$.*
- *Για κάθε QI-group $QI_j (1 \leq j \leq p)$ και για κάθε τιμή v της sensitive ιδιότητας A^s στο QI_j, το ST έχει μία εγγραφή της μορφής: $(j, v, c_j(v))$, όπου $c_j(u)$ ο αριθμός των εγγραφών $t \in QI_j$ έτσι ώστε $t[d+1] = u$.*

Patient Data				
	Non-sensitive			Sensitive
ID	Age	Sex	Zip code	Condition
1	[23-59]	M	[11000-59000]	pneumonia
2	[23-59]	M	[11000-59000]	dyspepsia
3	[23-59]	M	[11000-59000]	dyspepsia
4	[23-59]	M	[11000-59000]	pneumonia
5	[61-70]	F	[25000-54000]	prostate cancer
6	[61-70]	F	[25000-54000]	melanoma cancer
7	[61-70]	F	[25000-54000]	prostate cancer
8	[61-70]	F	[25000-54000]	colorectal cancer

Σχήμα 3.2: generalized tables

- *Εκτός από τις προηγούμενες εγγραφές δεν υπάρχουν άλλες στους δύο πίνακες.*

Την συνάρτηση αυτή θα την λέμε generalization to anatomy function.

Είναι προφανές ότι υπάρχει και η συνάρτηση η οποία υλοποιεί και την αντίστροφη διαδικασία. Η ύπαρξη της αντίστροφης συνάρτησης δεν έχει και μεγάλη σημασία. Ο λόγος είναι ότι ένας anatomized table προσφέρει στον ερευνητή μεγαλύτερη πληροφορία. Αν γενικεύσουμε το quasi-identifier τότε άμεσα χάνουμε και πληροφορία, αφού πια κάθε εγγραφή δεν αντιστοιχεί σε ένα σημείο του χώρου αλλά σε μία περιοχή στον χώρο, μειώνοντας μας έτσι την δυνατότητα να εξάγουμε ασφαλείς πληροφορίες από την anatomized όψη. Το Κεφάλαιο 2.6.2 εξηγεί και με ένα πιο formal τρόπο γιατί η γενίκευση μειώνει την πληροφορία. Μπορεί και κανείς να παρατηρήσει γενικότερα γιατί το anatomy είναι προτιμότερο αν ρίξει μία ματιά στα σχήματα 3.1 και 3.2. Είναι προφανές ότι ο δεύτερος πίνακας περιέχει λιγότερη πληροφορία από τον πρώτο.

Στην πραγματικότητα όμως η μέθοδος του generalization, έχει ένα πλεονέκτημα, προσφέρει στην προστασία της ταυτότητας μίας εγγραφής. Δηλαδή, αν δεν γνωρίζουμε ότι η εγγραφή υπάρχει ήδη στην βάση, τότε εξαιτίας της γενίκευσης είναι δύσκολο να εξάγουμε κάποιο συμπέρασμα. Επίσης η γενίκευση μας προστατεύει έναντι και στην ανακάλυψη του lifespan μίας εγγραφής. Δηλαδή ο χρήστης δεν μπορεί να γνωρίζει πότε ακριβώς διαγράφτηκε μία εγγραφή, αφού πιθανόν να συσχετίζεται με άλλες l.Όμως η γενίκευση των QI-groups δεν αποτελεί κάποιο formal τρόπο προστασίας της ταυτότητας και του lifespan, αν θέλουμε να εμποδίσουμε τον αντίπαλο να αναγνωρίσει μία εγγραφή τότε πρέπει να εφαρμόσουμε το k-anonymity και το l-diversity μαζί. Μόνο σε περίπτωση, όπου $k = l$, μπορούμε να ισχυριστούμε ότι το generalization είναι μία ασφαλής μεθοδολογία για την προστασία της ταυτότητας και του lifespan μίας εγγραφής.

3.4 Anatomy και Utility

Αυτό όμως που αποτυγχάνει να ελέγξει η σύγκριση μέσω της pdf-function είναι κατά πόσο η κατασκευή των QI-groups επηρεάζει το utility. Το πρόβλημα μας γενικότερα, δεν είναι απλά να ανακατασκευάσουμε μία εγγραφή, αλλά με επιτυχία μία περιοχή. Ας θυμηθούμε ξανά το παράδειγμα με τις ασθένειες και τις πόλεις όπου υπάρχει στο Κεφάλαιο 2.6.4.

Ας φανταστούμε ότι έχουμε τέσσερις πόλεις την Α, Β, Γ και Δ, οι δύο πρώτες είναι στα Βόρεια Προάστια της Αθήνας και άλλες δύο στα Νότια. Στην πόλη Α παρουσιάζεται η ασθένεια α, στην πόλη Β η β, στην πόλη Γ η γ και στην πόλη Δ η δ και καμία άλλη. Θεωρούμε ότι

κάθε πόλη έχει το ίδιο πλήθος εγγραφών ακριβώς n και επιθυμούμε να πετύχουμε 2-diverse. Εάν εφαρμόσουμε κάποιο τυχαίο αλγόριθμο τότε είναι πολύ πιθανόν να πάρουμε την παρακάτω λύση:

- Κατασκευάζονται ακριβώς 2n groups, με δύο εγγραφές ακριβώς το καθένα.
- Κάθε group από τα πρώτα n έχει μία εγγραφή από την πόλη Α και μία από την πόλη Γ.
- Κάθε group από τα υπόλοιπα n έχει μία εγγραφή από την πόλη Β και μία από την πόλη Δ.

Έστω λοιπόν ότι τώρα επιθυμούμε να κάνουμε μία στατιστική μελέτη και προσπαθούμε να βρούμε κάποια στατιστικά στοιχεία για κάθε περιοχή. Με βάση τον παραπάνω δημοσιευμένο πίνακα, θα γνωρίζουμε ότι στα Βόρεια Προάστια και Νότια Προάστια εμφανίζονται οι ασθένειες α,β,γ,δ ισοπίθανα.

Αντίθετα αν είχαμε την παρακάτω λύση:

- Κατασκευάζουμε ακριβώς 2n groups, με δύο εγγραφές ακριβώς το καθένα.
- Κάθε group από τα πρώτα n έχει μία εγγραφή από την πόλη Α και μία από την πόλη Β.
- Κάθε group από τα υπόλοιπα n έχει μία εγγραφή από την πόλη Γ και μία από την πόλη Δ.

Θα μπορούσαμε με απόλυτη ακρίβεια να υπολογίσουμε τις ασθένειες για κάθε περιοχή. Στον πίνακα 3.3 δίνεται και ένα instance του παραδείγματος.

Patient Data			
City	Zip code	altitude	Disease
New York	10000	high	flu type A
New York	10000	high	flu type A
New Jersey	10050	low	prostate cancer
New Jersey	10050	low	prostate cancer
Los Angeles	20000	high	flu type B
Los Angeles	20000	high	flu type B
San Francisco	20050	low	melanoma cancer
San Francisco	20050	low	melanoma cancer

(α′) Ακατέργαστα Δεδομένα

Patient Data			
City	Zip code	altitude	Disease
New York	10000	high	flu type A
New York	10000	high	flu type A
Los Angeles	20000	high	flu type B
Los Angeles	20000	high	flu type B
New Jersey	10050	low	prostate cancer
New Jersey	10050	low	prostate cancer
San Francisco	20050	low	melanoma cancer
San Francisco	20050	low	melanoma cancer

(β′) Example 1

Patient Data			
City	Zip code	altitude	Disease
New York	10000	high	flu type A
New York	10000	high	flu type A
New Jersey	10050	low	prostate cancer
New Jersey	10050	low	prostate cancer
Los Angeles	20000	high	flu type B
Los Angeles	20000	high	flu type B
San Francisco	20050	low	melanoma cancer
San Francisco	20050	low	melanoma cancer

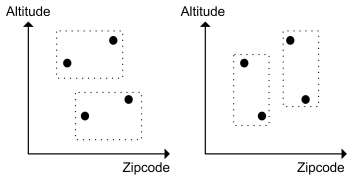
(γ′) Example 2

Σχήμα 3.3: anatomized tables

Το πρόβλημα μας σε αυτή την περίπτωση είναι η περίμετρος του QI-group. Με αλλά λόγια αν όλες οι εγγραφές σε ένα QI-group ήταν κοντά στον χώρο μεταξύ τους, τότε θα απαντούσαμε με μεγαλύτερη ασφάλεια range queries. Για παράδειγμα με βάση τον πίνακα,

αν τα range queries γίνονταν με βάση το zipcode τότε το example 2 είναι προφανέστατα μία καλύτερη προσέγγιση. Δηλαδή αν είχαμε κατασκευάσει τα QI-groups με βάση την απόσταση του zipcode τότε θα είμαστε σε θέση να απαντήσουμε καλύτερα τέτοια ερωτήματα.

Το πρόβλημα μας λοιπόν είναι να τοποθετήσουμε εγγραφές ώστε να μειώσουμε την περίμετρο. Στην πραγματικότητα όμως αυτό δεν είναι πάντα εφικτό και ούτε εύκολο να το ελέγξουμε. Δεν ξέρουμε από πριν την μορφή των queries. Δεν ξέρουμε ποια attributes αφορούν, δεν ξέρουμε την μορφή τους (δηλαδή πως ακριβώς μετράει την απόσταση ή την περίμετρο ο χρήστης) και παράλληλα δεν μπορούμε εύκολα να μοντελοποιήσουμε overlapping queries. Δείτε για παράδειγμα ξανά το Σχήμα 3.3. Τι θα γίνει αν το query γίνει με βάση το altitude; Προφανέστατα απαιτείται μία άλλη προσέγγιση. Όμως τα queries μπορούνε να αφορούν μία ιδιότητα ή συνδυασμό ιδιοτήτων. Επιπλέον η μορφή των queries μας επηρεάζει αρκετά. Δείτε το Σχήμα 3.4, είναι διαφορετικό τα queries μας να παίρνουν την μορφή τετραγώνου και διαφορετικό την μορφή ορθογωνίου. Παρ' όλα αυτά γενικότερα είναι προτιμότερο να προσπαθήσουμε να κατασκευάσουμε QI-groups με κριτήριο την ελαχιστοποίηση της περιμέτρου με βάση κάποια μετρική, από το να μην επιλέξουμε κανένα κριτήριο.



Σχήμα 3.4: Μία ιεραρχία κατηγορικών δεδομένων

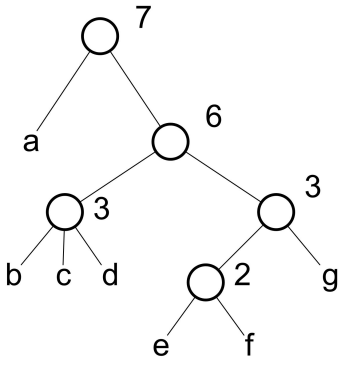
Όμως μία καλύτερη προσέγγιση είναι να επιλέξουμε να τοποθετήσουμε εγγραφές στα QI-groups των οποίων οι sensitive values έχουν μεγάλο βαθμό συσχετισμού. Πιο συγκεκριμένα ας δούμε ξανά τον πίνακα 3.3. Ναι μεν η δεύτερη προσέγγιση είναι καλύτερη σε περίπτωση που ρωτάμε με βάση το zipcode, παρ' όλα αυτά και η πρώτη μας εισάγει μικρό σφάλμα. Για την ακρίβεια ναι μεν δεν μπορεί να μας δώσει την ακριβή απάντηση σε range queries αυτής της μορφής αλλά μπορεί να μας δώσει μία ικανοποιητική προσεγγιστική απάντηση. Μπορούμε δηλαδή να γνωρίζουμε με πιθανότητα 100 τοις εκατό, ότι οι μισοί ασθενείς στην New York και το New Jersey έχουν flu και cancer. Μπορεί ναι μεν, να μην γνωρίζουμε τον ακριβή τύπο τους, αλλά η απάντηση αποτελεί μία καλή προσέγγιση. Μάλιστα αν έχουμε μία ιεραρχία γενίκευσης για τα δεδομένα μας, τότε είναι πολύ πιο εύκολο να ελέγξουμε αυτό το μέτρο, από ότι την περίμετρο, αφού στην τελευταία περίπτωση έχουμε πολλές περισσότερες ιδιότητες να ελέγξουμε. Προσέξτε όμως ότι υπάρχει ένα όριο, στο βαθμό συσχετισμού των sensitive values, όπως θα δειχθεί και παρακάτω.

Έτσι αν επιθυμούμε να απαντάμε με όσο το δυνατόν μικρότερο σφάλμα ένα range query πρέπει να διατηρούμε τις sensitive values κοντά μεταξύ στους σε ένα QI-group, η περίμετρος ενός QI-group να είναι όσο το δυνατόν μικρότερη και τέλος το μέγεθος του QI-group να είναι όσο το δυνατόν μικρότερο.

3.5 m-equality και m-unique

Μία υποπερίπτωση του m-equality είναι το m-unique. Πιο συγκεκριμένα:

Ορισμός 17. *m-Unique. Έστω ένας πίνακας T και μία m-equal όψη T^*. Έστω ότι η ευαίσθητη S παίρνει τις τιμές $s_1, ..., s_r$ σε κάποιο τυχαίο QI-group QI_j και ότι $|s_i|$ συμβολίζει τον πληθάριθμο της συγκεκριμένης τιμής στο QI_j. Θα λέμε ότι η όψη ικανοποιεί το m-unique αν $|s_i| = |s_k| = 1$ για κάθε i, k σε κάθε QI-group.*



Σχήμα 3.5: Μία ιεραρχία κατηγορικών δεδομένων

Η διαφορά τώρα είναι ότι τώρα το QI-group έχει το μικρότερο δυνατόν μέγεθος. Με βάση το προηγούμενο section αν αυτό γίνεται με κριτήριο της ελαχιστοποίησης της περιμέτρου, τότε το m-unique μας δίνει καλύτερα αποτελέσματα. Σε όλο το παρακάτω κεφάλαιο λοιπόν, θα χρησιμοποιήσουμε την μεθοδολογία του anatomy και θα κατασκευάζουμε m-unique views.

3.6 Correlation για Δυναμικά Δεδομένα

3.6.1 Κατηγορικά δεδομένα

Ένας τρόπος συσχέτισης των κατηγορικών δεδομένων προτάθηκε στο Utility Anonymization και αφορά όμως το k-anonymity και τις ιδιότητες του Quasi-Identifier. Ακολουθώντας την ίδια προσέγγιση, ορίζουμε τον συσχετισμό για την sensitive attribute.

Συνήθως για να γενικεύσουμε τα κατηγορικά δεδομένα έχουμε κάποια είδους ιεραρχία. Θα θεωρήσουμε ότι αυτή η ιεραρχία για μία κατηγορική τιμή είναι ένα δέντρο. Το πρόβλημα είναι πως ακριβώς μπορούμε να χρησιμοποιήσουμε αυτή την ιεραρχία για να μετρήσουμε τον συσχετισμό στα κατηγορικά δεδομένα.

Η πρώτη σκέψη είναι σε κάποια ιεραρχία να μετρήσουμε το ελάχιστο μονοπάτι. Δηλαδή έστω ότι έχω ένα Domain $D = d_1, ..., d_n$ και θέλω να μετρήσω την απόσταση των τιμών d_i και d_j. Τότε θα μπορούσα να θεωρήσω ως συσχετισμό το μήκος του ελάχιστου μονοπατιού το οποίο περνάει από αυτούς τους δύο κόμβους.

Αυτό το μέτρο όμως δεν είναι ικανοποιητικό. Δείτε το Σχήμα 3.5, με βάση αυτό το μέτρο το b είναι πιο κοντά με το a παρά με το e αφού το ελάχιστο μονοπάτι που ενώνει τα δύο πρώτα είναι μικρότερο. Στην πραγματικότητα όμως είναι καλύτερο το αντίθετο αφού το b και το e έχουν πιο κοντινό πρόγονο. Ένα καλύτερο μέτρο λοιπόν είναι για να μετρήσουμε τον συσχετισμό, είναι να επιλέξουμε το κοντινότερο πρόγονο. Ένας τρόπος να το κάνουμε αυτό είναι να μετρήσουμε ποιος πρόγονος έχει τα λιγότερα παιδιά, όσο τα λιγότερα τόσο μικρότερος ο συσχετισμός. Δίνεται ο ακριβής ορισμός:

Ορισμός 18. *Categorical Correlation Έστω ότι η ιδιότητα S είναι κατηγορική και παίρνει τις διακριτές τιμές $s_1, s_2, ..., s_n$. Τότε ορίζουμε ως συσχετισμό μεταξύ δύο τιμών u_i και u_j: $corr(u_i, u_j) = |S|/size(u)$ όπου u είναι ο κοντινότερος πρόγονος των u_i και u_j, $size(u)$ είναι το πλήθος των φύλλων που έχουν πρόγονο των u και $|S|$ είναι το πλήθος των διακριτών τιμών τις οποίες μπορείς να πάρει η attribute.*

Παρατηρήστε ότι με βάση με αυτό τον ορισμό είναι όντως πιο κοντά αυτή την φορά το b με το e. Προσέξτε ότι σε κάθε τιμή μπορούμε να δώσουμε και διαφορετική αξία αν βάζαμε βάρη

στις ακμές του δέντρου και έτσι για κάθε φύλλο μετά είχαμε 'άλλη αξία'. Επίσης προσέξτε ότι ο αριθμητής είναι πάντα σταθερός, άρα δεν χρειάζεται να τον υπολογίζουμε για να συγκρίνουμε συσχετίσεις μεταξύ τους.

3.6.2 Αριθμητικά δεδομένα

Με παρόμοιο τρόπο με πριν μπορούμε να ορίσουμε την συσχέτιση μεταξύ αριθμητικών δεδομένων. Το βασικό πρόβλημα στα αριθμητικά δεδομένα είναι με ποιο τρόπο θα μετρήσουμε την απόσταση. υπάρχουν διάφορα μέτρα, όπως η απόσταση Manhattan κτλ. Από εδώ και πέρα θα θεωρήσουμε ότι υπάρχει μία συνάρτηση απόστασης $f(x_1, x_2)$, η οποία έχει δοθεί από το πρόβλημα. (Θεωρούμε ότι για την συνάρτηση f ισχύει πάντα $f(x, y) = f(y, x)$.

Ορισμός 19. *Numeric Correlation. Έστω ότι η attribute S είναι αριθμητική. Τότε η συσχέτιση μεταξύ δύο τιμών u_i και u_j της S δίνεται από τον τύπο: $corr(u_i, u_j) = f(min, max)/f(u_j, u_i)$*

Όπου min, max η ελάχιστη και μέγιστη τιμή όπου μπορεί να πάρει η ιδιότητα.

3.7 Correlation based anonymization

Με βάση τα παραπάνω θα δώσουμε ένα νέο ορισμό για το privacy, ο οποίος έχει ως στόχο μία καλύτερη διαφύλαξη της ανωνυμίας μία εγγραφής.

Επιθυμούμε επίσης με βάση το ορισμό του m-invariance μία εγγραφή να έχει πάντα την ίδια signature. Όμως δεδομένου ότι πιθανόν τώρα να έχουμε και κάποια μορφή γενίκευσης, επεκτείνουμε τον ορισμό αυτό.

Ορισμός 20. *Έστω οι τιμές $s_1, ..., s_r$ της ευαίσθητης ιδιότητας. Θα λέμε ότι αυτές ανήκουν στο ίδιο σταθερό μονοπάτι αν υπάρχει μία ταξινόμηση της μορφής $s'_1, ..., s'_r$ έτσι ώστε :*

- *ο s_i να είναι ένα interval το οποίο καλύπτει το s_{i+1}.*

Με βάση τους παραπάνω ορισμούς είμαστε σε θέση να δώσουμε έναν ισχυρότερο ορισμό για να διατηρούμε το privacy.

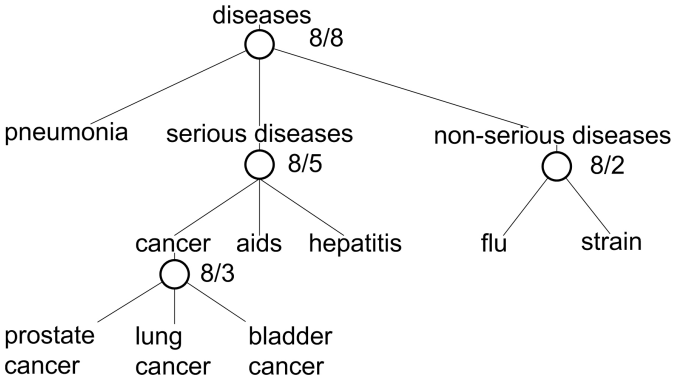
Ορισμός 21. *Θα λέμε ότι δύο signatures $s_a = s_{a_1}, ..., s_{a_n}$ και $s_b = s_{b_1}, ..., s_{b_k}$ είναι similar αν $k = n$ και για κάθε $1 \leq i \leq k$ οι s_{a_i} και s_{b_i} ανήκουν στο ίδιο σταθερό μονοπάτι.*

Ορισμός 22. *(e, m)-Correlation. Έστω ένας πίνακας T και μία l-diverse όψη του T^*. Έστω ότι η ευαίσθητη ιδιότητα S παίρνει τις τιμές $s_1, ..., s_r$ στο QI-group QI. θα λέμε ότι η όψη T^* ικανοποιεί το m-Correlation αν*

- *ικανοποιεί το m-equality*
- *και για κάθε εγγραφή η οποία βρίσκεται σε ένα $QI - group$ με κάποιες εγγραφές να έχει βαθμό συσχετισμού με κάθε μία από αυτές τις εγγραφές το πολύ e.*

Με άλλα λόγια, απαιτούμε

- κάθε όψη την οποία εκδίδουμε να είναι $m - equal$
- όλες οι εγγραφές να μην μπούνε στο QI-group με εγγραφές οι οποίες είναι αρκετά κοντά μεταξύ τους.



Σχήμα 3.6: Μία ιεραρχία κατηγορικών δεδομένων

Επίσης αν $e > 0$ τότε η πρώτη συνθήκη ορισμού καλύπτεται από την δεύτερη, με την προϋπόθεση να υπάρχουν τουλάχιστον l τιμές στο QI-group, ενώ αν $e = 0$ τότε η δεύτερη συνθήκη ικανοποιείται πάντα. Στα σχήματα 3.7 και 3.8 μπορούμε να δούμε ένα παράδειγμα της διαφοράς του ορισμού αυτού από τον κλασσικό του m-equal αν λάβουμε υπόψη μας την ιεραρχία του Σχήματος 3.6

Ορισμός 23. (e, m)-*Correlation Based Anonymity. Έστω μία αλληλουχία anonymous όψεων* $T^*(1), ..., T^*(n)$ *μίας βάσης* T. *Θα λέμε ότι ικανοποιούν το* (e, m)-*cba αν*

- *Κάθε όψη ικανοποιεί το (e.m)-correlation,*
- *Κάθε εγγραφή* $t \in T$ *η οποία υπάρχει σε κάποια από τις όψεις ως γενικευμένη εγγραφή* t^*, *έχει ένα σύνολο από signatures οι οποίες είναι similar μεταξύ τους.*
- *Και κάθε δύο εγγραφές* t_a, t_b *οι οποίες υπήρξαν στο ίδιο QI-group σε κάποια όψη* $T^*(j), 1 \leq j \leq n$ *έχουν ένα σύνολο από signatures οι οποίες είναι similar μεταξύ τους.*
- *Είναι πιθανόν σε αυτές τις όψεις να υπάρχουν counterfeits.*

Επί της ουσίας η διαφορά με τον ορισμό του m-invariance είναι ότι απαιτούμε οι τιμές της ευαίσθητης ιδιότητας σε ένα QI-group να έχουν μικρό βαθμό συσχετισμό μεταξύ τους. Επιπλέον δίνουμε την δυνατότητα της γενίκευσης (ή το αντίστροφο) στην τιμή της ευαίσθητης ιδιότητας, αρκεί πάντα να διατηρούμε τις signatures στο ίδιο σταθερό μονοπάτι.

3.8 Utility

Παράλληλα, εκτός από αυτή την διαφύλαξη, είναι επιθυμητό να διατηρήσουμε όσο το δυνατόν και μεγαλύτερο utility για τα δεδομένα. Με άλλα λόγια επιθυμούμε να έχουμε όσο το δυνατόν μικρότερη γενίκευση των δεδομένων. Έτσι ορίζουμε ένα βαθμό ποινής για την γενίκευση των δεδομένων μας.

Αρχικά ορίζουμε την ποινή για την γενίκευση μία κατηγορικής ιδιότητας

Ορισμός 24. *Έστω ένας πίνακας* T *με quasi-identifierr* $(X_1, \ldots, X_d)$, *όπου κάποιες από τις ιδιότητες πιθανόν να είναι κατηγορικές. Έστω ότι η ιδιότητα* X_i *είναι κατηγορική, θεωρούμε μία εγγραφή* $t = (x_1, \ldots, v, \ldots, x_d)$ *η οποία γενικεύεται στην εγγραφή* $t' = (x'_1, \ldots, u, \ldots, x'_d)$ *όπου* u *είναι ένας πρόγονος του* v *με κάποιο domain hierarchy. Για την ιδιότητα* X_i *η ποινή normalized certainty penalty ορίζεται ως*

Patient Data				
	Non-sensitive			Sensitive
ID	Age	Sex	Zipcode	Condition
1	23	M	11000	pneumonia
2	27	M	13000	flu
3	35	M	59000	flu
4	59	M	12000	strain
5	61	F	54000	strain
6	65	F	25000	strain
7	65	F	25000	aids
8	70	F	30000	hepatitis
9	40	M	54000	prostate cancer
10	32	F	15000	prostate cancer
11	45	M	25000	lung cancer
12	75	F	10300	bladder cancer

(α′) Ακατέργαστα Δεδομένα

Patient Data				
ID	Age	Sex	Zipcode	Group-ID
1	23	M	11000	2
2	27	M	13000	1
3	35	M	59000	1
4	59	M	12000	2
5	61	F	54000	1
6	65	F	25000	1
7	65	F	25000	3
8	70	F	30000	6
9	40	M	54000	3
10	32	F	15000	4
11	45	M	25000	4
12	75	F	10300	5

(β′) QIT table

Patient Data		
Group-ID	Disease	Count
1	flu	2
1	strain	2
2	strain	1
2	pneumonia	1
3	prostate cancer	1
3	aids	1
4	lung cancer	1
4	prostate cancer	1
5	bladder cancer	1
5	hepatitis	1

(γ′) ST table

Σχήμα 3.7: anatomized tables with l-diversity

$$NCP_{X_i}(t) = size(u)/|X_i|$$

όπου $|X_i|$ είναι όλες οι τιμές τις οποίες μπορεί να πάρει η ιδιότητα T στον πίνακα T και $size(u)$ είναι ίσο με το πλήθος των φύλλων που έχουνε πρόγονο το u.

Επί της ουσίας ο άνω τύπος είναι $1/corr(u, v)$. Με αλλά λόγια η ποινή μας είναι η αντίστροφη του συσχετισμού δύο τιμών.

Όμοια με τις κατηγορικές ιδιότητες ορίζουμε την ποινή για αριθμητικές:

Ορισμός 25. *Έστω ένας πίνακας T με quasi-identifierr $(X_1, \ldots, X_d)$, όπου κάποιες από τις ιδιότητες πιθανόν να είναι αριθμητικές. Έστω ότι η ιδιότητα X_i είναι αριθμητική, θεωρούμε μία εγγραφή $t = (x_1, \ldots, x_i, \ldots, x_d)$ η οποία γενικεύεται στην εγγραφή $t' = (x'_1, \ldots, [z_i, y_i], \ldots, x'_d)$ έτσι ώστε $y_i \leq x_i \leq z_i$. Για την ιδιότητα X_i η ποινή normalized certainty penalty ορίζεται ως*

$$NCP_{X_i}(t) = (z_i - y_i)/|X_i|$$

όπου $|X_i|$ είναι η διαφορά της μέγιστης και της ελάχιστης τιμής της ιδιότητας στον πίνακα T.

Αν $X_1, ..., X_d$ όλες οι ιδιότητες ενός QI τότε την ποινή για το QI $sum_{i=1}^{d} NPC_{X_i}(t)/d$ θα την συμβολίζουμε με $NCP_{qi}(t)$.

Patient Data				
	Non-sensitive			Sensitive
ID	Age	Sex	Zipcode	Condition
1	23	M	11000	pneumonia
2	27	M	13000	flu
3	35	M	59000	flu
4	59	M	12000	strain
5	61	F	54000	strain
6	65	F	25000	strain
7	65	F	25000	aids
8	70	F	30000	hepatitis
9	40	M	54000	prostate cancer
10	32	F	15000	prostate cancer
11	45	M	25000	lung cancer
12	75	F	10300	bladder cancer

(α′) Ακατέργαστα Δεδομένα

Patient Data				
ID	Age	Sex	Zipcode	Group-ID
1	23	M	11000	1
2	27	M	13000	4
3	35	M	59000	5
4	59	M	12000	3
5	61	F	54000	2
6	65	F	25000	3
7	65	F	25000	3
8	70	F	30000	5
9	40	M	54000	2
10	32	F	15000	2
11	45	M	25000	1
12	75	F	10300	4

(β′) QIT table

Patient Data		
Group-ID	Disease	Count
1	lung cancer	1
1	pneumonia	1
2	strain	2
2	prostate cancer	2
3	strain	1
3	aids	1
4	flu	1
4	bladder cancer	1
5	flu	1
5	hepatitis	1

(γ′) ST table

Σχήμα 3.8: anatomized tables with $corr \leq 8/5$

Όμως δεν πρέπει να μετρήσουμε το utility μόνο ως προς το βαθμός γενίκευσης του Quasi Identifier. Η τοποθέτηση των sensitive values σε groups και πιθανόν μάλιστα γενικευμένα μειώνει το utility. Δίνουμε την ποινή κατασκευής ενός QI-group για μία εγγραφή:

Ορισμός 26. *Έστω ένας πίνακας T με την sensitive ιδιότητα S. θεωρούμε μία εγγραφή t της οποίας η τιμή της sensitive attribute είναι s και εισάγεται σε ένα QI-group όπου η sensitive attribute παίρνει και τις τιμές $s_1, ..., s_r$, ενώ η τιμή s έχει γενικευτεί στην τιμή s^*. Για την ιδιότητα S η ποινή normalized certainty penalty ορίζεται ως*

$$NCP_S(t) = \frac{\frac{1}{corr(s,s^*)} + \sum_{i=1}^{r} \frac{1}{corr(s,s_i)}}{r+1}$$

.

Την τιμή $NCP_S(t)/(d+1)$ θα την συμβολίζουμε ως $NCP(S)$

Μέχρι τώρα έχουμε υπολογίσει την συνολική ποινή για την ιδιότητα μίας εγγραφής, είτε αυτή είναι κατηγορική είτε αριθμητική. Πως όμως υπολογίζουμε την συνολική ποινή για μία εγγραφή; Η πιο απλή λύση είναι προσθέσουμε την ποινή που προκύπτει από την γενίκευση για κάθε ιδιότητα ξεχωριστά στην συγκεκριμένη εγγραφή. Μπορούμε όμως να θεωρήσουμε ότι κάθε ιδιότητα σε μία εγγραφή έχει διαφορετικό βάρος. Έτσι μπορούμε να αντιστοιχήσουμε σε κάθε ιδιότητα ένα βάρος. Δίνεται ο ακριβής ορισμός.

Ορισμός 27. *Έστω ένας πίνακας T με quasi-identifier $(X_1, ..., X_d)$ και sensitive attribute S, και μία εγγραφή $t = (x_1, ..., x_d, s)$ η οποία γενικεύεται στην εγγραφή $t' = (x'_1, ..., x'_d, s^*)$ και αντιστοιχίζεται στο QI-group $s_1, ..., s_r$. Για την εγγραφή t η ποινή normalized certainty penalty θα είναι:*

$$NCP_\tau(t) = \left\{ w_s NCP_S(t) + \sum_{i=1}^{d} (w_i NCP_{X_i}(t)) \right\} \frac{1}{n+1}.$$

όπου $w_s + \sum_{i=1}^{d} w_i = 1$

Αν δούμε το παραπάνω τύπο, μπορούμε να παρατηρήσουμε ότι ο πρώτος όρος του αθροίσματος επί της ουσίας κοιτάει πόσο κοντά είναι μεταξύ τους οι sensitive values και ο δεύτερος ελέγχει την περίμετρο. Με βάση το section όπου εξετάζαμε το utility του anatomy είναι προτιμότερο να δώσουμε βάρος στην μείωση του πρώτου όρου από ότι του δεύτερου.

Παράλληλα όμως είναι πιθανόν μαζί με τις πραγματικές εγγραφές, να εκδίδουμε και εγγραφές οι οποίες δεν υπάρχουν στην πραγματικότητα στον πίνακα. Αυτό συμβαίνει κυρίως για να μειώσουμε την πιθανότητα ενός εξωτερικού αντιπάλου να ανακαλύψει την sensitive attribute μίας πραγματικής εγγραφής.

Ορισμός 28. *Έστω ένας πίνακας T και μία όψη $T*$ του πίνακα. Για την εγγραφή t, η οποία υπάρχει στον πίνακα $T*$ αλλά όχι στον πίνακα T, η ποινή normalized certainty penalty ορίζεται ως:*

$$NCP_c(t) = 1 - c + cNCP_\tau(t).$$

όπου $c \leq 1$, μία σταθερά

Την σταθερά c, την επιλέγουμε πάντα εμείς, εξαρτάται από την εφαρμογή και το βάρος το οποίο επιθυμούμε να δώσουμε στο penalty μία εγγραφής. Αν θέλουμε να δώσουμε το μέγιστο penalty δίνουμε την τιμή 0, γεγονός που σημαίνει ότι μία ψεύτικη εγγραφή αντιστοιχίζεται σε μία πλήρως γενικευμένη τιμή, αν θεωρούμε ότι το ψέμα δεν επηρεάζει το utility μας τότε βάζουμε 1. Για την ακρίβεια, θεωρούμε ότι ο βαθμός γενίκευσης της εγγραφής, παίζει ρόλο ο οποίος είναι ανάλογος του c. Δεν είναι το ίδιο να έχεις μία εγγραφή σε ένα σημείο του χώρου και το ίδιο σε όλο τον χώρο. Το δεύτερο συνήθως είναι χειρότερο.

3.9 Correlation Algorithm

Σε αυτό το σημείο θα παρουσιάσουμε τους αλγορίθμους ώστε να εκδίδουμε μία (e, m)-cba όψη. Στόχος του αλγορίθμου εκτός από την ικανοποίηση του privacy θα είναι να έχουμε και όσο το δυνατόν μικρότερο penalty. Θεωρούμε ότι έχουμε μία μόνο sensitive attribute. Αυτή μπορεί να είναι είτε κατηγορική είτε αριθμητική. Θα δούμε μόνο την περίπτωση όπου είναι κατηγορική και θα περιγράψουμε ένα αλγόριθμο για την εύρεση μίας (e, m)-cba όψης.

Ο αλγόριθμος εκμεταλλεύεται τις εξής τρεις ιδιότητες.

Πρόταση 1. *Έστω ένας κόμβος u του δέντρου ιεραρχίας. Για οποιοσδήποτε δύο απογόνους u_1, u_2 του u ισχύει $corr(u_1, u_2) \geq \frac{|S|}{size(u)}$.*

Με άλλα λόγια αν $\frac{|S|}{size(u)} > e$ δύο ή περισσότεροι απόγονοι του u δεν μπορούνε να μπούνε στο ίδιο QI-group.

Η δεύτερη ιδιότητα είναι η εξής:

Πρόταση 2. *Έστω ένας κόμβος u του δέντρου ιεραρχίας. Έστω ένας οποιοσδήποτε απόγονος u_1 του u και ένας οποιοσδήποτε κόμβος u_2 ο οποίος δεν είναι απόγονος του u, τότε ισχύει $corr(u_1, u_2) < \frac{|S|}{size(u)}$.*

Με άλλα λόγια αν $\frac{|S|}{size(u)} < e$ οποιοσδήποτε μη-απόγονος του u μπορεί να μπει στο ίδιο QI-group με κάποιο απόγονο του u.

Πρόταση 3. *Έστω ένας κόμβος u του δέντρου ιεραρχίας και δύο άμεσα παιδιά u_1, u_2 του u., Τότε για κάθε δύο απογόνους v_1, v_2 του u_1, u_2 αντίστοιχα ισχύει $corr(v_1, v_2) = \frac{|S|}{size(u)}$.*

Με άλλα λόγια αν $\frac{|S|}{size(u)} \leq e$ τότε ο v_1, v_2 μπορούνε να υπάρξουν στο ίδιο group.

Ο αλγόριθμος έχει έξι βασικά στάδια, το division, balancing, generalization, assignment, gsa-split, de-generalization και sa-balancing split. Το πρώτο είναι ακριβώς το ίδιο με του m-invariance. Εμείς θα εξετάσουμε μόνο τα επόμενα.

Άρα εμείς σε πρώτη φάση θέλουμε τις νέες εγγραφές να τις χωρίσουμε σε buckets ώστε να ικανοποιείται το (e, m)-cba. Αρχικά θα περιγράψουμε ένα αλγόριθμο ο οποίος εξετάζει εάν μπορούμε να εκδώσουμε μία (e, m)-cba όψη ενός πίνακα T. Ο αλγόριθμος εκμεταλλεύεται τις δύο παραπάνω ιδιότητες και το γεγονός ότι για να μπορούμε να κατασκευάσουμε μία m-unique όψη ενός πίνακα πρέπει για κάθε τιμή u της sensitive attribute να ισχύει $|u| < |T|/m$, όπου $|u|$ ο πληθάριθμος της sensitive attribute ως προς την τιμή u και $|T|$ το μέγεθος του πίνακα. Μπορεί να παρατηρήσει κανείς ότι για να μπορούμε να εκδώσουμε μία anonymous όψη, αρκεί για κάθε φύλλο της ιεραρχίας να υπάρχει ένας πρόγονος u για τον οποίο να ισχύει:

- $\frac{|S|}{size(v)} \leq e$, όπου v ο πατέρας του u.
- $|u| \leq |T|/m$

Άρα ο αλγόριθμος μας απλά σκανάρει το δέντρο και εξασφαλίζει ότι για κάθε φύλλο μπορούμε να βρούμε ένα τέτοιο πρόγονο. Προσέξτε ότι τα φύλλα αυτά μπορούμε να τα υπολογίσουμε μόνο μία φορά και θα ονομάζονται μάλιστα gsa nodes. Είναι εύκολο να δούμε ότι ο κοντινότερος κόμβος που ικανοποιεί την συνθήκη του correlation είναι πάντα ο ίδιος.

Ορισμός 1. *Έστω ένα δέντρο ιεραρχίας. Ονομάζουμε gsa-nodes τους κόμβους $u_1, ..., u_g$ για τους οποίους ισχύει:*

- $\frac{|S|}{size(v_i)} \leq e$, *όπου v_i ο πατέρας του u_i για κάθε $1 \leq i \leq g$.*
- *και για κάθε $1 \leq i \leq g$ έτσι ώστε $\frac{|S|}{size(u_i)} \leq e$ ο u_i δεν έχει κάποιο απόγονο*

Θα συμβολίζουμε με $gsa(u)$ τον gsa node της τιμή u.

Η πρώτη συνθήκη λοιπόν μας εξασφαλίζει ότι η gsa-nodes μπορούνε να μπούνε στο ίδιο QI-group. Η δεύτερη συνθήκη μας εξασφαλίζει ότι αυτοί οι κόμβοι είναι ελάχιστοι. Για να βρούμε τους gsa-nodes αρκεί να εκτελέσουμε τον παρακάτω αλγόριθμο:

Algorithm 8 Bottom-up method

```
INPUT : a tree t, parameter e
OUTPUT : a list of nodes
S = leaves; result = ∅
while S ≠ ∅ do
  pop lev from S
  find the nearest ancestor u of lev such as parent v of u has corr ≤ e
  result = result+(e,u);
  find all leaves L = (lev_1, ..., lev_m) such as v is their ancestor
  result=result+L;
  remove L from S
end while
return result
```

Η εύρεση του κοντινότερου προγόνου είναι μία bottom-up αναζήτηση Στο τέλος επιστρέφουμε απλά μία λίστα από ζευγάρια. Κάθε ζευγάρι είναι ένας κόμβος και των σύνολο των φύλλων τα οποία γενικεύονται σε αυτό τον κόμβο.

Άρα για να ελέγξουμε αν ένας πίνακας μπορεί να παράγει μία cba-view αρκεί να ικανοποιήσουμε την παρακάτω πρόταση:

Πρόταση 4. *Για να μπορεί ένας πίνακας να μας δώσει μία* $(e, m)-$*cba view, δοθέντος μίας ιεραρχίας στην ευαίσθητη ιδιότητα, αρκεί για τους gsa-nodes* $u_1, ..., u_g$ *να ισχύει:*

$$|u_i| \leq |T|/m \forall i$$

.

Με βάση όλα τα παραπάνω είμαστε σε θέση να περιγράψουμε τον αλγόριθμο.

Division Όπως προαναφέρθηκε αυτό το στάδιο είναι ακριβώς το ίδιο με αυτό που παρουσιάστηκε στο κεφάλαιο του m-invariance.

Balancing Στο προηγούμενο κεφάλαιο είχαμε δει ότι ο αλγόριθμος του m-invariance δεν μας δίνει πάντα λύση ενώ είναι πιθανό να υπάρχει. Σε αυτό το στάδιο θα εφαρμόσουμε μία παραλλαγή του σταδίου εκείνου του αλγορίθμου ώστε να έχουμε πάντα λύση αν υπάρχει.

Στο m-invariance εκτελούσαμε μία σειρά από επαναλήψεις ώστε να κάνουμε balanced τα buckets με την προϋποθεση να η αφαίρεση μίας εγγραφής να μην οδηγεί σε ένα μη m-eligible S_-. Αντίθετα εδώ θα εκτελέσουμε την εξής τακτική. Θα κατασκευάσουμε πάλι το σύνολο S_-, αλλή αυτή την φορά θα αφαιρούμε τις εγγραφές με διαφορετική τακτική.

Πιο συγκεκριμένα, μπορούμε να παρατηρήσουμε το εξής. Έστω ότι στο S_- εμφανίζονται οι gsa nodes $gs_1, ..., gs_n$. Θεωρούμε ότι μάλιστα αυτοί ότι είναι ταξινομημένοι με δύο κριτήρια. Το πρώτο είναι ως προς την συχνότητα εμφάνισης και εν συνεχεία αν δύο κόμβοι έχουν την ίδια συχνότητα εμφάνισης, τότε θα θεωρούμε ότι πρώτος θα μπει ο κόμβος του οποίου η τιμή δεν λείπει σε κάποιο unbalanced bucket. Αν και οι δύο λείπουνε ή δεν λείπουνε από κάποιο bucket τότε η σειρά τους είναι τυχαία. υπάρχουν λοιπόν δύο ενδεχόμενα:

- Ο S_- να μπορεί να μας δώσει κάποια (e, m)-cba view. Τότε μπορούμε να αφαιρούμε συνέχεια τυχαία (ή με κάποιο άλλο κριτήριο) εγγραφές έτσι ώστε να ισχύει για το πιο συχνά εμφανιζόμενο gsa node gs_1:

  $$|gs_1| \leq |T|/m$$

  Επί της ουσίας μέχρι εδώ εφαρμόζουμε τον αλγόριθμο του m-invariance. Αν έχουμε αφαιρέσει όσες εγγραφές μπορούμε, τότε αποθηκεύουμε την λύση ως την τελευταία λύση που βρήκαμε και αφαιρούμε μία εγγραφή η οποία να έχει gsa node gs_1. Και εν συνεχεία ταξινομούμε και εξετάζουμε ξανά τα ενδεχόμενα.

- Εδώ απλά αφαιρούμε μία εγγραφή η οποία να έχει gsa node gs_1. Και εν συνεχεία ταξινομούμε και εξετάζουμε ξανά τα ενδεχόμενα όπως έγινε και στο τέλος του προηγούμενου σταδίου.

Η τελευταία λύση την οποία αποθηκεύσαμε είναι η λύση. Η λύση αυτή προσέξτε ότι μπορεί να μην είναι μοναδική αλλά μας δίνει αρκετά μεγαλύτερη γκάμα αποτελεσμάτων από αυτή που προτάθηκε στον αλγόριθμο του m-invariance. Μάλιστα αν στο πρώτο ενδεχόμενο δεν επιλέγουμε τυχαία τις εγγραφές αλλά τις πιο συχνά εμφανιζόμενες τότε θα ελαχιστοποιήσουμε το πλήθος των counterfeits, ο οποίος είναι και ο στόχος μας.

Generalization. Σε κάθε εγγραφή t προσθέτουμε ακόμα μία ιδιότητα, την generalized sensitive attribute (gsa). Αν η τιμή της sensitive attribute είναι u, η τιμή της gsa είναι ο προγόνους u^* για του οποίου τον πατέρα ισχύει $corr \leq e$ και παράλληλα το $corr$ είναι ελάχιστο. Με άλλα λόγια κάθε sensitive value αντικαθίσταται από τον κοντινότερο πρόγονο στο δέντρο ιεραρχίας με την προϋπόθεση ο συσχετισμός να είναι το πολύ e. Προσέξτε ότι οι τιμές της gsa είναι επί της ουσίας οι gsa-nodes του προηγούμενου σταδίου και δεν αναγκαίο να υπολογιστούν ξανά.

Assignment. Σε αυτό το βήμα επιλέγουμε να χωρίσουμε τα δεδομένα σε QI-groups τα οποία να ικανοποιούν το l-diversity. Μπορούμε να εκτελέσουμε οποιοδήποτε γνωστό αλγόριθμο θεωρώντας όμως ως sensitive attribute την gsa. Επί της ουσίας είναι το γνωστό στάδιο του m-invariance μόνο που τώρα γίνεται με βάση την gsa.

GSA-split. Αυτό το στάδιο είναι ακριβώς το ίδιο με του m-invariance μόνο που το split γίνεται με βάση το gsa.

De-Generalization. Αντικαθιστούμε τις τιμές των gsa στα QI-groups με τις κανονικές τιμές της sensitive attribute.

SA-balancing split. Αυτό το στάδιο είναι ακριβώς το ίδιο με του m-invariance δηλαδή το split γίνεται με βάση την sensitive attribute.

Σε πρώτη φάση, μπορεί να αναρωτηθεί κανείς γιατί κάνουμε δύο φορές split. Στο πρώτο split δεν μπορούμε δυστυχώς να λάβουμε υπόψη μας τις παλιές εγγραφές, αφού για αυτές δεν υπάρχει gsa. Πρέπει όμως κατά κάποιο τρόπο να κατασκευαστούν QI-groups, ώστε να έχουμε μη γενικευμένα signatures. Επειδή σε αυτό το στάδιο έχουμε σχετικά μεγάλα bucket μας συμφέρει να κάνουμε τον έλεγχο με βάση την περίμετρο.

Αντίθετα το τελευταίο split του αλγορίθμου δεν είναι απαραίτητο και μπορεί να γίνεται ευριστικά. Πιο συγκεκριμένα, αν παρατηρήσουμε ότι η περίμετρος κάποιου QI-group (από αυτά που ήδη υπάρχουν) είναι πολύ μεγαλύτερη εν συγκρίσει με των καινούργιων, τότε θα επιλέξουμε να εκτελέσουμε αυτό το στάδιο.

Τέλος να σημειώσουμε ότι τα στάδια του generalization και του de-generalization του αλγορίθμου δεν επηρεάζουν καθόλου το utility. Για την ακρίβεια ναι μεν, επηρεάζουν το utility αποκλείοντας τον συσχετισμό κάποιων εγγραφών, αλλά για πετύχουμε μία (e, m)-cba όψη δεν γίνεται να επιτύχουμε μεγαλύτερο utility. Η αιτία είναι η εξής. Έστω μία εγγραφή η οποία έχει την τιμή u_t στην sensitive attribute. Τότε αυτή η εγγραφή δεν θα μπορεί να μπει στο ίδιο QI-group με οποιαδήποτε άλλη εγγραφή η οποία έχει την ίδια τιμή στην gsa με βάση την πρώτη ιδιότητα. Αντίθετα με βάση την ιδιότητα 2 μπορεί να μπει στο QI-group με οποιαδήποτε άλλη εγγραφή με διαφορετική τιμή στην gsa.

3.9.1 Assignment

Το στάδιο όμως του assignment μπορεί να επηρεάσει το utility και εξαρτάται από τον αλγόριθμο τον οποίο θα επιλέξουμε. Ένας αλγόριθμος ο οποίος μπορεί να χρησιμοποιηθεί είναι αυτός του anatomy ο οποίος στοχεύει στο να έχουμε όσο το δυνατόν μικρότερα QI-groups. Ένας δεύτερος αλγόριθμος ο οποίος μπορεί να χρησιμοποιηθεί είναι αυτός του σταδίου του assignment του m-invariance. Ο τελευταίος προσπαθεί να διατηρήσει όσο το δυνατόν μικρότερα QI-groups, αλλά παράλληλα προσπαθεί να συσχετίσει εγγραφές οι οποίες είναι κοντά μεταξύ τους ως προς το QI.

Ένα τρίτος αλγόριθμος ο οποίος μπορεί να χρησιμοποιηθεί στο στάδιο του anonymization έχει ως στόχο να διατηρήσει μικρά QI-groups, με κοντά μεταξύ τους εγγραφές με βάση την sensitive attribute.

Πριν δώσουμε την διαδικασία, θα δώσουμε ένα ορισμό:

Ορισμός 29. *Έστω ένα δέντρο t και ένας κόμβος u του δέντρου. Ορίζουμε ως άμεσα παιδιά του u, τους απογόνους του των οποίων το μονοπάτι από το u έχει μήκος ίσο με 1.*

Ο αλγόριθμος αυτός κάνει το εξής, δοθέντος του δέντρου ιεραρχίας για την sensitive attribute αναζητά τα δυνατότερα μικρά υποδέντρα για τα οποία να μπορούμε να έχουμε μία m-unique όψη. Πιο συγκεκριμένα, έστω ένας κόμβος v του δέντρου, με άμεσα παιδιά τα $u_1, ..., u_n$. Με βάση αυτά τα παιδιά μπορούμε να χωρίσουμε την βάση T στα υποσύνολα $T_1, ..., T_n$. Το πρόβλημα μας είναι για κάθε ένα από αυτά τα υποσύνολα να μπορούμε να πάρουμε μία m-unique view. Αν αυτό είναι εφικτό τότε όντως μπορούμε να χωρίσουμε τη βάση μας σε αυτά τα υποσύνολα.

Ο αλγόριθμος είναι απλός. Έστω ένας κόμβος v με άμεσα παιδιά $u_1, ..., u_n$ που ορίζουν ένα partition $(T_1, ..., T_n)$. Έστω ότι το παιδί u_i έχει πλήθος εγγραφών στην βάση $|u_i|$ και έχει ως φύλλα τους κόμβους $u_{i_1}, ..., u_{i_m}$ τότε για να είναι το partition αποδεκτό (δηλαδή να μας δίνει m-unique view πρέπει να ισχύει για κάθε φύλλο του παιδιού u_i η σχέση $|u_{i_j}| < |u_i|/m$. Αν αυτό ισχύει για κάθε παιδί του v τότε το partition είναι αποδεκτό. Ο αλγόριθμος επαναλαμβάνεται για κάθε στοιχείο του partition.

Να σημειώσουμε ότι ως φύλλα θεωρούνται οι gsa κόμβοι. Ο αλγόριθμος μπορεί να είναι bottom-up ξεκινώντας από φύλλο- φύλλο και κατασκευάζοντας το υποδέντρο. Σε αυτή την περίπτωση εκμεταλλευόμαστε την εξής ιδιότητα.

Πρόταση 5. *Εάν ένα φύλλο ικανοποιεί το l-diversity για ένα υποδέντρο με ρίζα το u τότε το ικανοποιεί για κάθε υποδέντρο με ρίζα v όπου το v είναι πρόγονος του u.*

Έτσι ανεβαίνοντας προς την κορυφή αν παρατηρήσουμε ότι ένα φύλλο ικανοποιεί το l-diversity δεν χρειάζεται να το ξαναελέγξουμε. Έτσι ο αλγόριθμος είναι ένας bottom up αλγόριθμος και ξεκινάει από ένα ένα φύλλο και αναζητά τον πρώτο κόμβο που ικανοποιεί το l-diversity, εν συνεχεία όλα τα φύλλα αυτού του κόμβου έχουν ελεγχθεί και δεν θα ξαναελεγχθούν. Η διαδικασία επαναλαμβάνεται για όσα φύλλα δεν έχουν ελεγχθεί. Δίνεται ο αλγόριθμος:

Algorithm 9 Bottom-up method

```
INPUT : a tree t, parameter l
OUTPUT : a list of nodes
S = leaves; result = ∅;
while S ≠ ∅ do
  pop e from S
  find the nearest ancestor u of e
  ul = find-leaves(u);
  remove ul from S
  if u satisfies l-diversity then
    push u to result;
    remove all children of u from result
  else
    push u to S
  end if
end while
return result
```

Αν μάλιστα έχουμε ταξινομήσει την βάση ως προς την sensitive attribute ξεκινώντας από το πιο αριστερό παιδί και πηγαίνοντας προς το δεξιότερο, τότε τα subsets θα είναι και αυτά ταξινομημένα και μάλιστα θα είναι διαδοχικά στην βάση μας.

Προσέξτε επίσης ότι ο αλγόριθμος δεν χρειάζεται όταν αλλάζει ένα κόμβο στην ιεραρχία να ελέγχει ξανά όλο το υποδέντρο, το οποίο βρίσκεται κάτω από αυτόν. Έστω ότι σε κάποια φάση είμαστε στον κόμβο u_i, ο οποίος έχει πατέρα τον v, ο οποίος έχει άμεσα παιδιά τα $u_1, ..., u_n$. Αν ο αλγόριθμος μας ξεκινάει από τα αριστερά προς τα δεξιά, τότε οι κόμβοι $u_1, ..., u_{i-1}$ δεν έχουν λόγο να προσπελαστούν, αφού οι τιμές αυτών με βάση την πρόταση 4 σίγουρα ικανοποιούν το l-diversity. Αντίθετα οι κόμβοι δεξιά του u_i δεν έχουν προσπελαστεί. Έτσι αν ο κόμβος u_i δώσει την γνώση του στον κόμβο u θα προσπελαστούν μόνοι οι υπόλοιποι κόμβοι. Έτσι η πολυπλοκότητα του αλγορίθμου μας είναι $O(k)$, όπου k το πλήθος των κόμβων του δέντρου.

Εν συνεχεία αφού έχουμε χωρίσει την βάση μας σε μικρότερες υποβάσεις απλά για κάθε υποβάση εφαρμόζουμε το στάδιο του assignment όπως αυτό προτάθηκε στο m-invariance προσπαθώντας να μειώσουμε την περίμετρο.

Ο αλγόριθμος είναι ευριστικός και στηρίζεται στο γεγονός ότι αν κατασκευάσουμε QI-groups στα οποία οι τιμές της sensitive attribute έχουν μεγάλο βαθμό συσχετισμού αυξάνουμε το utility.

3.10 Frequency για Δυναμικά Δεδομένα

Με βάση τον ορισμό του m-invariance, όταν μία εγγραφή διαγραφεί από ένα QI-group τότε θα πρέπει ή κάποια άλλη να μπει στην θέση της, ή να αντικατασταθεί με μία ψεύτικη. Το πρόβλημα έγκειται στο γεγονός ότι για να αντικατασταθεί από μία άλλη, τότε θα πρέπει να έρθει μία εγγραφή με την ίδια τιμή στην sensitive attribute, η κάποια τιμή η οποία να ανήκει στο σταθερό μονοπάτι της προηγούμενης τιμής. Αν όμως σε κάποια χρονική στιγμή στην βάση, οι διαγραφές ως προς μία τιμή της sensitive attribute είναι πιο συχνές από τις εισαγωγές ως προς την ίδια τιμή, τότε θα δημιουργηθούν αρκετά counterfeits, τα οποία θα οδηγήσουν σε μεγάλη μείωση του utility.

Μπορεί να παρατηρήσει όμως κανείς ότι αν γενικεύσουμε την sensitive attribute στην πλειοψηφία των περιπτώσεων είναι πιο πιθανόν να μας έρθει η νέα τιμή, αφού πια απευθυνόμαστε σε μεγαλύτερο εύρος δυνατών τιμών και παράλληλα σε μεγαλύτερο πλήθος σταθερών μονοπατιών. Επίσης μπορεί κανείς να παρατηρήσει, ότι μπορεί μία τιμή μία εγγραφής ως προς την sensitive attribute να την γενικεύσουμε όποια στιγμή εμείς επιθυμούμε, χωρίς να παραβιάζουμε το privacy. Δεν ισχύει το ίδιο αν θέλουμε να κατέβουμε επίπεδο στην ιεραρχία.

Πιο συγκεκριμένα ας εξετάσουμε μία προς μία τις δυνατές περιπτώσεις:

- Γενικεύουμε την sensitive attribute την χρονική στιγμή της διαγραφής. Δηλαδή, όταν μία εγγραφή διαγραφεί από την βάση, τότε γενικεύουμε την τιμή της, ώστε να αντικατασταθεί από κάποια άλλη εγγραφή, της οποία η τιμή να ανήκει στο υποδέντρο το οποίο ορίζει η νέα γενικευμένη τιμή. Αυτή η τακτική, αν και η πιο εύκολη και με το μικρότερο κόστος, δεν διασφαλίζει το privacy. Ο λόγος είναι, ότι αν ο αντίπαλος ξέρει πότε μία εγγραφή διαγράφεται και δει και σε ένα QI-group κάποια τιμή να γενικεύεται αμέσως μετά την διαγραφή τότε θα έχει ανακαλύψει την τιμή της sensitive attribute για την διαγραφείσα εγγραφή.

- Γενικεύουμε την sensitive attribute ανεξάρτητα με την χρονική στιγμή όπου θα γίνει η διαγραφή. Αυτό είναι μία αποδεκτή λύση, αφού επί της ουσίας ακόμα και αν γίνει κάποια στιγμή διαγραφή και εισαγωγή μίας εγγραφής, αυτό δεν επιτρέπει στον χρήστη να εξάγει κάποια συμπέρασμα.

- Ξε-γενικεύουμε την sensitive attribute σε τυχαία χρονική στιγμή. Η λύση αυτή είναι αποδεκτή μόνο αν κατέβουμε σε μονοπάτι που περιλάμβανε την προηγούμενη τιμή. Δη-

λαδή για παράδειγμα έστω ένα δέντρο ιεραρχίας, ένας κόμβος u αυτού και δύο παιδιά του v_1, v_2. Αν κάποια στιγμή γενικευτεί η τιμή v_1 στην u, αργότερα (σε περίπτωση όπου μία διαγραφή αυτής αντικατασταθεί από μία νέα) δεν γίνεται να κατέβουμε στην v_2.

Το πρόβλημα μας λοιπόν είναι με ποιο κριτήριο θα γενικεύουμε και με ποιο θα ξε-γενικεύουμε. Για το συγκεκριμένο πρόβλημα μπορούμε να εφαρμόσουμε δύο κριτήρια, το πρώτο είναι το κριτήριο της balanced μεταβολής του utility και το δεύτερο το κριτήριο του μέγιστου utility στην επόμενη όψη.

3.10.1 Balanced utility

Υπενθυμίζουμε ότι το πρόβλημα είναι να εξετάσουμε πότε μας συμφέρει να γενικεύσουμε και πότε όχι. Εξετάζοντας το πρώτο, μπορούμε να ισχυριστούμε ότι επιθυμούμε το σφάλμα στην βάση μας να είναι πάντα balanced, δηλαδή το συνολικό penalty μεταξύ διαφορών χρονικών στιγμών να παραμένει σταθερό ή καλύτερα να μην εξαρτάται από το πλήθος των counterfeits. Αν λοιπόν το penalty για μίας εγγραφής t στην έκδοση T_i της βάσης είναι $NCP_\tau(t)$ και η πιθανότητα να εμφανιστεί counterfeit για αυτή την εγγραφή είναι p_g, τότε για να μην έχουμε μεταβολή στο σφάλμα πρέπει $p_g(NCP_c(t) - NCP_\tau(t) = 0$. Αυτό σημαίνει ότι μία τιμή πρέπει να γενικευτεί τόσο, ώστε να έχει penalty ίσο με το counterfeit. Προφανέστατα αυτό είναι αρκετά περιοριστικό και δεν μας εξασφαλίζει σε περίπτωση που οι εισαγωγές αυξηθούν.

Αντίθετα μία καλύτερη λύση είναι να διατηρούμε την μεταβολή του penalty balanced. Με αλλά λόγια αν την χρονική στιγμή i επιλέξαμε να γενικεύσουμε την τιμή sensitive attribute ώστε να μειώσουμε τα counterfeits, απαιτούμε η μεταβολή του σφάλματος την επόμενη χρονική στιγμή ναι είναι η ίδια. Με άλλα λόγια για μία εγγραφή να ισχύει:

$$p_g(NCP_c(t^*) - NCP_\tau(t^*)) - (NCP_\tau(t^*) - NCP_\tau(t)) = 0.$$

όπου $NCP_\tau(t)$ το penalty της εγγραφής μας με βάση κάποια όψη την οποία έχουμε ως βάση, $NCP_\tau(t^*)$ το penalty της εγγραφής μας την χρονική στιγμή όπου εκδίδουμε την βάση, $NCP_c(t^*)$ το penalty μίας counterfeit εγγραφής, αφού την έχουμε γενικεύσει και p_g η πιθανότητα εμφάνισης counterfeit, αφού έχουμε γενικεύσει την τιμή .

Στην πραγματικότητα είναι αρκετά δύσκολο να κρατήσουμε αυτή την διαφορά πάντα ίση με μηδέν. Μπορούμε να επιλέξουμε να υπάρχει κάποιο όριο. Έτσι μπορούμε να απαιτήσουμε:

$$p_g(NCP_c(t^*) - NCP_\tau(t^*)) - (NCP_\tau(t^*) - NCP_\tau(t)) \leq f.$$

όπου f μία σταθερά επιλεγμένη από τον χρήστη. Μάλιστα αν θεωρήσουμε ότι δεν θα γενικεύσουμε καθόλου το quasi identifier ο τύπος μπορεί να γραφτεί ισοδύναμα:

$$p_g NCP_c(t^*) - p_g NCP_\tau(t^*) - \frac{\Delta NCP_S(t^*,t)}{|attr|} \leq f.$$

όπου $\Delta NCP_S(t^*, t)$ η μεταβολή του κόστους γενίκευσης της sensitive attribute και $|attr|$ το πλήθος των ιδιοτήτων στις οποίες εφαρμόζεται ο τύπος του penalty.

Μπορεί να παρατηρήσει από τα αθροίσματα του τύπου κάποια χαρακτηριστικά. Η πιθανότητα να εμφανιστεί ένα counterfeit είναι αρκετά σημαντική και επί της ουσίας εκφράζει την ανάγκη μας να μην αλλάξει αρκετά η κατανομή κάποιας τιμής της sensitive attribute στην βάση μας. Τέλος με βάση τον τύπο μας μπορεί να παρατηρήσει ότι όσο περισσότερο γενικεύει κανείς μία εγγραφή, και μάλιστα ιδιαίτερα ως προς την sensitive attribute τόσο πιο balanced παραμένει η μεταβολή του σφάλματος.

Στον αρχικό μας τύπο είχαμε θεωρήσει ότι η μεταβολή του σφάλματος σε δύο χρονικές στιγμές πρέπει να είναι ίση. Στην πραγματικότητα μπορούμε να απαιτήσουμε ότι υπάρχει μία σταθερή αναλογία, δηλαδή ο χρήστης να επιλέξει μία σταθερά b έτσι ώστε:

$$b(p_g NCP_c(t^*) - p_g NCP_\tau(t^*)) - \frac{\Delta NCP_S(t^*,t)}{|attr|} \leq f$$

Μάλιστα ανεξαρτήτως της τιμής της σταθεράς b το πρόβλημα έχει πάντα λύση. Αν γενικεύσουμε όλες τις ιδιότητες έτσι ώστε το penalty κάθε μίας από αυτής να είναι ίσο με 1, τότε ο όρος $b(p_g NCP(c) - p_g NCP(t^*)) < 0$ και άρα η ανίσωση ισχύει. Αν μάλιστα η σταθερά πάρει κάποια τιμή $b = \frac{v}{|attr|}, v \leq 1$, όπου v επίσης μία σταθερά, τότε η ανίσωση ισχύει αν γενικεύσουμε μόνο την ευαίσθητη ιδιότητα έτσι ώστε το penalty να είναι ίσο με 1. Μάλιστα σε αυτή την περίπτωση ο τύπος γίνεται:

$$v(p_g NCP_c(t^*) - p_g NCP_\tau(t^*)) - \Delta NCP_S(t *' t) \leq f$$

η ισοδύναμα:

$$v(p_g NCP_c(t^*) - p_g \frac{NCP_{qi}(t^*)}{|attr|}) - (p_g/|attr| + 1)\Delta NCP_S(t^*, t) \leq f$$

Το πλεονέκτημα αυτού του τύπου, είναι ότι δίνει μεγαλύτερη αξία στην τιμή της sensitive attribute καθώς και στον πλήθος των ιδιοτήτων. Εξ' αρχής εξάλλου το πρόβλημα μας ήταν να βρούμε πότε να γενικεύουμε μία ευαίσθητη ιδιότητα ώστε να μειώσουμε το πλήθος των counterfeits.

Τέλος αν θεωρήσουμε, για ευκολία ότι η χρονική στιγμή την οποία επιλέγουμε ως βάση είναι σταθερή και είναι η χρονική στιγμή 0, τότε ο τύπος γράφεται:

$$v(p_g NCP_c(t^*) - p_g \frac{NCP_{qi}(t^*)}{|attr|}) - (p_g/|attr| + 1)NCP_S(t^*) \leq f$$

Προσέξτε ότι ο παραπάνω τύπος καθώς και όλες οι ιδιότητες ισχύουν ανεξάρτητα το πως θα μετρήσει κανείς το utility για το counterfeit. Η μόνη προϋπόθεση είναι να τα penalty κανονικοποιημένα από 0 έως 1.

Μέχρι τώρα είδαμε πότε μπορούμε να γενικεύουμε εγγραφές. Το πρόβλημα μας όμως είναι πότε μπορούμε να επαναλάβουμε την ανάποδη διαδικασία. Πρώτον απαιτούμε να ισχύει ο τύπος όπου είχαμε βρει και πριν. Εκτός όμως από αυτόν τον τύπο, για να ξε-γενικεύσουμε μία εγγραφή, δεν πρέπει να παραβιάζουμε και το privacy. Θυμίζουμε ότι μία εγγραφή, πρέπει πάντα να έχει similar signature και μάλιστα όχι μόνο με τον εαυτό της αλλά με όσες εγγραφές έχει συσχετιστεί σε κάποιο QI-group. Έτσι αν κάποια στιγμή επιθυμούμε να ξε-γενικεύσουμε μία εγγραφή, απλά δεν δίνουμε την τιμή του φύλλου, αλλά του κοντινότερου προγόνου σε αυτό, ώστε να μην έχουμε privacy breach.

3.10.2 Minimal Penalty

Το κριτήριο της balanced μεταβολής έχει το μειονέκτημα ότι δεν ελέγχει πως θα ελαχιστοποιήσει το penalty αλλά πως θα διατηρήσει την μεταβολή του σταθερή στον χρόνο (αν βέβαια αυτή αυξάνει). Αντίθετα το επόμενο κριτήριο έχει ως στόχο το ελάχιστο penalty.

Έστω ότι εμείς θέλουμε να δούμε αν μας συμφέρει να γενικεύσουμε μία εγγραφή t, η ανισότητα την οποία ελέγχουμε είναι η εξής:

$$p_g NCP_c(t^*) + (1 - p_g)NCP_\tau(t^*) \leq p_g NCP_c(t) + (1 - p_g)NCP_\tau(t)$$

με αλλά λόγια απαιτούμε μία γενίκευση να οδηγήσει σε μικρότερο κόστος στην επόμενη όψη.

Το πρόβλημα αυτού του κριτηρίου, είναι ότι δεν το ενδιαφέρει τι συμβαίνει στην υπάρχουσα όψη και το γεγονός ότι μία γενίκευση μειώνει αρκετά το utility σε αυτή την όψη.

Το κριτήριο αυτό σε αντίθεση με το προηγούμενο έχει σίγουρα λύση αν δεν γενικεύσουμε την ευαίσθητη ιδιότητα. Μάλιστα αν $p_g > p$ τότε η ανίσωση δεν ισχύει. Μάλιστα αν θεωρήσουμε ότι γενικεύουμε ένα φύλλο, τότε ο τύπος παίρνει την μορφή:

$$p_g NCP_c(t^*) + (1 - p_g) NCP_\tau(t^*) \leq pNCP_c(t) + (1 - p) NCP_{qi}(t)/|attr|$$

Το κριτήριο της ξε-γενίκευσης, έχει την ίδια λογική με το κριτήριο του balanced. Κοιτάμε για ποιο κόμβο γενίκευσης πρέπει μας συμφέρει να κάνουμε ξε-γενίκευση και εν συνεχεία, απλά, ελέγχουμε να διατηρείτε η similar signature.

3.10.3 Balanced utility και Minimal Penalty

Τέλος μπορούμε να εφαρμόσουμε ένα συνδυασμό των δύο κριτηρίων. Δηλαδή αναζητάμε ένα κόμβο που να ικανοποιεί και τα δύο κριτήρια:

$$v(p_g NCP_c(t^*) - p_g \tfrac{NCP_{qi}(t^*)}{|attr|}) - (p_g/|attr| + 1) NCP_S(t^*) \leq f$$

και

$$p_g NCP_c(t^*) + (1 - p_g) NCP_\tau(t^*) \leq pNCP_c(t) + (1 - p) NCP_{qi}(t)/|attr|$$

Παίρνοντας τους τύπος μάλιστα που είχαμε ορίσει στο utility, για το ςουντερφειτ, οι τελευταίοι τύποι γίνονται:

$$vp_g(1 - c)\tfrac{NCP_{qi}(t^*}{|attr|} + ((p_g - c)/|attr| + 1) NCP_S(t^*) \geq vp_g(1 - c) - f$$

και

$$(1 - p_g + cp_g) NCP_S(t^*) + [(1 - p_g + cp_g) - (1 - p + cp)]\, NCP_{qi}(t) \leq (1 - c)|attr|(p - p_g)$$

Μπορεί να προσέξει ότι το κριτήριο του balanced utility μας εισάγει ένα lower bound ενώ το κριτήριο του minimal penalty μας εισάγει ένα upper bound. Το αποτέλεσμα αυτό είναι αναμενόμενο, αφού εμείς επιθυμούμε ένα balanced error χωρίς όμως να το αυξήσουμε ιδιαίτερο. Με άλλα λόγια προσπαθούμε να διατηρήσουμε το σφάλμα σταθερό, εμποδίζοντας όμως πιθανές απότομες αυξήσεις του.

Μάλιστα, αν θεωρήσουμε ότι $NCP_{qi}(t) \simeq 0$ και ότι $NCP_S(t^*) \simeq \frac{NCP(S)+(m-1)e^{-1}}{m}$ οι τύποι γίνονται:

$$\tfrac{vp_g(1-c)-f}{(1-vc)\frac{p_g}{|attr|}+1} m - (m-1)e^{-1} \leq NCP(S) \leq \tfrac{(1-c)|attr|(p-p_g)}{1-p_g+cp_g} m - (m-1)e^{-1}$$

Το πρόβλημα του συνδυασμένου κριτηρίου είναι ότι δεν έχει πάντα λύση. Δεν γίνεται δηλαδή κάποιος να μας εξασφαλίσει ότι θα βρεθεί σίγουρα λύση. Αυτό συμβαίνει συνήθως όταν θα έχουμε κάποια σημαντική αύξηση των counterfeits στην επόμενη όψη, οπότε δεν μπορούμε κατά κάποιο τρόπο να την εμποδίσουμε και άρα επί της ουσίας την αγνοούμε. Αυτό μεταφράζεται ότι θα εφαρμόζουμε οπότε μπορούμε το κριτήριο και όχι πάντα. Επίσης προσέξτε ότι η διπλή ανίσωση μας περιορίζει ιδιαίτερα στην άνοδο προς τα πάνω και αυτό είναι κάτι επιθυμητό, αφού η ξε-γενίκευση είναι αρκετά δύσκολο να υλοποιηθεί.

Τέλος επειδή η ξε-γενίκευση είναι αρκετά δύσκολη, θα γίνουμε ακόμα πιο περιοριστικοί απαιτώντας να ισχύει και

$$NCP(S) < e^{-1}$$

όπου e η γνωστή σταθερά του (e, m)-cba.

3.11 Frequency algorithm

Με βάση λοιπόν τα δύο παραπάνω κριτήρια, στον ήδη υπάρχοντα αλγόριθμο εισάγουμε ακόμα δύο στάδια. Στο πρώτο στάδιο, απλά εξετάζουμε ποιες εγγραφές πρέπει να γενικεύσουμε με βάση κάποιο κριτήριο (όπως ένα από τα δύο που προτάθηκαν παραπάνω). Στο δεύτερο εξετάζουμε ποιες εγγραφές πρέπει να ξεγενικεύσουμε με βάση τα παραπάνω κριτήρια (και με την προϋπόθεση βέβαια να μην έχουμε breach privacy.

Τα στάδια αυτά θα εκτελεστούν αμέσως πριν το στάδιο SA-split που παρουσιάστηκε στο correlation algorithm. Ο λόγος είναι ότι πια αλλάζει η συνθήκη με βάση την οποία γίνεται το split.

Η διαδικασία είναι απλή. Σε πρώτη φάση πρέπει να βρούμε τα φύλλα σε ποιους κόμβους αντιστοιχούν. Το βασικό μας πρόβλημα είναι ότι για κάθε φύλλο έχουμε άλλη πιθανότητα p. Για αυτό θα ακολουθήσουμε την εξής διαδικασία. Σε πρώτη φάση θα αφαιρέσουμε από το δέντρο τις ακμές για τις οποίες δεν ισχύει $NCP(S) < e^{-1}$. Επί της ουσίας με αυτό το τρόπο θα δημιουργήσουμε ένα σύνολο από νέα υποδέντρα. Κατά την διάρκεια που κάνουμε αυτό cutting μπορούμε να υπολογίσουμε ποιοι κόμβοι ικανοποιούν τον τύπο $\frac{vp_g(1-c)-f}{(1-vc)\frac{p_g}{|attr|}+1}m - (m-1)e^{-1}$ και επί της ουσίας να αφαιρέσουμε και άλλους κόμβους. Τέλος προσέξτε ότι το δεύτερο Με αυτό τον τρόπο θα έχουμε κρατήσει ένα πολύ μικρό κομμάτι του αρχικού μας δέντρου σε ένα σύνολο από δέντρα. Παράλληλα αν παρατηρήσει κανείς τον τύπο

$$NCP(S) \leq \frac{(1-c)|attr|(p-p_g)}{1-p_g+cp_g}m - (m-1)e^{-1}$$

μπορεί να δει ότι γράφεται ισοδύναμα:

$$(NCP(S) + (m-1)e^{-1})\frac{1-p_g+cp_g}{(1-c)|attr|m} + p_g \leq p$$

Έτσι παράλληλα με το στάδιο του cutting μπορούμε σε κάθε κόμβο να αποθηκεύουμε την τιμή $(NCP(S) + (m-1)e^{-1})\frac{1-p_g+cp_g}{(1-c)|attr|m} + p_g$, έχοντας κατά μία έννοια προϋπολογίσει την τιμή της πιθανότητας όπου απαιτείται.

Όλη η παραπάνω διαδικασία μπορεί να γίνει ακριβώς με τον ίδιο τρόπο όπου έγινε το στάδιο του generalization. Μάλιστα αν θέλουμε μπορεί να γίνει και παράλληλα και να την αποθηκεύσουμε ώστε να την χρησιμοποιήσουμε αργότερα. Δεν έχουμε όμως δείξει πως τελικά παίρνουμε το αποτέλεσμα.

Ανεβαίνοντας προς τα πάνω το δέντρο έχουμε κάνει precomputed και τις δύο πλευρές της ανίσωσης. Εν συνεχεία παίρνουμε τα αποτελέσματα με μία κατά βάθος εκτέλεση στα παιδιά ώστε αυτά να διαλέξουν αν με βάση το p θέλουνε το αποτέλεσμα. Με άλλα λόγια η διαδικασία είναι ίδια με αυτή του αλγορίθμου στο στάδιο του generalization με την διαφορά ότι στην κατά βάθος εκτέλεση κάθε πατέρας περνάει στο άμεσο παιδί του, όσες τιμές έχουν προϋπολογιστεί.

Μετά το τέλος της διαδικασίας γνωρίζουμε σε ποια τιμή πρέπει να ξε-γενικευτούν ή μη τα φύλλα. Η γενίκευση είναι απλή, σκανάρουμε την βάση και απλά αντικαθιστούμε. Η διαδικασία της ξε-γενίκευσης είναι αρκετά πιο σύνθετη, γιατί είναι πιθανόν μία εγγραφή να έχει αλλάξει QI-group, όμως εμείς πρέπει να λάβουμε υπόψη όλες τις εγγραφές με βάση τις οποίες βρέθηκε μαζί σε κάποιο QI-group ακόμα και αν αυτές έχουν διαγραφεί. Το πρόβλημα είναι ακόμα πιο σύνθετο αν παρατηρήσουμε ότι δεν μας επηρεάζουν μόνο οι τιμές της sensitive attribute για μία εγγραφή, αλλά όλη η υπογραφή της και πως αυτή μεταβλήθηκε στον χρόνο.

Έχουμε δύο επιλογές. Η πρώτη είναι να μην εκτελούμε το στάδιο SA-Split για όσα QI-groups περιέχουν κάποια sensitive value η οποία έχει γενικευτεί. Προφανέστατα αυτή η προσέγγιση δεν είναι ικανοποιητική. Αντίθετα είναι καλύτερο για κάθε QI-group να κρατάμε την ελάχιστη signature την οποία μπορούμε να έχουμε. Πιο συγκεκριμένα:

Ορισμός 30. *Minimal Signature. Έστω οι similar signatures* $s_1, s_2, ..., s_n$, *με* $s_i = (s_{i_1}, ..., s_{i_r}), 1 \leq i \leq n$. *Θα ονομάζουμε την signature* $s = (s_{s_1}, ..., s_{s_r})$ *minimal των υπογραφών* $s_1, ..., s_n$ *αν για κάθε* i, j *το* s_{i_j} *είναι ένα interval που καλύπτει το* s_{s_j} *και δεν υπάρχει άλλη τιμή* s'_{s_j} *έτσι ώστε αυτή να είναι ένα interval που να καλύπτει το* s_{s_j} *και ταυτόχρονα το* s_{i_j} *να είναι ένα interval που να καλύπτει το* s'_{s_j}.

Σε κάθε QI-group αντιστοιχούμε πάντα μία minimal signature. Αυτή προκύπτει ως εξής:

- Αν όλες οι εγγραφές σε αυτό το QI-group είναι καινούργιες τότε η minimal signature είναι η υπογραφή αυτών των εγγραφών.
- Έστω ότι υπάρχουν k εγγραφές στο QI-group, εκ των οποίων οι πρώτες l ανήκαν σε άλλα QI-group πιο πριν. Τότε ορίζουμε ως minimal signature, την minimal signature των minimal signature όλων των προηγουμένων QI-groups μαζί με την υπογραφή του υπάρχοντος.

Με άλλα λόγια, όταν δημιουργούμε ένα QI-group από εγγραφές οι οποίες πιο πριν υπήρχανε σε άλλα QI-groups κληρονομούμε τις minimal signature των προηγουμένων QI-groups. Προσέξτε μάλιστα ότι αυτή η συνθήκη μας δίνει και το κριτήριο με βάση μπορούμε να φτιάξουμε QI-groups. Μία εγγραφή μπορεί να μπει στο ίδιο QI-group με μία άλλη, μόνο αν οι minimal signatures των προηγουμένων QI-groups στα οποία ανήκαν είναι similar.

Τώρα μπορούμε να ορίσουμε πως θα γίνεται και η ξε-γενίκευση. Όταν θέλουμε να ξε-γενικεύσουμε μία τιμή, επιλέγουμε την τιμή εκείνη που αντιστοιχεί στην minimal signature. Προφανέστατα οι νέες εγγραφές δεν έχουν minimal signature, αλλά εξάλλου οι τιμές αυτές δεν έχουν γενικευτεί ποτέ ώστε να ξε-γενικευτούν. Επί της ουσίας με αυτή την διαδικασία ικανοποιούμε αυτή την συνθήκη *Και κάθε δύο εγγραφές* t_a, t_b *οι οποίες υπήρξαν στο ίδιο QI-group σε κάποια όψη* $T^*(j), 1 \leq j \leq n$ *έχουν ένα σύνολο από signatures οι οποίες είναι similar μεταξύ τους* του ορισμού του (e, m)-cba.

Τέλος πρέπει να ορίσουμε πως ακριβώς θα γίνει το στάδιο του SA-split. Το στάδιο αυτό θα είναι ακριβώς το ίδιο με του m-invariance μόνο που τα buckets δεν κατασκευάζονται με βάση την signature μίας εγγραφής, αλλά με βάση την minimal signature του QI-group στο οποίο άνηκε.

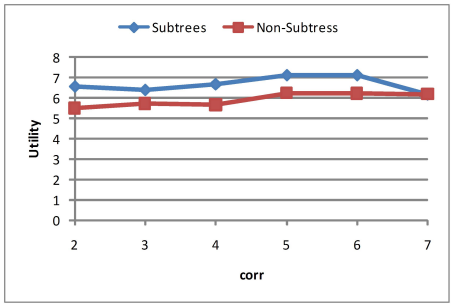
Να σημειώσουμε ότι όλοι οι αλγόριθμοι οι οποίοι παρουσιάστηκαν και αφορούν το σκανάρισμα των δέντρων, μπορούνε να γίνουνε παράλληλα, αφού είναι bottom-up διαδικασίες. Με αυτό τον τρόπο θα γλυτώσουμε σημαντικό χρόνο στην εκτέλεση.

3.12 Πειράματα

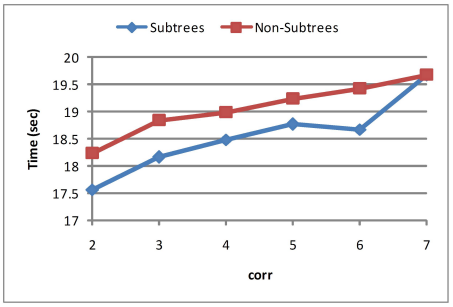
Για να ελέγξουμε την ορθότητα των συμπερασμάτων μας εκτελέσαμε μερικά πειράματα. Στα πειράματα αυτά δεν ελέγξαμε καθόλου τον αλγόριθμο του frequency. Για την ακρίβεια για διάφορες τιμές του m, e εκτελέσαμε τον αλγόριθμο που περιγράφτηκε στο 3.9. Εκτελέσαμε δύο αλγορίθμους ο πρώτος είναι αυτός που περιγράφτηκε με βάση τα sub trees και ο δεύτερος είναι χωρίς την χρήση sub trees. Ως δεδομένα εισόδου χρησιμοποιήσαμε τα ... και τα ελέγξαμε σε στατικό επίπεδο (χωρίς δηλαδή να έχουμε εισαγωγές ή διαγραφές)

Στο πρώτο μας πείραμα, κρατήσαμε σταθερό το $m = 4$ και για διάφορες τιμές του e μετρήσαμε το utility και το χρόνο εκτέλεσης. Δίνεται το αντίστοιχο διάγραμμα στο σχήμα 3.9.

Εν συνεχεία στο επόμενο πείραμα μας κρατήσαμε σταθερό το $e = 3$ και για διάφορες τιμές του m μετρήσαμε το utility και το χρόνο εκτέλεσης. Δίνεται το αντίστοιχο διάγραμμα στο σχήμα 3.10.

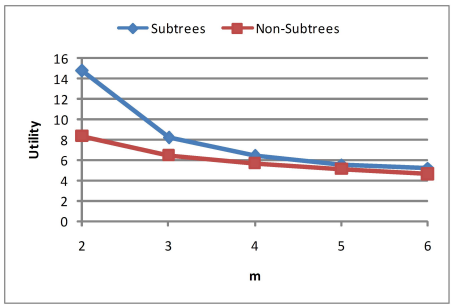


(α′) Το utility.

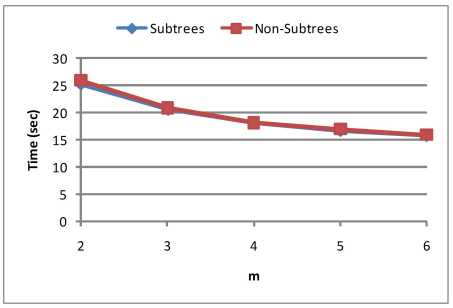


(β′) Ο χρόνος εκτέλεσης.

Σχήμα 3.9: Το utility και ο χρόνος εκτέλεσης για $m = 4$ και $e = 50/2$ έως $e = 50/7$



(α′) Το utility.



(β′) Ο χρόνος εκτέλεσης.

Σχήμα 3.10: Το utility και ο χρόνος εκτέλεσης για $m = 2$ έως $m = 6$ και $e = 50/3$

Μπορεί κανείς από τα πειράματα να παρατηρήσει το εξής. Η μέθοδος των sub trees δίνει καλύτερο utility από ότι χωρίς. Ο λόγος είναι ότι είναι δύσκολο να υπολογίσουμε την περίμετρο, αφού το πρόβλημα είναι NP-hard. Επίσης μπορεί κανείς να παρατηρήσει ότι για τον ίδιο ακριβώς λόγο η μέθοδος αυτή είναι και πιο γρήγορη. Όσο μεγαλύτερο αρχείο εισόδου έχουμε τόσο πιο πολύ μας κοστίζει ο υπολογισμός για τον έλεγχο της περιμέτρου. Μάλιστα η μέθοδος των sub trees επί της ουσίας έχει κόστος ανάλογο του βάθους του δέντρου σε αντίθεση με τον υπολογισμό της περιμέτρου το οποίο εξαρτάται από το μέγεθος της βάσης.

Βιβλιογραφία

[1] Kristen LeFerve, David J. DeWitt, Raghu Ramakrishnan, *Incognito: Efficient Full-Domain K-Anonymity*. University of Wisconsin-Madison, in ACM SIGMOD 2005.

[2] Kristen LeFerve, David J. DeWitt, Raghu Ramakrishnan, *Mondiran multidimensional k-anonymity*. University of Wisconsin-Madison, in ICDE 2006.

[3] Jian Xu, Wei Wang, Jain Pei, Xiaoyan Wang, Baile Shi, Ada Wai-Chee Fu *Utility-Based Anonymization for Privacy Preservation with Less Information Loss*. In SIGKDD Explorations 2006.

[4] Ashwin Machanavajjhala, Johannes Gehrke, Daniel Kifer *l-Diversity: Privacy Beyond k-Anonymity*. Department of Computer Science, Cornell University, in ICDE 2006.

[5] X. Xiao, Y. Tao *Anatomy: Simple and Effective Privacy Presevation*. Department of Computer Science and Engineering, Chinese University of Hong Kong, in VLDB, pages 139-150 2006.

[6] X. Xiao, Y. Tao *m-Invariance: Towards Privacy Preserving Re-publication of Dynamic Datasets*. Department of Computer Science and Engineering, Chinese University of Hong Kong, in SIGMOD 2007.