ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

ΚΜ-ΑΝΩΝΥΜΟΠΟΙΗΣΗ ΜΕ ΔΥΝΑΜΙΚΕΣ ΙΕΡΑΡΧΙΕΣ ΓΕΝΙΚΕΥΣΗΣ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

ΒΑΣΙΛΕΙΟΥ ΚΥΡΒΑΣΙΛΗ

Επιβλέπων : Ιωάννης Βασιλείου
Καθηγητής Ε.Μ.Π.

Αθήνα, Δεκέμβριος 2015

Η σελίδα αυτή είναι σκόπιμα λευκή.

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

ΚΜ – ΑΝΩΝΥΜΟΠΟΙΗΣΗ ΜΕ ΔΥΝΑΜΙΚΕΣ ΙΕΡΑΡΧΙΕΣ ΓΕΝΙΚΕΥΣΗΣ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

ΒΑΣΙΛΕΙΟΥ ΚΥΡΒΑΣΙΛΗ

Επιβλέπων : Ιωάννης Βασιλείου
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 21η Δεκεμβρίου 2015.

(Υπογραφή)	*(Υπογραφή)*	*(Υπογραφή)*
.....	.....	.....
Ιωάννης Βασιλείου	Στέφανος Κόλλιας	Κώστας Κοντογιάννης
Καθηγητής Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.	Αναπλ. Καθηγητής Ε.Μ.Π.

Αθήνα, Δεκέμβριος 2015

.....

ΒΑΣΙΛΕΙΟΣ ΚΥΡΒΑΣΙΛΗΣ

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

Περίληψη

Στην εποχή της πληροφορίας όπου ζούμε, η καταχώρηση και συλλογή προσωπικών πληροφοριών και στοιχείων λαμβάνει χώρα στο μεγαλύτερο εύρος των καθημερινών μας δραστηριοτήτων. Η δημοσίευση κάποιων εξ' αυτών των στοιχείων είναι πλέον συνήθης, με σκοπό την εκμετάλλευση αυτών για επιστημονικούς, στατιστικούς ή εμπορικούς λόγους. Η δημοσίευση αυτή πρέπει να γίνεται με τρόπο τέτοιον ώστε να διασφαλίζεται κατά το δυνατόν η ανωνυμία των προσώπων τα στοιχεία των οποίων δημοσιεύονται.

Η ανωνυμοποίηση των δεδομένων έχει ως στόχο την προστασία της ιδιωτικότητας και φυσικά έχει απασχολήσει ιδιαίτερα την επιστημονική κοινότητα. Έχουν προταθεί πλήθος αλγορίθμων που συμβάλλουν στην επίτευξη του στόχου αυτού. Στην παρούσα διπλωματική εργασία θα ασχοληθούμε με την διασφάλιση της ιδιωτικότητας με τη χρήση της k^m – ανωνυμίας.

Η k^m – ανωνυμία επιδιώκει τη προστασία της ιδιωτικότητας απαιτώντας τη μη δυνατή αναγνώριση μιας συγκεκριμένης εγγραφής των δεδομένων ανάμεσα σε k άλλες, με γνώση το πολύ m στοιχείων της εγγραφής αυτής. Κατά τον αλγόριθμο της k^m – ανωνυμοποίησης, ο παραπάνω στόχος επιτυγχάνεται μέσω της γενίκευσης στοιχείων της συλλογής δεδομένων βάσει μιας προκαθορισμένης ιεραρχίας γενίκευσης.

Στα πλαίσια της εργασίας αναπτύσσεται αλγόριθμος ο οποίος εξασφαλίζει την k^m – ανωνυμία χωρίς την ύπαρξη μια προκαθορισμένης ιεραρχίας γενίκευσης. Ο αλγόριθμος δημιουργεί την ιεραρχία γενίκευσης με δυναμικό τρόπο κατά την εκτέλεση του αλγορίθμου, γενικεύοντας μόνον όσα στοιχεία είναι απαραίτητο να γενικευτούν για την επίτευξη της ανωνυμίας.

Λέξεις Κλειδιά:

k^m – ανωνυμία,

δυναμικές ιεραρχίες γενίκευσης,

ανωνυμοποίηση δεδομένων

Abstract

In the information society in which we live today, collection and registration of personal information happens in the vast majority of our daily activities. Publishing some of this data is now often, and could be done for scientific, statistical or commercial purposes. Publishing of these data should be done in such a way that guarantees as much as possible the individual's of whom the data are published anonymity, without however losing the value of the information.

The goal of data anonymization is to preserve privacy, and it has of course been studied extensively by the scientific community. There are many algorithms that have been proposed in order to achieve the above. This diplomacy thesis deals with preserving privacy using km – anonymity.

km- anonymity seeks to preserve privacy by enforcing that a tuple of a data set cannot be distinguished among k others, while having knowledge of m items of that tuple. The above is achieved through the km – anonymization algorithm by generalizing items of the data set based on a predefined generalization hierarchy.

As part of this thesis, a new algorithm is developed that ensures km – anonymity without the need of a predefined generalization hierarchy. The algorithm dynamically builds the generalization hierarchy as executed, generalizing only items that truly need to be generalized in order to achieve km – anonymity.

Keywords:

km – anonymity

dynamic generalization hierarchy

data anonymity

Ευχαριστίες

Με την ολοκλήρωση της εργασίας αυτής θα ήθελα να ευχαριστήσω θερμά τον κ. Ιωάννη Βασιλείου, Καθηγητή ΕΜΠ, για την ευκαιρία που μου έδωσε να εργαστώ πάνω στο συγκεκριμένο τόσο ενδιαφέρον θέμα της ανωνυμοποίησης δεδομένων και τον χρόνο που διέθεσε για την επίβλεψη αυτού.

Επιπλέον θα ήθελα να ευχαριστήσω ιδιαιτέρως την κα. Όλγα Γκουντούνα για την ουσιαστική επιστημονική καθοδήγηση που με πολλή προθυμία παρείχε ανελλιπώς μέσα από την συνεργασία μας, συμβάλλοντας καθοριστικά στην ανάπτυξη της παρούσας εργασίας.

Τέλος, το μεγαλύτερο ευχαριστώ οφείλω στη σύζυγό μου Θεοδότη, για την αμέριστη στήριξή της καθ' όλη τη διάρκεια της εκπόνησης της εργασίας, και την ουσιαστική δική της συμβολή στην ολοκλήρωση αυτής της διπλωματικής εργασίας.

Πίνακας περιεχομένων

1

Εισαγωγή

1.1 Ανωνυμοποίηση δεδομένων

Ζούμε στην εποχή της πληροφορίας, όπου η καταχώρηση και συλλογή προσωπικών πληροφοριών και στοιχείων λαμβάνει χώρα στο μεγαλύτερο εύρος των καθημερινών μας συνηθειών, είτε αυτές αφορούν τις συναλλαγές μας με τη τράπεζα, τα ψώνια μας στο super market, την ιατρική εξέτασή μας σε ένα διαγνωστικό κέντρο ή ακόμα και τη συμμετοχή στο γυμναστήριο της γειτονιάς μας. Συχνά τα στοιχεία αυτά μπορεί να κοινοποιούνται σε τρίτους με σκοπό την εκμετάλλευσή τους προς χάριν κάποιας μελέτης ή έρευνας. Η δημοσίευση αυτών των στοιχείων, τα οποία μπορεί να περιέχουν συχνά ευαίσθητες προσωπικές πληροφορίες, θα πρέπει να γίνεται με τρόπο τέτοιο ώστε να προστατεύονται όλα τα εμπλεκόμενα πρόσωπα και τα προσωπικά τους στοιχεία.

Η ανωνυμοποίηση ενός συνόλου δεδομένων έχει ως στόχο την προστασία της ιδιωτικότητας των προσώπων που περιέχονται σε αυτό. Ο στόχος αυτός επιτυγχάνεται μέσα από την επεξεργασία των προς δημοσίευση στοιχείων, κατά τρόπο τέτοιο ώστε να μην είναι δυνατή η αναγνώριση και αντιστοίχησή τους σε κάποιο συγκεκριμένο πρόσωπο. Ανάλογα με την τεχνική ανωνυμοποίησης που χρησιμοποιείται, τίθενται διάφοροι περιορισμοί, βάσει των οποίων τροποποιούνται τα δεδομένα. Στην παρούσα εργασία επικεντρωνόμαστε κατά κύριο λόγω στην ανωνυμοποίηση δεδομένων ικανοποιώντας την εγγύηση της k^m-ανωνυμίας..

1.2 Αντικείμενο διπλωματικής

Η παρούσα εργασία ερευνά τον τομέα της ανωνυμοποίησης δεδομένων και επικεντρώνεται στην τεχνική της k^m-ανωνυμοποίησης, υλοποιώντας μια νέα μέθοδο εφαρμογής αυτής, με χρήση δυναμικών ιεραρχιών γενίκευσης.

Η υλοποίηση αυτή ξεφεύγει από τα πλαίσια της κλασσικής k^m-ανωνυμοποίησης, καθώς δεν ακολουθεί μια προκαθορισμένη ιεραρχία βάσει της οποίας γενικεύει τα στοιχεία που απαιτούνται για την επίτευξη της ανωνυμοποίησης, αλλά χτίζει την ιεραρχία αυτή δυναμικά, ανάλογα με τις εκάστοτε ανάγκες. Με τον τρόπο αυτό επιτυγχάνεται η ανωνυμοποίηση των δεδομένων με χρήση μιας ιεραρχίας που δημιουργήθηκε μέσα από το ίδιο το σύνολο των δεδομένων με στόχο την ανωνυμοποίησή τους. Η υλοποίηση αυτή αποτελεί έτσι μια βελτίωση της κλασσικής k^m-ανωνυμοποίησης, καθώς αναπτύχθηκε στη βάση της λογικής της κατά το δυνατόν μικρότερης απώλειας πληροφορίας.

1.2.1 Συνεισφορά

Η παρούσα εργασία επιχειρεί την βελτίωση του υπάρχοντος αλγορίθμου της k^m-ανωνυμοποίησης, εισάγοντας τη δυνατότητα δυναμικής δημιουργίας ιεραρχιών γενίκευσης, εντοπίζοντας τις απαραίτητες γενικεύσεις και μειώνοντας έτσι την απώλεια πληροφορίας. Επιπλέον καθιστά με αυτό τον τρόπο δυνατή την k^m-ανωνυμοποίηση δεδομένων για τις οποίες δεν ορίζεται κάποια προϋπάρχουσα ιεραρχία γενίκευσης.

Συγκεκριμένα κατά την εκπόνηση της εργασίας:

- Μελετήθηκε η σχετική βιβλιογραφία γύρω από το θέμα της ανωνυμοποίησης δεδομένων και της προστασίας της ιδιωτικότητας και διερευνήθηκαν οι πιθανοί τρόποι επίτευξης αυτών
- Αναπτύχθηκαν και υλοποιήθηκαν αλγόριθμοι που εξασφαλίζουν την k^m-ανωνυμοποίηση δεδομένων, με χρήση προκαθορισμένων ιεραρχιών γενίκευσης και με χρήση δυναμικών ιεραρχιών γενίκευσης.
- Διεξήχθησαν συγκριτικά πειράματα των δύο αλγορίθμων ώστε να αναδειχθεί η χρησιμότητα του νέου αλγορίθμου
- Αξιολογήθηκαν τα πειράματα καταδεικνύοντας τα σημεία όπου υστερεί και υπερέχει ο καθένας από τους δύο αλγορίθμους, ενώ επιβεβαιώθηκε και η χρησιμότητα του αλγορίθμου k^m-ανωνυμοποίησης με δυναμικές ιεραρχίες γενίκευσης.

1.3 Οργάνωση κειμένου

Το κείμενο της παρούσας εργασίας που αναπτύσσεται στα επόμενα κεφάλαια, ακολουθεί την παρακάτω δομή:

Στο δεύτερο κεφάλαιο αναλύεται η βιβλιογραφία που μελετήθηκε γύρω από το θέμα της ανωνυμοποίησης δεδομένων, της προστασίας της ιδιωτικότητας και των αλγορίθμων και τεχνικών που έχουν αναπτυχθεί γύρω από τα θέματα αυτά.

Στο τρίτο κεφάλαιο αναλύεται το πρόβλημα της προστασίας της ιδιωτικότητας και γίνεται μια εισαγωγή στην έννοια τις k^m-ανωνυμοποίησης δεδομένων.

Στο τέταρτο κεφάλαιο περιγράφεται ο αλγόριθμος που ερευνά την k^m-ανωνυμοποίηση και την επίτευξη αυτής με δυναμικές ιεραρχίες.

Στο πέμπτο κεφάλαιο παρουσιάζονται οι τεχνικές λεπτομέρειες της υλοποίησης των αλγορίθμων που υλοποιήθηκαν.

Στο έκτο κεφάλαιο περιγράφεται η διεξαγωγή των σχετικών πειραμάτων σύγκρισης των αλγορίθμων που αναπτύχθηκαν.

Στο έβδομο κεφάλαιο συνοψίζονται τα αποτελέσματα της εργασίας γύρω από το ζήτημα της km-ανωνυμοποίησης δεδομένων και της χρήσης δυναμικών ιεραρχιών γενίκευσης για την επίτευξη αυτής και προτείνονται πιθανές μελλοντικές επεκτάσεις της παρούσας εργασίας..

2

Σχετικές εργασίες

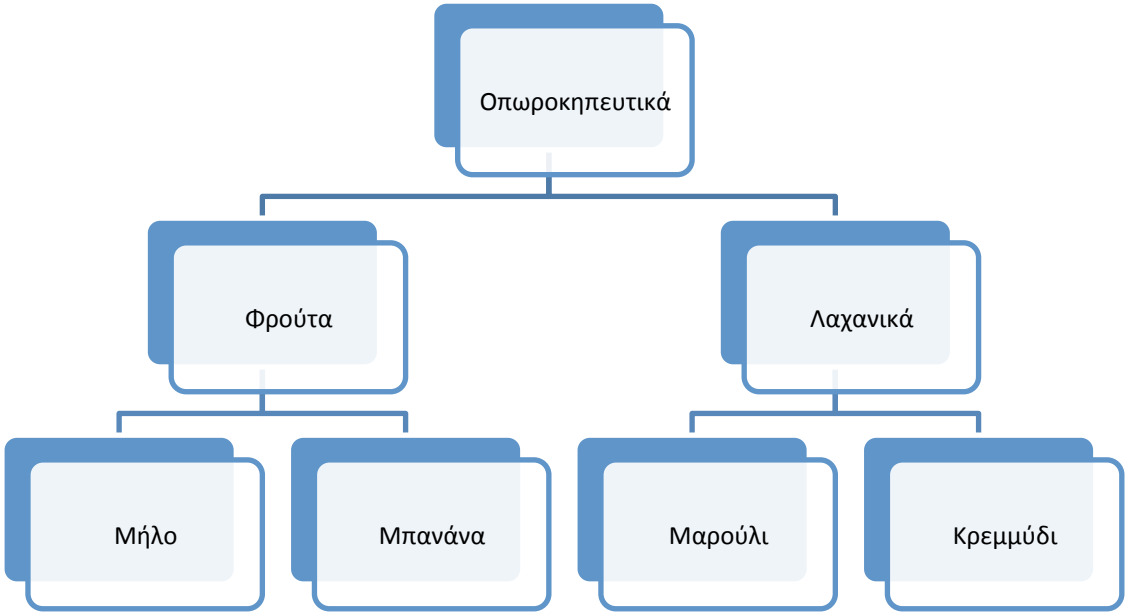
Σε μια εποχή της πληροφορίας και της ολοένα και αυξανόμενης παροχής και δυνατότητας πρόσβασης σε γνώσεις και πληροφορίες, το ζήτημα της ανωνυμοποίησης δεδομένων έχει σίγουρα απασχολήσει αρκετά. Η παροχή και δημοσίευση πληροφοριών είναι πλέον συνεχής, και μπορεί να γίνεται για παράδειγμα για εκπαιδευτικούς , για ερευνητικούς, για διαφημιστικούς ή για πάρα πολλούς άλλους σκοπούς. Η δημοσίευση πληροφοριών όμως γεννά αυτόματα και την ανάγκη προστασίας των εμπλεκομένων, στους οποίους αφορούν οι δημοσιευμένες πληροφορίες. Στο κεφάλαιο που ακολουθεί, θα αναφερθούμε σε εργασίες και μελέτες που έχουν ασχοληθεί με το ζήτημα της ανωνυμοποίησης δεδομένων και σχετίζονται με το πρόβλημα που μελετά και αναπτύσσει η παρούσα εργασία.

2.1 k - ανωνυμία

Το μοντέλο της k – ανωνυμίας (k - anonymity), εξασφαλίζει την ανωνυμία των δεδομένων, αποτρέποντας την άμεση αναγνώριση και συσχέτιση μίας πλειάδας, ανάμεσα σε k-1 άλλες πλειάδες. Πρακτικά ένας δημοσιευμένος πίνακας παρέχει k - ανωνυμία εάν οι πληροφορίες ενός προσώπου που βρίσκονται στο δημοσιευμένο πίνακα, δεν μπορεί αν αναγνωριστεί ανάμεσα σε k-1 άλλα πρόσωπα τα οποία επίσης βρίσκονται στο δημοσιευμένο πίνακα. Η k-ανωνυμία εστιάζει στη προστασία από την έκθεση της ταυτότητας (identity disclosure) μια εγγραφής.

2.1.1 Γενίκευση

Η μέθοδος της γενίκευσης, απαιτεί την παρουσία μιας ιεραρχίας γενίκευσης. Ιεραρχία γενίκευσης, αποτελεί μια δομή βάσει της οποίας κάθε τιμή του πίνακα, μπορεί να αντικατασταθεί από μια γενικότερη τιμή. Η τεχνική αυτή είναι ιδιαιτέρως χρήσιμη καθώς δίνει τη δυνατότητα να διαφυλάσσεται η πληροφορία της αρχικής τιμής, σε μια γενικότερη μορφή αυτής. Ταυτόχρονα, γενικεύοντας τιμές μέσω της τεχνικής της γενίκευσης, επιτυγχάνουμε να αντικαθίστανται δύο άνισες τιμές του πίνακα, με μια γενικευμένη τιμή, αυξάνοντας καθ' αυτόν τον τρόπο τις πλειάδες που περιέχουν την ανωτέρω τιμή που επιθυμούμε να αντικαταστήσουμε.



Στο παραπάνω διάγραμμα, βλέπουμε ένα παράδειγμα ιεραρχίας γενίκευσης. Βάσει της παραπάνω ιεραρχίας, δύο πλειάδες οι οποίες περιείχαν τις τιμές {Μήλο} και {Μπανάνα}, θα μπορούσαν να αντικαταστήσουν τις τιμές αυτές με τη γενικότερη τιμή {Φρούτα}.

2.1.2 Απόκρυψη

Κατά τη μέθοδο της απόκρυψης, επιλέγονται ορισμένες πλειάδες οι οποίες παραβιάζουν την k-ανωνυμία και αφαιρούνται εντελώς από τον προς δημοσίευση πίνακα. Η μέθοδος αυτή είναι χρήσιμη σε περιπτώσεις όπου δεν μπορεί να χρησιμοποιηθεί η τεχνική της γενίκευσης για την επίτευξη της k-ανωνυμίας, είτε διότι δεν υπάρχει κάποια γενίκευση για την εν λόγω

τιμή, είτε διότι μια ενδεχόμενη γενίκευση θα ευτέλιζε την πληροφορία καθιστώντας την μη αξιοποιήσιμη.

2.1.3 Αλγόριθμοι k-ανωνυμοποίησης πινάκων

Δύο από τους βασικούς αλγορίθμους k-ανωνυμοποίησης με τους οποίους θα ασχοληθούμε, αποτελούν οι αλγόριθμοι Incognito και Mondrian. Μέσω των αλγορίθμων αυτών επιδιώκεται η k-ανωνυμοποίηση ενός πίνακα, με την μικρότερη δυνατή απώλεια πληροφορίας.

2.1.3.1 Αλγόριθμος Incognito

Ο πρώτος αλγόριθμος τον οποίο και θα μελετήσουμε, είναι ο αλγόριθμος Incognito. Ο αλγόριθμος Incognito, που περιγράφεται αναλυτικά από τους LeFevre, DeWitt και Ramakrishnan, [LDR05], βασίζεται στη λογική της γενίκευσης πλήρους πεδίου (Full-Domain generalization) όπου κάθε τιμή ενός πίνακα αντιστοιχίζεται στην ίδια γενικευμένη τιμή βάσει της ιεραρχίας γενίκευσης.

Ξεκινώντας από την αρχική ιεραρχία γενίκευσης, ο αλγόριθμος κατασκευάζει ένα πλέγμα γενίκευσης, στο οποίο αναπαριστώνται όλες οι πιθανές γενικεύσεις κάθε στοιχείου του εκάστοτε ψευδοαναγνωριστικού. Για τον περιορισμό των πιθανών λύσεων, ο αλγόριθμος αξιοποιεί την ιδιότητα του υποσυνόλου (subset property) η οποία περιγράφει πως αν ένας πίνακας T με βάση ένα σύνολο στοιχείων Q είναι k-ανώνυμος τότε θα είναι k-ανώνυμος και με βάση κάθε σύνολο P, όπου το P είναι υποσύνολο του Q. Η παραπάνω ιδιότητα είναι χρήσιμη καθώς μας επιτρέπει να μπορούμε να απορρίπτουμε οποιαδήποτε πιο ειδική πιθανή λύση από μια γενικότερη η οποία έχει ήδη εξεταστεί και απορριφθεί, χωρίς να χρειάζεται να εξετάσουμε όλες τις πιθανές λύσεις. Παράλληλα αξιοποιείται και η ιδιότητα της γενίκευσης (generalization property) η οποία κατ' ουσίαν είναι η αντιστροφή της ιδιότητας του υποσυνόλου, και βάσει της οποίας αν ένας πίνακας T είναι k-ανώνυμος με βάση ένα σύνολο γνωρισμάτων P, υποσύνολο ενός συνόλου Q, τότε ο πίνακας T είναι k-ανώνυμος και με βάση το σύνολο Q. Η ιδιότητα αυτή μας δίνει τη δυνατότητα, εάν βρούμε μια k-ανώνυμη λύση, να μη χρειάζεται να εξετάσουμε τις γενικεύσεις αυτής, καθώς γνωρίζουμε πως θα είναι και αυτές k-ανώνυμες.

Έτσι, με βάση τα παραπάνω, ο αλγόριθμος στη πρώτη του επανάληψη ξεκινά και ελέγχει εάν και εφόσον μονοδιάστατα γνωρίσματα του ψευδοαναγωνριστικού ικανοποιούν την k-ανωνυμία. Σε κάθε επανάληψη i, εξετάζει μεγαλύτερα υποσύνολα, μεγέθους i, και κατασκευάζει γράφους με τις πιθανές γενικεύσεις με βάση όλα τα μεγέθους i υποσύνολα του ψευδοαναγνωριστικού, λαμβάνοντας φυσικά υπ' όψιν τυχόν αποκλεισμούς που έχουν ήδη

γίνει. Η τελευταία επανάληψη δίνει γράφο με όλες τις γενικεύσεις που μπορούσαν να ικανοποιήσουν την k-ανωνυμία, και από τις οποίες μπορεί να επιλεγεί η βέλτιστη.

Ο incognito αλγόριθμος όπως περιγράφηκε παραπάνω έχει τη δυνατότητα να υπολογίσει όλες τις k-ανώνυμες εκδοχές ενός πίνακα, βασισμένος σε μια αρχική ιεραρχία γενίκευσης. Ωστόσο το γεγονός ότι αναζητά όλες τις πιθανές λύσεις τον καθιστά χρονοβόρο σε περιπτώσεις όπου το ζητούμενο είναι απλά μια k-ανώνυμη λύση. Επιπλέον, ο αλγόριθμος incognito προϋποθέτει την εξ αρχής ύπαρξη μιας ιεραρχίας γενίκευσης. Αυτό μπορεί να είναι αρνητικό σε κάποιες περιπτώσεις , όπως στην περίπτωση αριθμητικών δεδομένων όπου οι γενικεύσεις συνήθως έχουν την μορφή ενός εύρους τιμών, και μια προκαθορισμένη ιεραρχία μπορεί να οδηγήσει σε γενικεύσεις πολύ μεγαλύτερες απ' ότι είναι πραγματικά απαραίτητο.

2.1.3.2 Αλγόριθμος Mondrian

Όπως είδαμε κατά την περιγραφή του αλγορίθμου Incognito, εφαρμόζεται μια γενίκευση πλήρους πεδίου, αντικαθιστώντας όλες τις τιμές με την ίδια γενίκευση. Ένα από τα βασικά μειονεκτήματα αυτού εμφανίζονταν στην περίπτωση αριθμητικών δεδομένων, όπου μπορούσε εύκολα να εμφανιστεί μια υπεργενίκευση των δεδομένων, αλλοιώνοντας καθ' αυτόν τον τρόπο χρήσιμη πληροφορία, και καθιστώντας τα δεδομένα συχνά δύσχρηστα.

Ο αλγόριθμος Mondrian ο οποίος περιγράφεται και πάλι από τους LeFevre, DeWitt και Ramakrishnan [LDR06] έρχεται να δώσει λύση στα βασικά μειονεκτήματα του αλγορίθμου incognito, όπως αυτά αναφέρθηκαν παραπάνω. Ο Mondrian, εκτελεί έναν άπληστο αλγόριθμο, ακολουθώντας ένα νέο πολυδιάστατο μοντέλο γενίκευσης.

Η διαφορά του μονοδιάστατου και πολυδιάστατου μοντέλου έγκειται στα όρια των τιμών γενίκευσης. Ο καθορισμός αυτών προκύπτει μέσα από τη διαμέριση (partitioning) του συνόλου των τιμών ώστε να προκύψει η βέλτιστη γενίκευση. Κατά τη μονοδιάστατη ανακωδικοποίηση (single dimensional recoding) χρησιμοποιείται μια συνάρτηση φ_t: D_{Xt} -> D' για κάθε γνώρισμα (attribute) X_t του ψευδοαναγνωριστικού, με D_X το πεδίο τιμών ενός γνωρίσματος X του ψευδοανγνωριστικού. Η ανωνυμία επιτυγχάνεται με την αντικατάσταση κάθε τιμής του X_t με την τιμή της συνάρτησης φ_t. Αντίθετα κατά την πολυδιάστατη ανακωδικοποίηση (multidimensional recoding) ορίζεται μια συνάντηση φ: D_{X1} x … x D_{Xn} -> D', η οποία χρησιμοποιείται για τη γενίκευση ενός συνόλου από πεδία του ψευδοαναγνωριστικού.

Ηλικία	Φύλο	ΤΚ	Ασθένεια
25	Άρρεν	53711	Γρίπη
25	Θήλυ	53712	Ηπατίτιδα
26	Άρρεν	53711	Βρογχίτιδα
27	Άρρεν	53710	Κάταγμα
27	Θήλυ	53712	AIDS
28	Άρρεν	53711	Υπέρταση

Πίνακας ασθενών

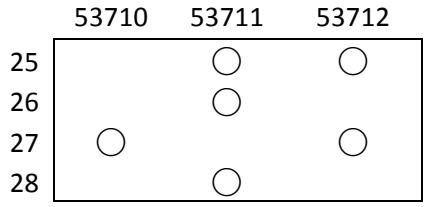
Ηλικία	Φύλο	ΤΚ	Ασθένεια
[25-28]	Άρρεν	[53710-53711]	Γρίπη
[25-28]	Θήλυ	53712	Ηπατίτιδα
[25-28]	Άρρεν	[53710-53711]	Βρογχίτιδα
[25-28]	Άρρεν	[53710-53711]	Κάταγμα
[25-28]	Θήλυ	53712	AIDS
[25-28]	Άρρεν	[53710-53711]	Υπέρταση

Μονοδιάστατη ανωνυμοποίηση

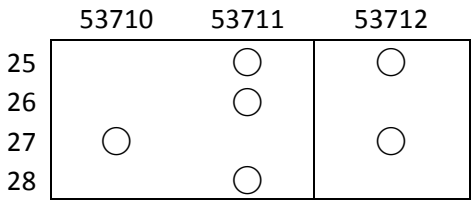
Ηλικία	Φύλο	ΤΚ	Ασθένεια
[25-26]	Άρρεν	53711	Γρίπη
[25-27]	Θήλυ	53712	Ηπατίτιδα
[25-26]	Άρρεν	53711	Βρογχίτιδα
[27-28]	Άρρεν	[53710-53711]	Κάταγμα
[25-27]	Θήλυ	53712	AIDS
[27-28]	Άρρεν	[53710-53711]	Υπέρταση

Πολυδιάστατη ανωνυμοποίηση

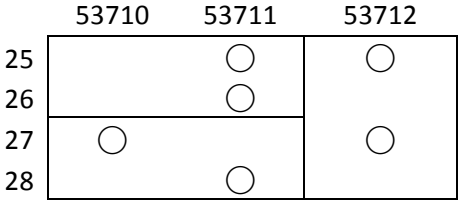
Στο παραπάνω παράδειγμα της πολυδιάστατης ανωνυμοποίησης μπορούμε να παρατηρήσουμε πως έχει γίνει μια μίξη των τιμών γενίκευσης του ΤΚ, ανάλογα με τις τιμές γενίκευσης της ηλικίας. Έτσι βλέπουμε τον ΤΚ 53711 να γενικεύεται με διαφορετικό τρόπο όταν πρόκειται για εγγραφές ατόμων ηλικίας 25-26 και διαφορετικό για ηλικίες 27-28. Μια χωρική αναπαράσταση είναι ιδιαιτέρως χρήσιμη για την κατανόηση της έννοιας.



Πίνακας ασθενών



Μονοδιάστατη ανωνυμοποίηση



Πολυδιάστατη ανωνυμοποίηση

Η εκτέλεση του Mondrian αλγορίθμου, βασίζεται στην αναδρομική διαμέριση του χώρου, εφόσον αυτή είναι δυνατή.

- Αρχικά ορίζονται οι πολυδιάστατες περιοχές που καλύπτουν το πεδίο του ψευδοαναγνωριστικού.
- Σε κάθε επανάληψη, επιλέγεται η διάσταση βάση της οποίας θα γίνει η διαμέριση. Ο τρόπος που μπορεί να γίνει η επιλογή αυτή δεν είναι αυστηρός, και υπάρχει μια σχετική ευελιξία. Ένας πιθανός τρόπος αποτελεί η επιλογή της διάστασης με το μεγαλύτερο εύρος τιμών, ή βάσει της εκτιμώμενης πολυπλοκότητας μιας επιλογής.
- Εάν η περαιτέρω διαμέριση είναι επιτρεπτή, τότε αυτή γίνεται με βάση τον στατιστικό μέσο (median) του υποχώρου στον οποίο βρισκόμαστε.
- Εάν δεν μπορεί να υπάρξει άλλη τομή, στη συγκεκριμένη διαμέριση, τότε επιστρέφεται η τελική διαμέριση.
- Τα παραπάνω βήματα επαναλαμβάνονται αναδρομικά μέχρι να μην είναι δυνατή άλλη διαμέριση σε καμία διάσταση.

2.1.4 Επιθέσεις κατά της k-ανωνυμίας

Παρά το ότι αποτρέπεται μια απευθείας σύνδεση των στοιχείων που ο επιτιθέμενος γνωρίζει, με τα δημοσιευμένα στοιχεία του πίνακα, εφόσον αυτός ικανοποιεί την k – ανωνυμία, υπάρχουν και άλλου είδους επιθέσεις, οι οποίες μπορούν να καταστήσουν τα δημοσιευμένα δεδομένα επιρρεπή ακόμη και σε περιπτώσεις που ικανοποιείται η k – ανωνυμία. Ορισμένες από αυτές περιγράφονται σύντομα παρακάτω.

2.1.4.1 Επίθεση σε ταξινομημένο πίνακα

Η επίθεση αυτή βασίζεται στην ενδεχόμενη δημοσίευση πολλαπλών πινάκων οι οποίοι ακολουθούν την ίδια ταξινόμηση των εγγραφών τους.

Εθνικότητα	Ημ. Γέννησης
Ελλάδα	1975
Ελλάδα	1985
Ελλάδα	1972
Ελλάδα	1988
Ισπανία	1975
Ισπανία	1985
Ισπανία	1972
Ισπανία	1988

Αρχικός Πίνακας

Εθνικότητα	Ημ. Γέννησης
Ευρωπαϊκή	1975
Ευρωπαϊκή	1985
Ευρωπαϊκή	1972
Ευρωπαϊκή	1988
Ευρωπαϊκή	1975
Ευρωπαϊκή	1985
Ευρωπαϊκή	1972
Ευρωπαϊκή	1988

Πίνακας 1

Εθνικότητα	Ημ. Γέννησης
Ελλάδα	197*
Ελλάδα	198*
Ελλάδα	197*
Ελλάδα	198*
Ισπανία	197*
Ισπανία	198*
Ισπανία	197*
Ισπανία	198*

Πίνακας 2

Στο παραπάνω παράδειγμα, παρά το οτι και οι δύο τροποποιημένοι πίνακες είναι k-ανώνυμοι (k=2), υπάρχει και ένας τρίτος παράγοντας ο οποίος μπορεί να οδηγήσει σε παραβίαση της ανωνυμίας. Και οι δύο πίνακες είναι ταξινομημένη με την ίδια σειρά. Συνεπώς, εάν δημοσιευτούν και οι δύο, είναι πολύ εύκολο με μια απευθείας αντιστοίχηση των μεταξύ των εγγραφών των δύο πινάκων, να αποκαλυφθεί ο αρχικός πίνακας. Η παραπάνω επίθεση μπορεί πολύ εύκολα να αποφευχθεί με την τυχαία ταξινόμηση κάθε πίνακα προτού δημοσιευτεί.

2.1.4.2 Επίθεση σε συμπληρωματική έκδοση

Είναι σύνηθες το ψευδοαναγνωριστικό να αποτελεί απλά ένα υποσύνολο των γνωρισμάτων ενός πίνακα. Είναι σημαντικό λοιπόν, όταν αναδημοσιεύεται ένας πίνακας, το ψευδοαναγνωριστικό του νέου πίνακα να λαμβάνει υπ' όψιν του τα ήδη δημοσιευμένα στοιχεία του προγενέστερου πίνακα.

Εθνικότητα	Ημ.Γέννησης	Γένος	ΤΚ	Ασθένεια
Ελλάδα	1965	Άρρεν	52141	δύσπνοια
Ελλάδα	1965	Άρρεν	52141	καρδιοπάθεια
Ελλάδα	1965	Θήλυ	52138	πονόλαιμος
Ελλάδα	1965	Θήλυ	52138	ίλιγγος
Ελλάδα	1964	Θήλυ	52138	παχυσαρκία
Ελλάδα	1964	Θήλυ	52138	καρδιοπάθεια
Ισπανία	196*	Άρρεν	52138	δύσπνοια
Ισπανία	196*	Α/Θ	52139	υπέρταση
Ισπανία	196*	Α/Θ	52139	παχυσαρκία
Ισπανία	196*	Α/Θ	52139	πυρετός
Ισπανία	196*	Άρρεν	52138	μαγουλάδες
Ισπανία	196*	Άρρεν	52138	κοιλόπονος

Πίνακας 1

Εθνικότητα	Ημ.Γέννησης	Γένος	ΤΚ	Ασθένεια
Ελλάδα	1965	Άρρεν	52141	δύσπνοια
Ελλάδα	1965	Άρρεν	52141	καρδιοπάθεια
Ευρωπαϊκή	1965	Θήλυ	5213*	πονόλαιμος
Ευρωπαϊκή	1965	Θήλυ	5213*	ίλιγγος
Ελλάδα	1964	Θήλυ	52138	παχυσαρκία
Ελλάδα	1964	Θήλυ	52138	καρδιοπάθεια
Ισπανία	1964	Άρρεν	5213*	δύσπνοια
Ευρωπαϊκή	1965	Θήλυ	5213*	υπέρταση
Ισπανία	1964	Άρρεν	5213*	παχυσαρκία
Ισπανία	1964	Άρρεν	5213*	πυρετός
Ισπανία	1967	Άρρεν	52138	μαγουλάδες
Ισπανία	1967	Άρρεν	52138	κοιλόπονος

Πίνακας 2

Στο παραπάνω παράδειγμα, δημοσιεύτηκαν οι πίνακες 1 και 2. Και οι δύο πίνακες είναι k-ανώνυμοι (k=2). Και οι δύο πίνακες χρησιμοποίησαν ως ψευδοαναγνωριστικό το υποσύνολο {Εθνικότητα, Ημ.Γέννησης, Γένος, ΤΚ}. Ωστόσο, μετά τη δημοσίευση και του πίνακα 2, με μια απλή αντιστοίχηση των Ασθενειών στους δύο πίνακες, μπορεί εύκολα να αποκαλυφθούν οι εγγραφές

[Ισπανία,1964,Άρρεν,52138,δύσπνοια] και [Ισπανία,1965,Θύλη,52139,υπέρταση].

	Ημ.Γέννησης	Γένος	ΤΚ	Ασθένεια
Ελλάδα	1965	Άρρεν	52141	δύσπνοια
Ελλάδα	1965	Άρρεν	52141	καρδιοπάθεια
Ελλάδα	1965	Θήλυ	52138	πονόλαιμος
Ελλάδα	1965	Θήλυ	52138	ίλιγγος
Ελλάδα	1964	Θήλυ	52138	παχυσαρκία
Ελλάδα	1964	Θήλυ	52138	καρδιοπάθεια
Ισπανία	1964	Άρρεν	52138	δύσπνοια
Ισπανία	1965	Θήλυ	52139	υπέρταση
Ισπανία	1964	Άρρεν	52139	παχυσαρκία
Ισπανία	1964	Άρρεν	52139	πυρετός
Ισπανία	1967	Άρρεν	52138	μαγουλάδες
Ισπανία	1967	Άρρεν	52138	κοιλόπονος

Αρχικός Πίνακας

Αυτό συνέβη διότι ο πίνακας 2 δεν έλαβε υπ' όψιν του τα ήδη δημοσιευμένα στοιχεία του πίνακα 1. Η παραπάνω επίθεση θα μπορούσε να αποφευχθεί εάν ο πίνακας 2 περιελάμβανε στο ψευδοαναγνωριστικό του και την {Ασθένεια} που ήταν ήδη δημοσιευμένη στον πίνακα 1, ή εάν έστω χρησιμοποιούσε τον πίνακα 1 ως βάση και δεν αποκάλυπτε στοιχεία που είχε αποκρύψει προηγουμένως ο πίνακας 1.

2.1.4.3 Χρονική επίθεση

Με την πάροδο του χρόνου, τα δεδομένα ενός πίνακα μπορεί να αλλάξουν με την προσθήκη ή αφαίρεση εγγραφών από τον πίνακα. Πολλές φορές, υπάρχει η ανάγκη αφότου δημοσιευτεί ένας πίνακας, να δημοσιευτεί μετά μια άλλη εκδοχή ή ένας συμπληρωματικός αυτού. Ωστόσο, όταν ο πίνακας έχει αλλάξει, δεν μπορεί να εξασφαλιστεί ότι τα ίδια στοιχεία θα απαιτούν γενίκευση για την ικανοποίηση της k-ανωνυμίας.

Έστω λοιπόν ένας δημοσιευμένος πίνακας T_1 ο οποίος ικανοποιεί την k-ανωνυμία. Έστω ότι μετά την πάροδο του χρόνου, στον αρχικό πίνακα έχουν προστεθεί ορισμένες εγγραφές, με τρόπο τέτοιο που δεν εξασφαλίζεται η k-ανωνυμία, χωρίς να είναι απαραίτητη μια γενίκευση η οποία είχε γίνει προγενέστερα. Η δημοσίευση του νέου πίνακα T_2 χωρίς την γενίκευση του T_1, αυτομάτως εκθέτει τον T_1, ο οποίος με αντιστοίχιση των στοιχείων των 2 πινάκων αποκαλύπτει τα στοιχεία τα οποία είχαν αρχικώς αποκρυφτεί, και έτσι παύει να είναι k-ανώνυμος.

Όπως και στη προηγούμενη περίπτωση, έτσι και εδώ, είναι σημαντικό πάντα όταν υπάρχει μια αναδημοσίευση ενός πίνακα, να λαμβάνονται πάντα υπ' οψιν τα ήδη δημοσιευμένα στοιχεία, και είτε να συμπεριλαμβάνονται όλα τα δημοσιευμένα γνωρίσματα στο ψευδοαναγνωριστικό του νέου πίνακα, είτε να μην εκτίθενται στοιχεία τα οποία είχαν αρχικώς αποκρυφτεί.

2.1.4.4 Επίθεση ομοιογένειας

Είναι πιθανό, παρά το ότι ένας πίνακας ικανοποιεί την k-ανωνυμία, ο τρόπος με τον οποίο είναι χωρισμένες οι κλάσεις ισοδυναμίας, να παρέχουν μια ομοιογένεια στα ευαίσθητα στοιχεία, καθιστώντας τα ευάλωτα σε επιθέσεις ομοιογένειας όπως αυτές περιγράφονται από τους A. Machanavajjhala, J. Gehrke, D. Kifer, M. Venkitasubramaniam [MGK+06], και αποκαλύπτοντας με αυτόν τον τρόπο στοιχεία που θα έπρεπε να είναι ανώνυμα.

	ΤΚ	Ηλικία	Φύλο	Φόρος
1	520**	< 30	*	4000
2	520**	< 30	*	4000
3	520**	< 30	*	5000
4	520**	< 30	*	5000
5	5385**	31 - 40	*	1000
6	5385**	31 - 40	*	4000
7	5385**	31 - 40	*	5000
8	5385**	31 - 40	*	5000
9	520**	> 40	*	1000
10	520**	> 40	*	1000
11	520**	> 40	*	1000
12	520**	> 40	*	1000

Στο παραπάνω παράδειγμα, ο δημοσιευμένος πίνακας από την ΔΟΥ περιέχει ευαίσθητα φορολογικά στοιχεία, και γι αυτό τον λόγο έχει τροποποιηθεί με προσοχή ώστε να ικανοποιεί την k-ανωνυμία (k=4). Ο επιτιθέμενος γνωρίζει ότι το πρόσωπο το οποίο αναζητά είναι πάνω από 40, και συνεπώς ανήκει σε μια εκ των εγγραφών 9-12. Επειδή ο πίνακας είναι 4-ανώνυμος, δεν μπορεί να γνωρίζει ποια από τις 4 εγγραφές αντιστοιχεί στο πρόσωπο το οποίο αναζητά, ωστόσο αυτό δεν έχει σημασία, καθώς και οι 4 εγγραφές, έχουν τον ίδιο φόρο.

Στην περίπτωση αυτή, παρά το γεγονός ότι ο πίνακας ικανοποιούσε την k-ανωνυμία, και ο επιτιθέμενος πράγματι δεν ήταν σε θέση να αντιστοιχίσει μια εγγραφή σε ένα συγκεκριμένο πρόσωπο, παρ' όλα αυτά, η ομοιογένεια ανάμεσα στα στοιχεία του πίνακα αποκάλυψε το ευαίσθητο γνώρισμα για το συγκεκριμένο πρόσωπο.

2.1.4.5 Επίθεση με πρότερη γνώση

Στην επίθεση με πρότερη γνώση [MGK+06], ο επιτιθέμενος γνωρίζει πως το πρόσωπο το οποίο αναζητά βρίσκεται στο δημοσιευμένο πίνακα, και επίσης έχει στην κατοχή του κάποια πληροφορία ή γενικότερη γνώση, η οποία του επιτρέπει να αποκλείσει πιθανές ευαίσθητες τιμές, και να καταλήξει με αυτόν τον τρόπο με βεβαιότητα στην εγγραφή που αντιστοιχεί στο πρόσωπο το οποίο αναζητά, παρ' ότι ο δημοσιευμένος πίνακας ικανοποιεί την k-ανωνυμία.

Στο προηγούμενο παράδειγμα, ο επιτιθέμενος γνωρίζει πως το πρόσωπο το οποίο αναζητά ανήκει στην κατηγορία 31-40. Εκ πρώτης όψεως δεν είναι δυνατή η απευθείας αναγνώριση της εγγραφής που αφορά στο πρόσωπο αυτό. Ο επιτιθέμενος όμως γνωρίζει επίσης, πως το πρόσωπο είναι άνεργος, και συνεπώς είναι πολύ απίθανο να πληρώνει φόρο τεσσάρων ή πέντε χιλιάδων ευρώ. Αποκλείοντας λοιπόν τις τιμές αυτές, καταλήγει με βεβαιότητα στο ότι το πρόσωπο το οποίο αναζητά πλήρωσε φόρο 1000 ευρώ.

Ο επιτιθέμενος αν και φαινομενικά δεν ήταν σε θέση να αναγνωρίσει το πρόσωπο το οποίο αναζητούσε, λόγω της πληροφορίας που είχε μπόρεσε να αποκλείσει τις υπόλοιπες εγγραφές και να καταλήξει σε αυτή που αντιστοιχούσε στο πρόσωπο που αναζητούσε.

2.2 l - Διαφορετικότητα

Στις δύο τελευταίες περιπτώσεις, αν και ο δημοσιευμένος πίνακας ικανοποιούσε την k-ανωνυμία, δεν υπήρχε η απαραίτητη διαφορετικότητα μεταξύ των ευαίσθητων τιμών που περιέχονταν στον πίνακα, και έτσι κατέστη δυνατή η παραβίαση της ιδιωτικότητας. Τις περιπτώσεις αυτές έρχεται να καλύψει η l-διαφορετικότητα (l - diversity).

Ανάλογα με την γνώση που έχει ο επιτιθέμενος, ο δημοσιευμένος πίνακας μπορεί να αποκαλύψει δεδομένα με δύο κυρίως τρόπους, τους οποίους ονομάζουμε θετική και αρνητική αποκάλυψη.

Κατά την θετική αποκάλυψη, ο επιτιθέμενος είναι σε θέση να αναγνωρίσει επιτυχώς το ευαίσθητο γνώρισμα με μεγάλη πιθανότητα. Στο παράδειγμα της προηγούμενης ενότητας κατά την επίθεση ομοιογένειας, ο επιτιθέμενος ήταν σε θέση να αναγνωρίσει με βεβαιότητα ο ευαίσθητο γνώρισμα του φόρου, για πρόσωπα τα οποία άνηκαν στην $3^{η}$ κλάση ισοδυναμίας.

Αντίστοιχα, κατά την αρνητική αποκάλυψη, ο επιτιθέμενος είναι σε θέση να αναγνωρίσει το ευαίσθητο γνώρισμα με μεγάλη πιθανότητα αποκλείοντας ορθώς κάποιες από τις πιθανές τιμές του γνωρίσματος. Στο παράδειγμα της προηγούμενης ενότητας, κατά την επίθεση με πρότερη γνώση, ο επιτιθέμενος ήταν σε θέση να αναγνωρίσει την τιμή του ευαίσθητου γνωρίσματος αποκλείοντας ορθώς τις υπόλοιπες τιμές της $2^{ης}$ κλάσης ισοδυναμίας.

Οι [MGK+06], έρχονται να δώσουν μια ενδιαφέρουσα προέκταση στον ορισμό της ιδιωτικότητας στις περιπτώσεις όπου υπάρχει κάποιας μορφής πρότερη γνώση. Σύμφωνα με τους παραπάνω, η αποκάλυψη κάποιας πληροφορίας δεν είναι πάντοτε αρνητική, και δεν καθιστά απαραίτητα από μόνη της παραβίασης της ιδιωτικότητας. Συγκεκριμένα αναφέρουν πως ένας δημοσιευμένος πίνακας, μπορεί να παρέχει επιπλέον πληροφορία, αρκεί αυτή να μην δημιουργεί μεγάλη διαφορά ανάμεσα στη πρότερη και την ύστερη γνώση του επιτιθέμενου. Για παράδειγμα, η ανακάλυψη στο παραπάνω παράδειγμα ότι κάποιος δεν πληρώνει φόρο 15000 ευρώ, επειδή δεν υπάρχει αυτή η τιμή σε καμία από τις εγγραφής της κλάσης ισοδυναμίας στην οποία ανήκει, δεν αποτελεί απαραίτητα μια αρνητική αποκάλυψη εάν ούτως ή άλλως εκ των προτέρων θεωρώντας αρκετά απίθανο να πληρώνει τόσο υψηλό φόρο. Επομένως ναι μεν ο επιτιθέμενος ήταν σε θέση να αποκτήσει αυτή τη πληροφορία, αλλά η διαφορά με την πρότερη γνώση του δεν ήταν μεγάλη, και συνεπώς δεν θεωρούμε ότι υπήρξε παραβίαση ιδιωτικότητας με βάση αυτό το γεγονός.

Η l-διαφορετικότητα, έρχεται να συμπληρώσει την k-ανωνυμία, αποτρέποντας την αναγνώριση της τιμής του ευαίσθητου γνωρίσματος μιας εγγραφής. Κατά την l-διαφορετικότητα, ορίζεται το πλήθος των τιμών του ευαίσθητου γνωρίσματος σε μια κλάση ισοδυναμίας, έτσι ώστε ακόμη και να μπορεί ο επιτιθέμενος να βρει την κλάση ισοδυναμίας στην οποία ανήκει η εγγραφή την οποία αναζητά, να μην είναι δυνατή η αναγνώριση της συγκεκριμένης εγγραφής, ανάμεσα σε l διαφορετικές τιμές του ευαίσθητου γνωρίσματος.

Η l-διαφορετικότητα δίνει το πολύ ισχυρό πλεονέκτημα σε όποιον δημοσιεύει το πίνακα, να μπορεί με ασφάλεια να αποτρέψει επιθέσεις με πρότερη γνώση, χωρίς να είναι απαραίτητο να κατέχει και ο ίδιος την ίδια γνώση. Οποιαδήποτε γνώση μπορεί ο επιτιθέμενος να έχει, θεωρείται απλώς ένας τρόπος να αποκλείσει κάποια τιμή, και με βάση την παράμετρο l, θα χρειάζεται να αποκλείσει άλλες l-1 τιμές για να μπορέσει να εντοπίσει την εγγραφή την οποία αναζητά.

Σύμφωνα με τους [MGK+06], μια κλάση ισοδυναμίας q* είναι l-διαφορετική εάν περιλαμβάνει τουλάχιστον l «καλώς αντιπροσωπούμενες» τιμές για τα ευαίσθητα γνωρίσματά της. Κατ' αντιστοιχία, ένας πίνακας θα θεωρείται l-διαφορετικός εάν κάθε κλάση ισοδυναμίας του είναι l-διαφορετική.

Ο όρος «καλώς αντιπροσωπούμενες» δεν είναι εξ αρχής ξεκάθαρος, και μπορεί να οριστεί με διαφορετικούς τρόπους ανάλογα με τις παραμέτρους που θέλουμε να λάβουμε υπ' όψιν μας. Παρακάτω περιγράφονται δύο βασικοί τρόποι με τους οποίους μπορούμε να ορίσουμε «καλώς αντιπροσωπούμενες» τιμές.

2.2.1 l – Διαφορετικότητα με εντροπία

Ένας πίνακας ικανοποιεί την l-διαφορετικότητα με εντροπία εάν για την εντροπία της κάθε κλάσης ισοδυναμίας q* ισχύει

$$Entropy(\mathrm{q}*) = -\sum_{s \in S} p(q*, s) \log(\quad p(q*, s)) \geq \log(l)$$

όπου p(q*,s) είναι κλάσμα των εγγραφών της κλάσης ισοδυναμίας q* τιμή του ευαίσθητου γνωρίσματος ίση με s.

Κατά την l-διαφορετικότητα με εντροπία, κάθε κλάση ισοδυναμίας q*, έχει τουλάχιστον l διαφορετικές μεταξύ τους τιμές στο ευαίσθητο γνώρισμα. Ο ορισμός των τιμών με βάση την l-διαφορετικότητα με εντροπία μπορεί σε κάποιες περιπτώσεις να είναι πολύ περιοριστική για κάποια γνωρίσματα, όπως για παράδειγμα στη περίπτωση όπου το μεγαλύτερο ποσοστό των ασθενών που επισκέπτονται μια κλινική έχουν για παράδειγμα κάποια καρδιακή πάθηση. Σε αυτή τη περίπτωση, μπορεί η θετική αποκάλυψη της παραπάνω πληροφορίας να είναι επιτρεπτή, καθώς ο επιτιθέμενος που γνωρίζει ότι ο ασθενής επισκέφτηκε την εν λόγω κλινική, ήδη γνωρίζει με μεγάλη πιθανότητα ότι πάσχει από κάποια καρδιακή πάθηση λόγω

του ότι είναι γνωστό το γεγονός ότι οι περισσότεροι ασθενείς που επισκέπτονται την κλινική, έχουν κάποια καρδιακή πάθηση

2.2.2 Αναδρομική (c,l)-διαφορετικότητα

Στις περιπτώσεις όπου κάποια τιμή εμφανίζεται με πολύ μεγαλύτερη συχνότητα από άλλες στο ευαίσθητο γνώρισμα, μπορούμε να εφαρμόσουμε την αναδρομική (c,l)-διαφορετικότητα, η οποία δίνει μια λιγότερο περιοριστική προσέγγιση σε τέτοιες περιπτώσεις.

Έστω ότι $s_1,\ldots,s_m$ οι πιθανές τιμές του ευαίσθητου γνωρίσματος σε μια κλάση ισοδυναμίας q*, ταξινομημένες κατά φθίνουσα συχνότητα εμφάνισης, και $r_1,\ldots,r_m$ οι συχνότητες εμφάνισής τους. Δοθείσης μιας σταθεράς C, θα λέμε ότι μια κλάση ισοδυναμίας q* ικανοποιεί την αναδρομική (c,l)-διαφορετικότητα εάν $r_1 < c(r_l+r_{l+1}\ldots+r_m)$. Ουσιαστικά μια κλάση ισοδυναμίας θα ικανοποιεί την αναδρομική (c,l)-διαφορετικότητα εάν αποκλείοντας μια πιθανή ευαίσθητη τιμή, η κλάση ισοδυναμίας εξακολουθεί και ικανοποιεί την αναδρομική (c,l-1)-διαφορετικότητα

Εφαρμόζοντας την αναδρομική (c,l)-διαφορετικότητα σε περιπτώσεις όπου μια θετική αποκάλυψη μπορεί να είναι επιτρεπτή, όπως στο παράδειγμα που είδαμε παραπάνω με μια τιμή να έχει πολύ μεγάλη συχνότητα εμφάνισης έχουμε τα εξής. Έστω Y ένα σύνολο από ευαίσθητες τιμές των οποίων μια θετική αποκάλυψη είναι επιτρεπτή, και s_y η πιο συχνή τιμή στην κλάση ισοδυναμίας που δεν ανήκει στο σύνολο Y, με r_y την συχνότητα αυτής. Η l-διαφορετικότητα σε αυτή την περίπτωση θα ικανοποιείται εφόσον μετά τον αποκλεισμό των $s_1,\ldots s_y$ η κλάση ισοδυναμίας παραμένει (l-y+1)-διαφορετική.

Επεκτείνοντας τον παραπάνω ορισμό σε περιπτώσεις αρνητικής αποκάλυψης, εισάγεται ο ορισμός της αναδρομικής (c_1,c_2,l)-διαφορετικότητας αρνητικής/θετικής αποκάλυψης. Έστω W το σύνολο των ευαίσθητων τιμών για τις οποίες η αρνητική αποκάλυψη δεν είναι επιτρεπτή. Ένας πίνακας θα ικανοποιεί την αναδρομικής(c_1,c_2,l)-διαφορετικότητα αρνητικής/θετικής αποκάλυψης εφόσον ικανοποιεί την αναδρομική (c,l)-διαφορετικότητα και κάθε τιμή S που ανήκει στο σύνολο W, εμφανίζεται λιγότερο από c_2% σε κάθε κλάση ισοδυναμίας q*.

Στο παράδειγμα της προηγούμενης ενότητας, στις περιπτώσεις της επίθεσης ομοιογένειας και της επίθεσης με πρότερη γνώση, ο δημοσιευμένος πίνακας θα μπορούσε να έχει δημοσιευτεί στην παρακάτω μορφή:

	ΤΚ	Ηλικία	Φύλο	Φόρος
1	5204*	≤40	*	4000
2	5204*	≤40	*	5000
3	5204*	≤40	*	1000
4	5204*	≤40	*	1000
5	5206*	≤40	*	4000
6	5206*	≤40	*	5000
7	5206*	≤40	*	1000
8	5206*	≤40	*	1000
9	5385**	> 40	*	1000
10	5385**	> 40	*	4000
11	5385**	> 40	*	5000
12	5385**	> 40	*	5000

Ο παραπάνω πίνακας, στη νέα του μορφή, ικανοποιεί την 3-διαφορετικότητα, καθώς σε κάθε κλάση ισοδυναμίας υπάρχουν 3 τουλάχιστον διαφορετικές τιμές στο ευαίσθητο γνώρισμα. Έτσι λοιπόν, ακόμη και να μπορέσει ο επιτιθέμενος να αποκλείσει μια τιμή, και πάλι θα υπάρχουν δύο ακόμη ανάμεσα στις οποίες μπορεί να βρίσκεται η εγγραφή την οποία αναζητά.

2.3 t - Εγγύτητα

Παρά το ότι η l-διαφορετικότητα αποτελεί σίγουρα μια βελτίωση της απλής k-ανωνυμίας σε ότι αφορά τη προστασία από αποκάλυψη γνωρισμάτων (attribute disclosure), εντούτοις παρουσιάζει κάποιες δυσκολίες, ορισμένες από τις οποίες περιγράφονται από τους N. Li, T. Li, S. Venkatasubramanian [LLV07] που την καθιστούν αποτρεπτική πολλές φορές όταν πρόκειται για πραγματικά δεδομένα.

Η πρώτη δυσκολία είναι το γεγονός ότι πολλές φορές μπορεί να είναι πολύ δύσκολο ή και αχρείαστο το να εφαρμοστεί. Εάν για παράδειγμα δούμε τη περίπτωση όπου εξετάζεται το αποτέλεσμα μια εργαστηριακής εξέτασης για μια συγκεκριμένη σπάνια πάθηση, η μεγάλη πλειοψηφία των εγγραφών θα έχει ως αποτέλεσμα της εξέτασης την τιμή «αρνητικό» και μόνο ένα μικρό ποσοστό θα είναι «θετικό». Οι δύο αυτές τιμές έχουν διαφορετικό βαθμό ευαισθησίας, καθώς κάποιος ο οποίος διαγνώστηκε με αρνητικά αποτελέσματα μπορεί να μην τον ενδιαφέρει να αποκαλυφθεί η πληροφορία καθώς αυτό ισχύει για τη συντριπτική πλειοψηφία του πληθυσμού. Αντιθέτως κάποιος τα αποτελέσματα του οποίου ήταν θετικά, δεν θα ήθελε να γίνει γνωστό αυτό. Στην παραπάνω περίπτωση, η l-διαφορετικότητα είναι αχρείαστη για μια κλάση ισοδυναμίας που περιέχει μόνο αρνητικά αποτελέσματα. Από την

άλλη, λόγω της πολύ μεγάλης διαφοράς στη συχνότητα μεταξύ θετικής και αρνητικής τιμής, θα έχουμε λιγότερες κλάσεις ισοδυναμίας ώστε να ικανοποιούν την l-διαφορετικότητα και επομένως θα χρειαστεί συχνά να γίνουν μεγάλες γενικεύσεις, με αποτέλεσμα την απώλεια πληροφορίας.

Ένα άλλο αρνητικό της l-διαφορετικότητας αποτελεί το γεγονός ότι δεν είναι πάντα σε θέση να αποτρέψει την έκθεση πληροφορίας, και είναι ευάλωτη σε κάποιου είδους επιθέσεις, όπως για παράδειγμα επιθέσεις αλλοίωσης (skewness attack) και επιθέσεις ομοιότητας (similarity attack).

Στην περίπτωση της επίθεσης αλλοίωσης, μπορεί να αλλοιώνονται τα αποτελέσματα και οι πληροφορίες μιας δημοσίευσης, βασισμένα σε λανθασμένα συμπεράσματα που μπορεί να βγάλει ο επιτιθέμενος. Για παράδειγμα εάν σε μια κλάση ισοδυναμίας τυγχάνει να βρίσκονται ίδιος αριθμός θετικών και αρνητικών αποτελεσμάτων, τότε ο επιτιθέμενος υποθέτει ότι η πιθανότητα κάποιος να είναι θετικός είναι 50%, το οποίο όμως στην πραγματικότητα δεν είναι αληθές στο γενικό σύνολο, και αποτελεί αλλοίωση της πληροφορίας που παρέχει ο πίνακας.

Στην περίπτωση της επίθεσης ομοιότητας, ένας πίνακας μπορεί να ικανοποιεί την l-διαφορετικότητα, και συνεπώς σε μια κλάση ισοδυναμίας να υπάρχουν πράγματι l διαφορετικές διακριτές τιμές, οι οποίες όμως να είναι σημασιολογικά παρόμοιες, και έτσι να αποκαλύπτεται και πάλι κάποια πληροφορία, παρόμοια με την περίπτωση της επίθεσης ομοιογένειας κατά της k-ανωνυμίας.

	ΤΚ	Ηλικία	Μισθός	Ασθένεια
1	476**	2*	300	έλκος
2	476**	2*	400	γαστρίτιδα
3	476**	2*	500	γαστρεντερίτιδα
4	4790*	≥40	600	γαστρίτιδα
5	4790*	≥40	1100	ίωση
6	4790*	≥40	800	βρογχίτιδα
7	476**	3*	700	βρογχίτιδα
8	476**	3*	900	πνευμονία
9	476**	3*	1000	γαστρεντερίτιδα

Ο παραπάνω πίνακας ικανοποιεί την 3-διαφορετικότητα, και πράγματι κάθε κλάση έχει 3 διαφορετικές διακριτές τιμές στα ευαίσθητα γνωρίσματα {Μισθός, Ασθένεια}. Ωστόσο εάν ο επιτιθέμενος γνωρίζει ότι κάποιος βρίσκεται στη $1^{η}$ κλάση για παράδειγμα, παρά το ότι αυτή έχει τρεις διαφορετικές τιμές στο γνώρισμα της ασθένειας, και οι τρείς ασθένειες έχουν μια

σημασιολογική συγγένεια καθώς σχετίζονται με στομαχικά προβλήματα. Η πληροφορία αυτή λοιπόν αποκαλύπτεται στον επιτιθέμενο, παρά το ότι ικανοποιείται η l-διαφορετικότητα

Γενικά βλέπουμε πως πίνακες με την ίδια διαφορετικότητα, μπορεί να παρέχουν πολύ διαφορετικά επίπεδα ιδιωτικότητας, είτε λόγω της σημασιολογικής συγγένειας των τιμών των γνωρισμάτων τους, είτε λόγω διαφοράς στο βαθμό ευαισθησίας της πληροφορίας. Η t-εγγύτητα (t - closeness) έρχεται να δώσει μια λύση στο παραπάνω πρόβλημα, εισάγοντας ως παράμετρο προς υπολογισμό τη διαφορά στη γνώση που αποκτά ο επιτιθέμενος από το τη μελέτη μιας κλάσης ισοδυναμίας σε σχέση με τη διαφορά της γνώσεις που αποκτά από τη μελέτη του συνόλου. Τίθεται δηλαδή προς υπολογισμό και η κατανομή των τιμών του ευαίσθητου γνωρίσματος ανάμεσα στις κλάσεις ισοδυναμίας και το σύνολο των δεδομένων.

Πιο συγκεκριμένα, και σύμφωνα με τους [LLV07], μια κλάση ισοδυναμίας ικανοποιεί την t-εγγύτητα όταν η απόσταση ανάμεσα στη κατανομή των τιμών του ευαίσθητου γνωρίσματος μέσα στη κλάση ισοδυναμίας, και τη κατανομή των τιμών του ευαίσθητου γνωρίσματος στο σύνολο του πίνακα, δεν υπερβαίνει ένα όριο t. Όπως και με την l-διαφορετικότητα, ένας πίνακας ικανοποιεί την t-εγγύτητα όταν όλες οι κλάσεις ισοδυναμίας του την ικανοποιούν. Όσο πιο κοντά βρίσκονται οι δύο κατανομές (δηλαδή όσο μικρότερο το t) τόσο αυστηρό είναι το κριτήριο της ιδιωτικότητας, αλλά και τόσο περιορίζεται και η χρηστικότητα της πληροφορίας.

Για τη μέτρηση της απόστασης των δύο κατανομών, προτείνεται η χρήση της μετρικής Earth's Mover's distance (EMD). Η μετρική αυτή βασίζεται στον υπολογισμό της ελάχιστης ενέργειας που απαιτείται για την μετατροπή της μίας κατανομής στην άλλη. Η μετρική EMD είναι δυνατόν να υπολογιστεί τόσο για αριθμητικά γνωρίσματα, όσο και κατηγορικά.

Είναι ενδιαφέρον να σημειωθεί πως η t-εγγύτητα αντιμετωπίζει τις επιθέσεις ομοιογένειας και επιθέσεις με πρότερη γνώση εναντίον της k-ανωνυμίας, όχι εξασφαλίζοντας ότι δεν μπορούν να συμβούν, αλλά εξασφαλίζοντας πως εάν συμβούν, τότε όμοια θα μπορούσαν να έχουν συμβεί και σε έναν πλήρως γενικευμένο πίνακα. Εξασφαλίζουν εν ολίγοις πως εάν αποφασιστεί η δημοσίευση ενός πίνακα, τότε ο πίνακας ο οποίος ικανοποιεί την t-εγγύτητα, είναι ο καλύτερος που μπορεί να επιτευχθεί ως προς αυτές τις επιθέσεις.

2.4 k^m - ανωνυμία

Η k^m – ανωνυμία, όπως έχει αναπτυχθεί από τους M. Terrovitis, N. Mamoulis, P. Kanlis [TMK08], δεν χωρίζει τα δεδομένα σε ευαίσθητα και μη ευαίσθητα, αλλά αντιμετωπίζει όλα τα δεδομένα ως πιθανά ψευδοαναγνωριστικά και πιθανά ευαίσθητα δεδομένα. Έτσι, εάν υποθέσουμε μια βάση δεδομένων η οποία να περιέχει πληροφορίες για προϊόντα τα οποία αγοράστηκαν από μια αλυσίδα σουπερμάρκετ, μπορούμε εύκολα να διαπιστώσουμε ότι η δημοσιοποίηση ενός πίνακα της εν λόγω βάσης θα μπορούσε να αποκαλύψει το πρόσωπο που συνδέεται μια ορισμένη πλειάδα στοιχείων, εάν ο επιτιθέμενος έχει κάποια γνώση ενός υποσυνόλου των προϊόντων που αγόρασε το πρόσωπο αυτό. Έτσι για παράδειγμα, υποθέτουμε πως κάποιος πραγματοποίησε κάποια ψώνια σε μια επίσκεψή του στο συγκεκριμένο σουπερμάρκετ, τα οποία περιελάμβαναν και καφέ, ψωμί, βούτυρο, γάλα, πράσινο τσάι και χαρτοπετσέτες. Στο δρόμο της επιστροφής, κάποιος γνωστός του είδε τα προϊόντα που βρίσκονταν πάνω πάνω στις σακούλες, τα οποία ήταν το ψωμί, το βούτυρο και οι χαρτοπετσέτες. Εάν στο δημοσιευμένο πίνακα με τις αγορές εκείνης της μέρας, υπήρχε μόνο μια εγγραφή που να περιείχε ψωμί, βούτυρο και χαρτοπετσέτες, αμέσως γνωστός αποκτά γνώση όλων των προϊόντων τα οποία είχε αγοράσει το αρχικό πρόσωπο.

Με βάση τον ορισμό των [TMK08] για την k^m – ανωνυμία, εάν ο επιτιθέμενος έχει μέγιστη γνώση το πολύ m στοιχείων, τότε ένας πίνακας ο οποίος ικανοποιεί την k^m – ανωνυμία θα αποτρέπει την αναγνώριση μια εγγραφής, ανάμεσα σε τουλάχιστον k άλλες. Με άλλα λόγια, για κάθε υποσύνολο m ή λιγότερων στοιχείων, θα πρέπει να υπάρχουν στον πίνακα τουλάχιστον k εγγραφές οι οποίες να περιέχουν το υποσύνολο αυτό. Έτσι, στο παραπάνω παράδειγμα, ο επιτιθέμενος που είχε γνώση τριών στοιχείων, δε θα μπορούσε να αναγνωρίσει τα ψώνια του γνωστού του ανάμεσα σε 5 άλλες συναλλαγές εάν ο δημοσιοποιημένος πίνακας ήταν 5^3 – ανώνυμος.

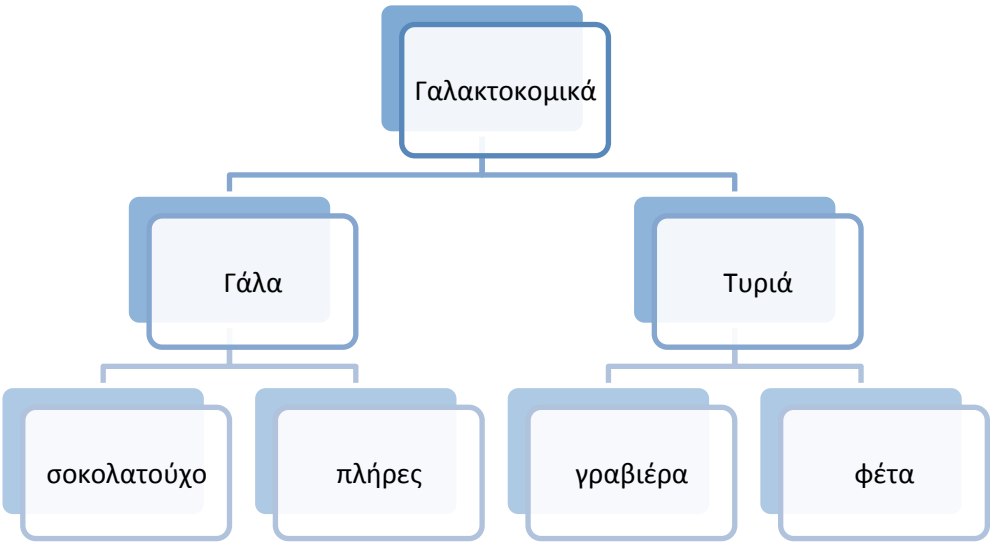
Έτσι για παράδειγμα, ο παρακάτω πίνακας δεν είναι 2^2 – ανώνυμος, καθώς ένας επιτιθέμενος με γνώση δύο συγκεκριμένων προϊόντων (σοκολατούχο γάλα και γραβιέρα) μπορεί να αναγνωρίσει την $1^η$ συναλλαγή ανάμεσα σε 2 άλλες.

Α/Α	Προϊόντα
1	Σοκολατούχο γάλα, γραβιέρα, φέτα
2	Πλήρες γάλα, γραβιέρα
3	Πλήρες γάλα, γραβιέρα, φέτα
4	Σοκολατούχο γάλα, πλήρες γάλα, φέτα

Για την επίτευξη της k^m – ανωνυμοποίησης προτείνεται η χρήση της μεθόδου της γενίκευσης. Συγκεκριμένα αξιοποιείται μια προκαθορισμένη ιεραρχία γενίκευσης, βάσει της οποίας τα πιο

ειδικά στοιχεία του πίνακα αντικαθίστανται με γενικότερα, έως ότου επιτευχθεί η k^m – ανωνυμοποίηση.

Έτσι, για παράδειγμα, μπορεί να αξιοποιηθεί η παρακάτω ιεραρχία για την k^m – ανωνυμοποίηση του πίνακα του παραπάνω παραδείγματος.



Στο παραπάνω παράδειγμα, και αξιοποιώντας την παραπάνω ιεραρχία γενίκευσης, γενικεύοντας το σοκολατούχο και πλήρες γάλα σε Γάλα, προκύπτει ο παρακάτω πίνακας:

A/A	Προϊόντα
1	Γάλα, γραβιέρα, φέτα
2	Γάλα, γραβιέρα
3	Γάλα, γραβιέρα, φέτα
4	Γάλα, φέτα

Ο πίνακας αυτός μπορούμε πλέον να διαπιστώσουμε πως ικανοποιεί την 2^2 – ανωνυμία, καθώς ένα επιτιθέμενος με γνώση το πολύ 2 στοιχείων του πίνακα, δεν είναι σε θέση να αναγνωρίσει μια συναλλαγή ανάμεσα σε τουλάχιστον 2 άλλες.

Ένα σημαντικό ζήτημα στην προσπάθεια να διαπιστώσουμε εάν μια συγκεκριμένη γενίκευση μπορεί να προσφέρει k^m – ανωνυμία, είναι η δυνατότητα εύκολου και αποτελεσματικού προσδιορισμού του πλήθους εμφάνισης όλων των δυνατών συνδυασμών m στοιχείων που εμφανίζονται στον πίνακα, με γρήγορο και αποτελεσματικό τρόπο, χωρίς να απαιτείται η σάρωσης ολόκληρου του πίνακα για τον έλεγχο κάθε πιθανής γενίκευσης. Για την επίτευξη αυτού, οι [TMK08] προτείνουν την χρήση ενός δέντρου μέτρησης. Το δέντρο μέτρησης αποτελεί μια δομή δεδομένων, η οποία μπορεί να αξιοποιηθεί για την διερεύνηση του πλήθους όλων των συνδυασμών m στοιχείων ενός πίνακα, αλλά και όλων των πιθανών

γενικεύσεων αυτών. Η δομή αυτή αναπαριστάται ως ένα δέντρο ύψους m, το οποίο περιέχει όλους τους πιθανούς συνδυασμούς στοιχείων και πιθανών γενικεύσεων. Κάθε κόμβος του δέντρου κρατά το πλήθος εμφάνισης του κλαδιού στον πίνακα. Το δέντρο αυτό μπορεί να δημιουργηθεί με μια σάρωση του πίνακα εάν τα στοιχεία του είναι γνωστά, και από κει και πέρα μπορεί να χρησιμοποιείται για τον υπολογισμό του πλήθους εμφάνισης ενός δεδομένου συνδυασμού, χωρίς να απαιτείται η σάρωση ολόκληρου του πίνακα κάθε φορά.

Φυσικά, η δημιουργία του δέντρου μέτρησης είναι από μόνη της απαιτητική και χρονοβόρα, καθώς πρέπει για κάθε γραμμή του πίνακα να βρεθούν όλοι οι συνδυασμοί στοιχείων και πιθανών γενικεύσεων μεγέθους έως m . Για τον αποδοτικότερο υπολογισμό του δέντρου μέτρησης προτείνεται η χρήση της apriori αρχής, βάσει της οποίας εκμεταλλευόμαστε το γεγονός όπου εάν ένας συνδυασμός μήκους i, παραβιάζει την k^m – ανωνυμία, τότε και κάθε υπερσύνολο αυτού του συνδυασμού θα την παραβιάζει επίσης. Έτσι, στο παραπάνω παράδειγμα, εάν η γνώση ότι τα προϊόντα σοκολατούχο γάλα και γραβιέρα περιέχονται σε μια συναλλαγή μπορεί να παραβιάσει την k^m – ανωνυμία του πίνακα, τότε την παραβιάζουν και όλοι οι υπόλοιποι συνδυασμοί που περιέχουν τα δύο αυτά προϊόντα.

Η χρήση της apriori αρχής είναι ιδιαιτέρως σημαντική, καθώς δίνει τη δυνατότητα περιορισμού των συνδυασμών που μελετώνται, και μειώνει τις περιπτώσεις που πρέπει να ληφθούν υπ' όψιν και να μελετηθούν. Ο αλγόριθμος που χρησιμοποιείται για την αξιοποίηση της αρχής, χτίζει το δέντρο μέτρησης ένα επίπεδο τη φορά, και ανάμεσα σε κάθε επίπεδο ελέγχει για τυχόν παραβιάσεις της km – ανωνυμίας. Πιο συγκεκριμένα:

- Για κάθε i από 1 έως m κατασκευάζουμε το δέντρο μέτρησης ύψους i, με όλους τους πιθανούς συνδυασμούς γενίκευσης.
- Ελέγχουμε εάν κάποιος από τους πιθανούς συνδυασμούς παραβιάζει την km – ανωνυμία και τον αφαιρούμε από τους πιθανούς συνδυασμούς αντικαθιστώντας τον με τη βέλτιστη γενίκευσή του. Με αυτόν τον τρόπο εξασφαλίζουμε ότι κάθε επίπεδο του δέντρου καθώς χτίζεται, αποτελείται μόνο από συνδυασμούς οι οποίοι ικανοποιούν την k^i – ανωνυμία.
- Στην επόμενη επανάληψη, το δέντρο μέτρησης κατασκευάζεται χωρίς τους πιθανούς συνδυασμούς που αφαιρέθηκαν σε αυτή την επανάληψη, μειώνοντας έτσι τις περιπτώσεις που έχουμε να αναπτύξουμε.

Παρά το γεγονός ότι με τον παραπάνω αλγόριθμο ο πίνακας πρέπει να σαρωθεί m φορές αντί για μία όπως κάνει ο αρχικός αλγόριθμος χωρίς την χρήση της apriori αρχής, εντούτοις αποδεικνύεται πως το όφελος από τον περιορισμό των συνδυασμών που πρέπει να εξετασθούν, και η δημιουργία μικρότερου δέντρου μέτρησης κάθε φορά, καθιστούν τον αλγόριθμο αυτόν γρηγορότερο.

3

Ορισμός προβλήματος

Η ανάγκη για προστασία της ιδιωτικότητας σε περιπτώσεις δημοσίευσης δεδομένων είναι αδιαμφισβήτητη. Στο κεφάλαιο που ακολουθεί θα παρουσιαστεί το πρόβλημα της προστασίας της ιδιωτικότητας, και η επίτευξη αυτής μέσω της ανωνυμοποίησης δεδομένων. Στο κεφάλαιο που προηγήθηκε αναλύθηκαν πολλοί διαφορετικοί αλγόριθμοι οι οποίοι έχουν προταθεί για την αντιμετώπιση του ζητήματος της προστασίας της ιδιωτικότητας κατά τη δημοσίευση δεδομένων. Στα πλαίσια της παρούσας εργασίας θα επικεντρωθούμε στην ανωνυμοποίηση δεδομένων ικανοποιώντας την εγγύηση της k^m-ανωνυμίας.

3.1 k^m-ανωνυμία

Ο περιορισμός που τίθεται για την επίτευξη της k^m-ανωνυμίας όπως έχει αναφερθεί, είναι να μην μπορεί να αναγνωριστεί μια συγκεκριμένη εγγραφή του συνόλου δεδομένων, ανάμεσα σε μια ομάδα k τουλάχιστον άλλων εγγραφών, με δεδομένη τη γνώση m το πολύ στοιχείων της συγκεκριμένης εγγραφής.

Εάν υποθέσουμε πως ένα νοσοκομείο για τις ανάγκες μιας στατιστικής ανάλυσης δημοσιεύει το ιστορικών των ασθενών που εισήχθησαν σε αυτό το τελευταίο μήνα, θα πρέπει ο εκδότης των δεδομένων (δηλαδή το νοσοκομείο) να φροντίσει και για την προστασία των ασθενών του, καθώς το ιατρικό τους ιστορικό εμπεριέχει προσωπικά δεδομένα τα οποία δεν θα έπρεπε να περιέλθουν σε γνώση οποιουδήποτε τρίτου.

Ονοματεπώνυμο	Ηλικία	Ιατρικό Ιστορικό
Δημητρίου Νίκος	27	Ανεμοβολογιά, Φυματίωση, Λαρυγγίτιδα
Αντωνόπουλος Τάκης	32	Ακμή, Ιλαρά, Πνευμονία, Ερπητοϊός
Παπαδάκη Μαρία	44	Ανεμοβλογιά, Διφθερίτιδα, Λαρυγγίτιδα
Καλλίκα Ευτυχία	36	Έκζεμα, Τέτανος, Διφθερίτιδα
Σωτηρίου Ιωάννης	51	Ανεμοβολογιά, Ηπατίτιδα C, Λαρυγγίτιδα
Ευτυχίδου Δήμητρα	78	Έκζεμα, Πνευμονία, Φαρυγγίτιδα

Η πρώτη σκέψη θα ήταν να αποκρύψουμε τα βασικά αναγνωριστικά στοιχεία, όπως το όνομα και η ηλικία των ασθενών. Είναι όμως αυτό αρκετό;

Ας υποθέσουμε ότι κάποιος φίλος τους κ. Τάκη Αντωνόπουλου, γνωρίζει ότι ο φίλος του είχε παλαιότερα εμφανίσει ακμή, καθώς επίσης και ότι είχε περάσει κάποτε πνευμονία. Με μια απλή ματιά στα δημοσιευμένα στοιχεία όμως, μπορούμε να διαπιστώσουμε ότι μόνο ένας ασθενής του νοσοκομείου είχε παρουσιάσει κατά το παρελθόν και ακμή και πνευμονία.

Ονοματεπώνυμο	Ηλικία	Ιατρικό Ιστορικό
Δημητρίου Νίκος	27	Ανεμοβολογιά, Φυματίωση, Λαρυγγίτιδα
Αντωνόπουλος Τάκης	32	Ακμή, Ιλαρά, Πνευμονία, Ερπητοϊός
Παπαδάκη Μαρία	44	Ανεμοβλογιά, Διφθερίτιδα, Λαρυγγίτιδα
Καλλίκα Ευτυχία	36	Έκζεμα, Τέτανος, Διφθερίτιδα
Σωτηρίου Ιωάννης	51	Ανεμοβολογιά, Ηπατίτιδα C, Λαρυγγίτιδα
Ευτυχίδου Δήμητρα	78	Έκζεμα, Πνευμονία, Φαρυγγίτιδα

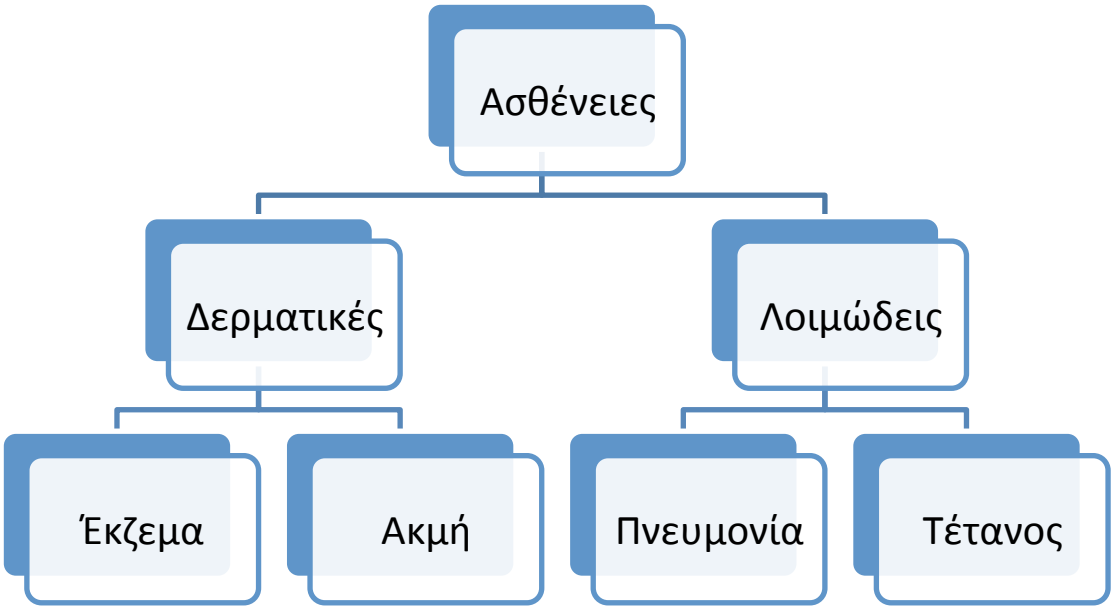
Αν και το όνομα του κ. Αντωνόπουλου δεν εμφανίζεται στα δημοσιευμένα στοιχεία, ο φίλος του έχει ήδη καταφέρει να τον αναγνωρίσει ανάμεσα στους ασθενείς, και έχει πλέον γνώση του ιατρικού του ιστορικού, γεγονός ανεπιθύμητο.

Αντίθετα αν κάποιος γνωστός της κας. Παπαδάκη γνωρίζει πως κάποτε είχε περάσει ανεμοβλογιά και λαρυγγίτιδα, δεν θα ήταν σε θέση να την αναγνωρίσει ανάμεσα σε άλλους δύο ασθενείς και συνεπώς να αποκτήσει πρόσβαση στο ιατρικό της ιστορικό.

Ονοματεπώνυμο	Ηλικία	Ιατρικό Ιστορικό
Δημητρίου Νίκος	27	Ανεμοβολογιά, Φυματίωση, Λαρυγγίτιδα
Αντωνόπουλος Τάκης	32	Ακμή, Ιλαρά, Πνευμονία, Ερπητοϊός
Παπαδάκη Μαρία	44	Ανεμοβλογιά, Διφθερίτιδα, Λαρυγγίτιδα
Καλλίκα Ευτυχία	36	Έκζεμα, Τέτανος, Διφθερίτιδα
Σωτηρίου Ιωάννης	51	Ανεμοβολογιά, Ηπατίτιδα C, Λαρυγγίτιδα
Ευτυχίδου Δήμητρα	78	Έκζεμα, Πνευμονία, Φαρυγγίτιδα

Αυτός είναι εν τέλει και ο απώτερος σκοπός της k^m-ανωνυμοποίησης. Η δυνατότητα προστασίας των προσώπων κατά τη δημοσίευση δεδομένων. Στη πρώτη περίπτωση του παραδείγματός μας, ο επιτιθέμενος είχε γνώση 2 στοιχείων μιας εγγραφής, και αυτό του αρκούσε για να την αναγνωρίσει ανάμεσα στις υπόλοιπες. Αντίστοιχα, στο δεύτερο παράδειγμα ο επιτιθέμενος, αν και είχε γνώση 2 στοιχείων μιας συγκεκριμένης εγγραφής, δεν κατάφερε να αναγνωρίσει τη συγκεκριμένη εγγραφή ανάμεσα σε άλλες.

Με στόχο την ανωνυμοποίηση των δεδομένων, η τεχνική της km-ανωνυμοποίησης χρησιμοποιεί μια προκαθορισμένη ιεραρχία, ώστε να γενικεύσει τα δεδομένα που δημοσιεύονται, και να πετύχει με αυτό τον τρόπο την ύπαρξη περισσότερων κοινών συνδυασμών μέσα στο σύνολο αυτών.



Έτσι αν υποθέσουμε την παραπάνω ιεραρχία, τα στοιχεία, προτού δημοσιευτούνε θα τροποποιούνταν ως ακολούθως:

Ονοματεπώνυμο	Ηλικία	Ιατρικό Ιστορικό
Δημητρίου Νίκος	27	Ανεμοβολογιά, Φυματίωση, Λαρυγγίτιδα
Αντωνόπουλος Τάκης	32	Δερματικές, Ιλαρά, Λοιμώδεις, Ερπητοϊός
Παπαδάκη Μαρία	44	Ανεμοβλογιά, Διφθερίτιδα, Λαρυγγίτιδα
Καλλίκα Ευτυχία	36	Δερματικές, Λοιμώδεις, Διφθερίτιδα
Σωτηρίου Ιωάννης	51	Ανεμοβολογιά, Ηπατίτιδα C, Λαρυγγίτιδα
Ευτυχίδου Δήμητρα	78	Δερματικές, Λοιμώδεις, Φαρυγγίτιδα

Με την τελευταία τροποποίηση των δεδομένων, ο γνωστός του κ. Τάκη μπορεί μεν να γνωρίζει ότι είχε άσθμα και πνευμονία, όμως πλέον δεν μπορεί να τον αναγνωρίσει αμέσως ανάμεσα στους υπολοίπους 3 ασθενείς που είχαν επίσης περάσει κάποια δερματική και λοιμώδη νόσο.

3.2 Δυναμικές Ιεραρχίες

Όπως διαπιστώνουμε, βασική παράμετρος της k^m – ανωνυμοποίησης όπως αυτή αναλύθηκε παραπάνω, αποτελεί η ύπαρξη μιας ιεραρχίας γενίκευσης, βάσει της οποίας θα πρέπει να γενικευτούν τα στοιχεία τα οποία προκαλούν παραβιάσεις στην ικανοποίηση της k^m – ανωνυμίας.

Η ύπαρξη μιας προκαθορισμένης ιεραρχίας μπορεί να μας βοηθήσει κατευθύνοντάς μας κατά την ομαδοποίηση και γενίκευση των στοιχείων ενός συνόλου δεδομένων, στη προσπάθειά μας για επίτευξη της k^m – ανωνυμίας. Δίνονται συγκεκριμένες ομάδες στοιχείων οι οποίες γενικεύονται ώστε να επιτύχουν το επιθυμητό αποτέλεσμα της ανωνυμοποίησης. Ωστόσο, όπως διακρίναμε και προηγουμένως, η γενίκευση των στοιχείων αυτών γίνεται αν συνόλω για κάθε γενίκευση. Στη περίπτωση δηλαδή που ένα στοιχείο γενικεύεται σε ένα πιο γενικό, ταυτόχρονα γενικεύονται και όλα τα υπόλοιπα στοιχεία τα οποία βρίσκονται κάτω από το νέο γενικευμένο στοιχείο.

Το πρόβλημα λοιπόν με την ύπαρξη μιας προκαθορισμένης ιεραρχίας γενίκευσης, αποτελεί το γεγονός ότι η γενίκευση ενός στοιχείου για την επίτευξη της km – ανωνυμίας, προκαλεί ταυτόχρονα και την γενίκευση όλων των αδελφών – στοιχείων τα οποία βρίσκονται στον ίδιο κλάδο του δέντρου της ιεραρχίας γενίκευσης. Αυτό ως συνέπεια οδηγεί σε άσκοπες γενικεύσεις στοιχείων τα οποία υπό άλλες συνθήκες δεν θα υπήρχε ανάγκη να γενικευτούν, καθώς αυτά από μόνα τους δεν προκαλούν παραβιάσεις στην k^m – ανωνυμία.

Ένα άλλο σημαντικό πρόβλημα με την ύπαρξη προκαθορισμένης ιεραρχίας γενίκευσης αποτελεί το γεγονός ότι η k^m – ανωνυμία δεν επιτυγχάνεται με τον ίδιο τρόπο για τα ίδια σύνολα δεδομένων, αλλά εξαρτάται κάθε φορά από την διάταξη της ιεραρχίας γενίκευσης. Έτσι για παράδειγμα, το ίδιο σύνολο δεδομένων, μπορεί έπειτα από την k^m – ανωνυμοποίησή

του να εξασφαλίσει την k^m – ανωνυμία με κάποιον Α τρόπο, γενικεύοντας συγκεκριμένα στοιχεία σε άλλα γενικότερά του, ακολουθώντας την ιεραρχία γενίκευσης η οποία έχει δοθεί. Ωστόσο, το ίδιο σύνολο δεδομένων, δοθείσης μιας διαφορετικής ιεραρχίας, θα μπορούσε να γενικευτεί με έναν διαφορετικό τρόπο Β, γενικεύοντας πιθανώς διαφορετικά στοιχεία σε διαφορετικές γενικεύσεις, ανάλογα με τη διάταξη της νέας ιεραρχίας που δόθηκε ως για το σύνολο αυτό.

Η διαφοροποίηση αυτή, καθιστά σαφές το γεγονός ότι η k^m – ανωνυμοποίηση, επιτυγχάνεται με διαφορετικούς τρόπους για διαφορετικές ιεραρχίες, και επομένως και με διαφορετικά κόστη απώλειας πληροφορίας για διαφορετικές ιεραρχίες. Συνεπώς τίθεται το ερώτημα της επιβεβαίωσης ύπαρξης μια βέλτιστης λύσης. Μπορεί λοιπόν να είμαστε σίγουροι ότι με τον αλγόριθμο που θα χρησιμοποιήσουμε για την k^m – ανωνυμοποίηση του συνόλου δεομένων μας θα μπορούμε να έχουμε εξασφαλίσει τη βέλτιστη λύση του προβλήματος δοθείσης μιας συγκεκριμένης ιεραρχίας, όμως μπορεί το ίδιο σύνολο δεδομένων, με της χρήση μιας ενδεχομένως διαφορετικής διάταξης ιεραρχίας γενίκευσης να ήταν σε θέση να εξασφαλίσει την km – ανωνυμία με ακόμη αποδοτικότερο τρόπο, έχοντας ακόμη μικρότερο κόστος απώλειας πληροφορίας.

Καθίσταται σαφές λοιπόν, πως αν και η ύπαρξη μιας προκαθορισμένης ιεραρχίας γενίκευσης αποτελεί έναν καλό οδηγό για τη γενίκευση στοιχείων κατά την k^m – ανωνυμοποίηση ενός συνόλου δεδομένων, εντούτοις ενέχει και ορισμένους περιορισμούς. Στα πλαίσια της παρούσας εργασίας θα εισάγουμε έναν νέο τρόπο υπολογισμού των γενικεύσεων που είναι απαραίτητο να γίνουν, με στόχο να κάμψουμε τους παραπάνω περιορισμούς που περιγράφηκαν. Θα μελετηθεί η δυνατότητα επίτευξης της k^m – ανωνυμίας, χωρίς την ύπαρξη κάποιας προκαθορισμένης ιεραρχίας γενίκευσης.

Πιο συγκεκριμένα θα επιχειρηθεί η επίτευξη της k^m – ανωνυμίας με την ανάπτυξη μιας ιεραρχίας γενίκευσης με δυναμικό τρόπο κατά την εκτέλεση του αλγορίθμου της ανωνυμοποίησης. Με αυτόν τον τρόπο εξασφαλίζεται ότι κάθε φορά γενικεύονται μόνο τα στοιχεία του συνόλου δεδομένων τα οποία είναι απαραίτητο να γενικευτούν προκειμένου να επιτευχθεί η km – ανωνυμία, αποτρέποντας με αυτόν τον τρόπο τις άσκοπες γενικεύσεις οι οποίες συνέβαιναν κατά την γενίκευση στοιχείων βάσει μιας προκαθορισμένης ιεραρχίας γενίκευσης και επιτυγχάνοντας καλύτερα αποτελέσματα ως προς την απώλεια πληροφορίας, του αρχικού συνόλου.

Η ανάπτυξη της απαραίτητης ιεραρχίας με δυναμικό τρόπο, αποδεσμεύει επίσης τα αποτελέσματα της k^m – ανωνυμοποίησης από την εκάστοτε δοθείσα ιεραρχία γενίκευσης. Με αυτόν τον τρόπο εξασφαλίζεται το γεγονός ότι η k^m – ανωνυμία επιτυγχάνεται με τον ίδιο τρόπο πάντοτε για τα ίδια σύνολα, απομακρύνοντας τις αμφιβολίες περί ύπαρξης μιας ακόμη

καλύτερης λύσης με χαμηλότερο κόστος απώλειας πληροφορίας σε περίπτωση χρήσης μιας διαφορετικής ιεραρχίας.

Τέλος, η δυναμική ανάπτυξη ιεραρχιών γενίκευσης κατά την εκτέλεση του αλγορίθμου της k^m – ανωνυμοποίησης, θα δώσει επιπλέον τη δυνατότητα εκτέλεσης της ανωνυμοποίησης και σε σύνολα για τα οποία δεν υπάρχει κάποια προκαθορισμένη ιεραρχία, διευρύνοντας καθ' αυτόν τον τρόπο ακόμη περισσότερο το πεδίο εφαρμογής της τεχνικής της k^m – ανωνυμοποίησης για την εξασφάλιση της προστασίας της ιδιωτικότητας κατά τη δημοσίευση δεδομένων.

4

Περιγραφή αλγορίθμου

Στο κεφάλαιο αυτό θα αναλύσουμε τον αλγόριθμο που χρησιμοποιήθηκε και πως αυτός υλοποιήθηκε.

4.1 Περιγραφή αλγορίθμου

Ο αλγόριθμός μας αποσκοπεί στην k^m-ανωνυμοποίηση ενός δοθέντος συνόλου δεδομένων. Κατά την εκτέλεση του αλγορίθμου δημιουργείται δυναμικά μια ιεραρχία γενίκευσης των δεδομένων. Η γενίκευση των δεδομένων βάσει της ιεραρχίας που δημιουργείται είναι και αυτή που εξασφαλίζει την k^m-ανωνυμοποίηση του συνόλου εν τέλει.

Υποθέτοντας ότι ο επιτιθέμενος γνωρίζει το πολύ m στοιχεία μιας συγκεκριμένης εγγραφής των δεδομένων, με την k^m-ανωνυμοποίηση θέλουμε να επιτύχουμε να μην μπορεί να αναγνωρίσει την εγγραφή αυτή ανάμεσα σε k ξεχωριστές εγγραφές. Με άλλα λόγια, για οποιοδήποτε ομάδα στοιχείων μεγέθους μικρότερο ή ίσο του m, θα πρέπει να υπάρχουν τουλάχιστον k εγγραφές μέσα στις οποίες να εμφανίζεται. Για να το πετύχουμε αυτό, εφόσον το αρχικό μας σύνολο δεν εξασφαλίζει την km-ανωνυμία, αντικαθιστούμε ορισμένα από τα στοιχεία του συνόλου με κάποια γενίκευσή τους από την ιεραρχία γενίκευσης. Με αυτόν τον τρόπο επιτυγχάνουμε διαφορετικά στοιχεία του συνόλου να γενικεύονται σε ίδια, καθιστώντας με αυτόν τον τρόπο δυνατή την ομαδοποίησή τους σε τουλάχιστον k ίδιες ομάδες μεγέθους το πολύ m. Στόχος μας είναι κατά την διερεύνηση των πιθανών γενικεύσεων των στοιχείων του συνόλου δεδομένων, να βρούμε μια γενίκευση που να

εξασφαλίζει την k^m-ανωνυμοποίηση όπως αυτή περιγράφηκε παραπάνω, επιτυγχάνοντας την κατά το δυνατόν μικρότερη απώλεια πληροφορίας από το αρχικό μας σύνολο.

Η ιεραρχία γενίκευσης χτίζεται και αυτή δυναμικά κατά την εκτέλεση του αλγορίθμου. Αρχικά κανένα στοιχείο της βάσης δεν είναι γενικευμένο. Εκτελώντας τον αλγόριθμο και αναζητώντας μια γενίκευση που να εξασφαλίζει την k^m-ανωνυμία, ελέγχουμε τους πιθανούς συνδυασμούς ομαδοποίησης των προς γενίκευση στοιχείων, χτίζοντας καθ' αυτόν τον τρόπο την ιεραρχία μας. Και εδώ, η επιλογή της κατάλληλης ομαδοποίησης των στοιχείων για τη δημιουργία μιας γενίκευσης γίνεται με κριτήριο τους δύο στόχους μας, δηλαδή την εξασφάλιση της k^m-ανωνυμίας και την κατά το δυνατό μικρότερη απώλεια πληροφορίας από το αρχικό μας σύνολο.

Η είσοδος του αλγορίθμου είναι το σύνολο των δεδομένων $RT(A_1,A_2,\ldots,A_n)$, όπου $A_1,A_2,\ldots,A_n$ οι διαφορετικές εγγραφές, η παράμετρος m που αντιπροσωπεύει την μέγιστη γνώση του επιτιθέμενου των στοιχείων μιας εγγραφής του συνόλου, και η παράμετρος ανωνυμίας k που υποδηλώνει τον ελάχιστο αριθμό εγγραφών ανάμεσα στις οποίες δεν θα πρέπει ο επιτιθέμενος να μπορεί να αναγνωρίσει την εγγραφή μέρος της οποίας έχει γνώση. Με την αλλαγή των παραμέτρων k και m μπορεί να καθορίζεται ο βαθμός προστασίας των δεδομένων, αυξάνοντας λιγότερο ή περισσότερο τις απαιτήσεις για την αναγνώριση μιας εγγραφής του συνόλου από τον επιτιθέμενο. Σαφώς η αλλαγή των παραμέτρων αυτών επηρεάζει άμεσα τόσο τον χρόνο εκτέλεσης του αλγορίθμου, όσο και την απώλεια πληροφορίας της αρχικής βάσης καθώς όσο αυξάνεται για παράδειγμα ο αριθμός εγγραφών k στις οποίες πρέπει να απαντάται κάθε σύνολο m στοιχείων, τόσο περισσότερες γενικεύσεις ενδεχομένως να πρέπει να γίνουν ώστε να αυξάνονται οι ομάδες όμοιων στοιχείων στις εγγραφές του συνόλου.

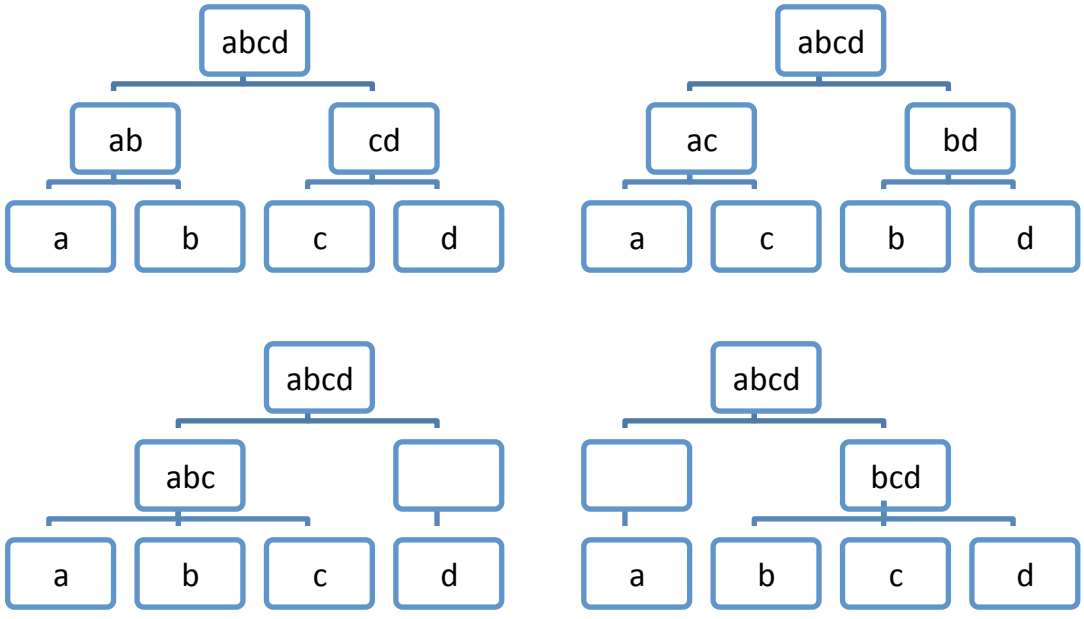
Καθώς το πλήθος των δεδομένων αυξάνεται, η διερεύνηση όλων των πιθανών γενικεύσεων για την εύρεση της καταλληλότερης γίνεται εξαντλητική. Για αυτό το λόγο ο αλγόριθμος χρησιμοποιεί ευριστικές μεθόδους ώστε να περιορίσει τις προς διερεύνηση πιθανές γενικεύσεις, εντοπίζοντας τα στοιχεία εκείνα που παραβιάζουν την k^m-ανωνυμοποίηση και επικεντρώνεται σε αυτά ψάχνοντας τοπικές λύσεις που εξασφαλίζουν την απαιτούμενη ανωνυμοποίηση και σε μεγαλύτερα σύνολα.

4.2 Βοηθητικοί αλγόριθμοι

Ο αλγόριθμος της k^m-ανωνυμοποίησης στηρίζεται στην υλοποίηση τεσσάρων βοηθητικών αλγορίθμων τους οποίους και θα αναλύσουμε παρακάτω. Ο πρώτος κατασκευάζει ένα δέντρο μέτρησης το οποίο μας δίνει πληροφορίες για το πλήθος εμφάνισης των πιθανών

συνδυασμών των στοιχείων σε εγγραφές της βάσης. Ο δεύτερος εξετάζει τις πιθανές γενικεύσεις της ιεραρχίας και ποιες από αυτές εξασφαλίζουν την βέλτιστη ανωνυμοποίηση. Ο τρίτος αλγόριθμος εντοπίζει τις πιθανές παραβιάσεις στη βάση και αναζητά μέσω του δεύτερου αλγορίθμου τη βέλτιστη λύση για κάθε επιμέρους πρόβλημα. Τέλος, ο τέταρτος αλγόριθμος που χρησιμοποιείται εξετάζει προοδευτικά τις πιθανές παραβιάσεις, επιλύοντάς τες με τη βοήθεια του τρίτου αλγορίθμου εκ των προτέρων, ώστε να μην χρειάζεται να εξετάσει όλες τις νέες πιθανές παραβιάσεις που θα προέκυπταν από τις προηγούμενες εάν αυτές δεν είχαν επιλυθεί.

Στο σημείο αυτό, και πριν προχωρήσουμε στην ανάλυση των προαναφερθέντων αλγορίθμων, θα ήταν καλό να ορίσουμε τι εννοούμε με τον όρο «πιθανές γενικεύσεις». Η ιεραρχία γενίκευσης στον αλγόριθμό μας χτίζεται δυναμικά κατά τη διάρκεια εκτέλεσής του. Αυτό σημαίνει ότι δεν υπάρχει μια εκ των προτέρων καθορισμένη ιεραρχία, αλλά μόνο υποψήφιες πιθανές γενικεύσεις των στοιχείων της βάσης. Έτσι, εάν για παράδειγμα η βάση μας αποτελείται από συνδυασμούς των στοιχείων a,b,c,d τότε υπάρχουν κάποιοι πιθανοί συνδυασμοί των στοιχείων αυτών που θα μπορούσαν να δημιουργήσουν μια ιεραρχία όπως φαίνεται στις παρακάτω εικόνες.



Συνολικά όλες οι πιθανές γενικεύσεις για κάθε ένα από τα στοιχεία της εν λόγω βάσης είναι οι ακόλουθες:

a	b	c	d
ab	ab	ac	ad
ac	bc	bc	bd
ad	bd	cd	cd
abc	abc	abc	abd
abd	abd	acd	acd
acd	bcd	bcd	bcd
abcd	abcd	abcd	abcd

4.2.1 Δέντρο Μέτρησης (count tree)

Για να μπορέσουμε να επιβεβαιώσουμε το κατά πόσο μια γενίκευση εξασφαλίζει την k^m-ανωνυμοποίηση ενός συνόλου δεδομένων, θα πρέπει να μπορούμε να γνωρίζουμε το πλήθος εμφάνισης όλων των πιθανών συνδυασμών στοιχείων της βάσης, μεγέθους μικρότερου ή ίσου του m. Επιπλέον, για να αποφύγουμε να διαβάζουμε ολόκληρο το σύνολο των δεδομένων μας κάθε φορά που θέλουμε να ελέγξουμε μια πιθανή γενίκευση, θα πρέπει να μπορούμε να γνωρίζουμε πως η κάθε γενίκευση θα την επηρεάσει και πως θα αλλάξουν οι πιθανοί συνδυασμοί και το πλήθος εμφάνισής τους σε αυτή. Για τον λόγο αυτό κατασκευάζουμε μια δομή δεδομένων στην οποία θα περιέχονται το πλήθος όλων των πιθανών συνδυασμών μεγέθους m των στοιχείων της βάσης και των πιθανών γενικεύσεων αυτών. Τονίζεται πως η καταγραφή του πλήθους των συνδυασμών στοιχείων μεγέθους m εξασφαλίζει και την δυνατότητα ελέγχου των συνδυασμών μικρότερου μεγέθους καθώς το πλήθος εμφάνισης ενός συγκεκριμένου συνδυασμού στοιχείων θα είναι πάντα μικρότερο ή το πολύ ίσο με το πλήθος εμφάνισης ενός υποσυνόλου του συνδυασμού αυτού.

Για την καταγραφή των πιθανών συνδυασμών και του πλήθους εμφάνισης αυτών στο σύνολο, θα χρησιμοποιήσουμε ένα n-αδικό δέντρο μέτρησης. Η δημιουργία του δέντρου προϋποθέτει την ύπαρξη μιας διάταξης για όλα τα στοιχεία toy συνόλου μας και τις πιθανές γενικεύσεις αυτών. Η διάταξη αυτή μπορεί να είναι βάσει της συχνότητας εμφάνισης του κάθε στοιχείου, βάσει του πλήθους των στοιχείων που γενικεύονται, αλφαβητική ή ακόμη και τυχαία. Για το παραπάνω πρώτο παράδειγμα ιεραρχίας της ενότητας έχουμε μια διάταξη της μορφής {abcd, ab, cd, a, b, c, d}. Κάθε κόμβος του δέντρου μέτρησης θα περιέχει το πλήθος εμφάνισης του συνδυασμού του μέχρι εκεί μονοπατιού. Για παράδειγμα ο κόμβος [a] του

πρώτου επιπέδου περιέχει το πλήθος εμφάνισης του στοιχείου a, ενώ ο κόμβος [cd] που βρίσκεται κάτω από τον κόμβο [a] περιέχει το πλήθος εμφάνισης του συνδυασμού {a, cd} στη βάση. Ακολουθώντας λοιπόν τα μονοπάτια του δέντρου μέτρησης μπορούμε να υπολογίσουμε το πλήθος όλων των συνδυασμών των στοιχείων μεγέθους έως m.

Κατά την εκτέλεση του αλγορίθμου σαρώνεται το δοθέν σύνολο, και κάθε εγγραφή του συνόλου επεκτείνεται προσθέτοντας τις πιθανές γενικεύσεις όλων των στοιχείων αυτής. Για κάθε επεκταμένη εγγραφή πλέον, ο αλγόριθμος βρίσκει όλα τα δυνατά υποσύνολα μεγέθους έως m, υπό την προϋπόθεση να μην υπάρχουν στο υποσύνολο δύο στοιχεία εκ των οποίων το ένα να αποτελεί πιθανή γενίκευση του άλλου, και αναζητά το αντίστοιχο μονοπάτι στο δέντρο μέτρησης, αυξάνοντας το πλήθος του τελικού κόμβου κατά ένα. Ο περιορισμός του να μην υπάρχουν στο υποσύνολο στοιχεία που το ένα να είναι πιθανή γενίκευση του άλλου οφείλεται στο γεγονός ότι σε μια εγγραφή δεν μπορεί να εμφανίζεται ταυτόχρονα ένα στοιχείο με τη γενίκευσή του, καθώς εάν έχει γενικευτεί το συγκεκριμένο στοιχείο, αυτό σημαίνει πως θα έχει αντικατασταθεί από την γενίκευσή του.

Με ένα πέρασμα του συνόλου καθ' αυτόν τον τρόπο, κατασκευάζουμε το δέντρο μέτρησης στο οποίο πλέον υπάρχουν αποθηκευμένα το πλήθος εμφάνισης όλων των συνδυασμών των στοιχείων που εμφανίζονται στο σύνολο δεδομένων, καθώς και των πιθανών γενικεύσεων αυτών. Είναι εύκολο πλέον, με μια ματιά στο δέντρο να διαπιστώσουμε ποιοι συνδυασμοί εμφανίζονται τουλάχιστον k φορές, και άρα εξασφαλίζουν k-ανωνυμία, και ποιοι όχι.

Ακολουθεί ο ψευδοκώδικας του αλγορίθμου δημιουργίας του δέντρου μέτρησης:

Είσοδος: $RT(A_1,A_2,\ldots,A_n)$, αρχικό σύνολο δεδομένων.

k: παράμετρος ανωνυμίας,

m: μέγιστη γνώση επιτιθέμενου

Έξοδος: *CT δέντρο μέτρησης

Βήματα αλγορίθμου

Για κάθε εγγραφή $t = (t_1,t_2,\ldots,t_n) \in RT(A_1,A_2,\ldots,A_n)$

Επέκτεινε την εγγραφή προσθέτοντας τις πιθανές γενικεύσεις των $(t_1,t_2,\ldots,t_n)$

Για κάθε υποσύνολο της επεκταμένης εγγραφής $c \leq m$

Αν $\nexists$ i, j ∈ c τέτοια ώστε το i να αποτελεί γενίκευση του j

Πρόσθεσε το c στο CT δέντρο

Αύξησε το πλήθος του τελευταίου κόμβου κατά ένα

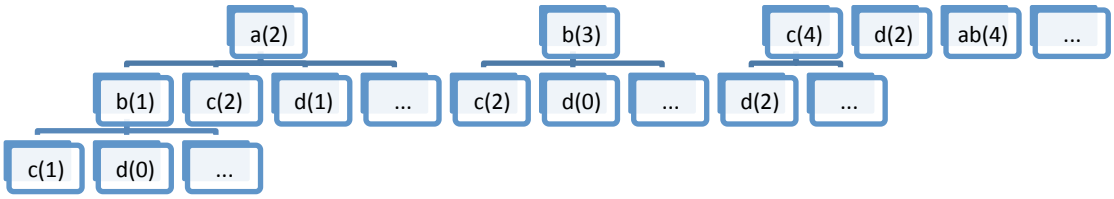
Έστω ότι έχουμε τα ακόλουθα δεδομένα:

t	εγγραφή
t_1	a, b, c
t_2	a, c, d
t_3	b, c
t_4	c, d

Το εκτεταμένο σύνολο βάσει της πρώτης ιεραρχίας του παραδείγματος θα είναι τότε:

t	εγγραφή
t_1	a, b, c, ab, cd, abcd
t_2	a, c, d, ab, cd, abcd
t_3	b, c, ab, cd, abcd
t_4	c, d, cd, abcd

Και το δέντρο μέτρησης θα είναι:



4.2.2 Βέλτιστη ανωνυμοποίηση

Όπως αναφέραμε, στόχος είναι να επιτευχθεί η k^m – ανωνυμία με τη μικρότερη δυνατή απώλεια πληροφορίας. Για τον λόγο αυτό, εκτελούμε τον αλγόριθμο της βέλτιστης ανωνυμοποίησης, ο οποίος αναζητά τη βέλτιστη γενίκευση που εξασφαλίζει την k^m – ανωνυμία με τη μικρότερη δυνατή απώλεια πληροφορίας.

Αρχικά εντοπίζουμε τις πιθανές γενικεύσεις και τμηματικά τις εξετάζουμε. Τοποθετούμε πρώτα σε μια λίστα Q τις πιθανές γενικεύσεις που αντιστοιχούν σε καμία γενίκευση – τα

αρχικά μη γενικευμένα στοιχεία του πίνακα δηλαδή. Για μια μια τις πιθανές γενικεύσεις της λίστας, εξετάζουμε εάν μπορούν να ικανοποιήσουν την k^m – ανωνυμία. Εάν ναι, τότε καθίστανται μια υποψήφια λύση και όλες οι γενικεύσεις αυτών απορρίπτονται ως μη βέλτιστες. Αφότου υπολογίσουμε το κόστος της γενίκευσης, είμαστε σε θέση να το συγκρίνουμε με τη βέλτιστη λύση που έχει βρεθεί έως τώρα, και εφόσον το κόστος είναι μικρότερο, να καταστήσουμε τη νέα λύση ως τη βέλτιστη.

Εάν η πιθανή γενίκευση που εξετάσαμε δεν ικανοποιεί την k^m – ανωνυμία, τότε προσθέτουμε στη λίστα όλες τις άμεσες πιθανές γενικεύσεις αυτής ώστε να εξεταστούν κατά τον ίδιο τρόπο. Όταν η λίστα αδειάσει, δηλαδή αφότου έχουν εξεταστεί ή αποκλειστεί όλες οι πιθανές γενικεύσεις, τότε έχουμε βρει την βέλτιστη λύση, με το μικρότερη κόστος γενίκευσης.

Ακολουθεί ο ψευδοκώδικας του αλγορίθμου Βέλτιστης Ανωνυμοποίησης:

Είσοδος: $RT(A_1,A_2,\ldots,A_n)$, αρχικό σύνολο δεδομένων.

k: παράμετρος ανωνυμίας,

m: μέγιστη γνώση επιτιθέμενου

Έξοδος: c_{opt}: βέλτιστη γενίκευση

Βήματα αλγορίθμου

Προσθέτουμε τη μικρότερη δυνατή γενίκευση σε μια λίστα Q

Όσο η λίστα Q είναι ακόμη γεμάτη

Έλεγξε την επόμενη πιθανή γενίκευση

Αν ικανοποιεί την k^m – ανωνυμία

Απόρριψε όλες τις άμεσες γενικεύσεις της

Αν το κόστος είναι μικρότερο από της έως τώρα βέλτιστης λύσης

Σημείωσε αυτήν ως την έως τώρα βέλτιστη λύση

Αλλιώς

Πρόσθεσε στη λίστα Q όλες τις άμεσες πιθανές γενικεύσεις της

(που δεν έχουν ήδη απορριφθεί)

Επανάλαβε

Επέστρεψε τη βέλτιστη λύση c_{opt} που βρήκες

Σαφώς αντιλαμβανόμαστε πως όσα περισσότερα τα διαφορετικά στοιχεία του πίνακα, τόσες περισσότερες και οι πιθανές γενικεύσεις οι οποίες θα πρέπει να προστεθούν στη λίστα Q και να εξεταστούν. Επομένως όσο αυξάνονται τα στοιχεία του πεδίου ορισμού I, τόσο πιο

απαιτητικός γίνεται και ο παραπάνω αλγόριθμος. Έτσι αξιοποιούνται οι επόμενοι ευριστικοί αλγόριθμοι για την αποδοτικότερη επίτευξη της k^m – ανωνυμίας

4.2.3 Ευθεία ανωνυμοποίηση (Direct Anonymization)

Ο αλγόριθμος της ευθείας ανωνυμοποίησης στοχεύει στον περιορισμό των πιθανών γενικεύσεων που εξετάζουμε σε αυτούς τους οποίους πραγματικά χρειαζόμαστε να εξετάσουμε. Έτσι ο αλγόριθμος σαρώνει το δέντρο μέτρησης, αναζητώντας για πιθανές παραβιάσεις της k^m – ανωνυμίας, και έπειτα αναζητά λύσεις μόνο για τις συγκεκριμένες περιπτώσεις, εξετάζοντας έτσι λιγότερες πιθανές γενικεύσεις.

Πιο συγκεκριμένα, ο αλγόριθμος σαρώνει το δέντρο μέτρησης και ελέγχει εάν υπάρχουν κλαδιά μήκους m, τα οποία να έχουν πλήθος εμφάνισης μικρότερο του k. Εφόσον βρει τέτοια κλαδιά, τα οποία και παραβιάζουν την k^m – ανωνυμία, αναζητά τη βέλτιστη γενίκευση για την επίλυση αυτών μέσω του αλγορίθμου της βέλτιστης ανωνυμοποίησης που περιγράφηκε παραπάνω, περιορίζοντας όμως την αναζήτηση μόνο σε πιθανές γενικεύσεις των στοιχείων που βρίσκονται στον προβληματικό κλάδο που εντόπισε.

Επιπλέον, εάν κατά τη σάρωσης του δέντρου μέτρησης, ο αλγόριθμος βρεθεί σε κόμβο ο οποίος έχει ήδη γενικευτεί, τότε δεν εξετάζει το υπόλοιπο κλαδί, καθώς όλοι οι συνδυασμοί που περιέχουν τον επόμενο κόμβο δεν θα αποτελούν μέρος της λύσης ούτως ή άλλως. Με τον τρόπο αυτόν κερδίζουμε ακόμη καλύτερο υπολογιστικό κόστος, καθώς εάν ο αλγόριθμος εξετάσει κάποιον κόμβο ο οποίος είναι γενίκευση άλλων στοιχείων που δεν έχουν εξεταστεί ακόμη, και χρειαστεί να γενικεύσει περαιτέρω το στοιχείο αυτό, τότε αυτό σημαίνει πως ο αλγόριθμος δεν θα χρειαστεί να εξετάσει καθόλου κόμβους οι οποίοι βρίσκονται χαμηλότερα στην ιεραρχία γενίκευσης.

Ακολουθεί ο ψευδοκώδικας του αλγορίθμου Ευθείας Ανωνυμοποίησης:

Είσοδος: $RT(A_1,A_2,\ldots,A_n)$, αρχικό σύνολο δεδομένων.

k: παράμετρος ανωνυμίας,

m: μέγιστη γνώση επιτιθέμενου

Έξοδος: c: γενίκευση

Βήματα αλγορίθμου

Σαρώνουμε το αρχικό σύνολο δεδομένων κατασκευάζοντας το δέντρο μέτρησης

Για κάθε κόμβο του δέντρου μέτρησης μέσα από τη σάρωσή του

Αν το στοιχείο του κόμβου έχει ήδη γενικευτεί

Υποχώρησε από τον κλάδο αγνοώντας τους υπόλοιπους κόμβους του

Αν το πλήθος εμφάνισης είναι μικρότερο του k

Βρες τις γενικεύσεις του κλάδου J που καθιστούν το J k-ανώνυμο

Πρόσθεσε τις γενικεύσεις που βρήκε στις ήδη υπάρχουσες γενικεύσεις

Υποχώρησε έως ότου κανένα στοιχείο του κλάδου δεν έχει γενικευτεί

Επέστρεψε τη λύση που βρήκες

Ο αλγόριθμος της ευθείας ανωνυμοποίησης σίγουρα χρειάζεται να εξετάσει λιγότερες πιθανές γενικεύσεις από τον αλγόριθμο της βέλτιστης ανωνυμοποίησης, αλλά εξακολουθεί να είναι απαιτητικός καθώς αναγκάζεται να σαρώνει το δέντρο μήκους m για να βρει πιθανές παραβιάσεις της k^m – ανωνυμίας, τη στιγμή κατά την οποία πολλές από τις παραβιάσεις θα μπορούσαν να έχουν βρεθεί και από κλάδους μικρότερου μήκους. Για την αντιμετώπιση του προβλήματος αυτού, αξιοποιούμε τον επόμενο αλγόριθμο, ο οποίος αναμένεται να καταστήσει την εύρεση της k^m – ανώνυμης λύσης που αναζητούμε ακόμη αποδοτικότερη.

4.2.4 Apriori Αλγόριθμος

Ο αλγόριθμος αυτός στηρίζεται στην apriori αρχή βάσει της οποίας εάν ένα σύνολο στοιχείων J μεγέθους i παραβιάζει την k^m – ανωνυμία, τότε και κάθε υπερσύνολο αυτού θα την παραβιάζει επίσης. Έτσι, αξιοποιώντας την αρχή αυτή, μπορούμε να εξετάζουμε το δέντρο γενίκευσης για πιθανές παραβιάσεις σταδιακά, ένα επίπεδο τη φορά, εντοπίζοντας και επιλύοντας τα προβλήματα νωρίτερα στο δέντρο. Με αυτόν τον τρόπο περιορίζονται οι αναζητήσεις που πρέπει να γίνουν σε κάθε επίπεδο αφού, κάποιοι κλάδοι έχουν ήδη περιοριστεί εντελώς από το προηγούμενο κιόλας επίπεδο.

Για την υλοποίηση του αλγορίθμου, ουσιαστικά κατασκευάζουμε κάθε φορά ένα παραπάνω επίπεδο στο δέντρο μέτρησης, και εφαρμόζουμε τον αλγόριθμο της ευθείας ανωνυμοποίησης. Εξασφαλίζουμε έτσι πριν την κατασκευή του επόμενου επιπέδου, ότι όλα τα προηγούμενα επίπεδα ικανοποιούν την k^m – ανωνυμία, έχοντας ήδη κάνει ορισμένες γενικεύσεις και συνεπώς έχοντας ήδη αποκλείσει κάποιους κλάδους οι οποίοι εν συνεχεία αγνοούνται κατά την κατασκευή του επόμενου επιπέδου.

Ακολουθεί ο ψευδοκώδικας του αλγορίθμου Apriori Ανωνυμοποίησης:

Είσοδος: $RT(A_1,A_2,\ldots,A_n)$, αρχικό σύνολο δεδομένων.

k: παράμετρος ανωνυμίας,

m: μέγιστη γνώση επιτιθέμενου

Έξοδος: c: γενίκευση

Βήματα αλγορίθμου

Για i = 1 έως m

Κατασκεύασε τον εκτεταμένο πίνακα με βάση τις έως τώρα γενικεύσεις

Κατασκεύασε το δέντρο μέτρησης ύψους i του εκτεταμένου πίνακα

Εκτέλεσε τον αλγόριθμο Ευθείας Ανωνυμοποίησης στο δέντρο βάθους i

Πρόσθεσε τις γενικεύσεις που βρήκες στις ήδη υπάρχουσες γενικεύσεις.

Επέστρεψε τη λύση που βρήκες

4.3 Δυναμικές ιεραρχίες

Για την υλοποίηση του αλγορίθμου με δυναμικές ιεραρχίες , τροποποιήσαμε την παραπάνω υλοποίηση της k^m-ανωνυμοποίησης, έτσι ώστε να εκτελείται χωρίς να ακολουθεί μια προκαθορισμένη ιεραρχία γενίκευσης, αλλά αναζητώντας και επιλέγοντας την βέλτιστη γενίκευση κάθε φορά.

Για την υλοποίηση του αλγορίθμου αυτού βασιζόμαστε και πάλι στην a priori λογική ότι εάν ένα στοιχείο του δέντρου μέτρησης έχει γενικευτεί, δεν υπάρχει λόγος να εξεταστούν τα παιδιά του, καθώς δεν θα υπάρχουν σα συνδυασμοί στο ανωνυμοποιημένο σύνολο των δεδομένων μας. Έτσι, κάθε επίπεδο του δέντρου μέτρησης εξετάζεται κι εδώ ξεχωριστά.

Σε κάθε επανάληψη του αλγορίθμου, επεκτείνεται το δέντρο μέτρησης, το οποίο περιλαμβάνει όλους τους πιθανούς συνδυασμούς των στοιχείων του συνόλου, μεγέθους κατά ένα μεγαλύτερο από τους συνδυασμούς της προηγούμενης εκτέλεσης, και μέχρι το πολύ μεγέθους m. Όταν κάποιο φύλλο του δέντρου εντοπιστεί να έχει πλήθος εμφάνισης μικρότερο του k, τότε εξετάζονται οι κόμβοι-αδελφοί του συγκεκριμένου φύλλου εάν το πλήθος εμφάνισης κάποιου από τους αδελφούς κόμβους προστιθέμενο στο πλήθος εμφάνισης του «προβληματικού» κόμβου είναι μεγαλύτερο του k, τότε η γενίκευση των δύο κόμβων θα μπορούσε να είναι μια υποψήφια γενίκευση που να εξασφάλιζε την k^m-ανωνυμία για το συγκεκριμένο συνδυασμό. Εάν πάλι το άθροισμα είναι μικρότερο του k, δεν θα μπορούσε η γενίκευση αυτή να αποτελεί λύση του προβλήματος, καθώς το πλήθος εμφάνισής της θα εξακολουθούσε να είναι μικρότερο του k. Με το κριτήριο αυτό κατασκευάζουμε όλες τις πιθανές γενικεύσεις που θα μπορούσαν να επιλύσουν τοπικά σε κάθε κλάδο την παραβίαση της km-ανωνυμίας που εντοπίσαμε.

Αφότου βρεθούν όλες οι υποψήφιες λύσεις, τότε αυτές εξετάζονται, και επιλέγεται η γενίκευση που εξασφαλίζει τη μικρότερη δυνατή απώλεια πληροφορίας. Καθ' αυτόν τον

τρόπο, σε κάθε επανάληψη του αλγορίθμου, χτίζεται δυναμικά η ιεραρχία γενίκευσης που εξασφαλίζει την k^m-ανωνυμοποίηση των δεδομένων μας.

Προφανώς όταν κάποιο στοιχείο του δέντρου μέτρησης έχει ήδη γενικευτεί δεν επεκτείνεται στην επόμενη επανάληψη, μειώνοντας έτσι τα προς εξέταση στοιχεία και εξασφαλίζοντας μικρότερες απαιτήσεις σε χρόνο και μνήμη για την εκτέλεση του αλγορίθμου.

Ακολουθεί ο ψευδοκώδικας του αλγορίθμου:

Είσοδος: $RT(A_1,A_2,\ldots,A_n)$, αρχικό σύνολο δεδομένων

k: παράμετρος ανωνυμίας

m: μέγιστη γνώση επιτιθέμενου

Έξοδος: Generalization Ιεραρχία γενίκευσης

Βήματα αλγορίθμου

Για i = 1 έως m

Επέκτεινε το δέντρο μέτρησης κατά ένα επίπεδο για όλα τα μη-γενικευμένα στοιχεία του δέντρου

Αναζήτησε φύλλα του δέντρου με πλήθος εμφάνισης < k

Αν το πλήθος εμφάνισης του κόμβου supp1 < k

Μέχρις ότου βρεθούν υποψήφιες λύσεις ή αναζητηθούν όλοι οι συνδυασμοί

Αναζήτησε αδελφούς κόμβους με supp2: supp1+supp2 ≥ k

Αν supp1+supp2 ≥ k

Καταχώρησε το συνδυασμό κόμβων ως υποψήφια λύση

Αν δεν βρεθούν υποψήφιες λύσεις

Αύξησε κατά 1 το μέγεθος συνδυασμών κόμβων που αναζητούνται

Επανάλαβε

Εξέτασε υποψήφιες λύσεις

Πρόσθεσε στην ιεραρχία γενίκευσης τη λύση με το μικρότερο κόστος

Ακολουθεί ένα παράδειγμα εκτέλεσης του παραπάνω αλγορίθμου.

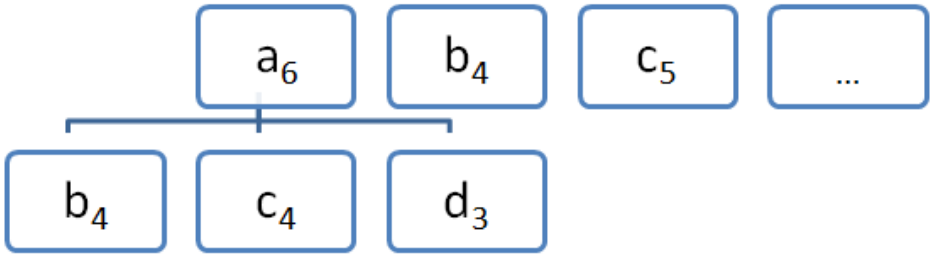
Δίδεται σύνολο δεδομένων που περιέχει τα στοιχεία {a,b,c,d,e,f} και ζητείται η ιεραρχία γενίκευσης που εξασφαλίζει την 3^3-ανωνυμία.

Ο αλγόριθμος θα κατασκευάσει τμηματικά το δέντρο μέτρησης βάθους τριών επιπέδων, και θα δημιουργήσει δυναμικά τις γενικεύσεις που εξασφαλίζουν την 3^3-ανωνυμία

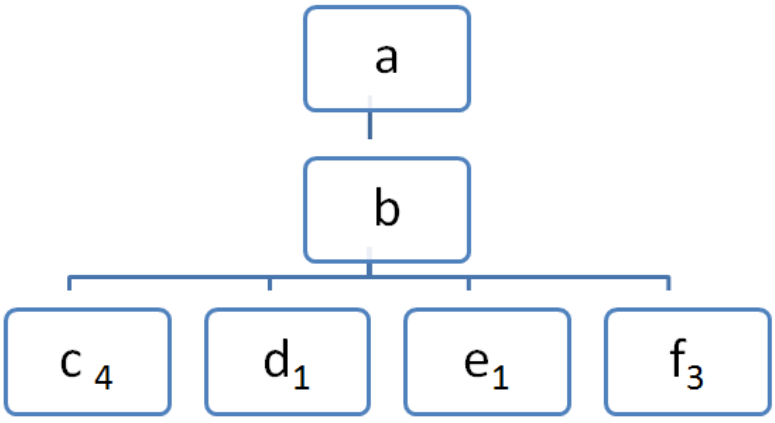
Κατασκευάζουμε το πρώτο επίπεδο του δέντρου μέτρησης, το οποίο περιέχει όλα τα διαφορετικά στοιχεία του συνόλου, και το πλήθος εμφάνισής τους σε αυτό.

Ελέγχουμε τα φύλλα του δέντρου, και διαπιστώνουμε ότι κανένα στοιχείο δεν έχει πλήθος εμφάνισης μικρότερο του 3. Επομένως καμία γενίκευση δεν απαιτείται σε αυτό το επίπεδο.

Επεκτείνουμε το δέντρο μας στο δεύτερο επίπεδο, το οποίο περιέχει όλους τους πιθανούς συνδυασμούς μεγέθους δύο των στοιχείων του συνόλου.



Εξετάζουμε πάλι τα φύλλα όλα του δέντρου , αναζητώντας και πάλι για συνδυασμούς με πλήθος εμφάνισης μικρότερο του k. Εφόσον δεν υπάρχουν τέτοια στοιχεία επεκτείνουμε το δέντρο γενίκευσης και σε τρίτο επίπεδο.



Αναζητούμε και πάλι φύλλα τα οποία να έχουν πλήθος εμφάνισης μικρότερο του k. Στο παράδειγμά μας εντοπίζεται το φύλλο d που αντιστοιχεί στο συνδυασμό a – b – d. Ο συνδυασμός αυτός είναι προβληματικός καθώς εμφανίζεται μόνο 1 φορά στο σύνολο, και επομένως δεν επιτρέπει την 3^3-ανωνυμία. Αναζητούμε λοιπόν ανάμεσα στους αδελφούς κόμβους κάποιον κόμβο του οποίου το πλήθος εμφάνισης να είναι μεγαλύτερο ή ίσο του 2, ώστε το άθροισμα των δύο να είναι μεγαλύτερο ή ίσο του 3.

Διαβάζεται πρώτα ο κόμβος e, ο οποίος έχει πλήθος εμφάνισης 1, και συνεπώς δεν μπορεί να αποτελέσει υποψήφια λύση του προβλήματος. Ακολουθεί ο κόμβος f, ο οποίος έχει πλήθος εμφάνισης για τον συνδυασμό a –b – f. Έτσι η γενίκευση df σημειώνεται ως μια υποψήφια λύση η οποία εξασφαλίζει την 3^3-ανωνυμία του συνόλου δεδομένων. Ελέγχονται και οι υπόλοιποι κόμβοι-αδελφοί κατά τον ίδιο ακριβώς τρόπο, και εφόσον ολοκληρωθεί η αναζήτηση, έχουμε το σύνολο των υποψηφίων λύσεων, έστω df, dh, dj.

Διαβάζουμε το σύνολο των δεδομένων, επεκτείνοντας τα στοιχεία d, f, h και j, και αναζητούμε το πλήθος εμφάνισης των γενικεύσεων df, dh, dj. Πλέον είμαστε σε θέση να γνωρίζουμε το κόστος NCP για κάθε μια από τις υποψήφιες λύσεις. Αυτή με το μικρότερο NCP, έστω η df, γίνεται εν τέλει και η γενίκευση των στοιχείων αυτών (d → df , f → df), και καταχωρείται στις ήδη αποφασισμένες γενικεύσεις.

Συνεχίζουμε πλέον να εξετάζουμε το υπόλοιπο δέντρο μέτρησης, δρώντας με το ίδιο ακριβώς τρόπο. Εάν στη συνέχεια της αναζήτησης συναντήσουμε μονοπάτι που περιέχει τα d ή f, τότε απλά τα αγνοούμε, ενώ εάν το δέντρο επεκτεινόταν και σε επόμενο επίπεδο (4), τότε δεν θα επεκτείνονται μονοπάτια που περιέχουν τα d ή f.

4.4 Υπολογισμός Κόστους

Όπως είναι φυσικό, και έχει προαναφερθεί, με κάθε γενίκευση κάποιου στοιχείου του αρχικού πίνακα, με στόχο την επίτευξη της k^m – ανωνυμίας, προκύπτει κάποια απώλεια πληροφορίας. Η απώλεια αυτή είναι φυσική, καθώς κάποιες τιμές του πίνακα οι οποίες αρχικά ήταν γνωστές, μετά την εκτέλεση του αλγορίθμου της k^m – ανωνυμοποίησης γενικεύονται και συνεπώς χάνεται μέρος της συγκεκριμένης πληροφορίας. Στόχος πρέπει να είναι η κατά το δυνατόν μικρότερη απώλεια πληροφορίας, ώστε να μπορεί να αξιοποιηθεί στο μέγιστο η πληροφορία που παρέχει ο δημοσιευμένος πίνακας.

Την απώλεια αυτή της πληροφορίας θα θεωρήσουμε ως το κόστος της k^m – ανωνυμίας και όπως είναι φυσικό, χρειαζόμαστε κάποιον τρόπο για να μπορέσουμε να υπολογίζουμε την απώλεια της πληροφορίας που προκύπτει από κάθε εκτέλεση του αλγορίθμου μας, ώστε να μπορούμε κατ' επέκταση και να υπολογίσουμε ποιο το κόστος αυτών των γενικεύσεων.

Για τον υπολογισμό του κόστους της απώλειας πληροφορίας θα χρησιμοποιήσουμε ως μέτρο την τιμή Κανονικοποιημένης Ποινής Βεβαιότητας (Normalized Certainty Penalty - NCP). Η Κανονικοποιημένη Ποινή Βεβαιότητας υπολογίζεται λαμβάνοντας υπ' όψιν την ιεραρχία που χρησιμοποιείται κατά την ανωνυμοποίηση.

Έτσι, αν:

- p είναι ένα στοιχείο του πεδίου ορισμού I,
- $|u_p|$ είναι ο αριθμός των φύλλων του δέντρου που βρίσκονται κάτω από τη γενίκευση u_p
- |I| είναι ο αριθμός όλων των στοιχείων του πεδίου ορισμού

Τότε η Κανονικοποιημένη Ποινή Βεβαιότητας (NCP) ενός στοιχείου p του προς ανωνυμοποίηση πίνακα μπορεί να υπολογιστεί ως εξής:

$$NCP(p) = \begin{cases} 0, & για\ |\text{up}| = 1 \\ \frac{|\text{up}|}{|\text{I}|}, & \text{για κάθε άλλο} \end{cases}$$

Διαπιστώνουμε εύκολα πως η Κανονικοποιημένη Ποινή Βεβαιότητας ενός στοιχείου, προσπαθεί να υπολογίσει το κόστος της απώλειας πληροφορίας λαμβάνοντας υπ' όψιν το ποσοστό του συνόλου των στοιχείων που γενικεύτηκαν ως προς το σύνολο των στοιχείων που περιέχει ο πίνακας (πεδίο ορισμού I).

Μέσω του $|u_p|$ προσμετρείται το πλήθος όλων των φύλλων που βρίσκονται στο δέντρο γενίκευσης κάτω από μια ορισμένη γενίκευση. Με την μέτρηση αυτή επιβεβαιώνουμε πόσο επηρεάζεται το κόστος της απώλειας πληροφορίας στις περιπτώσεις μιας προκατασκευασμένης ιεραρχίας γενίκευσης, όπου μια αναγκαία γενίκευση ενός στοιχείου του πίνακα εξαναγκάζει σε γενίκευση και τα αδελφά-στοιχεία στην ιεραρχία, προσμετρώντας και αυτά στο κόστος της απώλειας, ενισχύοντας καθ' αυτόν τον τρόπο την ανάγκη διερεύνησης μεθόδων ανωνυμοποίησης χωρίς τη χρήση τέτοιων προκαθορισμένων ιεραρχιών, όπως ακριβώς αποσκοπεί και η παρούσα εργασία με τη δημιουργία δυναμικών ιεραρχιών γενίκευσης.

Για τον υπολογισμό της Κανονικοποιημένης Ποινής Βεβαιότητας ολόκληρου του πίνακα, θα χρειαστεί να λάβουμε υπ' όψιν και το ειδικό βάρος που έχει η κάθε γενίκευση. Αυτό θα επιτευχθεί μέσω του υπολογισμού του πλήθους εμφάνισης στον αρχικό πίνακα του κάθε στοιχείου που γενικεύεται από τον αλγόριθμο.

Έτσι, αν:

- C_p είναι το πλήθος εμφάνισης του στοιχείου p στον αρχικό προς ανωνυμοποίηση πίνακα

Τότε η Κανονικοποιημένη Ποινή Βεβαιότητας (NCP) ενός προς ανωνυμοποίηση πίνακα D, μπορεί να υπολογιστεί ως εξής:

$$NCP(D) = \frac{\sum_{p \in I} Cp \cdot NCP(p)}{\sum_{p \in I} Cp}$$

Με τον τρόπο αυτό κάθε στοιχείο του πίνακα δεν θεωρείται πως έχει την ίδια βαρύτητα σε ότι αφορά την απώλεια πληροφορίας, καθώς η γενίκευση ενός στοιχείου το οποίο έχει υψηλότερο πλήθος εμφάνισης στον αρχικό πίνακα, σαφώς και συμβάλλει σε μεγαλύτερη απώλεια πληροφορίας συγκριτικά με ένα στοιχείο με μικρότερο πλήθος εμφάνισης.

5

Υλοποίηση αλγορίθμου

Σε αυτό το κεφάλαιο περιγράφονται οι τεχνικές λεπτομέρειες γύρω από την υλοποίηση των περιγραφέντων αλγορίθμων.

5.1 Λεπτομέρειες υλοποίησης

Η υλοποίηση του αλγορίθμου έγινε με χρήση της γλώσσας C++, ενώ η σύνταξή του για λόγους χρήσης των σχετικών διευκολύνσεων κατά τη συγγραφή κώδικα, έγινε μέσω του περιβάλλοντος Code::Blocks.

5.1.1 Χαρακτηριστικά της υλοποίησης

5.1.1.1 Είσοδος-Έξοδος

Η είσοδος των δεδομένων γίνεται με τη μορφή αρχείου απλού κειμένου (.txt). Κάθε γραμμή του αρχείου είναι και από μια ξεχωριστή εγγραφή, ενώ τα στοιχεία κάθε εγγραφής διαχωρίζονται με ένα κενό χαρακτήρα.

Κατά την εκτέλεση του αλγορίθμου δημιουργείται ένα νέο αρχείο κειμένου που περιέχει την τη τελική ιεραρχία γενίκευσης που έχει δημιουργηθεί και η οποία εξασφαλίζει την k^m-ανωνυμία, καθώς και το κόστος της γενίκευσης εκφρασμένο μέσω της τιμής της Κανονικοποιημένης Ποινής Βεβαιότητας (NCP) και το χρόνο εκτέλεσης του αλγορίθμου.

5.1.1.2 Μεταγλώττιση και εκτέλεση της εφαρμογής

Η εφαρμογή μεταγλωττίζεται και εκτελείται μέσα από το περιβάλλον του Code::Blocks η οποία χρησιμοποιεί τον μεταγλωττιστή g++ της Mingw32.

Για την εκτέλεση του αλγορίθμου ζητούνται ως είσοδοι:

- *Όνομα αρχείου δεδομένων:* Το όνομα του αρχείου κειμένου στο οποίο βρίσκεται αποθηκευμένη η βάση για την οποία θέλουμε να επιτευχθεί η k^m-ανωνυμοποίηση.
- *Όνομα αρχείου εξόδου:* Το όνομα του αρχείου στο οποίο θα αποθηκεύεται η εκτεταμένη βάση.
- *Τιμή παραμέτρου m:* Η τιμή της παραμέτρου m που υποδηλώνει την μέγιστη γνώση του επιτιθέμενου. Η παράμετρος m είναι ακέραιος αριθμός, μεγαλύτερος του μηδενός και προφανώς μικρότερος του πλήθους των διαφορετικών στοιχείων που εμφανίζονται στη βάση μας.
- *Τιμή παραμέτρου ανωνυμίας k:* Η τιμή της παραμέτρου k που υποδηλώνει το πλήθος των εγγραφών ανάμεσα στις οποίες ο επιτιθέμενος δεν θα πρέπει να μπορεί να αναγνωρίσει την εγγραφή, μέρος της οποίας κατέχει γνώση. Και η παράμετρος k είναι επίσης ακέραιος αριθμός, μεγαλύτερος του μηδενός, και προφανώς μικρότερος του πλήθους των εγγραφών της βάσης.

5.1.1.3 Δομές δεδομένων

Κατά την υλοποίηση του αλγορίθμου χρησιμοποιήθηκαν για την αποθήκευση και επεξεργασία των δεδομένων κατά κύριο λόγο οι δομές τύπου set, και δευτερευόντως όπου κρίθηκε απαραίτητο χρησιμοποιήθηκαν και δομές τύπου λίστας. Τα sets προτιμήθηκαν καθώς αποτελούν δομές αποθήκευσης μοναδικών δεδομένων σε ορισμένη διάταξη, κάτι το οποίο ήταν επιθυμητό και απαραίτητο για τη δημιουργία, επεξεργασία και χρήση του δέντρου μέτρησης και κατά συνέπεια και όλων των υπολοίπων υπό-αλγορίθμων που αναπτύχθηκαν και στηρίζονταν στη χρήση του δέντρου μέτρησης. Οι λίστες χρησιμοποιήθηκαν κατά κύριο λόγο σε συναρτήσεις που στηρίζονταν σε εφαρμογή πρακτικών FIFO, LIFO, όπως για παράδειγμα στη δημιουργία και χρήση των διαφορετικών υποσυνόλων μεγέθους έως m των εγγραφών της βάσης.

Και οι δύο τύποι δομών δεδομένων προτιμήθηκαν σε σχέση με τη χρήση πινάκων για παράδειγμα, καθώς η χρήση τους δεν προϋποθέτει τον εκ των προτέρων ορισμό προκαθορισμένου μεγέθους, αλλά το μέγεθός τους μπορεί να διαμορφώνεται δυναμικά ανάλογα με τις διαφορετικές εισόδους κάθε φορά, εξασφαλίζοντας σίγουρα μεγαλύτερη εξοικονόμηση μνήμης για τον αλγόριθμό μας.

5.1.1.4 Συναρτήσεις & Μέθοδοι

Κατά την ανάπτυξη και υλοποίηση του αλγορίθμου, χρησιμοποιήθηκαν ορισμένες συναρτήσεις και μέθοδοι οι οποίες παρουσιάζονται επιγραμματικά παρακάτω:

- *read_database:* Κάνει μια ανάγνωση της βάσης, ώστε να γνωρίζουμε τι στοιχεία αυτή περιέχει αποθηκεύοντάς τα σε ένα set, ώστε να μπορούν να δημιουργηθούν όλες οι πιθανές γενικεύσεις.
- *AA_algo:* Εκτελεί τον apriori αλγόριθμο ανωνυμοποίησης, κατασκευάζοντας και ελέγχοντας το δέντρο μέτρησης με βάθος αυξημένο κατά ένα κάθε φορά.
- *build_ctree:* Δημιουργεί το δέντρο μέτρησης
- *DA_algo:* Εκτελεί παραλλαγή του αλγορίθμου της ευθείας ανωνυμοποίησης αναζητώντας μονοπάτια ενός υποδέντρου του δέντρου μέτρησης με πλήθος μικρότερο του k.
- *checkLeafs_algo:* Ελέγχει τους αδελφούς κόμβους ενός φύλλου του δέντρου μέτρησης, για να εντοπίσει ποιοι εξ' αυτών αποτελούν προβληματικούς κόμβους καθώς και τις πιθανές λύσεις αυτών.
- *find_possible_solutions:* Βρίσκει όλες τις πιθανές λύσεις για τους προβληματικούς κόμβους του δέντρου μέτρησης, βάσει του αθροίσματος του πλήθους εμφάνισής των αδελφών κόμβων αυτού.
- *check_solut:* Ελέγχει το δέντρο μέτρησης ώστε να διαπιστώσει εάν οι πιθανές λύσεις που έχουν ήδη βρεθεί, είναι πράγματι λύσεις εξασφαλίζοντας την k^m – ανωνυμία για το μονοπάτι το οποίο εξετάζεται.
- *resolve_problematics:* Επιλύει τις παραβιάσεις της k-ανωνυμίας σαρώνοντας τους προβληματικούς κόμβους που έχουν εντοπιστεί και βρίσκοντας τη βέλτιστη από τις λύσεις που έχουν προσδιοριστεί για κάθε έναν από αυτούς.

Επιπλέον των παραπάνω κύριων συναρτήσεων και μεθόδων του αλγορίθμου, υλοποιήθηκαν και οι κατωτέρω οι οποίες είτε εκτελούν μικρότερα τμήματα των παραπάνω είτε λειτουργούν υποστηρικτικά σε αυτές.

- *generalized_path:* Ελέγχει κατά τη διάρκεια εκτέλεσης του DA_algo εάν κάποιο στοιχείο του τρέχοντος μονοπατιού στο δέντρο μέτρησης έχει ήδη γενικευτεί, ώστε να επιστρέψει συνεχίζοντας από εκεί, καθώς δεν υπάρχει λόγος ελέγχου ενός μονοπατιού που περιέχει ήδη γενικευμένα στοιχεία.

- *update_fcut:* Ανανεώνει τους τελικούς κανόνες γενίκευσης εάν έχουν βρεθεί νέοι που πρέπει να καταχωρηθούν.
- *create_counttree:* Κατασκευάζει το επίπεδο του δέντρου μέτρησης
- *create_sub_ctree:* Επεκτείνει σε βάθος το δέντρο μέτρησης ανάλογα με το επίπεδο που έχει φτάσει ο apriori αλγόριθμος ανωνυμοποίησης.
- *add_siblings_ctree:* Προσθέτει ως «αδέλφια» όλα τα στοιχεία που βρίσκονται δεξιά του «γονιού» ενός κόμβου στο δέντρο μέτρησης, υπό την προϋπόθεση ότι σε κάθε μονοπάτι που δημιουργείται δεν υπάρχει στοιχείο που να αποτελεί πιθανή γενίκευση κάποιου άλλου στοιχείου του μονοπατιού.
- *check_generalization:* Ελέγχει δύο στοιχεία για το εάν κάποιο από τα δύο αποτελεί πιθανή γενίκευση του άλλου.
- *expand_database:* Κατασκευάζει την εκτεταμένη βάση, διαβάζοντας και επεκτείνοντας τις εγγραφές της βάσης ώστε να περιέχουν και τις πιθανές γενικεύσεις των στοιχείων τους σε αυτές.
- *subset:* Βρίσκει όλα τα υποσύνολα μεγέθους έως m των στοιχείων κάθε εκτεταμένης εγγραφής της βάσης.
- *checkbuild_ctree:* Ελέγχει εάν ένα υποσύνολο των στοιχείων της εκτεταμένης βάσης υπάρχει ως μονοπάτι στο δέντρο μέτρησης, και εάν ναι, τότε αυξάνει το πλήθος του κατά ένα.
- *search_ctree:* Αναζητά για κάποιο στοιχείο στο δέντρο μέτρησης.
- *generalized_fcut:* Ελέγχει εάν κάποιο στοιχείο έχει γενικευτεί βάσει των κανόνων γενίκευσης που έχουν οριστεί μέχρι τώρα.
- *find_item_set:* Επιστρέφει ένα set με όλα τα στοιχεία των οποίων γενίκευση αποτελεί μια δοθείσα γενίκευση ή με άλλα λόγια επιστρέφει τα στοιχεία τα οποία γενικεύονται σε αυτή.
- *free_ctree:* Διαγράφει το δέντρο μέτρησης απελευθερώνοντας έτσι τη δεσμευμένη μνήμη.
- *add_sol_to_final:* Προσθέτει μια λύση, αφότου αυτή βρεθεί, στη λίστα με τις συνολικές λύσεις, και ταυτοχρόνως αφαιρεί τα στοιχεία τα οποία γενικεύει από τη λίστα των προβληματικών στοιχείων.
- *calc_NCP:* Υπολογίζει την τιμή της Κανονικοποιημένης Ποινής Βεβαιότητας (NCP) μια δοθείσης γενίκευσης
- *CP:* Επιστρέφει το πλήθος εμφάνισης ενός στοιχείου μέσα στη βάση, αναζητώντας το στοιχείο αυτό στο δέντρο μέτρησης.

- *create_GenMap:* Χτίζει μια δομή δεδομένων τύπου map, η οποία περιέχει όλες τις πιθανές γενικεύσεις των στοιχείων της βάσης, βάσει των πιθανών λύσεων που έχουν βρεθεί.
- *Main συνάρτηση:* Αποτελεί τη βασική συνάρτηση του κώδικα, η οποία και οργανώνει την εκτέλεση του αλγορίθμου.
- *Print* συναρτήσεις: Πέραν όλων των προηγούμενων συναρτήσεων, δημιουργήθηκαν επιπλέον και κάποιες συναρτήσεις εκτύπωσης στο τερματικό ή εξαγωγής σε αρχεία κειμένου των λιστών, των sets, των ιεραρχιών, του δέντρου μέτρησης, των κανόνων γενίκευσης κλπ, για την καλύτερη παρακολούθηση και έλεγχο της εκτέλεσης και ορθότητας του αλγορίθμου.

5.1.2 Ανάλυση βασικών μεθόδων του κώδικα

5.1.2.1 Main συνάρτηση

Κατά την εκκίνηση του αλγορίθμου, και αφού έχουν δοθεί τα απαραίτητα στοιχεία εισόδου όπως αυτά έχουν αναλυθεί και προηγουμένως (όνομα αρχείου εισόδου, αρχείου εξόδου, παράμετροι k, m) η main καλεί την συνάρτηση read_database η οποία κάνει μια ανάγνωση των δεδομένων από το αρχείο εισόδου ώστε να γνωρίζουμε τα διαφορετικά στοιχεία του συνόλου που θα μας χρειαστούν, και τα τοποθετεί σε μια ορισμένη διάταξη για την κατασκευή του δέντρου μέτρησης.

Είμαστε τώρα έτοιμοι για την εκτέλεση του κύριου σταδίου του αλγορίθμου, καθώς η main καλεί την AA_algo που αναλύεται παρακάτω.

Ουσιαστικά με την ολοκλήρωση της AA_algo ο αλγόριθμος έχει ολοκληρωθεί και έχουν βρεθεί οι απαραίτητοι κανόνες γενίκευσης για την εξασφάλιση της k^m-ανωνυμοποίησης της βάσης. Γίνονται οι απαραίτητες εκτυπώσεις των αποτελεσμάτων και η εφαρμογή τερματίζει. Να σημειώσουμε ότι η μέτρηση του χρόνου εκτέλεσης της εφαρμογής γίνεται στη main και αφορά ουσιαστικά τον χρόνο από την έναρξη της εφαρμογής που εισέρχεται στην συνάρτηση main, μέχρι και την εκτύπωση και των τελευταίων αποτελεσμάτων όπου ετοιμάζεται για τον τερματισμό αυτής.

5.1.2.2 AA_algo

Η συνάρτηση AA_algo αποτελεί ουσιαστικά την κεντρική συνάρτηση του κώδικα, υλοποιώντας τον αλγόριθμο της apriori ανωνυμοποίησης και ακολουθώντας τα παρακάτω βήματα:

- Καλεί την συνάρτηση build_ctree η οποία κατασκευάζει το δέντρο μέτρησης με τον τρόπο που θα περιγραφεί παρακάτω, και για βάθος ίσο με 1 (δηλαδή το δέντρο μέτρησης έχει αρχικά μόνο ένα επίπεδο).
- Έχοντας πλέον έτοιμο και το δέντρο μέτρησης, η AA_algo καλεί την DA_algo για να εκτελέσει τον αλγόριθμο της ευθείας ανωνυμοποίησης για το δέντρο που κατασκευάστηκε, εντοπίζοντας τους προβληματικούς κόμβους του δέντρου για το επίπεδο αυτό, καθώς και τις πιθανές λύσεις αυτών.
- Καλεί ξανά την συνάρτηση build_ctree η οποία θα καλέσει την expand_database και η οποία θα επεκτείνει την βάση προσθέτοντας τις πιθανές γενικεύσεις των στοιχείων κάθε εγγραφής, και από αυτή θα χτίσει ξανά το δέντρο μέτρησης περιλαμβάνοντας αυτή τη φορά σε αυτό και τις πιθανές λύσεις.
- Καλεί την resolve_problematics η οποία θα σαρώσει τους προβληματικούς κόμβους που έχουν εντοπιστεί, θα ελέγξει ποιες από τις πιθανές λύσεις που βρέθηκαν αποτελούν πράγματι λύσεις, και θα επιλέξει τη βέλτιστη από τις λύσεις που έχουν προσδιοριστεί για κάθε έναν από αυτούς.
- Η παραπάνω διαδικασία επαναλαμβάνεται αυξάνοντας κάθε φορά το βάθος του δέντρου μέτρησης κατά ένα, και έως ότου εκτελεστούν οι παραπάνω μέθοδοι για δέντρο μέτρησης βάθους m.

5.1.2.3 build_ctree

Η build_counttree καλεί την create_counttree και είναι η μέθοδος που κατασκευάζει το δέντρο μέτρησης με βάση την ορισμένη διάταξη των στοιχείων της βάσης.

Το πρώτο επίπεδο ουσιαστικά του δέντρου θα είναι η ίδια η διάταξη των στοιχείων της βάσης. Για τη κατασκευή του δευτέρου επιπέδου η συνάρτηση καλεί τις συναρτήσεις create_sub_ctree και add_siblings_ctree οι οποίες φτιάχνουν το πρώτο «παιδί» κάθε κόμβου, και προσθέτουν για «αδέλφια» του τα «αδέλφια» που βρίσκονται στα δεξιά του «γονέα», προσέχοντας όμως σε κάθε μονοπάτι που δημιουργείται να μην υπάρχει στοιχείο που να αποτελεί πιθανή γενίκευση κάποιου άλλου στοιχείου του μονοπατιού. Με τον τρόπο αυτό χτισίματος του δέντρου, επιτυγχάνουμε κάθε μονοπάτι του δέντρου να είναι μοναδικό όχι μόνο ως προς τα στοιχεία του αλλά και ως προς τις γενικεύσεις αυτών. Υπενθυμίζεται ότι το δέντρο μέτρησης στην υλοποίησή μας είναι ένα n-αδικό δέντρο όπου κάθε κόμβος «γονέας» έχει ένα «παιδί» με πολλά «αδέλφια».

Αφότου κατασκευαστεί ο σκελετός του δέντρου μέτρησης από την create_counttree όπως περιγράφεται παραπάνω, η build_counttree καλεί την expand_database για να χτίσει το δέντρο προσθέτοντας τα πλήθη εμφάνισης κάθε κόμβου.

5.1.2.4 expand_database

Η μέθοδος expand_database είναι αυτή που χτίζει ουσιαστικά το δέντρο μέτρησης υπολογίζοντας και προσθέτοντας τα πλήθη εμφάνισης του κάθε κόμβου.

- Αρχικά διαβάζει μια μια τις εγγραφές της βάσης. Κατά τη δεύτερη εκτέλεση του αλγορίθμου σε κάθε επίπεδο του AA_algo, και αφότου έχουν βρεθεί οι προβληματικοί κόμβοι και οι πιθανές λύσεις αυτών, για κάθε στοιχείο μίας εγγραφής το οποίο δεν έχει ήδη γενικευτεί βάσει των κανόνων γενίκευσης fcut, προσθέτει τις πιθανές γενικεύσεις του, επεκτείνοντας καθ' αυτόν τον τρόπο τις εγγραφές.
    - Για κάθε μια εγγραφή καλείται η συνάρτηση subset η οποία με αναδρομικό τρόπο βρίσκει όλα τα δυνατά υποσύνολα της εγγραφής, μεγέθους μικρότερου ή ίσου του i, ανάλογα με τον αριθμό της επανάληψης στην οποία βρίσκεται η συνάρτηση AA_algo. Για κάθε ένα υποσύνολο που δημιουργείται:
        - η subset καλεί την checkbuild_ctree η οποία σαρώνει το ήδη κατασκευασμένο δέντρο μέτρησης, αναζητώντας για μονοπάτι το οποίο να αποτελείται από τα στοιχεία του υποσυνόλου που ελέγχεται.
        - Εάν αυτό βρεθεί, τότε αυξάνεται κατά ένα ο μετρητής που βρίσκεται στο τελευταίο κόμβο του μονοπατιού που βρήκαμε, υποδηλώνοντας ότι βρέθηκε ακόμη μια επιπλέον φορά ο συγκεκριμένος συνδυασμός στη βάση.

Από την παραπάνω διαδικασία αξίζει να επισημανθούν δύο σημεία:

Πρώτον τονίζεται η σπουδαιότητα ύπαρξης ορισμένης διάταξης για την κατασκευή του δέντρου μέτρησης και των υποσυνόλων των εγγραφών, καθώς διευκολύνεται η αναζήτηση στο δέντρο μέτρησης η οποία γίνεται ανά επίπεδο. Κάθε στοιχείο του υποσυνόλου που διερευνάται αναζητείται και σε ξεχωριστό επίπεδο, τα στοιχεία του οποίου βρίσκονται σε ορισμένη διάταξη, κάνοντας έτσι πιο γρήγορη την αναζήτηση.

Δεύτερον επισημαίνεται ότι ο έλεγχος κάθε υποσυνόλου που κατασκευάζει η subset, και η αναζήτησή του στο δέντρο μέτρησης, γίνεται κατά την κατασκευή του υποσυνόλου, κερδίζοντας έτσι σε χρόνο και μνήμη καθώς τα υποσύνολα που κατασκευάζονται δεν αποθηκεύονται κάπου για να διερευνηθούν μετά.

5.1.2.5 DA_algo

Η συνάρτηση DA_algo υλοποιεί παραλλαγή του αλγορίθμου ευθείας ανωνυμοποίησης. για το δέντρο που έχει βρεθεί στην AA_algo.

- Η συνάρτηση σαρώνει με αναδρομή το δέντρο μέτρησης, κρατώντας σε μια λίστα το μονοπάτι πάνω στο οποίο κινείται και εφόσον αυτό δεν έχει στοιχεία που έχουν ήδη γενικευτεί από τους κανόνες γενίκευσης fcut.
- Εάν φτάσει σε φύλλο, του οποίου το πλήθος εμφάνισης είναι μικρότερο του k, σημαίνει πως για το συγκεκριμένο μονοπάτι δεν ικανοποιείται η k^m-ανωνυμία και έτσι:
    - Καλεί την checkLeafs_algo συνάρτηση η οποία θα αναζητήσει τους υπόλοιπους αδελφούς κόμβους για να βρει κι άλλους προβληματικούς κόμβους που παραβιάζουν την k^m-ανωνυμία, καθώς και τις πιθανές λύσεις αυτών, όπως αναλύεται παρακάτω.
    - Ανανεώνει τους πιθανούς κανόνες γενίκευσης που προέκυψαν κατά την εκτέλεση της checkLeafs_algo μέσω της update_fcut

Αξίζει να σημειωθεί πως εάν κατά τη σάρωση του δέντρου μέτρησης ο αλγόριθμος εντοπίσει κάποιο άλλο γενικευμένο στοιχείο, θα το παρακάμψει και δεν θα συνεχίσει προς τα κάτω το μονοπάτι εκείνο, καθώς όπως έχουμε ήδη αναλύσει, δεν υπάρχει λόγος να εξεταστεί μονοπάτι που περιέχει στοιχείο το οποίο έχει ήδη γενικευτεί από τους κανόνες fcut. Γλυτώνουμε λοιπόν με αυτόν τον τρόπο περιττή σπατάλη πόρων και εξοικονομούμε τον χρόνο εκτέλεσης που θα αντιστοιχούσε σε αυτό το μονοπάτι.

Επίσης βλέπουμε ότι η checkLeafs_algo δεν εκτελείται για όλα τα πιθανά μονοπάτια σπαταλώντας έτσι και πάλι άσκοπα πόρους, αλλά μόνο για τα μονοπάτια στα οποία εντοπίζονται πιθανές παραβιάσεις της ιδιωτικότητας. Αυτό είναι άλλωστε και το πλεονέκτημα που μας εξασφαλίζει η εκτέλεση του αλγορίθμου ευθείας ανωνυμοποίησης μέσω της DA_algo μεθόδου.

5.1.2.6 checkLeafs_algo

Η μέθοδος checkLeafs_algo ελέγχει τους αδελφούς κόμβους ενός κόμβου του δέντρου μέτρησης, τον οποίο η DA_algo έχει εντοπίσει ως προβληματικό, προκειμένου να εντοπίσει και άλλους προβληματικούς κόμβους, καθώς και τις πιθανές λύσεις αυτών.

- Η συνάρτηση σαρώνει τους αδελφούς κόμβους ενός κόμβου, και ελέγχει το πλήθος εμφάνισής τους. Εάν αυτό είναι μικρότερο του k τότε τους προσθέτει και αυτούς στη λίστα των προβληματικών κόμβων.

- Αφού βρει όλους τους προβληματικούς κόμβους, καλεί την find_possible_solutions, η οποία θα ερευνήσει για πιθανές λύσεις και θα τις προσθέσει στη λίστα των πιθανών λύσεων.

5.1.2.7 find_possible_solutions

Η find_possible_solutions αναζητά πιθανές λύσεις για τη λίστα των προβληματικών κόμβων που εντόπισε η checkLeafs_algo, μέσα από τους αδελφούς κόμβους αυτών.

Η λογική της πιθανής λύσης στηρίζεται στο γεγονός ότι μια πιθανή λύση θα είναι μια γενίκευση η οποία και θα πρέπει να έχει πλήθος εμφάνισης μεγαλύτερο του k. Μία πιθανή λύση λοιπόν, δεν μπορεί να πρόκειται για μια γενίκευση η οποία αποτελείται από κόμβους το άθροισμα του πλήθους εμφάνισης των οποίων είναι μικρό από k, καθώς σε αυτή τη περίπτωση προφανώς ούτε και η γενίκευσή τους θα εμφανίζεται περισσότερες από k φορές.

Στηριζόμενη η μέθοδος στην παραπάνω λογική:

- Σαρώνει τους αδελφούς κόμβους των προβληματικών κόμβων, ελέγχοντας το πλήθος εμφάνισής τους.
- Εάν το άθροισμα του πλήθους εμφάνισης του προβληματικού κόμβου, με το συνδυασμό άλλων κόμβων είναι μεγαλύτερο του k, τότε ο συνδυασμό αυτός αποτελεί μία γενίκευση η οποία μπορεί να αποτελέσει μια πιθανή λύση και συνεπώς προστίθεται στη λίστα των πιθανών λύσεων.
- Ο έλεγχος των συνδυασμών γίνεται τμηματικά. Ελέγχονται δηλαδή πρώτα οι συνδυασμοί 2 κόμβων, έπειτα 3 κόμβων, 4 κ.ο.κ. Αυτό γίνεται διότι εάν μια γενίκευση η οποία είναι συνδυασμός 2 κόμβων μπορεί να αποτελέσει λύση στον προβληματικό κόμβο, τότε η λύση αυτή θα είναι προτιμότερη από μια γενίκευση 3 κόμβων καθώς η γενίκευση θα έχει μικρότερο κόστος.

5.1.2.8 resolve_problematics

Η μέθοδος resolve_problematics είναι αυτή η οποία θα δώσει τελικά τις οριστικές λύσεις στις πιθανές παραβιάσεις της k^m – ανωνυμίας για αυτό το επίπεδο του AA_algo.

- Αρχικά θα καλέσει την check_solut η οποία θα σαρώσει το νέο δέντρο μέτρησης το οποίο χτίστηκε από την εκτεταμένη βάση και το οποίο περιλαμβάνει και τις πιθανές λύσεις των προβληματικών κόμβων, ώστε να εξακριβώσει ποιές από τις πιθανές λύσεις που εντοπίστηκαν από την find_possible_solutions κατά την εκτέλεση του DA_algo αποτελούν πράγματι λύσεις. Εν ολίγοις θα ελέγξει ποιες από τις πιθανές

γενικεύσεις έχουν πράγματι πλήθος εμφάνισης μεγαλύτερο του k, και συνεπώς μπορούν να αξιοποιηθούν για την επίλυση των προβληματικών κόμβων. Οι λύσεις θα ταξινομηθούν σε αύξουσα σειρά με βάση το κόστος των γενικεύσεών τους.

- Έπειτα θα αντιπαραβάλει τη λίστα των λύσεων, με τη λίστα των προβληματικών κόμβων, και θα επιλέξει τη βέλτιστη λύση για κάθε προβληματικό κόμβο.
- Τέλος, εάν δεν καταστεί δυνατή η εύρεση μιας λύσης για κάποιον προβληματικό κόμβο, τότε θα επιχειρηθεί να δημιουργήσει γενίκευση με τον κόμβο γονιό, ο οποίος λόγο της apriori λογικής του αλγορίθμου AA_algo, θα έχει ήδη πλήθος εμφάνισης μεγαλύτερο ή ίσο του k, και συνεπώς θα αποτελεί σίγουρα λύση του προβληματικού κόμβου.

5.1.3 Κλασσικός Αλγόριθμος k^m – ανωνυμοποίησης

Τα παραπάνω αποτελούν την ανάλυση των βασικών σημείων υλοποίησης του αλγορίθμου της k^m-ανωνυμοποίησης με υπολογισμό και χρήση δυναμικών ιεραρχιών γενίκευσης. Για λόγους σύγκρισης του παραπάνω αλγορίθμου, υλοποιήθηκε και ο κλασσικός αλγόριθμος της k^m – ανωνυμοποίησης. Κατά την υλοποίηση του αλγορίθμου αυτού, χρησιμοποιήθηκαν αρκετές από τις παραπάνω ήδη ανεπτυγμένες συναρτήσεις, πολλές εκ των οποίων ελαφρώς τροποποιημένες και προσαρμοσμένες στις ανάγκες του κλασσικού αλγορίθμου k^m – ανωνυμοποίησης. Επίσης υλοποιήθηκαν και κάποιες επιπλέον συναρτήσεις, οι οποίες παρουσιάζονται στη συνέχεια:

- *OA_algo:* Εκτελεί τον αλγόριθμο βέλτιστης ανωνυμοποίησης ελέγχοντας για κάθε μονοπάτι που δεν εξασφαλίζει λύση, τις πιθανές γενικεύσεις των στοιχείων του.
- *create_cans:* Κατασκευάζει τα sets όλων των πιθανών γενικεύσεων του μονοπατιού που εξετάζεται στην OA_algo() .
- *use_cans:* Ελέγχει εάν μια πιθανή γενίκευση του μονοπατιού που εξετάζεται στην OA_algo() , προσφέρει k^m-ανωνυμία ή όχι, αναζητώντας τη γενίκευση του μονοπατιού στο δέντρο μέτρησης.
- *check_cans:* Ελέγχει αν υπάρχει κάποιο στοιχείο μιας γενίκευσης ενός μονοπατιού που να αποτελεί γενίκευση κάποιου άλλου στοιχείου, και επιστρέφει τη γενίκευση του μονοπατιού κρατώντας το πιο γενικευμένο στοιχείο από τα δύο.
- *create_crule:* Δημιουργεί έναν πιθανό κανόνα γενίκευσης, βάσει μιας πιθανής γενίκευσης ενός μονοπατιού του δέντρου μέτρησης, και υπολογίζει του κανόνα

αυτού εκφρασμένο μέσω της τιμής της Κανονικοποιημένης Ποινής Βεβαιότητας (NCP).

- *delete_Q: Αναζητά και διαγράφει από τη λίστα Q μια γενίκευση μονοπατιού.*
- *check_H:* Ελέγχει εάν μια γενίκευση ενός μονοπατιού έχει ήδη εξεταστεί (οπότε βρίσκεται στο set H) ειδάλλως την προωθεί στη λίστα Q όπου βρίσκονται οι προς εξέταση γενικεύσεις μονοπατιών.
- *create_hier_set:* Δημιουργεί την ορισμένη διάταξη που περιέχει όλα τα στοιχεία της βάσης και θα χρησιμοποιηθεί για την κατασκευή του δέντρου μέτρησης.

5.1.3.1 OA_algo

Η συνάρτηση OA_algo είναι η συνάρτηση που υλοποιεί τον αλγόριθμο της βέλτιστης ανωνυμοποίησης για το μονοπάτι που βρέθηκε από την DA_algo να παραβιάζει την k^m-ανωνυμία. Κατά την εκτέλεση της OA_algo:

- Το μονοπάτι σπρώχνεται σε μια λίστα Q η οποία περιέχει τα προς εξέταση μονοπάτια.
- Όσο η Q δεν είναι άδεια
    - Εξετάζεται το μπροστινό στοιχείο της Q το οποίο προστίθεται στο set H με τα ήδη εξετασμένα μονοπάτια.
    - Καλεί την create_cans η οποία κατασκευάζει όλα τα δυνατά μονοπάτια που προκύπτουν από τις πιθανές γενικεύσεις των στοιχείων του, και αναζητά μέσω της use_cans ποιο από αυτά δίνει λύση, με τον τρόπο που περιγράφεται παρακάτω.

5.1.3.2 use_cans

Η use_cans συνάρτηση καλείται από την OA_algo για να διαχειριστεί τα διάφορα μονοπάτια που δημιουργούνται από τις πιθανές γενικεύσεις του αρχικού μονοπατιού που εξετάζεται. Τα βήματα που ακολουθεί η use_cans για τη διαχείριση των μονοπατιών είναι τα εξής:

- Αρχικά ελέγχει μέσω της check_cans τα στοιχεία του νέου πιθανού μονοπατιού που προέκυψε από τις πιθανές γενικεύσεις. Πρέπει να εξασφαλίζεται ότι κανένα στοιχείο του μονοπατιού δεν αποτελεί πιθανή γενίκευση κάποιου άλλου στοιχείου, καθώς σε αυτή τη περίπτωση δεν υπάρχει καν μονοπάτι στο δέντρο μέτρησης. Εάν βρεθούν δύο τέτοια στοιχεία, τότε κρατάμε το γενικότερο εκ των δύο και συνεχίζουμε.

- Στη συνέχεια αναζητείται το νέο μονοπάτι στο δέντρο μέτρησης και εφόσον βρεθεί τότε:
    - Αν ο μετρητής του τελευταίου κόμβου του μονοπατιού είναι μικρότερος του k, και άρα δεν εξασφαλίζεται η k^m-ανωνυμία τότε:
        - Ελέγχεται αν το μονοπάτι έχει εξεταστεί ξανά (δηλαδή αν βρίσκεται στο H set) και εφόσον όχι τότε σπρώχνεται στη λίστα Q για να εξεταστούν οι πιθανές γενικεύσεις του
    - Αν ο μετρητής είναι μεγαλύτερος του k και άρα εξασφαλίζεται η k^m-ανωνυμία τότε :
        - Κατασκευάζεται ο κανόνας γενίκευσης που προκύπτει υπολογίζοντας παράλληλα και το κόστος του.
        - Εάν το κόστος του νέου κανόνα είναι μικρότερο του βέλτιστου κόστους έως τώρα, τότε ο νέος κανόνας τον αντικαθιστά και καθίσταται αυτός ο βέλτιστος κανόνας.

6

Αξιολόγηση

Στο κεφάλαιο αυτό θα αναλυθεί η πειραματική διαδικασία που ακολουθήθηκε για την μελέτη και αξιολόγηση του αλγορίθμου που υλοποιήθηκε. Παράλληλα θα αναλυθούν και οι παράμετροι που χρησιμοποιήθηκαν για την αξιολόγηση αυτή και ο τρόπος με τον οποίο υπολογίστηκαν.

6.1 Παράμετροι αξιολόγησης

Για την αξιολόγηση των αλγορίθμων που υλοποιήθηκαν και εκτελέστηκαν, χρησιμοποιήθηκαν 2 βασικές παράμετροι αξιολόγησης. Πρώτη παράμετρος αποτέλεσε η μετρική της απώλειας πληροφορίας, ενώ επιπλέον παράμετρος που αξιολογήθηκε κατά την επεξεργασία των αποτελεσμάτων των εκτελέσεων ήταν σαφώς και ο χρόνος της κάθε εκτέλεσης.

6.1.1 Μετρική Απώλειας πληροφορίας

Η πρώτη παράμετρος που χρησιμοποιήθηκε για την αξιολόγηση των αποτελεσμάτων της εκτέλεσης των αλγορίθμων που υλοποιήθηκαν είναι η μετρική απώλειας πληροφορίας. Κατά την ανωνυμοποίηση ενός πίνακα δεδομένων, ο υπολογισμός της απώλειας της πληροφορίας που προκύπτει λόγω της ανωνυμοποίησης αυτού αποτελεί πολύ σημαντικό παράγοντα

αξιολόγησης, καθώς δίνει πληροφορίες για την αποδοτικότητα της ανωνυμοποίησης και την χρηστικότητα της πληροφορίας που παρέχει ο πίνακας μετά την ανωνυμοποίηση.

Ως μετρική της απώλειας πληροφορίας χρησιμοποιήθηκε η Κανονικοποιημένη Ποινή Βεβαιότητας (Normalized Certainty Penalty - NCP), όπως αυτή έχει περιγραφεί προηγουμένως στο κεφάλαιο 4.

Η Κανονικοποιημένη Ποινή Βεβαιότητας (NCP) ενός στοιχείου p ενός προς ανωνυμοποίηση πίνακα υπολογίζεται ως εξής:

$$NCP(p) = \begin{cases} 0, & \gamma\iota\alpha\ |\mathrm{up}| = 1 \\ \dfrac{|\mathrm{up}|}{|\mathrm{I}|}, & \text{για κάθε άλλο} \end{cases}$$

Όπου:

- p είναι ένα στοιχείο του πεδίου ορισμού I,
- $|u_p|$ είναι ο αριθμός των φύλλων του δέντρου που βρίσκονται κάτω από τη γενίκευση u_p
- |I| είναι ο αριθμός όλων των στοιχείων του πεδίου ορισμού

Για τον υπολογισμό της Κανονικοποιημένης Ποινής Βεβαιότητας ολόκληρου του πίνακα λαμβάνουμε υπ' όψιν και το ειδικό βάρος που έχει η κάθε γενίκευση μέσω του υπολογισμού του πλήθους εμφάνισης στον αρχικό πίνακα του κάθε στοιχείου που γενικεύεται από τον αλγόριθμο.

Έτσι η Κανονικοποιημένη Ποινή Βεβαιότητας (NCP) ενός προς ανωνυμοποίηση πίνακα D, υπολογίζεται ως:

$$NCP(D) = \frac{\sum_{p \in I} Cp \cdot NCP(p)}{\sum_{p \in I} Cp}$$

Όπου: :

- C_p είναι το πλήθος εμφάνισης του στοιχείου p στον αρχικό προς ανωνυμοποίηση πίνακα

6.1.2 Χρόνος εκτέλεσης

Η δεύτερη παράμετρος η οποία ελήφθη υπ 'όψιν κατά την αξιολόγηση των αποτελεσμάτων των αλγορίθμων που εκτελέστηκαν είναι ο χρόνος εκτέλεσης. Ο χρόνος εκτέλεσης δεν μπορεί να θεωρηθεί μια αντικειμενική παράμετρος για την αξιολόγησης ενός αλγορίθμου, καθώς εξαρτάται και επηρεάζεται από την υλοποίηση του αλγορίθμου, καθώς και το περιβάλλον εκτέλεσης του αλγορίθμου, την υπολογιστική ισχύ του μηχανήματος στο οποίο

πραγματοποιείται η εκτέλεση κ.α. Ωστόσο μπορεί να αποτελέσει σίγουρα ένα σημαντικό στοιχείο κατά τη σύγκριση των διαφορετικών αλγορίθμων, γι αυτό και μετρήθηκε και χρησιμοποιήθηκε κατά την διαδικασία της αξιολόγησης των αλγορίθμων που μελετά η παρούσα εργασία.

6.2 Πειραματική διαδικασία

6.2.1 Δεδομένα

Για την διεξαγωγή των πειραμάτων χρησιμοποιήθηκε γεννήτρια παραγωγής τυχαίων δεδομένων, με τη βοήθεια της οποίας κατασκευάστηκαν οι υποτιθέμενοι προς δημοσίευση πίνακες δεδομένων. Έτσι δημιουργήθηκαν δεδομένα τα οποία και χρησιμοποιήθηκαν ως είσοδοι των αλγορίθμων που αναπτύχθηκαν, και πάνω στα οποία βασίστηκε η εκτέλεση και μετέπειτα αξιολόγηση αυτών.

Πιο συγκεκριμένα κατασκευάστηκε αρχικός πίνακας με πλήθος εγγραφών |D|=10.000 και ο πίνακας αυτός χρησιμοποιήθηκε για την εκτέλεση διαφόρων επαναλήψεων των αλγορίθμων τόσο στον αρχικό πίνακα όσο και σε υποσύνολα αυτού.

Τα δεδομένα τα οποία δημιουργήθηκαν αποτελούνταν από μονοψήφιους ακεραίους αριθμούς μεγαλύτερους του μηδενός. Τα υποσύνολα του αρχικού πίνακα που δημιουργήθηκε είχαν πλήθος εγγραφών μεγέθους |D|={10.000,5.000,4.000,2.000,1.000,750,500}. Η κάθε μία από τις εγγραφές του πίνακα είχαν μέγιστο πλήθος στοιχείων n=5.

6.2.2 Διαδικασία εκτέλεσης

Κατά την διεξαγωγή των πειραμάτων εκτελέστηκαν επαναλήψεις των δύο αλγορίθμων της k^m – ανωνυμοποίησης με χρήση προκαθορισμένης ιεραρχίας γενίκευσης και με χρήση δυναμικών ιεραρχιών, όπως αυτοί περιγράφηκαν στα κεφάλαια 4 και 5. Οι εκτελέσεις των αλγορίθμων έγιναν για διαφορετικό πλήθος εγγραφών |D| και διαφορετικές τιμές των παραμέτρων k και m. Για κάθε μια από τις διαφορετικές επαναλήψεις υπολογίσθηκαν και κατεγράφησαν η Κανονικοποιημένη Ποινή Βεβαιότητας (NCP) καθώς και ο χρόνος εκτέλεσης όπως αυτά περιγράφηκαν στην προηγούμενη ενότητα του κεφαλαίου. Τα δεδομένα που προέκυψαν από τους δύο αλγορίθμους συγκρίθηκαν και αξιολογήθηκαν ως προς τις παραπάνω παραμέτρους της απώλειας πληροφορίας με βάση την τιμή της NCP και του χρόνου εκτέλεσης.

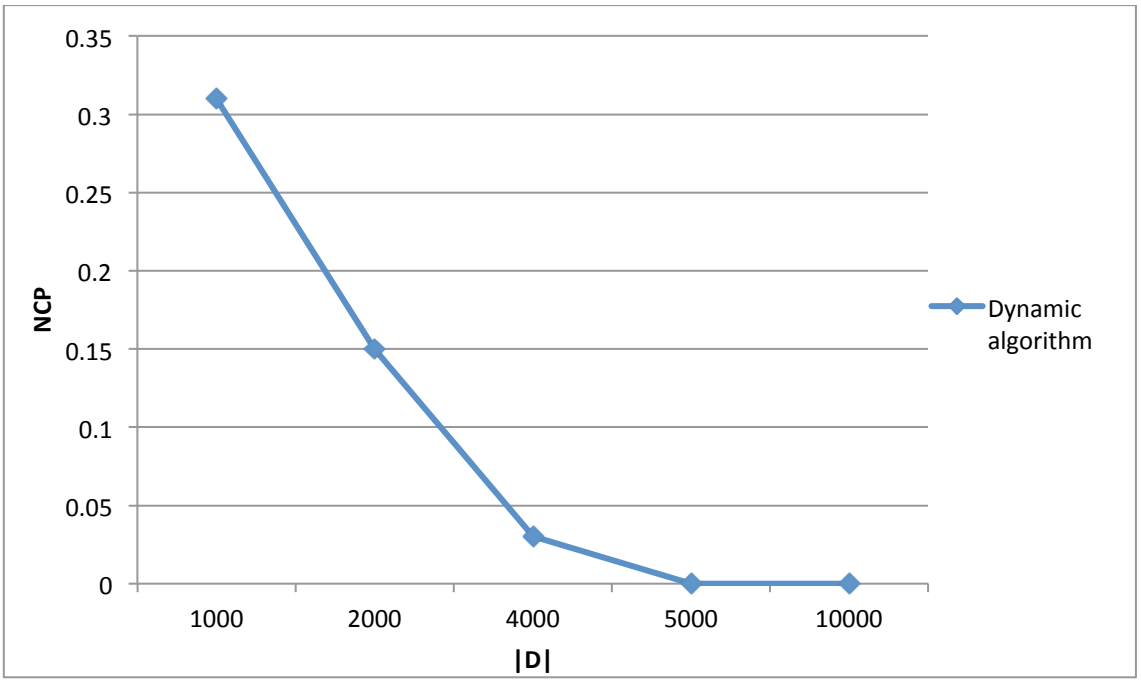
Πιο συγκεκριμένα πραγματοποιήθηκε πλήθος επαναλήψεων των εκτελέσεων των δύο αλγορίθμων για πίνακες μεγέθους |D|={10.000,5.000,4.000,2.000,1.000,750,500}. Επίσης έγιναν διαφορετικές επαναλήψεις με τροποποιήσεις στις παραμέτρους k={200,150,100,50,25,15} καθώς και m={1.2.3).

Όλα τα πειράματα εκτελέστηκαν σε προσωπικό υπολογιστή με επεξεργαστή Intel Core i5-4210M, με CPU 2,60GHz, RAM 8,00 GB και λειτουργικό σύστημα Windows 8.1. Για την ανάπτυξη και την εκτέλεση των πειραμάτων, χρησιμοποιήθηκε το IDE περιβάλλον Code::Blocks.

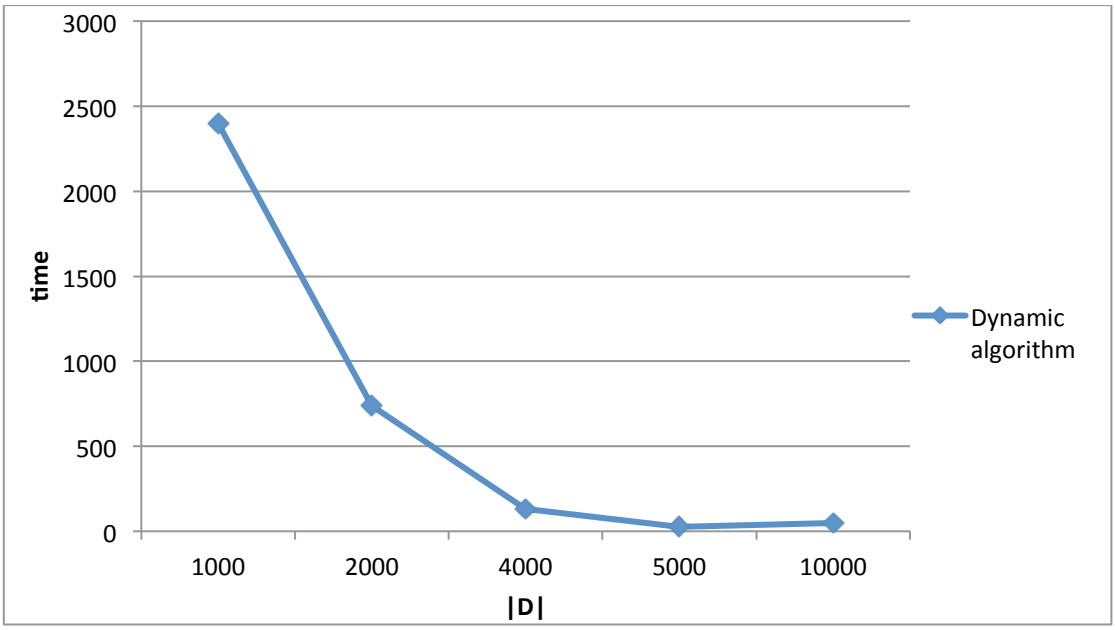
6.3 Αποτελέσματα

Κατά την εκτέλεση των διαφόρων επαναλήψεων των αλγορίθμων που υλοποιήθηκαν, μετρήθηκαν και αναλύθηκαν όπως αναφέρθηκε το κόστος της απώλειας πληροφορίας μέσω της χρήσης της Κανονικοποιημένης Ποινής Βεβαιότητας, καθώς και ο χρόνος εκτέλεσης αυτών. Παρακάτω αναλύονται κάποια εκ των βασικών αποτελεσμάτων τα οποία κρίθηκαν ενδιαφέροντα και άξια σχολιασμού.

Στο πρώτο πείραμα μελετήθηκε η απώλεια πληροφορίας αναλογικά με τον αριθμό των εγγραφών του αρχικού πίνακα. Για σταθερά k=150 και m=3, εκτελέστηκε διαδοχικά ο αλγόριθμος της k^m – ανωνυμοποίησης με χρήση δυναμικών ιεραρχιών γενίκευσης για διάφορα σύνολα με πλήθος εγγραφών |D|={1000,2000,4000,5000,10000} και καταγράφηκε η απώλεια πληροφορίας NCP για κάθε μια από τις παραπάνω εκτελέσεις. Από το παραπάνω πείραμα παρατηρήθηκε πως για σταθερές παραμέτρους k,m το κόστος NCP μειώνεται όσο αυξάνεται το πλήθος των εγγραφών του πίνακα. Το αποτέλεσμα αυτό είναι λογικό και αναμενόμενο καθώς αυξάνοντας τον αριθμό των εγγραφών αυξάνεται και το πλήθος εμφάνισης των διαφόρων συνδυασμών των στοιχείων του πίνακα. Κρατώντας σταθερές τις απαιτήσεις για το μέγεθος των συνδυασμών (m) καθώς και το απαραίτητο πλήθος εμφάνισης αυτών (k), οι συνδυασμοί που δεν ικανοποιούν την k^m – ανωνυμία και συνεπώς χρήζουν γενίκευσης μειώνονται, δικαιολογώντας καθ' αυτόν τον τρόπο την μείωση του κόστους η οποία παρατηρήθηκε στην εν λόγω εκτέλεση.



Εν συνεχεία μελετήθηκε ο χρόνος εκτέλεσης του αλγορίθμου, σε σύγκριση με τον αριθμό των εγγραφών του αρχικού πίνακα. Και σε αυτή τη πειραματική διαδικασία, εκτελέστηκε διαδοχικά ο αλγόριθμος της k^m – ανωνυμοποίησης με χρήση δυναμικών ιεραρχιών γενίκευσης για διάφορα σύνολα με πλήθος εγγραφών |D|={1000,2000,4000,5000,10000} κρατώντας σταθερές τις παραμέτρους k και m (k=150 και m=3), και καταγράφηκε ο χρόνος εκτέλεσης του αλγορίθμου για κάθε μια από τις παραπάνω εκτελέσεις. Τα αποτελέσματα του συγκεκριμένου πειράματος ήταν ιδιαιτέρως ενδιαφέροντα, καθώς φάνηκε πως με την αύξηση του πλήθους των εγγραφών, υπήρχε σε γενικές γραμμές μια πτωτική τάση του χρόνου εκτέλεσης. Ο λόγος και πάλι για τον οποίο παρατηρείται αυτή η τάση έγκειται στο γεγονός ότι με την αύξηση του πλήθους εγγραφών αυξάνεται και το πλήθος εμφάνισης συνδυασμών στοιχείων, και συνεπώς κρατώντας σταθερές τις παραμέτρους k και m, υπάρχουν λιγότερες παραβιάσεις της k^m – ανωνυμίας, μειώνοντας έτσι και τις περιπτώσει που ο αλγόριθμος πρέπει να εξετάσει και να αναζητήσει λύσεις.

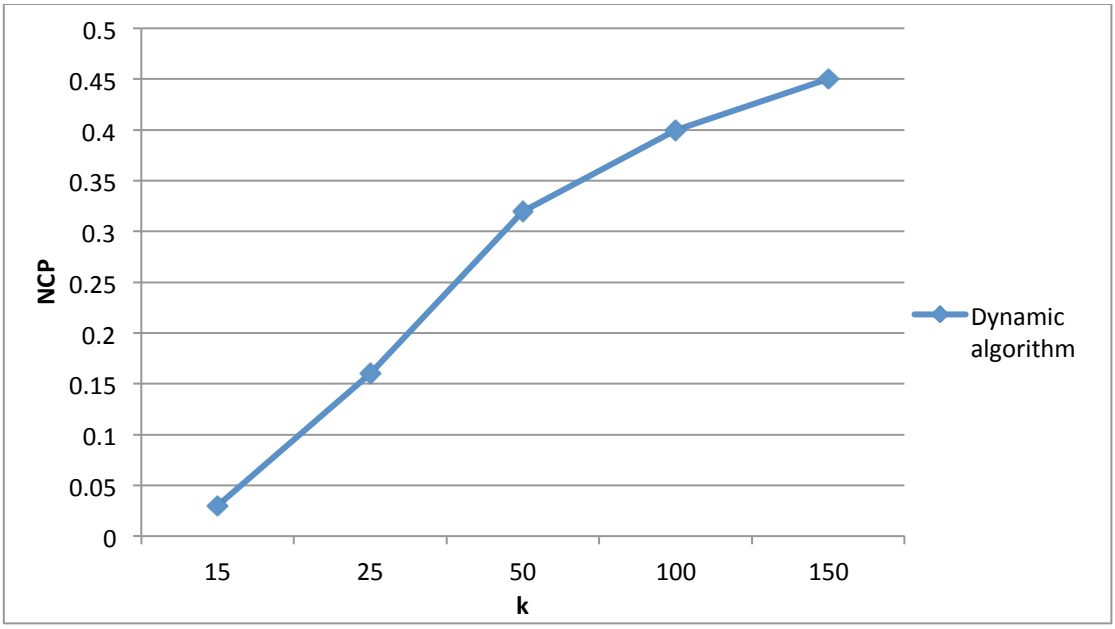


Από το παραπάνω πείραμα καταδεικνύεται επίσης και το γεγονός ότι το χρονοβόρο κομμάτι της εκτέλεσης του αλγορίθμου μας, αποτελεί όπως είναι αναμενόμενο η προσπάθεια εξέτασης των προβληματικών μονοπατιών του δέντρου και η εξεύρεση όλων των πιθανών λύσεων αυτών, καθιστώντας δυνατή επί της ουσίας ακόμη και τη μείωση του χρόνου εκτέλεσης του αλγορίθμου παρά την αύξηση του πλήθους των δεδομένων, αρκεί να μειώνονται οι περιπτώσεις που πρέπει να ερευνηθούν.

Αξίζει πάντως να σημειωθεί ότι η φθίνουσα τάση στο χρόνο εκτέλεσης δεν παρουσιάζει πάντα αυτή τη γνησίως μονοτονία. Ο χρόνος εκτέλεσης, όπως θα φανεί και σε επόμενα διαγράμματα στη συνέχεια, μπορεί να παρουσιάζει και διακυμάνσεις κάποιες φορές, εμφανίζοντας αυξομειώσεις. Το γεγονός αυτό οφείλεται ακριβώς στο ότι ο χρόνος εκτέλεσης εξαρτάται κατά βάση από τις περιπτώσεις που πρέπει να ερευνηθούν, και πόσο νωρίς θα μπορούν να περιοριστούν κάποιοι κλάδοι του δέντρου. Η διάταξη εμφάνισης και εκτέλεσης ορισμένων συνδυασμών μπορεί δηλαδή να παίξει ρόλο επηρεάζοντας τους απόλυτους χρόνους εκτέλεσης. Παρ' όλα αυτά η πτωτική τάση στον χρόνο εκτέλεσης φαίνεται να αποτελεί τη γενικότερη τάση κατά τις εκτελέσεις των διαφόρων πειραμάτων, και γι αυτό και γίνεται ιδιαίτερη μνεία σε αυτή.

Εν συνεχεία μελετήθηκε η απώλεια πληροφορίας ανάλογα με τη διακύμανση της παραμέτρου k. Κατά τη διάρκεια του εν λόγω πειράματος, εκτελέστηκαν διαδοχικά εκτελέσεις του αλγορίθμου της km – ανωνυμοποίησης με χρήση δυναμικών ιεραρχιών γενίκευσης σε πίνακα με σταθερό πλήθος εγγραφών |D|=500 και διατηρώντας σταθερή τη παράμετρο m=3 για διάφορες τιμές της παραμέτρου k (k={15,25,50,100,150}). Κατά την εκτέλεση αυτών των

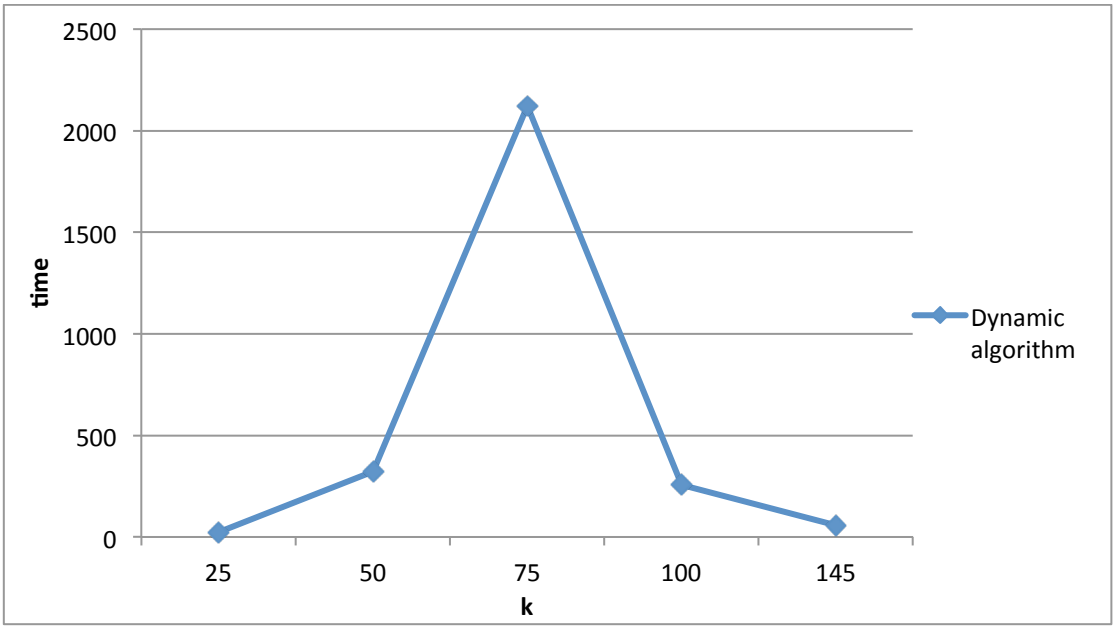
πειραμάτων, παρατηρήθηκε πως όσο αυξάνεται η τιμή k, τόσο αυξάνεται και το κόστος NCP της απώλειας της πληροφορίας. Η παρατήρηση αυτή δεν παρεκκλίνει από ότι αναμέναμε να δούμε, καθώς αυξάνοντας την απαίτηση του πλήθους εμφάνισης ενός συνδυασμού προκειμένου για την ικανοποίηση της k^m – ανωνυμίας, αυξάνεται και το πλήθος των συνδυασμών στον αρχικό πίνακα που δεν καταφέρνουν να ικανοποιήσουν την k^m – ανωνυμία, και κατ' επέκταση και το πλήθος των στοιχείων που πρέπει να γενικευτούν προκειμένου να επιτευχθεί αυτή, αυξάνοντας έτσι και την απώλεια πληροφορίας η οποία αντανακλάται μέσω της τιμής NCP.



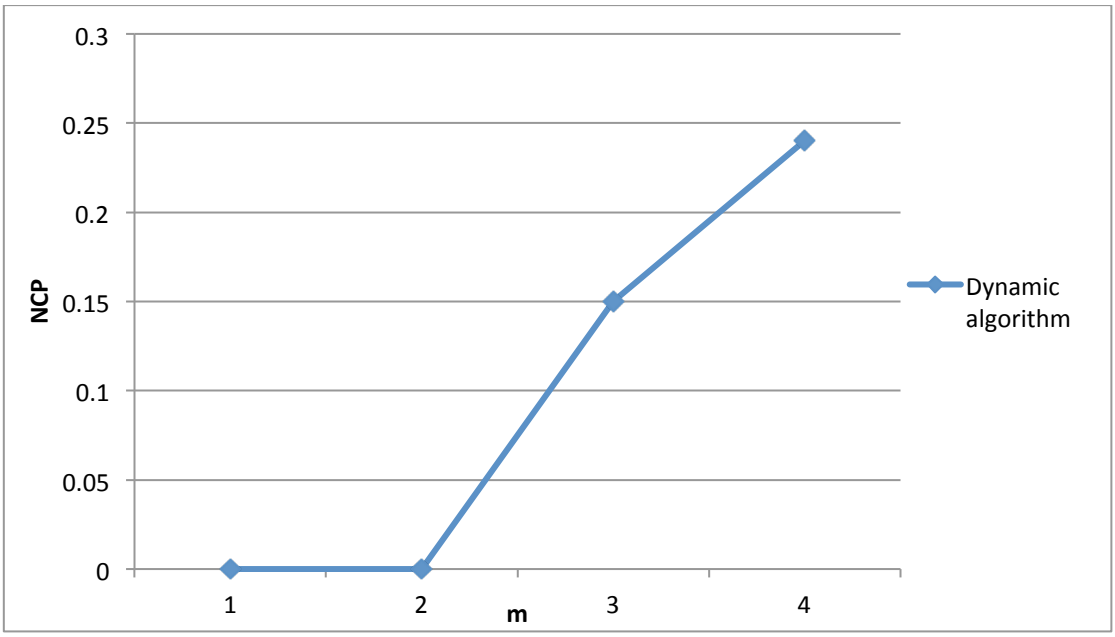
Σε συνέχεια του προηγούμενου πειράματος, μελετήθηκε έπειτα και η σύνδεση ανάμεσα στην παράμετρο k και τον χρόνο εκτέλεσης του αλγορίθμου. Κατά τη συγκεκριμένη πειραματική διαδικασία, δεν φάνηκε να υπάρχει μια συγκεκριμένη αναλογικότητα του χρόνου σε σχέση με την παράμετρο k. Γενικότερα παρατηρήθηκαν συχνές διακυμάνσεις του χρόνου εκτέλεσης για διάφορες τιμές του k. Ωστόσο σε γενικές γραμμές, φαίνεται να υπάρχει μια τάση αύξησης του χρόνου εκτέλεσης γύρω από ορισμένες τιμές του k, και μείωσης έξω από τα όρια αυτά, όπως παρατηρείται και στο παρακάτω διάγραμμα. Με την αύξηση της παραμέτρου k, αυξάνεται το πλήθος εμφάνισης που απαιτείται προκειμένου ένας συνδυασμός να ικανοποιεί την k^m – ανωνυμία, αυξάνοντας έτσι και το πλήθος των συνδυασμών που παραβιάζουν την k^m – ανωνυμία. Η αύξηση αυτή των περιπτώσεων που πρέπει να μελετηθούν αυξάνει συνεπώς και το χρόνο εκτέλεσης του αλγορίθμου. Ωστόσο, φαίνεται αυτό να επηρεάζει μέχρι ενός σημείου και από εκεί και πέρα ο χρόνος εκτέλεσης αρχίζει να πέφτει ξανά. Η υλοποίηση του αλγορίθμου μας αναζητά πιθανές λύσεις και εξετάζει μόνο αυτές για να βρει τη λύση με το χαμηλότερο κόστος. Έτσι, παρά το γεγονός ότι συνεχίζουν να αυξάνονται οι παραβιάσεις της k^m – ανωνυμίας, από ένα σημείο κι έπειτα η αυστηροποίηση της παραμέτρου k προκαλεί

σαφώς και τον περιορισμό των πιθανών λύσεων, μειώνοντας έτσι ξανά τις περιπτώσεις που μπορούν να ερευνηθούν ως λύσεις, προκαλώντας έτσι αυτή τη μείωση στο χρόνο εκτέλεσης που παρατηρούμε στο παρακάτω διάγραμμα.

Στο πείραμα αυτό, κρατήσαμε σταθερό το πλήθος εγγραφών του πίνακα (|D|=750) και τη παράμετρο m=3 και εκτελέστηκε διαδοχικά και πάλι ο αλγόριθμος της km – ανωνυμοποίησης με χρήση δυναμικών ιεραρχιών γενίκευσης για διάφορες τιμές της παραμέτρου k (k={25,50,75,100,150}). ΤΑ αποτελέσματα όπως αναλύθηκε και προηγουμένως παρουσιάζουν μια αυξητική τάση του χρόνου μέχρι ενός μέγιστου σημείου, και από εκεί και πέρα φαίνεται να υπάρχει μείωση του χρόνου εκτέλεσης του αλγορίθμου, παρά την αύξηση της παραμέτρου k.



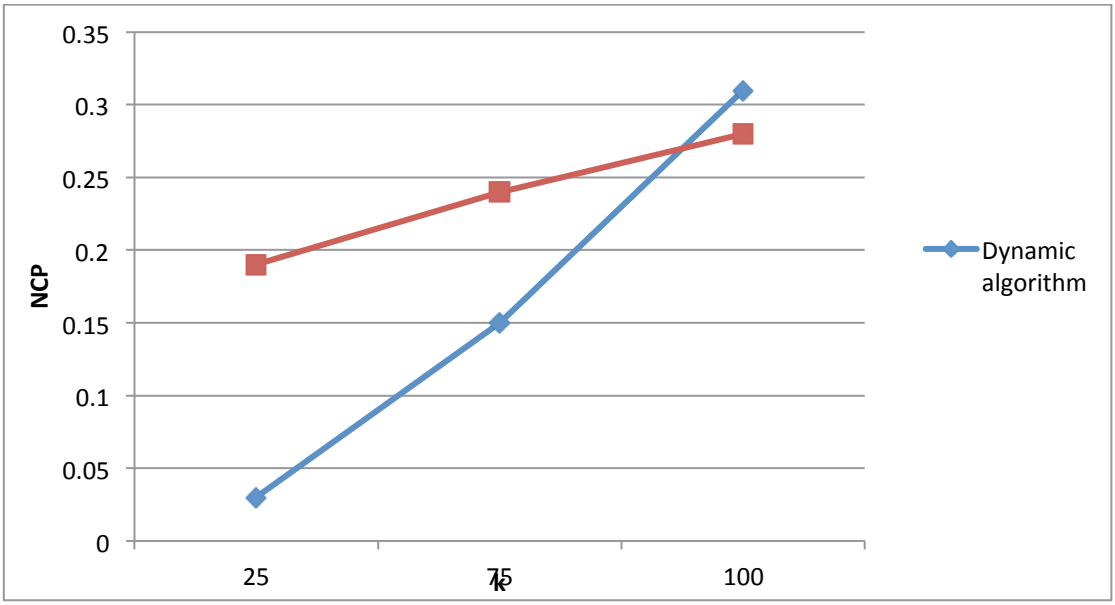
Το επόμενο πείραμα που διεξήχθη είχε ως αντικείμενο μελέτης την επήρεια της παραμέτρου m στο κόστος της απώλειας πληροφορίας. Στο συγκεκριμένο πείραμα, για σταθερό αριθμό k=50 εκτελέστηκε σε διάφορες επαναλήψεις ο αλγόριθμος σε πίνακα με πλήθος εγγραφών |D|=750, αλλάζοντας κάθε φορά τη μεταβλητή m (m={1,2,3,4}). Από την εκτέλεση των πειραμάτων παρατηρούμε πως η τιμή της Κανονικοποιημένης Ποινής Βεβαιότητας αυξάνεται καθώς αυξάνεται και η παράμετρος m. Τα αποτελέσματα όπως ήταν αναμενόμενο επιβεβαιώνουν το γεγονός ότι με την αύξηση του μεγέθους των συνδυασμών που μπορεί ο επιτιθέμενος να γνωρίζει, αυξάνονται και οι γενικεύσεις που πρέπει να πραγματοποιηθούν προκειμένου η γνώση αυτή του επιτιθέμενου να μην μπορεί να προκαλέσει παραβίαση της k^m – ανωνυμίας.



Τέλος, ως κλείσιμο της πειραματικής διαδικασίας που διεξήχθη, θεωρήθηκε σκόπιμη και η σύγκριση μεταξύ του νέου αλγορίθμου της k^m – ανωνυμίας με χρήση δυναμικών ιεραρχιών γενίκευσης που υλοποιήθηκε κατά τη παρούσα εργασία, και του κλασσικού αλγορίθμου της k^m – ανωνυμίας, ώστε να μπορούν να βρεθούν τα δυνατά και αδύναμα σημεία του κάθε αλγορίθμου.

Κατά το πρώτο σκέλος του πειράματος αυτού, συγκρίθηκε η εκτέλεση των δύο αλγορίθμων ως προς το κόστος της απώλειας πληροφορίας που προκαλούν. Για το λόγο αυτό, εκτελέστηκαν επαναλήψεις τόσο του αλγορίθμου της k^m – ανωνυμίας με χρήση δυναμικών ιεραρχιών γενίκευσης, όσο και ο κλασσικός αλγόριθμος της k^m – ανωνυμίας με χρήση προκαθορισμένων ιεραρχιών γενίκευσης. Οι εκτελέσεις όλες έγιναν σε πίνακα με πλήθος εγγραφών |D|=750 και για παράμετρο m=3. Κατά τις διάφορες επαναλήψεις τροποποιούνταν η τιμή της παραμέτρου k (k={25,75,100) και καταγράφονταν η απώλεια πληροφορίας μέσω της τιμής της Κανονικοποιημένης Ποινής Βεβαιότητας. Τα συμπεράσματα του συγκεκριμένου πειράματος είναι ιδιαιτέρως ενδιαφέροντα. Πιο συγκεκριμένα παρατηρήθηκε όπως αναμενόταν ότι η χρήση δυναμικών ιεραρχιών είναι σε θέση να παράγει καλύτερα αποτελέσματα από τον κλασσικό αλγόριθμο της k^m – ανωνυμίας. Η παρατήρηση αυτή ήταν αναμενόμενη, καθώς το όλο νόημα της χρήσης δυναμικών ιεραρχιών γενίκευσης είναι ακριβώς η δυνατότητα δημιουργίας της ιεραρχίας γενίκευσης με δυναμικό τρόπο, ανάλογα με τις ανάγκες για εξασφάλιση της km-ανωνυμίας, αποφεύγοντας καθ' αυτόν τον τρόπο την αναγκαστική γενίκευση στοιχείων του πίνακα που δεν είναι απαραίτητο να γενικευτούν, αλλά παρά ταύτα γενικεύονται κατά τον κλασσικό αλγόριθμο διότι τυγχάνει να βρίσκονται σε σημείο της ιεραρχίας που πρέπει να γενικευτεί λόγω άλλου στοιχείου που παραβιάζει την k^m

– ανωνυμία. Παρ' όλα αυτά, παρατηρούμε ότι αυτό δεν είναι πάντα απαραίτητο. Διαπιστώνουμε λοιπόν πως υπάρχουν περιπτώσεις όπου ο κλασσικός αλγόριθμος μπορεί να έχει και καλύτερα αποτελέσματα όσον αφορά την απώλεια πληροφορίας, από τον αλγόριθμο των δυναμικών ιεραρχιών. Ο λόγος που συμβαίνει αυτό, έχει να κάνει με την υλοποίηση του αλγορίθμου των δυναμικών ιεραρχιών. Ο αλγόριθμος που αναπτύχθηκε, κατά την εκτέλεσή του, δεν αναζητά πάντοτε την βέλτιστη λύση ανάμεσα σε όλες τις λύσεις, αλλά αναζητά τη βέλτιστη λύση ανάμεσα στις λύσεις ενός συγκεκριμένου μεγέθους κάθε φορά, και που προκύπτουν από την ανάλυση των αδελφών κόμβων του κόμβου που προκαλεί την παραβίαση της k^m – ανωνυμίας. Η λύση αυτή λοιπόν μπορεί να μην αποτελεί τη βέλτιστη λύση γενικότερα για το συγκεκριμένο πρόβλημα. Μια προκατασκευασμένη ιεραρχία η οποία είναι ταξινομημένη με τέτοιο τρόπο ώστε η λύση που δίδεται για ένα πρόβλημα να αποτελεί πράγματι τη βέλτιστη λύση αυτού του προβλήματος, μπορεί με αυτόν τον τρόπο να επιστρέψει καλύτερα αποτελέσματα από την συγκεκριμένη υλοποίηση του αλγορίθμου των δυναμικών ιεραρχιών.



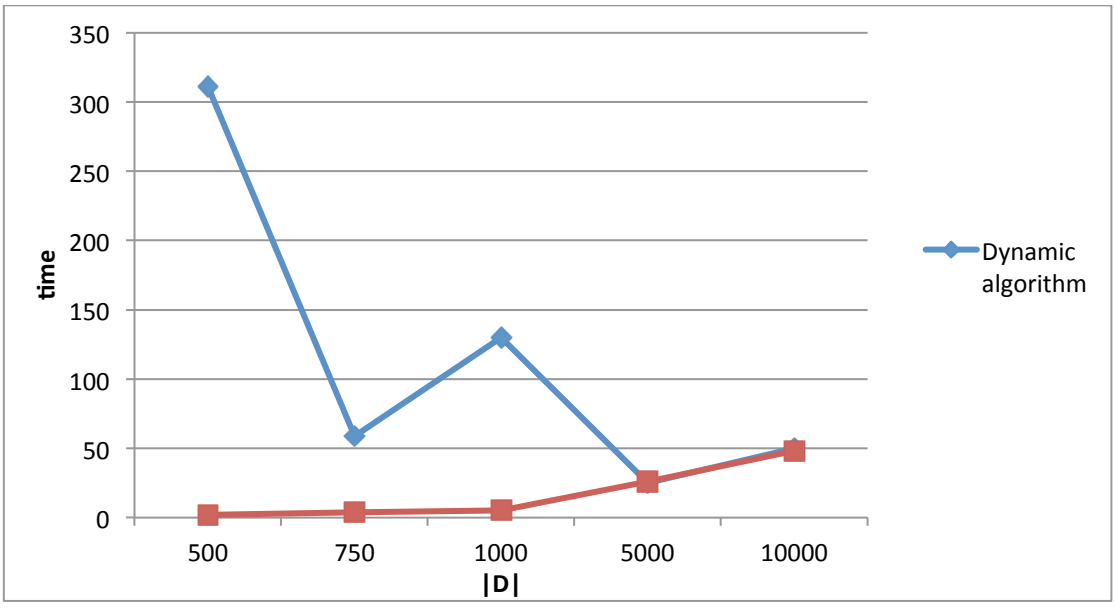
Γενικότερα μπορούμε να διαπιστώσουμε ότι η απώλεια πληροφορίας στον κλασσικό αλγόριθμο εξαρτάται κατά μεγάλο λόγο από τη δομή του δέντρου της ιεραρχίας γενίκευσης που υπάρχει πριν την εκτέλεση. Η ανάλυση του συγκεκριμένου ζητήματος δεν κρίνεται σκόπιμη στα πλαίσια της παρούσας εργασίας η οποία εστιάζει περισσότερο στον αλγόριθμο των δυναμικών ιεραρχιών.

Το τελευταίο πείραμα το οποίο διεξήχθη, προσπάθησε να κάνει μια γενικότερη σύγκριση του χρόνου εκτέλεσης των δύο αλγορίθμων. Ο χρόνος εκτέλεσης δεν μπορεί να συγκριθεί με απόλυτο τρόπο για τους δύο αλγορίθμους, καθώς εξαρτάται σε μεγάλο βαθμό από την

υλοποίηση του κάθε αλγορίθμου, καθώς και την δομή της ιεραρχίας που θα χρησιμοποιηθεί στον κλασσικό αλγόριθμο. Ωστόσο επιχειρήθηκε μια γενικότερη σύγκριση των δύο αλγορίθμων χωρίς να μπούμε σε λεπτομέρειες επί ακριβών τιμών.

Έτσι, γενικότερα μπορεί να παρατηρηθεί πως η εκτέλεση του αλγορίθμου της k^m – ανωνυμίας με χρήση δυναμικών ιεραρχιών γενίκευσης είναι σε γενικές γραμμές πιο χρονοβόρος από τον κλασσικό αλγόριθμο. Η παρατήρηση αυτή είναι αναμενόμενη καθώς ο αλγόριθμος των δυναμικών ιεραρχιών δεν ακολουθεί μια προκαθορισμένη ιεραρχία, αλλά καλείται να αναζητήσει πλήθος πιθανών λύσεων που μπορεί να επιλύουν τους προβληματικούς κόμβους που έχουν βρεθεί, γεγονός που εκ των πραγμάτων θα αποτελεί και πιο χρονοβόρα διαδικασία. Ωστόσο παρατηρήθηκε κι εδώ η συμπεριφορά του αλγορίθμου των δυναμικών ιεραρχιών που συζητήθηκε και προηγουμένως να εμφανίζει μια φθίνουσα τάση ο χρόνος εκτέλεσης καθώς αυξάνεται το πλήθος των εγγραφών. Αντίστοιχα ο χρόνος εκτέλεσης του κλασσικού αλγορίθμου τείνει να έχει αυξητική τάση με την αύξηση του πλήθους των εγγραφών. Τα δύο αυτά φαινόμενα δεν αποκλείεται λοιπόν να φέρνουν πιο κοντά τους χρόνους εκτέλεσης των δυο αλγορίθμων καθώς αυξάνεται το πλήθος εγγραφών |D|, όπως μπορεί να παρατηρηθεί και στο παρακάτω διάγραμμα.

Για το παρακάτω πείραμα εκτελέστηκαν διάφορες επαναλήψεις και των δύο αλγορίθμων για διαφορετικές τιμές του πλήθους |D| των εγγραφών του πίνακα (|D|={500,750,1000,5000,10000}) και για σταθερές τις παραμέτρους k και m (k=150 και m=3).



7

Συμπεράσματα

7.1 Σύνοψη και συμπεράσματα

Η παρούσα εργασία ασχολήθηκε με το ζήτημα της προστασίας της ιδιωτικότητας μέσω του αλγορίθμου της k^m – ανωνυμίας χωρίς ωστόσο να αξιοποιείται κάποια προκαθορισμένη ιεραρχία γενίκευση όπως συμβαίνει στην περίπτωση του κλασσικό αλγορίθμου της k^m - ανωνυμοποίησης..

Πιο συγκεκριμένα, στα πλαίσια της εργασία αναπτύχθηκε αλγόριθμος ο οποίος εξασφαλίζει την k^m – ανωνυμοποίηση ενός συνόλου δεδομένων, χωρίς ωστόσο να αξιοποιεί κάποια προκαθορισμένη ιεραρχία γενίκευσης. Αντιθέτως ο αλγόριθμος αναπτύσσει τις απαραίτητες ιεραρχίες γενίκευσης με δυναμικό τρόπο, γενικεύοντας στοιχεία τα οποία είναι απαραίτητα για την εξασφάλιση της k^m – ανωνυμίας. Παράλληλα υλοποιήθηκε και ο κλασσικός αλγόριθμος της k^m – ανωνυμοποίησης. Οι δύο αλγόριθμοι που υλοποιήθηκαν, εκτελέστηκαν σε διάφορες παραλλαγές, με διαφορετικές εισόδους και παραμέτρους, και συγκρίθηκαν ως προς τον χρόνο εκτέλεσής τους, καθώς και ως προς την απώλεια πληροφορίας που επιφέρουν στο αρχικό σύνολο δεδομένων, μέσω της μετρικής απώλειας πληροφορίας της Κανονικοποιημένης τιμής Βεβαιότητας.

Από τη διεξαγωγή των πειραμάτων προέκυψε πως αν και είναι πιο χρονοβόρος ο νέος αλγόριθμος που αναζητά την k^m – ανωνυμία χτίζοντας δυναμικά τις ιεραρχίες γενίκευσης,

εντούτοις εξασφαλίζει μικρότερη απώλεια πληροφορίας από τον κλασσικό αλγόριθμο της km- ανωνυμοποίησης. Η διαφορά στο χρόνο εκτέλεσης των δύο αλγορίθμων ήταν αναμενόμενη καθώς ο αλγόριθμος της k^m – ανωνυμίας με χρήση δυναμικών ιεραρχιών γενίκευσης δεν εξετάζει συγκεκριμένες προκαθορισμένες γενικεύσεις, αλλά πλήθος πιθανών γενικεύσεων, για την εξεύρεση της βέλτιστης δυνατής πιθανής λύσης στα προβλήματα που προκαλούν την παραβίαση της k^m - ανωνυμίας. Το τίμημα αυτό ανταλλάσσεται με την επίτευξη μικρότερης απώλειας πληροφορίας στο τελικό ανωνυμοποιημένο σύνολο δεδομένων.

Η δυνατότητα επίτευξης μικρότερης απώλειας πληροφορίας από το νέο αλγόριθμο που αναπτύχθηκε αποδίδεται στο γεγονός ότι ενώ ο πρώτος αλγόριθμος αναγκάζεται να ακολουθεί προκαθορισμένες γενικεύσεις, γενικεύοντας έτσι συχνά και στοιχεία που δεν απαιτούν γενίκευση μόνο και μόνο γιατί τυγχάνει να βρίσκονται στον ίδιο κλάδο με κάποιο στοιχείο που τη χρειάζεται, ο δεύτερος αλγόριθμος της k^m – ανωνυμοποίησης με χρήση δυναμικών ιεραρχιών γενίκευσης δεν επιδέχεται τέτοιων περιορισμών, αναζητώντας κάθε φορά και προχωρώντας μόνο στις απαραίτητες για την επίτευξη της k^m-ανωνυμίας γενικεύσεις.

7.2 Μελλοντικές επεκτάσεις

Ο αλγόριθμος ο οποίος αναπτύχθηκε στα πλαίσια της παρούσας εργασίας, αναζητά και εφαρμόζει τις γενικεύσεις εκείνες οι οποίες είναι απαραίτητες ώστε να επιτευχθεί η k^m – ανωνυμοποίηση ενός συνόλου δεδομένων.

Βασικό σημείο στο οποίο υστερεί ο αλγόριθμος αυτός αποτελεί ο χρόνος εκτέλεσής του, ο οποίος και μπορεί ενδεχομένως να περιορίζει τις δυνατότητες της πρακτικής εφαρμογής αυτού. Κατά συνέπεια θα άξιζε ως μια μελλοντική επέκταση του εν λόγω αλγορίθμου να ερευνηθεί η δυνατότητα χρήσης επιπλέον μεθόδων που θα μπορέσουν να το καταστήσουν ακόμη αποδοτικότερο από πλευράς χρόνου εκτέλεσης, καθιστώντας τον έτσι ακόμη πιο προσιτό και δελεαστικό προς χρήση σε πρακτικές εφαρμογές κατά την ανωνυμοποίηση δεδομένων.

Επιπλέον, ο αλγόριθμος που αναπτύχθηκε αναζητά βέλτιστες λύσεις σε τοπικό επίπεδο, εξετάζοντας γενικεύσεις με τους αδελφούς κόμβους των προβληματικών κόμβων που προκαλούν παραβιάσεις της k^m – ανωνυμοποίησης. Το γεγονός αυτό καθιστά τον αλγόριθμο αποδοτικότερο από πλευράς χρόνου, αλλά η αναζήτηση λύσεων σε μεγαλύτερο εύρος πιθανών γενικεύσεων θα μπορούσε να εξασφαλίσει ακόμη χαμηλότερη απώλεια της αρχικής

πληροφορίας. Θα μπορούσε λοιπόν η παραπάνω προσθήκη να αποτελέσει επίσης μια αξιόλογη μελλοντική επέκταση τους αλγορίθμου.

Τέλος, μια ενδιαφέρουσα επέκταση του αλγορίθμου που αναπτύχθηκε, θα μπορούσε να αποτελέσει η επέκτασή της λογικής της ανάπτυξης δυναμικών ιεραρχιών και σε άλλους αλγορίθμους ανωνυμοποίησης δεδομένων, όπως για παράδειγμα της απλής k – ανωνυμοποίησης, ή παραλλαγών αυτής που επίσης βασίζονται στη γενίκευση των στοιχείων του αρχικού συνόλου βάσει μιας προκαθορισμένης ιεραρχίας γενίκευσης. Με αυτόν τον τρόπο μπορούν να αξιοποιηθούν και τα πλεονεκτήματα που παρέχουν και άλλοι αλγόριθμοι ανωνυμοποίησης, δίνοντας έτσι τη δυνατότητα αξιοποίησης και της δυναμικής δημιουργίας των ιεραρχιών γενίκευσης και στη πρόληψη διαφορετικού τύπου επιθέσεων που αντιμετωπίζουν οι αλγόριθμοι αυτοί.

8

Βιβλιογραφία

[LDR05] K. LeFevre, D. J. DeWitt, R. Ramakrishnan, Incognito: Efficient Full-Domain k-Anonymity, In Proc. Special Interest Group on Management Data, 2005

[LDR06] K. LeFevre, D. J. DeWitt, R. Ramakrishnan, Mondrian : Multidimensional k-Anonymity, In Proc. IEEE Intl. Conference on Data Engineering, 2006

[LLV07] N. Li, T. Li, S. Venkatasubramanian. t-Closeness: Privacy Beyond k-Anonymity and l-Diversity, In Proc. Intl. Conference on Data Engineering, 2007

[MGK+06] A. Machanavajjhala, J. Gehrke, D. Kifer, M. Venkitasubramaniam. l-Diversity: Privacy Beyond k-Anonymity, In Proc. IEEE Intl. Conference on Data Engineering, 2006

[NAC07] M.E. Nergiz, M. Atzori, C. Clifton. Hiding the Presence of Individuals from Shared Databases, In Proc. Special Interest Group on Management of Data, 2007

[NC06] M. Nergiz, C. Clifton. Thoughts on k-Anonymization, In Proc. IEEE Intl. Conference on Data Engineering Workshops, 2006

[Swe02] L. Sweeney. *k*-Anonymity: A Model for Protecting Privacy. International Journal on Uncertainty, Fuzziness and Knowledge-based Systems, Volume 10, no. 5, 2002

[TMK08] M. Terrovitis, N. Mamoulis, P. Kanlis, Privacy-preserving Anonymization of Set-valued Data, Very Large Data Bases,2008

[XT06] X. Xiao, Y. Tao. Anatomy: Simple and Effective Privacy Preservation, In Proc. Very Large Data Bases, 2006

[XT07] X. Xiao, Y. Tao , m-Invariance: Towards Privacy Preserving Re-publication
of Dynamic Datasets, In Proc. Special Interest Group on Management of Data, 200

[XWP+06] J. Xu, W. Wang, J. Pei, X. Wang, B. Shi, A. Fu, Utility-Based Anonymization Using Local Recoding, Special Interest Group KDD, 2006