Ανοικτό Πανεπιστήμιο Κύπρου

Σχολή Θετικών και Εφαρμοσμένων Επιστημών

Μεταπτυχιακή Διατριβή
στα Πληροφοριακά Συστήματα

Απόκρυψη Κανόνων Συσχέτισης (Association Rule Hiding) με Τεχνικές Αναδόμησης Βάσης Δεδομένων

Ιωάννης Μουμούρης

Επιβλέπων Καθηγητής
Βασίλειος Βερύκιος

Μάιος 2013

Ανοικτό Πανεπιστήμιο Κύπρου

Σχολή Θετικών και Εφαρμοσμένων Επιστημών

Απόκρυψη Κανόνων Συσχέτισης (Association Rule Hiding) με Τεχνικές Αναδόμησης Βάσης Δεδομένων

Ιωάννης Μουμούρης

Επιβλέπων Καθηγητής
Βασίλειος Βερύκιος

Η παρούσα μεταπτυχιακή διατριβή υποβλήθηκε
προς μερική εκπλήρωση των απαιτήσεων για απόκτηση

μεταπτυχιακού τίτλου σπουδών
στα Πληροφοριακά Συστήματα

από τη Σχολή Θετικών και Εφαρμοσμένων Επιστημών
του Ανοικτού Πανεπιστημίου Κύπρου

Ιούνιος 2013

Περίληψη

Από τη στιγμή της γέννησής της, η διατήρηση της ιδιωτικότητας κατά την εξόρυξη δεδομένων είναι ένα πολύ ενεργό και ενδιαφέρον πεδίο έρευνας στην ευρύτερη περιοχή του Data Mining, που επικεντρώνεται στην διερεύνηση εκείνων των συνεπειών και παρενεργειών της υπάρχουσας τεχνολογίας του Data Mining, οι οποίες πηγάζουν από την διείσδυση στην ιδιωτικότητα προσώπων και οργανισμών. Η απόκρυψη των κανόνων συσχέτισης, αποτελεί προφανώς ένα από τα επί μέρους προβλήματα που συναντώνται σ' αυτήν ακριβώς την επιστημονική περιοχή (της διατήρησης της ιδιωτικότητας κατά την εξόρυξη δεδομένων), για την επίλυση του οποίου έχουν προταθεί μέχρι σήμερα διάφορες τεχνικές.

Σκοπός της διατριβής, είναι η συγκριτική μελέτη ορισμένων τεχνικών που έχουν προταθεί για την επίλυση του προβλήματος της απόκρυψης των κανόνων συσχέτισης, οι οποίες στηρίζονται στην μέθοδο της Αναδόμησης της Βάσης Δεδομένων, αλλά και η πρόταση νέων βελτιωμένων τεχνικών με βάση πάντα την Αναδόμηση της Βάσης Δεδομένων.

Για την επίτευξη του παραπάνω στόχου, αφού μελετήθηκαν αρκετές τεχνικές, επιλέχθηκαν δύο που έχουν παρουσιασθεί αρκετά πρόσφατα, οι οποίες αφού μελετήθηκαν αναλυτικά, εντοπίσθηκαν τα προβλήματα και οι αδυναμίες του και στη συνέχεια οι βελτιωμένοι αλγόριθμοι που προτείνουμε, υλοποιήθηκαν στην γλώσσα προγραμματισμού R και δοκιμάσθηκαν σε αρκετά σύνολα συναλλαγών, ώστε να παραχθούν όσο το δυνατόν πιο αντιπροσωπευτικά αποτελέσματα και να εξαχθούν πιο αξιόπιστα αποτελέσματα.

Επίσης, ακριβώς λόγω της μεγάλης πολυπλοκότητας των διαφόρων τεχνικών, είναι πολύ σημαντικό να υπάρξει ένα εργαλείο που θα δίνει στον χρήστη την δυνατότητα να αξιολογήσει τις διάφορες τεχνικές σε σχέση με την ακρίβεια των αποτελεσμάτων αλλά και την πολυπλοκότητά τους, ώστε να είναι σε θέση να επιλέξει αυτήν που ταιριάζει καλύτερα στις δικές του ανάγκες κάθε φορά.

Summary

Since its first appearance, Privacy Preserving Data Mining, has been a very interesting research field in the Data Mining community. It investigates the side effetcts of data mining methods proceeding from the penetration into the privacy of individuals and organizations. Association Rule Hiding constitutes one of the problems that can be identified in this scientific area (Privacy Preserving Data Mining) for the solution of which, a number of methods have been proposed until nowdays.

The aim of this work is the relative stydy of specific techniques that have been proposed for the solution of the Association Rule Hiding problem, which are based on the method of Database Reconstruction but also the proposal of new upgraded tecniques which are focused on Database Reconstruction.

For the attainment of the target mentioned above and after having studied certain tecniques, two have been chosen, having been extensively examined and their problems and weaknesses were identified. After this, the improved algorithms that we introduced were depicted in the programming language R, and were tested in a number of transactions, so as to produce as many representative results as possible and to ensure the extraction of more reliable results.

Besides due to the great complexity of various tecnhiques, it is greatly important for a tool to exist, which will give the user the possibility to considerate several tecniques in relation to the accuracy of their results and their complexity itself, so as to be able to choose the one which best responds to his needs each time.

Ευχαριστίες

Η εκπόνηση της παρούσας διατριβής θα ήταν αδύνατο να πραγματοποιηθεί χωρίς τις συμβουλές, τις χρήσιμες υποδείξεις και γενικότερα την πολύτιμη βοήθεια, που μου προσέφερε ο επιβλέπων καθηγητής κος Βασίλειος Βερύκιος. Θα ήθελα να του εκφράσω με ειλικρίνεια και θέρμη, ένα μεγάλο «Ευχαριστώ».

Περιεχόμενα

Κεφάλαιο 1
Εισαγωγή

Στο 1ο Κεφάλαιο γίνεται μία εισαγωγή στο ευρύτερο θέμα της διατριβής, δηλαδή στην Απόκρυψη των Κανόνων Συσχέτισης. Στην ενότητα 1.1 κάνουμε μία ιστορική αναδρομή της Εξόρυξης Δεδομένων ενώ στην ενότητα 1.2, κάνουμε μία σύντομη παρουσίασή της. Στις δύο επόμενες ενότητες παρουσιάζουμε την αναγκαιότητα για την προστασία της ιδιωτικότητας στην Εξόρυξη Δεδομένων γενικά και στην Εξόρυξη Κανόνων Συσχέτισης ειδικότερα. Τέλος, στην ενότητα 1.5, αναφέρουμε μέρος της σχετικής έρευνας που έχει γίνει μέχρι σήμερα.

1.1 Γενικά

Οι ραγδαίες εξελίξεις στη συλλογή και αποθήκευση δεδομένων, μας επιτρέπουν να αποθηκεύουμε τεράστιες ποσότητες δεδομένων και να εξάγουμε χρήσιμες πληροφορίες μέσα από τις βάσεις δεδομένων (ή αποθήκες πληροφοριών) που έχουμε δημιουργήσει. Εξόρυξη δεδομένων είναι η εξεύρεση μιας ενδιαφέρουσας, αυτονόητης, μη προφανούς και πιθανώς χρήσιμης πληροφορίας, από πολύ μεγάλες Βάσεις Δεδομένων, με χρήση σύγχρονων, κατάλληλων αλγορίθμων και των αρχών και κανόνων της Στατιστικής. Η ουσιαστική διαφορά με την παραδοσιακή επιστημονική περιοχή της Διαχείρισης των Βάσεων Δεδομένων, είναι ότι με τα Συστήματα Διαχείρισης Βάσεων Δεδομένων γίνεται εξαγωγή χρήσιμων πληροφοριών που

ήδη υπάρχουν μέσα στις βάσεις δεδομένων ως αποθηκευμένα δεδομένα, ενώ με την Εξόρυξη Δεδομένων, πραγματοποιείται ανακάλυψη νέας πληροφορίας (ή αλλιώς γνώσης) μέσα από τις υπάρχουσες βάσεις δεδομένων. Έτσι, οι τεχνικές της εξόρυξης δεδομένων εφαρμόζονται για να ερευνήσουν σε βάθος μεγάλες βάσεις δεδομένων με σκοπό την ανακάλυψη νέων και χρήσιμων προτύπων που σε διαφορετική περίπτωση θα παρέμεναν άγνωστα, αλλά και την παροχή δυνατοτήτων πρόβλεψης μιας μελλοντικής παρατήρησης.

1.2 Εργασίες της εξόρυξης δεδομένων

Οι εργασίες της εξόρυξης δεδομένων, χωρίζονται σε δύο κύριες κατηγορίες:

1. Προγνωστικές εργασίες Στόχος αυτών των εργασιών είναι να προβλέψουν την τιμή ενός συγκεκριμένου χαρακτηριστικού, βασιζόμενες στις τιμές άλλων χαρακτηριστικών.

2. Περιγραφικές εργασίες. Στόχος αυτών των εργασιών είναι να εξάγουν υποδείγματα (όπως συσχετίσεις, συστάδες, τροχιές, ανωμαλίες) που συνοψίζουν τις βασικές σχέσεις που υπάρχουν στα δεδομένα.

Οι τέσσερις βασικές εργασίες της εξόρυξης δεδομένων είναι οι ακόλουθες:

1. **Προγνωστική μοντελοποίηση**. Αναφέρεται στην εργασία δημιουργίας ενός μοντέλου για τη μεταβλητή στόχο σαν συνάρτηση των ανεξάρτητων μεταβλητών. Υπάρχουν δύο τύποι εργασιών προγνωστικής μοντελοποίησης:

    1.1. Κατηγοριοποίηση: Ανάθεση αντικειμένων σε προκαθορισμένες κλάσεις, επομένως χρησιμοποιείται για διακριτές μεταβλητές στόχους (π.χ. η πρόβλεψη για το εάν ένας δανειολήπτης θα ανταποκριθεί ή όχι στην υποχρέωση αποπληρωμής του δανείου)

    1.2. Παλινδρόμηση: Χρησιμοποιείται για συνεχείς μεταβλητές στόχους (π.χ. η πρόβλεψη της μελλοντικής τιμής μιας μετοχής)

2. **Ανάλυση Συστάδων (Συσταδοποίηση ή Ομαδοποίηση)**. Στην εργασία αυτή αναζητούνται συστάδες (ομάδες) από στενά συσχετιζόμενες παρατηρήσεις, ώστε αυτές

που ανήκουν στην ίδια συστάδα, να είναι περισσότερο παρεμφερείς μεταξύ τους σε σχέση με παρατηρήσεις που ανήκουν σε άλλες συστάδες.

3. **Ανίχνευση Ανωμαλιών**. Είναι η εργασία προσδιορισμού εκείνων των παρατηρήσεων, των οποίων τα χαρακτηριστικά διαφέρουν σημαντικά από τα αντίστοιχα χαρακτηριστικά των υπόλοιπων δεδομένων.

4. **Ανάλυση Συσχέτισης**. Η εργασία αυτή χρησιμοποιείται για να ανακαλυφθούν υποδείγματα, τα οποία περιγράφουν έντονα συσχετιζόμενα χαρακτηριστικά των δεδομένων. Τα υποδείγματα που ανακαλύπτονται, συνήθως αναπαρίστανται με τη μορφή κανόνων συνεπαγωγής (συσχέτισης).

1.3 Προστασία της ιδιωτικότητας κατά την εξόρυξη δεδομένων.

Η σημαντική πρόοδος στη συλλογή και αποθήκευση πληροφοριών, έχει οδηγήσει στην εύκολη και χωρίς ιδιαίτερο κόστος διατήρηση τεράστιων ποσοτήτων δεδομένων, τόσο από εταιρείες και επιχειρήσεις όσο και από δημόσιους οργανισμούς [09] .

Εκτός από τα πλεονεκτήματα χρησιμοποίησης των δεδομένων αυτών καθαυτών, η εξόρυξη δεδομένων μπορεί ν' αποκαλύψει ανεκτίμητη νέα γνώση στον κάτοχο της βάσης δεδομένων. Τα πρότυπα που εξάγονται μπορούν να παρέχουν πολύτιμη γνώση στους ιδιοκτήτες της βάσης δεδομένων και να είναι ανεκτίμητα σε σημαντικές διεργασίες όπως η λήψη αποφάσεων και ο στρατηγικός σχεδιασμός [07].

Πολλές φορές επίσης, οι εταιρείες είναι πρόθυμες να συνεργασθούν με άλλους οργανισμούς που ασχολούνται με παρόμοιες δραστηριότητες, με σκοπό την απόκτηση αμοιβαίων προνομίων στις εργασίες τους. Πολύτιμα πρότυπα γνώσης (νέα γνώση) μπορούν να παραχθούν και να μοιρασθούν ανάμεσα στους συνεταίρους (συνεργαζόμενους φορείς), μέσω της κοινής εξόρυξης δεδομένων στα δεδομένα τους. Ακόμα πιο σημαντικό είναι το γεγονός ότι δημόσιοι οργανισμοί και υπηρεσίες, συχνά είναι υποχρεωμένοι να μοιρασθούν ένα τμήμα των δεδομένων τους ή της γνώσης που έχουν συλλέξει, με άλλους οργανισμούς που έχουν ίδιο ή κοινό σκοπό ή να είναι υποχρεωμένοι να δημοσιοποιήσουν τα δεδομένα και τη γνώση τους ώστε να συμμορφώνονται και να είναι συνεπείς με συγκεκριμένους κανονισμούς ή τη νομοθεσία γενικότερα [06].

Σε πολλές περιπτώσεις όμως, είναι δυνατόν τα δεδομένα που έχουν συλλεχθεί ή η γνώση που έχει εξαχθεί (εξορυχθεί), να πρέπει να μοιρασθούν με άλλους ενδεχομένως αναξιόπιστους οργανισμούς ή εταιρείες. Η διανομή των δεδομένων ή/και της γνώσης, ενδέχεται να περιορίζεται από κάποιο βαθμό στην ιδιωτικότητα, που οφείλεται σε δύο κυρίως λόγους: i) εάν τα δεδομένα αφορούν φυσικά πρόσωπα, τότε η δημοσιοποίησή τους ή η κοινοποίηση σε τρίτους μπορεί να παραβιάζει τα δικαιώματα και την ιδιωτικότητα των ατόμων που καταγράφονται στα δεδομένα ή η εφαρμογή εργασιών εξόρυξης σ' αυτά, μπορεί να αποκαλύπτει ευαίσθητα προσωπικά δεδομένα ii) εάν τα δεδομένα αναφέρονται σε επιχειρηματικές πληροφορίες, τότε η αποκάλυψη αυτών των δεδομένων ή της γνώσης που μπορεί να εξαχθεί από αυτά, ενδέχεται ν' αποκαλύψει ευαίσθητα επαγγελματικά μυστικά, των οποίων η γνώση παρέχει σημαντικά πλεονεκτήματα στους ανταγωνιστές [06][09]. Από όλα τα παραπάνω, είναι προφανής η ανάγκη για την προστασία της ιδιωτικότητας κατά την εξόρυξη δεδομένων.

Η διατήρηση της ιδιωτικότητας στην εξόρυξη δεδομένων, είναι μία σχετικά νέα περιοχή έρευνας στο ευρύτερο πεδίο της εξόρυξης δεδομένων και μετράει περίπου μία δωδεκαετία ύπαρξης. Διερευνά τις *παρενέργειες* (*side effects)* της εξόρυξης δεδομένων, που προέρχονται από την διείσδυση στην ιδιωτική «περιοχή» των φυσικών προσώπων και των οργανισμών. Από την πρωτοπόρο εργασία των Agrawal & Srikant [01] και των Lindell & Pinkas [11] το 2000, έχουν προταθεί αρκετές προσεγγίσεις για την προστασία της ιδιωτικότητας στην εξόρυξη δεδομένων. Η πλειονότητα των προτεινόμενων προσεγγίσεων, μπορούν να ταξινομηθούν σε δύο κατηγορίες: i) προσεγγίσεις απόκρυψης δεδομένων και ii) προσεγγίσεις απόκρυψης γνώσης.

Η πρώτη κατηγορία (απόκρυψη δεδομένων), περιλαμβάνει όλες τις μεθοδολογίες που διερευνούν με ποιον τρόπο η ιδιωτικότητα των ακατέργαστων δεδομένων μπορεί να επιτευχθεί, πριν την εξόρυξη δεδομένων. Οι προσεγγίσεις αυτής της κατηγορίας, στοχεύουν στην απομάκρυνση ιδιωτικών ή εμπιστευτικών πληροφοριών από τα αρχικά δεδομένα πριν από την αποκάλυψή τους και ενεργούν εφαρμόζοντας τεχνικές όπως η δειγματοληψία, γενίκευση κλπ, με σκοπό να παράγουν ένα εξομαλυμένο μέρος της αρχικής βάσης δεδομένων. Ο βασικός τους στόχος είναι να παρέχουν στον κάτοχο της βάσης δεδομένων, την δυνατότητα να λάβει ακριβή αποτελέσματα εξόρυξης δεδομένων όταν δεν έχει τα πραγματικά δεδομένα [06].

Η δεύτερη κατηγορία (απόκρυψη γνώσης), περιλαμβάνει τις μεθοδολογίες που σκοπεύουν να προστατεύσουν τα ευαίσθητα αποτελέσματα που παρήχθησαν με την εφαρμογή των εργαλείων εξόρυξης δεδομένων, παρά τα δεδομένα αυτά καθαυτά. Οι προσεγγίσεις αυτές εστιάζουν σε τεχνικές διαστρέβλωσης (αλλαγής) και μπλοκαρίσματος των δεδομένων, που απαγορεύουν τη

διαρροή ευαίσθητων προτύπων από τα νέα τροποποιημένα δεδομένα, καθώς και σε τεχνικές μείωσης της αποτελεσματικότητας των κατηγοριοποιητών στις διεργασίες κατηγοριοποίησης, έτσι ώστε οι παραγόμενοι κατηγοριοποιητές να μην αποκαλύπτουν ευαίσθητη γνώση [06].

1.4 Απόκρυψη κανόνων συσχέτισης

Όπως αναφέρθηκε και παραπάνω, οι κανόνες συσχέτισης είναι υποδείγματα που περιγράφουν συσχετιζόμενα χαρακτηριστικά μεταξύ των δεδομένων.

Εξάγονται από μία βάση δεδομένων (κυρίως συναλλαγών), με βάση δύο συγκεκριμένες παραμέτρους (την *υποστήριξη* και την *εμπιστοσύνη*), που καθορίζει ο χρήστης και μετρούν την σπουδαιότητα και την αξιοπιστία τους.

Οι κανόνες συσχέτισης παρέχουν σημαντική γνώση αφού συνοψίζουν τα δεδομένα ενώ αποκαλύπτουν κρυμμένες συσχετίσεις ανάμεσα στα διάφορα στοιχεία (αντικείμενα) που υπάρχουν στα δεδομένα [21].

Ο όρος Απόκρυψη Κανόνων Συσχέτισης, αναφέρθηκε για πρώτη φορά το 1999 από τον M. Atallah et al. [03]. Οι συγγραφείς έθεσαν το πρόβλημα της τροποποίησης της αρχικής βάσης δεδομένων με τέτοιον τρόπο, ώστε κάποιοι συγκεκριμένοι κανόνες συσχέτισης που ορίζονται ως *ευαίσθητοι*, να μην μπορούν να αποκαλυφθούν χωρίς ωστόσο να επηρεάζονται σημαντικά τόσο τα δεδομένα όσο και οι μη ευαίσθητοι κανόνες. Πρότειναν κάποιες τεχνικές οι οποίες λύνουν το πρόβλημα ευριστικά. Την ίδια προσέγγιση (δηλαδή λύση με ευριστικό τρόπο) ακολούθησαν και οι περισσότερες τεχνικές που προτάθηκαν στη συνέχεια από άλλους ερευνητές. Αυτές οι τεχνικές αναζητούν την βέλτιστη λειτουργία σε κάθε βήμα του αλγορίθμου και συνεπώς δεν εγγυώνται την εύρεση της βέλτιστης λύσης για ολόκληρο το πρόβλημα.

Πρόσφατα προτάθηκαν και κάποιες άλλες τεχνικές, που παρουσιάζουν μεγαλύτερη πολυπλοκότητα και απαιτήσεις σε χρόνο επεξεργασίας και μνήμη, αλλά προσφέρουν μεγαλύτερη εγγύηση και ακρίβεια στην τελική λύση.

1.5 Σχετική Έρευνα

Από την πρώτη της εμφάνιση το 1999, η Απόκρυψη Κανόνων Συσχέτισης, έχει μελετηθεί εκτενώς από την επιστημονική κοινότητα, οδηγώντας σε ένα ευρύ πεδίο έρευνας στα χρόνια που ακολούθησαν. Οι τεχνικές που προτάθηκαν, ξεκινούν από απλές και μη απαιτητικές ευριστικές, οι οποίες επιλέγουν συγκεκριμένες συναλλαγές και αντικείμενα για να τροποποιήσουν, και καταλήγουν σε πιο σύνθετες και εκλεπτυσμένες λύσεις που αντιμετωπίζουν την διαδικασία απόκρυψης σαν ένα πρόβλημα αναζήτησης βέλτιστης λύσης, το οποίο προσπαθούν να επιλύσουν χρησιμοποιώντας συγκεκριμένες τεχνικές βελτιστοποίησης [06].

Πιο συγκεκριμένα, εκτός από τους Atallah et al. [03] που όπως προαναφέρθηκε ήταν οι πρώτοι που πρότειναν έναν συγκεκριμένο αλγόριθμο για την απόκρυψη ευαίσθητων κανόνων συσχέτισης, μπορούμε ν' αναφέρουμε τις εργασίες (και τις τεχνικές που προτείνονται εκεί), των Dasseni et al. [05] οι οποίοι γενίκευσαν το πρόβλημα της απόκρυψης με την έννοια ότι θεώρησαν από κοινού την απόκρυψη των ευαίσθητων συχνών στοιχειοσυνόλων και των ευαίσθητων κανόνων, των Verykios et al. [23] που επέκτειναν την προαναφερθείσα εργασία [05] βελτιώνοντας τους αλγόριθμους που προτάθηκαν εκεί, των Oliveira & Zaïane [16] που ήταν οι πρώτοι που παρουσίασαν προσεγγίσεις απόκρυψης πολλαπλών κανόνων, του Amiri [02] που πρότεινε τρεις αποτελεσματικούς ευριστικούς αλγόριθμους απόκρυψης πολλαπλών κανόνων, των Wu et al. [26] που πρότειναν μία πιο εκλεπτυσμένη μεθοδολογία που αφαιρεί την υπόθεση που παρουσιάζεται στο [05] αναφορικά με την μη προφανή (μη αιτιατή) σχέση ανάμεσα στα αντικείμενα των διαφόρων ευαίσθητων κανόνων. Οι Pontikakis et al. [17] πρότειναν δύο ευριστικές τεχνικές που βασίζονται στην *παραμόρφωση* (*distortion*) ώστε να αποκρύψουν επιλεγμένους ευαίσθητους κανόνες, οι Wang & Jafari [24][25] παρουσίασαν δύο αλγορίθμους τροποποίησης δεδομένων που στοχεύουν στην απόκρυψη ευαίσθητων κανόνων που μπορούν να προβλεφθούν (π.χ. κανόνες που περιλαμβάνουν τα ευαίσθητα αντικείμενα στο αριστερό τους μέρος), ενώ οι Lee et al. [10] έδωσαν μία προσέγγιση παραμόρφωσης που λειτουργεί κατασκευάζοντας αρχικά έναν εξομαλυμένο πίνακα από τα αρχικά δεδομένα και στη συνέχεια πολλαπλασιάζει την αρχική βάση με τον πίνακα αυτόν, ώστε να προκύψει η εξομαλυμένη βάση.

Οι Saygin et al. [19][20] ήταν οι πρώτοι που πρότειναν το *μπλοκάρισμα* (*blocking*) αντί της παραμόρφωσης, με τη χρησιμοποίηση αγνώστων τιμών (?) στις συναλλαγές αντί της εναλλαγής των 0 με 1 και αντίστροφα, ενώ οι Pontikakis et al [18] ισχυρίστηκαν ότι το βασικό μειονέκτημα της τεχνικής του μπλοκαρίσματος είναι ότι η αρχική βάση, εκτός των αγνώστων τιμών που εισήχθησαν, δεν έχει ουσιαστικά τροποποιηθεί, με αποτέλεσμα την εύκολη αποκάλυψη των

κρυμμένων κανόνων αν αναγνωρίσουμε αυτά τα στοιχειοσύνολα που περιέχουν ερωτηματικά (?) και οδηγούν σε κανόνες με εμπιστοσύνη που είναι πάνω από το κατώφλι minconf.

Οι Moustakides & Verykios [14][15] δίνουν δύο ευριστικούς αλγόριθμους που βασίζονται στο κριτήριο max-min. Και οι δύο αλγόριθμοι χρησιμοποιούν το αναθεωρημένο θετικό όριο των συχνών στοιχειοσυνόλων, για να διερευνήσουν την επίδραση που έχει η τροποποίηση του κάθε υποψήφιου θύματος ξεχωριστά.

Στην δική τους εργασία, οι Chen et al. [04], προτείνουν μία μεθοδολογία η οποία χρησιμοποιεί μία τεχνική ανάστροφης εξόρυξης πλέγματος στοιχειοσυνόλων για την αναδόμηση (ανακατασκευή) της αρχικής βάσης δεδομένων σε μία νέα, η οποία διατηρεί την ιδιωτικότητα κατά την εξόρυξη δεδομένων. Τέλος, ιδαίτερο ενδιαφέρον παρουσιάζει η εργασία του Guo [07] στην οποία παρουσιάζεται μία τεχνική που χρησιμοποιεί την ανάστροφη ανάπτυξη του FP-δένδρου για την αναδόμηση της βάσης δεδομένων. Η εργασία αυτή διαφοροποιείται από την προηγούμενη των Chen et al. [04] στο ότι στοχεύει στην απόκρυψη των ευαίσθητων κανόνων αντί για την απόκρυψη των ευαίσθητων στοιχειοσυνόλων.

Κεφάλαιο 2
Ορισμοί

Στο κεφάλαιο αυτό δίνουμε την ορολογία και το υπόβαθρο που είναι απαραίτητα για την κατανόηση εκ μέρους του αναγνώστη, της Απόκρυψης Κανόνων Συσχέτισης. Στην ενότητα 2.1 δίνουμε τους ορισμούς και τους τύπους που χρησιμοποιούνται στην Εξόρυξη των Κανόνων Συσχέτισης, ενώ στα επόμενα δίνουμε επακριβώς τους στόχους της Απόκρυψης Κανόνων Συσχέτισης και κατηγοριοποιούμε τις μεθολογίες που έχουν προταθεί σύμφωνα με διάφορα κριτήρια.

2.1 Θεμελιώδεις έννοιες – Ορισμοί

Πολλές εμπορικές επιχειρήσεις, συσσωρεύουν τεράστιες ποσότητες δεδομένων από τις καθημερινές τους λειτουργίες. Κλασικό παράδειγμα οι αλυσίδες super-markets που συγκεντρώνουν δεδομένα από τα ταμεία τους, τα οποία είναι κοινώς γνωστά, ως *συναλλαγές καλαθιού αγοράς (market basket transactions)*. Κάθε συναλλαγή αποτελείται από έναν μοναδικό κωδικό και ένα σύνολο *αντικειμένων* που αγόρασε ο πελάτης. Ένα σύνολο από τέτοια *αντικείμενα (items)*, ονομάζεται *στοιχειοσύνολο (itemset)*. Η ανάλυση συσχέτισης, υποδεικνύει μία ισχυρή σχέση μεταξύ διαφόρων αντικειμένων και εκφράζεται όπως έχει ήδη αναφερθεί, με τη βοήθεια *κανόνων συσχέτισης* της μορφής $I \Rightarrow J$, όπου I, J είναι *συχνά στοιχειοσύνολα*.

Στοιχειοσύνολο (itemset)

Εάν $I=\{i_1, i_2, \ldots, i_n\}$ το σύνολο όλων των *αντικειμένων (items)* που περιέχονται σε μία βάση δεδομένων καλαθιού αγοράς, και $T=\{t_1, t_2, \ldots, t_n\}$ όλων των συναλλαγών της βάσης. Όπως είναι προφανές, κάθε συναλλαγή t_i περιέχει ένα υποσύνολο αντικειμένων από το *I*, το οποίο στην ανάλυση συσχέτισης ονομάζεται *στοιχειοσύνολο*.

Στοιχειοσύνολο-k (itemset-k)

Είναι ένα *στοιχειοσύνολο* που αποτελείται από k *αντικείμενα (items)*.

Δηλαδή στοιχειοσύνολο-k : $\{ I_k \mid I_k = \{i_1, i_2, \ldots, i_k\}, k \leq n \}$.

Δοθέντος ενός *στοιχειοσυνόλου X*, λέμε ότι μία συναλλαγή t_i περιλαμβάνει το *X*, αν $X \subseteq t_i$.

Μέτρηση Υποστήριξης (support count)

Μία βασική ιδιότητα ενός στοιχειοσυνόλου *X*, είναι η *μέτρηση υποστήριξης*, η οποία αναφέρεται στο πλήθος των συναλλαγών που περιέχουν το συγκεκριμένο *στοιχειοσύνολο*.

$$\sigma(X) = |\{t_i \mid X \subseteq t_i, t_i \in T\}| \tag{2.1}$$

Συχνό στοιχειοσύνολο (Frequent Itemset)

Στοιχειοσύνολα των οποίων η *μέτρηση υποστήριξης* ξεπερνά μία ελάχιστη τιμή - κατώφλι *minsup*, ονομάζονται *συχνά στοιχειοσύνολα*.

Κανόνας συσχέτισης (Association Rule)

Κανόνας συσχέτισης είναι μία πρόταση συνεπαγωγής της μορφής $X \Rightarrow Y$, όπου $X \cap Y = \varnothing$ (τα *X* και *Y* είναι ξένα μεταξύ τους στοιχειοσύνολα).

Η ισχύς ενός κανόνα συσχέτισης μετράται με βάση δύο παραμέτρους:

1. *Υποστήριξη (support)*

$$s(X \Rightarrow Y) = \sigma(X \cup Y) / N \tag{2.2}$$

Η *υποστήριξη* καθορίζει πόσο συχνά εμφανίζεται ο κανόνας σε ένα σύνολο δεδομένων (συναλλαγών). Κανόνας χαμηλής υποστήριξης μπορεί να εμφανίζεται τυχαία ή μπορεί να μην παρουσιάζει ενδιαφέρον (αδιάφορος κανόνας).

2. *Εμπιστοσύνη (confidence)*

$$c(X \Rightarrow Y) = \sigma(X \cup Y) / \sigma(X) \quad (2.3)$$

Η *εμπιστοσύνη* καθορίζει πόσο συχνά τα αντικείμενα του στοιχειοσυνόλου *Y*, εμφανίζονται σε συναλλαγές που περιέχουν το στοιχειοσύνολο *X* και μετράει την αξιοπιστία του κανόνα.

Η εξόρυξη των κανόνων συσχέτισης είναι μία συχνή και συνήθης διεργασία στην εξόρυξη δεδομένων. Είναι η διαδικασία ανακάλυψης κανόνων συσχέτισης, των οποίων η υποστήριξη και η εμπιστοσύνη ξεπερνούν δοθείσες προκαθορισμένες τιμές (κατώφλια), *minsup* και *minconf* αντίστοιχα.

Η εξόρυξη κανόνων συσχέτισης διαχωρίζεται σε δύο υποδιεργασίες:

1. Παραγωγή συχνών στοιχειοσυνόλων

2. Παραγωγή κανόνων από τα συχνά στοιχειοσύνολα που προέκυψαν στο προηγούμενο βήμα. (Από ένα στοιχειοσύνολο-k, είναι δυνατόν να παραχθούν έως $2^k - 2$ κανόνες).

2.2 Στόχοι της Απόκρυψης των Κανόνων Συσχέτισης

Οι μεθοδολογίες και τεχνικές που εφαρμόζονται στην Απόκρυψη Κανόνων Συσχέτισης, στοχεύουν στο να τροποποιήσουν την αρχική βάση δεδομένων, με τέτοιον τρόπο ώστε να επιτυγχάνεται τουλάχιστον ένας από τους παρακάτω στόχους:

1. Κανένας από τους ευαίσθητους κανόνες δεν πρέπει να εξάγεται από την τροποποιημένη βάση δεδομένων, όταν σ' αυτήν πραγματοποιείται εξόρυξη κανόνων συσχέτισης κάτω από τις ίδιες ή ψηλότερες τιμές υποστήριξης και εμπιστοσύνης *minsup* και *minconf*.

2. Όλοι οι μη ευαίσθητοι κανόνες που εξάγονται από την εξόρυξη κανόνων στην αρχική βάση δεδομένων κάτω από συγκεκριμένες τιμές υποστήριξης και εμπιστοσύνης, να μπορούν να

εξάγονται επιτυχώς και από την τροποποιημένη βάση και κάτω από τις ίδιες τιμές υποστήριξης και εμπιστοσύνης.

3. Κανένας κανόνας που δεν είχε εξαχθεί από την αρχική βάση με δεδομένες τιμές κατωφλίων *minsup* και *minconf*, να μην εξάγεται από την τροποποιημένη βάση με τις ίδιες ή ψηλότερες τιμές κατωφλίων υποστήριξης και εμπιστοσύνης.

Ο πρώτος στόχος προφανώς αντανακλά τη βασική διεργασία της Απόκρυψης Κανόνων Συσχέτισης, που είναι ακριβώς η μη εξαγωγή κανόνων που δεν επιθυμούμε να εξαχθούν.

Ο δεύτερος στόχος σχετίζεται με τους μη ευαίσθητους κανόνες συσχέτισης που μπορούν να χαθούν στη νέα βάση. Το επιθυμητό είναι να μην χάνεται κανένας από τους μη ευαίσθητους κανόνες.

Ο τρίτος στόχος σχετίζεται με μη ευαίσθητους κανόνες που μπορούν να εξαχθούν από τη νέα βάση, ενώ δεν είχαν εξαχθεί στην αρχική. Τέτοιοι κανόνες ονομάζονται *ghost rules*. (*κανόνες φαντάσματα*). Το επιθυμητό είναι να μην παράγεται κανένας *ghost rule*.

Τόσο η απώλεια μη ευαίσθητων κανόνων συσχέτισης στη νέα βάση δεδομένων, όσο και η εμφάνιση ορισμένων κανόνων που δεν υπήρχαν στην αρχική, ονομάζονται *παρενέργειες (side effects)* και στόχος είναι να ελαχιστοποιούνται όσο το δυνατόν περισσότερο.

Με βάση τους τρεις προαναφερόμενους στόχους, ένας αλγόριθμος απόκρυψης κανόνων συσχέτισης πρέπει να τροποποιεί την βάση δεδομένων με τέτοιον τρόπο, ώστε να επηρεάζει όσο το δυνατόν λιγότερο την αρχική βάση δεδομένων, να διατηρεί τα γενικά πρότυπα και τάσεις και να επιτυγχάνει την απόκρυψη όλων των ευαίσθητων κανόνων. Μία λύση που επιτυγχάνει τον πρώτο βασικό στόχο της απόκρυψης των ευαίσθητων κανόνων, λέγεται *εφικτή (feasible).* Μία λύση που επιτυγχάνει και τους τρεις στόχους που τέθηκαν παραπάνω, λέγεται *ακριβής (exact)*. Μία ακριβής λύση που προκαλεί την ελάχιστη δυνατή μεταβολή στην αρχική βάση δεδομένων, λέγεται *ιδανική* ή *βέλτιστη* (*ideal* or *optimal*). Τέλος, οι λύσεις που είναι *εφικτές* αλλά όχι *ακριβείς* (δηλαδή επιτυγχάνουν τον πρώτο στόχο, αλλά παρουσιάζουν *παρενέργειες (side effects)*) λέγονται *προσεγγιστικές (approximate)*.

Σαν τελικό σχόλιο πρέπει να παρατηρήσουμε ότι οι μεθοδολογίες της απόκρυψης κανόνων συσχέτισης, συνήθως διαφέρουν στον τρόπο με τον οποίο ιεραρχούν τους τρεις

προαναφερόμενους στόχους και ειδικότερα τον δεύτερο και τρίτο. Αναφορικά με τον πρώτο στόχο, είναι ενδιαφέρον να παρατηρήσουμε ότι για κάθε βάση δεδομένων και για κάθε σύνολο ευαίσθητων κανόνων, υπάρχει μία *εφικτή* λύση, δηλαδή λύση που αποκρύπτει αποτελεσματικά όλους τους ευαίσθητους κανόνες της βάσης δεδομένων. Αυτό σημαίνει ότι ο πρώτοε στόχος είναι δυνατόν να επιτευχθεί πάντα και ανεξάρτητα από τις ιδιότητες της βάσης δεδομένων ή τις ιδιαιτερότητες του προβλήματος της απόκρυψης. Ο πιο κοινός τρόπος για να επιτύχουμε μία λύση σε μία βάση δεδομένων και να αποκρύψουμε όλους τους ευαίσθητους κανόνες, είναι αυτός σύμφωνα με τον οποίο, για κάθε ευαίσθητο κανόνα επιλέγουμε ένα αντικείμενο (*item*) από το στοιχειοσύνολο που παράγει αυτόν τον ευαίσθητο κανόνα και το διαγράφουμε από όλες τις συναλλαγές της βάσης.

2.3 Ορισμός του προβλήματος

Η απόκρυψη των κανόνων συσχέτισης έχει διερευνηθεί σε βάθος, με βάση δύο διαφορετικές προσεγγίσεις (κατευθύνσεις).

1. Οι μεθοδολογίες που ακολουθούν την πρώτη προσέγγιση, στοχεύουν στην απόκρυψη συγκεκριμένων *κανόνων συσχέτισης*, ανάμεσα σε αυτούς που εξάγονται από την αρχική βάση δεδομένων.

2. Από την άλλη μεριά, οι τεχνικές που υιοθετούν την δεύτερη προσέγγιση, στοχεύουν στην απόκρυψη συγκεκριμένων *συχνών στοιχειοσυνόλων* από όλα αυτά που βρέθηκαν όταν εφαρμόσθηκε η εξόρυξη κανόνων συσχέτισης στην αρχική βάση δεδομένων (και ειδικότερα το πρώτο βήμα της εξόρυξης όπως αναφέρθηκε στην ενότητα 2.1).

Οι δύο προσεγγίσεις είναι παρόμοιες. Πράγματι, η απόκρυψη των ευαίσθητων κανόνων συσχέτισης μέσω της απόκρυψης των *συχνών στοιχειοσυνόλων* από τα οποία προκύπτουν, είναι μία συνήθης πρακτική που υιοθετείται από την πλειονότητα των ερευνητών. Αν εξασφαλίσουμε ότι τα *συχνά στοιχειοσύνολα* που οδηγούν στην παραγωγή ευαίσθητων κανόνων συσχέτισης, θα είναι μη συχνά στην τροποποιημένη βάση, τότε είναι προφανές ότι οι ευαίσθητοι κανόνες δεν θα μπορούν να εξαχθούν από τη νέα βάση και συνεπώς η ευαίσθητη γνώση θα παραμείνει κρυφή και προστατευμένη.

2.3.1 Απόκρυψη ευαίσθητων στοιχειοσυνόλων

Υποθέτουμε ότι μας δίνεται μία βάση δεδομένων D_0, που αποτελείται από N συναλλαγές. Μάς δίνεται επίσης ένα κατώφλι υποστήριξης *minsup*. Μετά την εφαρμογή εξόρυξης συχνών στοιχειοσυνόλων στην D_0 και κάτω από την *minsup*, παράγουμε ένα σύνολο συχνών στοιχειοσυνόλων, έστω F_{D_0}. Έστω ένα υποσύνολο S των συχνών στοιχειοσυνόλων ($S \subseteq F_{D_0}$) που αποτελείται από στοιχειοσύνολα που θεωρούνται ευαίσθητα.

Δοθέντος του συνόλου των ευαίσθητων στοιχειοσυνόλων S, ο στόχος των μεθοδολογιών απόκρυψης ευαίσθητων στοιχειοσυνόλων, είναι να κατασκευάσουν από την αρχική βάση δεδομένων D_0 μία νέα βάση D, από την οποία δεν θα είναι δυνατή η εξόρυξη των ευαίσθητων στοιχειοσυνόλων, ενώ παράλληλα θα επιτυγχάνεται η εξόρυξη όσο το δυνατόν περισσότερων από τα μη ευαίσθητα στοιχειοσύνολα (δηλαδή τα στοιχειοσύνολα που ανήκουν στο $F_{D_0} - S$).

Για να κρύψει ένα ευαίσθητο στοιχειοσύνολο ο αλγόριθμος, πρέπει να τροποποιήσει την αρχική βάση D_0 με τέτοιον τρόπο, ώστε όταν γίνεται η παραγωγή συχνών στοιχειοσυνόλων στη νέα βάση D με βάση την ίδια ή ψηλότερη τιμή υποστήριξης *minsup*, τα παραγόμενα *συχνά στοιχειοσύνολα* να είναι όλα μη ευαίσθητα.

2.3.2 Απόκρυψη ευαίσθητων κανόνων συσχέτισης

Υποθέτουμε ότι μας δίνεται μία βάση δεδομένων D_0 που αποτελείται από N συναλλαγές και τιμές κατωφλίων για την υποστήριξη και εμπιστοσύνη, *minsup* και *minconf*, αντίστοιχα Μετά την εφαρμογή εξόρυξης κανόνων συσχέτισης κάτω από τις τιμές των *minsup* και *minconf*, παράγουμε ένα σύνολο κανόνων συσχέτισης έστω R, ανάμεσα στους οποίους ένα υποσύνολο του R, R_s ($R_s \subseteq R$), περιλαμβάνει τους κανόνες που θεωρούνται ευαίσθητοι.

Δοθέντος του συνόλου των ευαίσθητων κανόνων R_s, ο στόχος των μεθοδολογιών που εφαρμόζονται, είναι να κατασκευάσουν από την αρχική βάση D_0 μία νέα βάση D, από την οποία δεν θα είναι δυνατή η εξόρυξη όλων των ευαίσθητων κανόνων του *Rs*, ενώ θα επιτυγχάνεται η εξόρυξη όσο το δυνατόν περισσότερων από τους μη ευαίσθητους κανόνες (αυτούς που ανήκουν στο *R-Rs*).

Επομένως, ο αλγόριθμος θα πρέπει να διαμορφώσει την αρχική βάση δεδομένων D_0 με τέτοιον τρόπο, ώστε όταν γίνεται εξόρυξη κανόνων συσχέτισης στη νέα βάση D με βάση τις τιμές των

minsup και *minconf* (ή ψηλότερες), οι κανόνες συσχέτισης που προκύπτουν να είναι όλοι μη ευαίσθητοι.

2.4 Μεθοδολογίες Επίλυσης- Κατηγορίες

Από την πρώτη παρουσίαση της απόκρυψης των κανόνων συσχέτισης ως επιστημονικό πεδίο έρευνας και μέχρι σήμερα, έχουν προταθεί πολλοί αλγόριθμοι τόσο για την απόκρυψη των συχνών στοιχειοσυνόλων όσο και για την απόκρυψη των κανόνων συσχέτισης. Οι αλγόριθμοι αυτοί μπορούν να ταξινομηθούν με βάση τις εξής παραμέτρους:

2.4.1 Στρατηγική Απόκρυψης

Με βάση αυτήν την παράμετρο οι αλγόριθμοι χωρίζονται σε αυτούς που χρησιμοποιούν την υποστήριξη του κανόνα και σε αυτούς που χρησιμοποιούν την εμπιστοσύνη του κανόνα, κατά τη διαδικασία απόκρυψης.

1. Αλγόριθμοι που χρησιμοποιούν την υποστήριξη

   Αυτοί αποκρύπτουν έναν ευαίσθητο κανόνα συσχέτισης, μειώνοντας την υποστήριξη είτε του αριστερού μέρους του κανόνα είτε του δεξιού μέρους του κανόνα ή ελαττώνοντας την υποστήριξη του στοιχειοσυνόλου από το οποίο παράγεται ο ευαίσθητος κανόνας, μέχρι το σημείο που η υποστήριξη του κανόνα να πέσει κάτω από το κατώφλι υποστήριξης *minsup*.

2. Αλγόριθμοι που χρησιμοποιούν την εμπιστοσύνη

   Αυτοί οι αλγόριθμοι μειώνουν την εμπιστοσύνη του ευαίσθητου κανόνα, είτε αυξάνοντας την μέτρηση υποστήριξης του αριστερού μέρους του κανόνα, είτε μειώνοντας την μέτρηση υποστήριξης του δεξιού μέρους.

2.4.2 Στρατηγική Τροποποίησης Δεδομένων

Με βάση το είδος της τροποποίησης των δεδομένων, οι αλγόριθμοι χωρίζονται σε αυτούς που χρησιμοποιούν την *παραμόρφωση* των δεδομένων (*distortion*) και σε αυτούς που χρησιμοποιούν το *μπλοκάρισμα* των δεδομένων (*blocking*).

1. *Παραμόρφωση* (*distortion*)

   Είναι η διαδικασία αντικατάστασης των 0 με 1 και των 1 με 0, σε συγκεκριμένα αντικείμενα (*items*) επιλεγμένων συναλλαγών, σε μία βάση δυαδικών δεδομένων (βάση όπου τα χαρακτηριστικά είναι δυαδικά, δηλαδή παίρνουν την τιμή 0 ή 1).

2. *Μπλοκάρισμα* (*blocking*)

   Είναι η διαδικασία αντικατάστασης, των αρχικών τιμών (0 ή 1) με ερωτηματικό (?), σε συγκεκριμένα αντικείμενα σε συγκεκριμένες συναλλαγές.

2.4.3 Αριθμός Κανόνων που Αποκρύπτονται σε Κάθε Επανάληψη

Εδώ οι αλγόριθμοι διακρίνονται σε δύο κατηγορίες ανάλογα με το αν σε κάθε επανάληψη του αλγορίθμου αποκρύπτεται ένας μόνο κανόνας συσχέτισης ή ένα σύνολο κανόνων.

1. Απόκρυψη απλού κανόνα

   Αυτοί οι αλγόριθμοι εξετάζουν έναν κανόνα τη φορά και τροποποιούν τα δεδομένα κατάλληλα ώστε να κρύψουν αυτόν τον κανόνα.

2. Απόκρυψη πολλών κανόνων

   Αυτοί οι αλγόριθμοι επιλέγουν αντικείμενα των οποίων η τροποποίηση επηρεάζει περισσότερους από έναν κανόνες συσχέτισης.

2.4.4 Φύση του αλγορίθμου

Οι αλγόριθμοι απόκρυψης κανόνων συσχέτισης, μπορεί να είναι ευριστικοί ή ακριβείς.

1. Ευριστικοί

   Προσπαθούν να πετύχουν στη διαδικασία απόκρυψης, τη βέλτιστη λύση συγκεκριμένων στόχων από τους τρεις που αναφέρθηκαν στην ενότητα 2.2, ενώ δεν εγγυώνται τη βέλτιστη λύση του συνολικού προβλήματος (συνολικής απόκρυψης).

2. Ακριβείς

   Εστιάζουν στο να διαμορφώσουν το πρόβλημα της απόκρυψης κανόνων συσχέτισης με τέτοιον τρόπο, ώστε να βρεθεί μία λύση που να ικανοποιεί και τους τρεις στόχους της απόκρυψης.

Κεφάλαιο 3
Τεχνικές Αναδόμησης Βάσης Δεδομένων

Στο κεφάλαιο αυτό παρουσιάζουμε και αναλύουμε δύο τεχνικές Αναδόμησης Βάσης Δεδομένων που έχουν προταθεί από άλλους ερευνητές. Στην ενότητα 3.1 παρουσιάζουμε τον πρώτο αλγόριθμο, αναλύουμε τα προβλήματα που εμφανίζει, παρουσιάζουμε τρεις δικές μας τροποποιήσεις για να ξεπεράσουμε τα προβλήματα αυτά και δίνουμε κάποια παραδείγματα. Στην ενότητα 3.2, κάνουμε τα ίδια για τον δεύτερο αλγόριθμο.

3.1 Πρώτος Βασικός Αλγόριθμος *(reconstruction_by_cardinality)*

3.1.1 Θεωρητική Ανάλυση

Στην εργασία που παρουσίασαν το 2004 οι Xia Chen, Maria Orlowska και Xue Li [04], δίνουν ένα πλαίσιο και προτείνουν μία τεχνική που βασίζονται στην Αναδόμηση της Βάσης Δεδομένων, με σκοπό την απόκρυψη των κανόνων συσχέτισης και τελικά την δημοσιοποίηση των δεδομένων με διατήρηση της ιδιωτικότητας.

Πριν συνεχίσουμε στην παρουσίαση του αλγορίθμου, θα σημειώσουμε ότι επειδή αυτός επιτυγχάνει την Αναδόμηση της Βάσης Δεδομένων με χρήση της *πληθικότητας (cardinality)* (βλ. παρακάτω), τον ονομάζουμε *reconstruction_by_cardinality*. Σε αυτήν τη διατριβή λοιπόν, όταν θα αναφέρουμε τον αλγόριθμο *reconstruction_by_cardinality,* θα εννοούμε τον αλγόριθμο των Chen, Orlowska και Li.

Η μεθοδολογία που προτείνουν οι συγγραφείς, χρησιμοποιεί μία τεχνική ανάστροφης εξόρυξης πλέγματος στοιχειοσυνόλων, και βασίστηκε στο γενικότερο πρόβλημα της *ανάστροφης εξόρυξης συχνών στοιχειοσυνόλων (inverse frequent set mining)* [13].

Με τον όρο *ανάστροφη εξόρυξη* που αποτελεί ένα νέο αναπτυσσόμενο επιστημονικό πεδίο στο ευρύτερο πεδίο της περιοχής της απόκρυψης κανόνων συσχέτισης, εννοείται η δημιουργία μίας βάσης δεδομένων από ένα δοθέν πλέγμα συχνών στοιχειοσυνόλων και των μετρήσεων υποστήριξης αυτών, τέτοιας ώστε να συμφωνεί απόλυτα με τις μετρήσεις υποστήριξης των στοιχειοσυνόλων που δόθηκαν, ενώ οι μετρήσεις υποστήριξης των υπόλοιπων στοιχειοσυνόλων να είναι μικρότερες από το προσδιορισμένο κατώφλι [13].

Αποδεικνύεται ότι υπάρχει μία 1-1 αντιστοίχιση μεταξύ ενός συνόλου συναλλαγών και του πλέγματος στοιχειοσυνόλων με δοθείσες μετρήσεις υποστήριξης. Αν λοιπόν, από μία πραγματική βάση δεδομένων μας δώσουν ένα πλέγμα συχνών στοιχεισυνόλων με τις μετρήσεις υποστήριξης αυτών, θα υπάρχει μόνο ένα συμβατό σύνολο δεδομένων (συναλλαγών). Το προφανές ερώτημα όμως που ανακύπτει, είναι: «Πώς μπορούμε να μετατρέψουμε αυτά τα στοιχειοσύνολα με τις αντίστοιχες μετρήσεις υποστήριξής τους, σε ένα σύνολο συναλλαγών;»

Οι Chen, Orlowska και Li, ορίζουν για κάθε στοιχειοσύνολο *X*, ένα νέο μέγεθος *f(x)* που το ονομάζουν *cardinality* (ο αντίστοιχος όρος στα Ελληνικά είναι *πληθικότητα*) και ορίζει τον αριθμό συναλλαγών μέσα σε ολόκληρο το σύνολο των συναλλαγών, οι οποίες περιλαμβάνουν ακριβώς το στοιχειοσύνολο *X* και μόνο αυτό και όχι κάποιο υπερσύνολο αυτού. Με άλλα λόγια η *πληθικότητα* ενός στοιχειοσυνόλου *X* μας δείχνει πόσες φορές εμφανίζεται στις συναλλαγές το στοιχειοσύνολο *X* μόνο του και όχι σαν μέρος κάποιου ευρύτερου στοιχειοσυνόλου (υπερσυνόλου του).

Έστω λοιπόν ότι έχουμε ένα σύνολο συναλλαγών $T = \{t_1,t_2,\ldots,t_n\}$ και ένα σύνολο αντικειμένων $I = \{i_1,i_2,\ldots i_n\}$ τα οποία μπορεί να συμμετέχουν σε κάθε συναλλαγή, δηλαδή κάθε συναλλαγή

αποτελείται από ένα υποσύνολο του *I*. Το *P(T)* υποδηλώνει το δυναμοσύνολο των συναλλαγών *T* και το *P(I)* υποδηλώνει το δυναμοσύνολο των αντικειμένων του *I*.

Η πληθικότητα ορίζεται σαν:

$$f(X) = |T(X)| = Support\_count(X) - \sum_{X \subset I' \in P(I)} f(I') \quad (3.1)$$

Όπως φαίνεται από τον τύπο (3.1) και από όσα προαναφέρθηκαν, η πληθικότητα ενός στοιχειοσυνόλου Χ, προκύπτει αν από τη μέτρηση υποστήριξης του στοιχειοσυνόλου Χ αφαιρέσουμε τις πληθικότητες όλων των υπερσυνόλων του Χ, που εμφανίζονται στο σύνολο των συναλλαγών.

Για παράδειγμα έστω ότι έχουμε το παρακάτω σύνολο συναλλαγών: D1 = {ABD, BD, BC, ABD, AC, BC, ABCD, ABC, ABCD, ABC}. Όπως μπορούμε εύκολα να διαπιστώσουμε, οι μετρήσεις υποστήριξης για κάθε στοιχειοσύνολο του συνόλου συναλλαγών D1, είναι αυτές που δίνονται στον ακόλουθο Πίνακα 3.1:

Πίνακας 3.1 Μετρήσεις Υποστήριξης συνόλου συναλλαγών D1

Στοιχειο σύνολο	Μέτρηση Υποστήριξης	Στοιχειο σύνολο	Μέτρηση Υποστήριξης
ABCD	2	BC	6
ABC	4	BD	5
ABD	4	CD	2
ACD	2	A	7
BCD	2	B	9
AB	6	C	7
AC	5	D	5
AD	4		

Υπολογίζοντας λοιπόν την πληθικότητα για κάθε στοιχειοσύνολο, έχουμε:

$$f(ABCD) = Support\_count(ABCD) - f(\varnothing) = 2$$

$$f(ABC) = Support\_count(ABC) - f(ABCD) = 4 - 2 = 2$$

$$f(ABD) = Support\_count(ABD) - f(ABCD) = 4 - 2 = 2$$

$$f(ACD) = Support\_count(ACD) - f(ABCD) = 2 - 2 = 0$$

$$f(BCD) = Support\_count(BCD) - f(ABCD) = 2 - 2 = 0$$

$$f(AB) = Support\_count(AB) - (f(ABC) + f(ABD) + f(ABCD)) = 6 - (2 + 2 + 2) = 0$$

κοκ

$$f(AC) = Support\_count(AC) - (f(ABC) + f(ACD) + f(ABCD)) = 5 - (2 + 0 + 2) = 1$$

$$f(AD) = Support\_count(AD) - (f(ABD) + f(ACD) + f(ABCD)) = 4 - (2 + 0 + 2) = 0$$

$$f(A) = Support\_count(A) - (f(AB) + f(AC) + f(AD) + f(ABC) + f(ABD) + f(ABCD)) =$$
$$= 7 - (0 + 1 + 0 + 2 + 2 + 2) = 0$$

Από τους υπολογισμούς στο παραπάνω παράδειγμα, γίνεται εμφανές ότι στην αρχική βάση, υπάρχουν δύο συναλλαγές που περιέχουν το σύνολο ABCD, δύο συναλλαγές που περιέχουν το σύνολο ABC, δύο συναλλαγές που περιλαμβάνουν το ABD και καμία συναλλαγή που να περιλαμβάνει το σύνολο AB ή το σύνολο A. Επομένως, χρησιμοποιώντας το μέγεθος της πληθικότητας όπως ορίσθηκε πιο πάνω, μπορούμε να προσδιορίσουμε επακριβώς τις συναλλαγές μέσα σε μία βάση δεδομένων.

Υποθέτοντας περαιτέρω ότι $f(X) = n$ (όπου n ένας ακέραιος), μπορεί ν' αποδειχθεί ότι:

- Εάν $n > 0$, τότε υπάρχουν μέσα στο σύνολο συναλλαγών, n συναλλαγές που περιλαμβάνουν ακριβώς το στοιχειοσύνολο *X*.

- Εάν $n = 0$, τότε δεν υπάρχει καμία συναλλαγή που να περιλαμβάνει ακριβώς το στοιχειοσύνολο *X*. (Σ' αυτήν την περίπτωση το X είναι υποσύνολο μέσα σε κάποιες συναλλαγές).

- Εάν $n < 0$, τότε καταδεικνύεται ότι οι δοθείσες μετρήσεις υποστήριξης του πλέγματος των στοιχειοσυνόλων, δεν είναι δυνατόν να αποτελούν έγκυρες εξόδους μιας εργασίας εξόρυξης συχνών στοιχειοσυνόλων από μία πραγματική βάση, και επομένως δεν μπορούμε να βρούμε ένα σύνολο συναλλαγών που να ανταποκρίνεται σε αυτό το πλέγμα (το πρόβλημα αυτό συζητείται εκτενέστερα παρακάτω). Συνεπώς απαιτείται κάποιου είδους αντιστάθμιση, για την αναδόμηση της βάσης δεδομένων.

Τελικά λοιπόν η σχέση (3.1) που ορίζει το μέγεθος της πληθικότητας, αντανακλά τη *σχέση συνέπειας* μεταξύ των μετρήσεων υποστήριξης διαφόρων στοιχειοσυνόλων.

3.1.2 Παρουσίαση Αλγορίθμου

Στη συνέχεια δίνουμε τον αλγόριθμο Αναδόμησης Βάσης Δεδομένων, όπως παρουσιάζεται στο [04].

Ο αλγόριθμος ξεκινά με το στοιχειοσύνολο μεγαλύτερης τάξης (δηλαδή αυτό που αποτελείται από τα περισσότερα αντικείμενα) και οι υπολογισμοί συνεχίζουν να εκτελούνται βαθμιαία σε στοιχειοσύνολα μικρότερης τάξης. Υποτίθεται ακόμα ότι τα στοιχειοσύνολα μεγαλύτερης τάξης είναι στην κορυφή του πλέγματος και τα στοιχειοσύνολα μικρότερης τάξης στη βάση του πλέγματος.

Ο αλγόριθμος αρχικά βρίσκει όλα τα μέγιστα μη κενά στοιχειοσύνολα τα οποία βρίσκονται στο k-επίπεδο (στοιχειοσύνολα-k). Από τη στιγμή λοιπόν που όλα αυτά τα στοιχειοσύνολα-k δεν έχουν υπερσύνολα, η πληθικότητά τους θα είναι ίση με την *μέτρηση υποστήριξής* τους (*support_count*).

Αφού η πληθικότητα των στοιχειοσυνόλων-k έχει προσδιοριστεί, γράφουμε αυτά τα στοιχειοσύνολα-k, τόσες φορές όσο είναι η πληθικότητά τους και τα προσαρτούμε στην νέα αναδομημένη βάση δεδομένων, με αντίστοιχους κωδικούς συναλλαγών. Στη συνέχεια, για κάθε ένα από τα στοιχειοσύνολα-k, ο αλγόριθμος βρίσκει όλα τα μη κενά υποσύνολά του και αφαιρεί τις δικές τους μετρήσεις υποστήριξης, με σκοπό να βρει την πληθικότητα αυτών των υποσυνόλων.

Στη συνέχεια ο αλγόριθμος προχωράει με το (k-1) επίπεδο, στη συνέχεια με το (k-2) επίπεδο, κοκ μέχρι να φτάσει στα στοιχειοσύνολα-1 στη βάση του πλέγματος.

Αλγόριθμος 3.1 *– Algorithm of Chen, Orlowska & Li*

```
1:  D ← ∅;
2:  Q ← ∅;
3:  S ← ∅;
4:  k ← the length of the largest non-empty itemsets
5:  Repeat
6:      Q ← { I_i | |I_i| = k && support_count (I_i) > 0 }
7:      While non empty (Q)
8:          { p ← remove (Q)
9:              duplicate p for support_count (p) times and add them into D
10:         S ← { f_i | f_i ⊂ p } ;  /* find all subsets of p */
11:         for each q ∈ S do
12:             { support_count (q) ← support_count (q) – support_count (p)
```

```
13:                    if support_count (q) < 0
14:                        fail (0) }
15:          k = k-1
16:    until k=0
```

ΠΑΡΑΔΕΙΓΜΑ 3.1

Έστω το σύνολο συναλλαγών D2, που δίνεται στον Πίνακα 3.2. Οι αντίστοιχες τιμές μέτρησης υποστήριξης για τα στοιχειοσύνολα του D2, φαίνονται στον Πίνακα 3.3:

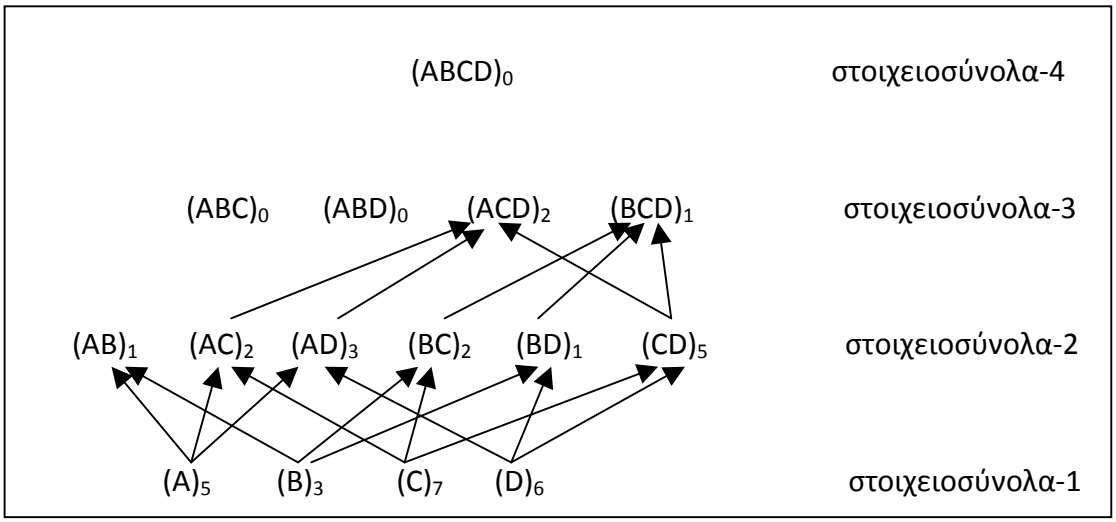
Πίνακας 3.2: Σύνολο Συναλλαγών D2

Συναλλαγή	Αντικείμενα
T1	ACD
T2	C
T3	ACD
T4	AB
T5	CD
T6	CD
T7	BC
T8	AD
T9	BCD
T10	A

Πίνακας 3.3: Μέτρηση Υποστήριξης Στοιχειοσυνόλων Συνόλου Συναλλαγών D2

Στοιχειο-σύνολο	Μέτρηση Υποστήριξης	
ABCD	0	
ABC	0	
ABD	0	
ACD	2	
BCD	1	
AB	1	
AC	2	
AD	3	
BC	2	
BD	1	
CD	5	
A	5	
B	3	
C	7	
D	6	

Το πλέγμα στοιχειοσυνόλων για το παραπάνω σύνολο συναλλαγών με τις αντίστοιχες τιμές μέτρησης υποστήριξης, δίνεται παρακάτω στο Σχήμα 3.1:



Σχήμα 3.1: Πλέγμα στοιχειοσυνόλων συνόλου συναλλαγών D2

Στο παραπάνω σύνολο συναλλαγών D2 και το αντίστοιχο πλέγμα στοιχειοσυνόλων, ο αλγόριθμος αρχικά προσπελαύνει τους κόμβους που βρίσκονται σε υψηλότερο επίπεδο στο πλέγμα. Για το πρώτο μη κενό στοιχειοσύνολο που βρίσκει (το ACD) υπολογίζει το μήκος του (που είναι 3) και το θέτει στη μεταβλητή k. Παράγει δύο σύνολα ACD (όσα και η μέτρηση υποστήριξης του ACD), τα οποία και προσθέτει στη νέα βάση δεδομένων D. Στη συνέχεια βρίσκει όλα τα υποσύνολα του ACD και τροποποιεί την μέτρηση υποστήριξης αντίστοιχα, για το καθένα από αυτά.

Ακολούθως, συνεχίζει την ίδια διαδικασία με το επόμενο μη κενό στοιχειοσύνολο-3 (το BCD) και όταν δεν βρίσκει άλλο στοιχειοσύνολο-3, μειώνει το k σε 2 και συνεχίζει την ίδια διαδικασία. Μόλις το k πάρει την τιμή 0, τερματίζεται η επανάληψη και επιστρέφεται η αναδομημένη βάση D.

Ο παρακάτω Πίνακας 3.4, δείχνει πώς μεταβάλλεται η μέτρηση υποστήριξης του κάθε υποσυνόλου, σε κάθε επανάληψη του αλγορίθμου.

Πίνακας 3.4: Παρουσίαση του Αλγορίθμου 3.1 για το πλέγμα στοιχειοσυνόλων του Σχήματος 3.1

Στοιχειοσύνολο	Αρχική Μέτρηση Υποστήριξης	ACD	BCD	AB	AC	AD	BC	BD	CD	A	B	C	D
ACD	**2**	0											
BCD	1	**1**	0										
AB	1	1	**1**	0									
AC	2	0	0	**0**	0								
AD	3	1	1	1	**1**	0							
BC	2	2	1	1	1	**1**	0						
BD	1	1	0	0	0	0	**0**	0					
CD	5	3	2	2	2	2	2	**2**	0				
A	5	3	3	2	2	1	1	1	**1**	0			
B	3	3	2	1	1	1	0	0	0	**0**	0		
C	7	5	4	4	4	4	3	3	1	1	**1**	0	
D	6	4	3	3	3	2	2	2	0	0	0	**0**	0

Τα σύνολα που προσθέτει ο αλγόριθμος στη βάση δεδομένων D στο τέλος κάθε επανάληψης, φαίνονται με έντονα γράμματα στον παραπάνω πίνακα και είναι δύο ACD, ένα BCD, ένα AB κοκ. Η νέα αναδομημένη βάση δεδομένων D, που κατασκευάζει και επιστρέφει ο αλγόριθμος είναι: D = {ACD, ACD, BCD, AB, AD, BC, CD, CD, A, C}

Προφανώς η αναδομημένη βάση D είναι ίδια με την αρχική. Αυτό βέβαια ήταν αναμενόμενο, αφού το πλέγμα στοιχειοσυνόλων και οι τιμές των μετρήσεων υποστήριξης, δεν μεταβλήθηκαν από τις αρχικές που είχαν παραχθεί από την αρχική βάση δεδομένων.

3.1.3 Προβλήματα – Αποτυχίες του Αλγόριθμου reconstruction_by_cardinality

Από την προηγούμενη ανάλυση, παρατηρούμε ότι υπάρχει μία *σχέση συνέπειας* (*consistent relationship*), μεταξύ των τιμών μέτρησης υποστήριξης των στοιχειοσυνόλων. *Οι τιμές αυτές πρέπει να είναι τέτοιες, ώστε να οδηγούν σε μη αρνητικές τιμές της πληθικότητας για κάθε στοιχειοσύνολο.*

Πράγματι, δεν είναι δυνατόν ένα στοιχειοσύνολο να έχει τιμή μέτρησης υποστήριξης μικρότερη από το άθροισμα όλων των τιμών πληθικότητας των υπερσυνόλων του. Για παράδειγμα αν *support_count(AB)=2* και *support_count(AC)=1*, τότε προφανώς η *support_count*(A) δεν μπορεί να είναι μικρότερη από 3, αφού το αντικείμενο A εμφανίζεται σε τρεις τουλάχιστον συναλλαγές με βάση τις *support_count* των AB και AC.

Η σχέση συνέπειας λοιπόν που πρέπει να ικανοποιούν οι τιμές της μέτρησης υποστήριξης των στοιχειοσυνόλων, εκφράζεται με τον παρακάτω τύπο:

$$\forall(X \in T)\ \ Support\_count(X) \geq \sum_{X \subset I' \in P(I)} f(I') \qquad (3.2)$$

Στο παράδειγμα 3.1 που δώσαμε παραπάνω, οι μετρήσεις υποστήριξης των στοιχειοσυνόλων προφανώς ικανοποιούσαν την σχέση συνέπειας, αφού προέρχονταν από μία πραγματική βάση δεδομένων. Το γεγονός αυτό της συνέπειας των μετρήσεων υποστήριξης, φάνηκε και στην εκτέλεση του αλγορίθμου, όπου σε καμία επανάληψη δεν προέκυψε αρνητική τιμή για την πληθικότητα κανενός στοιχειοσυνόλου (βλ. Πίνακα 3.2). Αν σε κάποιο βήμα του αλγόριθμου, προκύψει αρνητική τιμή της μέτρησης υποστήριξης για κάποιο στοιχειοσύνολο, τότε συμβαίνει όπως λέμε *αποτυχία* (*fail*).

Για παράδειγμα έστω το παρακάτω σύνολο συχνών στοιχειοσυνόλων με τις τιμές μέτρησης υποστήριξης για το καθένα από αυτά:

$\{(A)_{10}, (B)_{11}, (C)_{10}, (D)_{8}, (AB)_{7}, (AC)_{6}, (AD)_{6}, (BC)_{7}, (BD)_{6}, (CD)_{4}, (ABC)_{4}, (ABD)_{4}\}$

Εκτελώντας τον Αλγόριθμο reconstruction_by_cardinality, έχουμε:

$$f(ABC) = Support\_count(ABC) = 4$$

$$f(ABD) = Support\_count(ABD) = 4$$

$$f(AB) = Support\_count(AB) - (f(ABC) + f(ABD)) = 7 - (4+4) = 7 - 8 = -1$$ ***(fail)***

Για να οδηγηθεί ο αλγόριθμος σε αποτυχία, προφανώς δεν ικανοποιούνταν η σχέση συνέπειας της (3.2).

Πράγματι, αν ικανοποιούνταν αυτή η σχέση, αν για παράδειγμα η μέτρηση υποστήριξης του συνόλου AB ήταν support_count(AB)= 8, τότε

$$f(AB) = Support\_count(AB) - (f(ABC) + f(ABD)) = 8 - (4+4) = 8 - 8 = 0$$

και δεν θα συνέβαινε αποτυχία του αλγορίθμου (τουλάχιστον σε αυτό το βήμα).

3.1.4 Άρση Αποτυχιών του Αλγορίθμου reconstruction_by_cardinality

Όπως αναλύσαμε παραπάνω, οι δοθείσες μετρήσεις υποστήριξης των στοιχειοσυνόλων αντιπροσωπεύουν μία έγκυρη έξοδο της εξόρυξης συχνών στοιχειοσυνόλων και κατά συνέπεια θα είμαστε σε θέση να κατασκευάσουμε μία αναδομημένη βάση δεδομένων, μόνο όταν αυτές (οι μετρήσεις υποστήριξης) ικανοποιούν τη σχέση συνέπειας (3.2).

Με δεδομένο ότι για να μπορέσει να επιτευχθεί η απόκρυψη των ευαίσθητων κανόνων συσχέτισης, θα πρέπει να τροποποιηθεί ανάλογα η μέτρηση υποστήριξης ορισμένων συχνών στοιχειοσυνόλων, το προφανές ερώτημα που ανακύπτει είναι το εξής: *«Είναι δυνατόν να κατασκευάσουμε μία βάση δεδομένων όταν οι δοθείσες τιμές των μετρήσεων υποστήριξης δεν ικανοποιούν αυτήν την σχέση συνέπειας;»*

Εξίσου προφανής είναι και η απάντηση στο παραπάνω ερώτημα, ότι κάτι τέτοιο δεν είναι δυνατόν να επιτευχθεί για τις συγκεκριμένες τιμές μέτρησης υποστήριξης.

Στη συνέχεια, θα προσπαθήσουμε να δώσουμε κάποιους τρόπους με τους οποίους είναι εφικτή η κατασκευή μιας αναδομημένης βάσης δεδομένων, για την οποία οι τιμές των μετρήσεων υποστήριξης των στοιχειοσυνόλων, θα είναι όσο το δυνατόν πιο κοντά στις δοθείσες τιμές. Με άλλα λόγια θα προσπαθήσουμε να ξεπεράσουμε τις αποτυχίες του Αλγορίθμου 3.1 και να οδηγηθούμε στην επιτυχή ολοκλήρωσή του.

Από τη στιγμή λοιπόν που στην εκτέλεση του αλγορίθμου συμβεί fail σε κάποιο στοιχειοσύνολο-k δηλαδή η πληθικότητα του στοιχειοσυνόλου είναι αρνητική (έστω *Y* ένα τέτοιο στοιχειοσύνολο), για να μην έχουμε *inconsistency* (*ασυνέπεια*), πρέπει προφανώς η πληθικότητα του συγκεκριμένου στοιχειοσυνόλου *Y*, να γίνει θετική ή μηδέν.

Λαμβάνοντας υπ' όψιν μας τον τύπο (3.1), καταλήγουμε στο συμπέρασμα η άρση της ασυνέπειας, μπορεί να επιτευχθεί με τρεις τρόπους συνολικά:

1. Αύξηση της μέτρησης υποστήριξης (*support_count*) του συγκεκριμένου στοιχειοσυνόλου *Y* όσο η απόλυτη τιμή της πληθικότητας του στοιχειοσυνόλου *Y*

2. Μείωση του αθροίσματος των τιμών πληθικότητας όλων των στοιχειοσυνόλων-k+1 που είναι υπερσύνολα του στοιχειοσυνόλου *Y*, τόσο όσο η απόλυτη τιμή της πληθικότητας του στοιχειοσυνόλου *Y*.

   Η μείωση αυτή μπορεί να επιτευχθεί με δύο τρόπους:

   2.1. Μείωση της μέτρησης υποστήριξης (*support_count*) κάποιου ή κάποιων υπερσυνόλων του *Y*, τάξης k+1

   2.2. Αύξηση της μέτρησης υποστήριξης (support_count) κάποιου υπερσυνόλου του *Y*, τάξης k+2

Άρση αποτυχίας μέσω αύξησης της μέτρησης υποστήριξης του στοιχειοσυνόλου *Y* στο οποίο συνέβη αποτυχία

Είναι ο πιο απλός και εμφανής τρόπος για άρση της αποτυχίας.

Η μέτρηση υποστήριξης του στοιχειοσυνόλου *Y*, πρέπει να αυξηθεί όσο η απόλυτη τιμή της πληθικότητας. Έτσι η πληθικότητα του στοιχειοσυνόλου θα είναι τώρα 0 και αν ο αλγόριθμος ξαναεκτελεσθεί δεν θα έχουμε αποτυχία (fail).

Στο παράδειγμα που είχαμε δώσει παραπάνω στην υπο-ενότητα3.1.3, δείξαμε ότι αν η μέτρηση υποστήριξης του συνόλου AB, αυξηθεί από 7 που είναι αρχικά σε 8, τότε:

$$f(AB) = 8 - (4 + 4) = 0$$

κι έτσι δεν θα έχουμε αποτυχία στην εκτέλεση του αλγορίθμου.

Με βάση λοιπόν τα παραπάνω, προτείνουμε δύο τροποποιήσεις στην εκτέλεση του Αλγορίθμου 1:

Τροποποίηση του Αλγορίθμου reconstruction_by_cardinality

Κατά την εκτέλεση του αλγορίθμου όπως περιγράφηκε και δόθηκε από τους Chen, Orlowska και Li, κάθε φορά που συμβαίνει fail σε κάποιο στοιχειοσύνολο, διακόπτουμε την εκτέλεση του αλγορίθμου, διορθώνουμε την μέτρηση υποστήριξης του αντίστοιχου στοιχειοσυνόλου (την αυξάνουμε όσο η απόλυτη τιμή της πληθικότητας του στοιχειοσυνόλου) και ξαναρχίζουμε την εκτέλεση του αλγορίθμου από την αρχή, με τις τροποποιημένες όμως τιμές μέτρησης υποστήριξης.

Στη συνέχεια δίνουμε την τροποποίηση του αλγορίθμου:

Αλγόριθμος 3.2 *– Πρώτη Τροποποίηση Αλγορίθμου reconstruction_by_cardinality*

```
1:  function increase_ itemset-k_ support (itemsets I , itemsets support counts SUP)
2:      NSUP ← SUP   /* Create new support counts copying support counts */
3:      D ← ∅
4:      Q ← ∅
5:      S ← ∅
6:      k ← the length of the largest non-empty itemsets ;
7:      Repeat
8:          Q ← { I_i | |I_i| = k && new_support_count (I_i) > 0 }
9:          while non empty (Q)
10:             p ← remove (Q)
11:             duplicate p for new_support_count (p) times and add them into D ;
12:             S ← { f_i | f_i ⊂ p } /* find all subsets of p */
13:             for each q ∈ S do
14:                 new_support_count (q) ← new_support_count (q) – new_support_count (p) ;
```

```
15:                         if new_support_count (q) < 0
16:                             support_count (q) ←support_count (q) + abs(new_support_count (q))
17:                             break
18:                         end if
19:                     end for
20:                     if (∃j in I : new_support_count (j)) < 0 then
21:                         break
22:                     end if
23:                 end while
24:                 k = k-1
25:             until k=0
26:             if (∃j in I : new_support_count (j) < 0) then
27:                 increase_ itemset-k_ support ( I, SUP)
28:             else
29:                 return (D)
30:             end if
31:  end function
```

Άρση αποτυχίας μέσω μείωσης της μέτρησης υποστήριξης κάποιων υπερσυνόλων τάξης k+1 του στοιχειοσυνόλου *Y* στο οποίο συνέβη αποτυχία

Από τον τύπο (3.1) είναι προφανές ότι μειώνοντας την μέτρηση υποστήριξης (*support_count*) κάποιου στοιχειοσυνόλου *X*, θα μειωθεί και η τιμή της πληθικότητα *f(X)* αυτού του στοιχειοσυνόλου.

Έστω λοιπόν ότι κατά την εκτέλεση του αλγορίθμου, προκύπτει αρνητική πληθικότητα για κάποιο στοιχειοσύνολο-k, έστω *Y*, (*f(Y) < 0*, οπότε έχουμε fail). Αν μειώσουμε την μέτρηση υποστήριξης κάποιου στοιχειοσυνόλου-k+1 υπερσυνόλου του *Y*, έστω *Z*, (οπότε θα μειωθεί και η πληθικότητα *f(Z)*), τότε από τον τύπο (3.1) καταλήγουμε στο συμπέρασμα ότι τελικά θα έχουμε αύξηση της πληθικότητα *f(Y)*.

Επιλέγοντας λοιπόν στοιχειοσύνολα-k+1, υπερσύνολα του στοιχειοσυνόλου *Y*, και μειώνοντας κατάλληλα την μέτρηση υποστήριξής τους (*support_count*), θα επιτύχουμε τελικά την αύξηση της πληθικότητας *f(Y)* τουλάχιστον μέχρι το 0. Εκτελώντας λοιπόν, ξανά τον αλγόριθμο από την αρχή, δεν θα έχουμε αποτυχία στο ίδιο στοιχειοσύνολο Y.

Πράγματι, στο παράδειγμα που είχαμε δει παραπάνω στην υπο-ενότητα 3.1.3, συνέβη αποτυχία στο στοιχειοσύνολο-2 *AB*, αφού $f(AB) = -1$.

Αν μειώσουμε σε 3 την μέτρηση υποστήριξης του στοιχειοσυνόλου-3 *ABC,* το οποίο είναι επιπλέον και υπερσύνολο του *AB*, θα έχουμε:

$$f(ABC) = Support\_count(ABC) = 3$$

$$f(ABD) = Support\_count(ABD) = 4$$

$$f(AB) = Support\_count(AB) - (f(ABC) + f(ABD)) = 7 - (3+4) = 7 - 7 = 0$$

Έτσι τώρα δεν συμβαίνει αποτυχία στο στοιχειοσύνολο *AB*.

ΣΗΜ: Στο ίδιο αποτέλεσμα θα είχαμε καταλήξει αν είχαμε μειώσει σε 3, την μέτρηση υποστήριξης του στοιχειοσυνόλου-3 ABD.

ΠΕΡΙΟΡΙΣΜΟΙ

Από την παραπάνω ανάλυση, είναι προφανές ότι:

1. Τα στοιχειοσύνολα-k+1 που θα επιλέξουμε να μειώσουμε τις μετρήσεις υποστήριξής τους, θα πρέπει να έχουν μη μηδενική πληθικότητα, διότι σε αντίθετη περίπτωση είναι προφανές ότι κατά την εκ νέου εκτέλεση του αλγορίθμου, θα συμβεί αποτυχία (fail) σε αυτά.

2. Σε κάθε στοιχειοσύνολο-k+1 που επιλέγεται για να γίνει μείωση της μέτρησης υποστήριξης, η μείωση αυτή πρέπει να είναι <u>το πολύ</u> όσο η τιμή της πληθικότητας του συγκεκριμένου στοιχειοσυνόλου, διότι αλλιώς είναι προφανές ότι κατά την νέα εκτέλεση του αλγορίθμου θα συμβεί αποτυχία (fail) σε αυτό.

Ακολουθεί ο αλγόριθμος που προτείνουμε:

Αλγόριθμος 3.3 *– Δεύτερη Τροποποίηση Αλγορίθμου reconstruction_by_cardinality*

```
1:  function decrease_itemset-k+1_support(itemsets I, itemsets support counts SUP)
2:      D ← ∅
3:      Q ← ∅
4:      S ← ∅
5:      U ← ∅
6:      W ← ∅
7:      k ← the length of the largest non-empty itemsets
8:      Repeat
9:          Q ← { I_i | |I_i| = k && support_count (I_i) > 0 }
10:         while non empty (Q)
11:             p ← remove (Q) ;
```

```
12:                 duplicate p for support_count (p) times and add them into D
13:                 S ←{ f_i | f_i ⊂ p } ; /* find all subsets of p */
14:                 for each q ∈ S do
15:                     support_count (q) ←support_count (q) – support_count (p)
16:                     if (support_count (q)) < 0 then
17:                         U ←insert (q)
18:                     end if
19:                 end for
20:             end while
21:             if not empty (U) then
22:                 break
23:             end if
24:             k = k-1
25:         until k=0
26:         if empty (U) then
27:             return D
28:         else
29:             while not empty (U)
30:                 u ←remove (U)
31:                 w ←first f ∈ I : |f| = k && f ⊃ u  /* find first superset of u, with length k */
32:                 support_count (w) ←support_count (w)+min(support_count (w), abs(support_count(u))
33:                 support_count (u) ←support_count (u)+min(support_count (w), abs(support_count(u)
34:                 for each u ∈ U : u ⊂ w do
35:                   u ←remove (U)
36:                   support_count (u) ←support_count (u)+min(support_count (w), abs(support_count(u)))
37:                 end for
38:             end while
39:             decrease_itemset-k+1 _support (I, SUP)
40          end if
41: end function
```

Άρση αποτυχίας μέσω αύξησης της μέτρησης υποστήριξης κάποιου υπερσυνόλου τάξης k+2 του στοιχειοσυνόλου *Y* στο οποίο συνέβη αποτυχία

Από την σχέση (1) που ορίζει το μέγεθος της πληθικότητας, παρατηρούμε ότι υπάρχει μία μορφή «*αναδρομικότητας*» στις αντίστοιχες τιμές των στοιχειοσυνόλων διαφορετικών επιπέδων (διαφορετικής τάξης).

Πράγματι, η πληθικότητα ενός στοιχειοσυνόλου-k, εξαρτάται άμεσα από τις τιμές της πληθικότητες των υπερσυνόλων του, τάξης k+1. Οι τιμές αυτές με τη σειρά τους, εξαρτώνται άμεσα από τις τιμές της πληθικότητας των αντίστοιχων υπερσυνόλων τάξης k+2, κοκ. Τελικά

λοιπόν η πληθικότητα ενός στοιχειοσυνόλου k τάξης, εξαρτάται από την πληθικότητα όλων των υπερσυνόλων του οποιασδήποτε τάξης, και επηρεάζει την πληθικότητα όλων των υποσυνόλων του, οποιασδήποτε τάξης.

Με βάση τα παραπάνω λοιπόν, μπορούμε να πούμε ότι η μείωση της πληθικότητας ενός στοιχειοσυνόλου-k+1 (η οποία είναι επιθυμητή διότι θα μας οδηγήσει σε αύξηση της πληθικότητας του στοιχειοσυνόλου-k στο οποίο συνέβη αποτυχία), μπορεί να επιτευχθεί με αύξηση της μέτρησης υποστήριξης (*support_count*) κάποιου στοιχειοσυνόλου-k+2.

Πράγματι, η αύξηση της μέτρησης υποστήριξης ενός οποιουδήποτε στοιχειοσυνόλου, κατά x μονάδες, έχει σαν αποτέλεσμα την αύξηση της πληθικότητας γι αυτό το στοιχειοσύνολο επίσης κατά x μονάδες. Η αύξηση της πληθικότητας κατά x μονάδες, θα έχει σαν αποτέλεσμα τη μείωση της πληθικότητας του κάθε υποσυνόλου-k+1, κατά x μονάδες.

Αν λάβουμε υπ' όψιν μας το γεγονός ότι μεταξύ ενός στοιχειοσυνόλου-k+2, (έστω *A*) και ενός υποσυνόλου-k, (έστω *B*), υπάρχουν μόνο δύο στοιχειοσύνολα-k+1 (έστω C και D) τα οποία είναι υποσύνολα του *A* και ταυτόχρονα υπερσύνολα του B, η μείωση αυτή θα έχει σαν τελικό αποτέλεσμα την αύξηση της πληθικότητας του *B, f(B).*

Πράγματι, ξέρουμε ότι ισχύει:

$$f(B) = Support\_count(B) - (f(C) + f(D) + f(A))$$

Μετά την αύξηση της *f(A)* κατά x, έχουμε μείωση των *f(C)* και *f(D)* κατά x (η κάθε μία). Άρα:

$f'(A) = f(A) + x$, $f'(C) = f(C) - x$, $f'(D) = f(D) - x$ και τελικά:

$$f'(B) = Support\_count(B) - (f'(C) + f'(D) + f'(A)) =$$
$$= Support\_count(B) - (f(C) - x + f(D) - x + f(A) + x) =$$
$$= Support\_count(B) - (f(C) + f(D) + f(A)) + x = f(B) + x$$

Δηλαδή η τιμή της πληθικότητας για το στοιχειοσύνολο *B, f(B),* αυξήθηκε κατά x μονάδες.

Με βάση όλα τα παραπάνω, καταλήγουμε στο συμπέρασμα ότι μπορούμε να επιτύχουμε αύξηση της πληθικότητας του στοιχειοσυνόλου-k στο οποίο συνέβη αποτυχία του αλγορίθμου, (έτσι ώστε να ξεπεράσουμε την αποτυχία), αν αυξήσουμε την μέτρηση υποστήριξης ενός

στοιχειοσυνόλου-k+2 το οποίο θα είναι: Υπερσύνολο δύο (2) στοιχειοσυνόλων-k+1, τα οποία θα είναι υπερσύνολα του στοιχειοσυνόλου-k στο οποίο παρατηρήθηκε η αποτυχία του αλγορίθμου και επιπλέον θα έχουν μη μηδενική πληθικότητα.

Περιορισμοι

1. Η αύξηση της μέτρησης υποστήριξης (*support_count*) του στοιχειοσυνόλου-k+2, θα πρέπει να είναι το πολύ όσο η ελάχιστη τιμή της πληθικότητας των 2 στοιχειοσυνόλων-k+1 (αλλιώς θα συμβεί νέα αποτυχία του αλγορίθμου αφού η πληθικότητα του στοιχειοσυνόλου-k+1 θα μειωθεί όσο η αύξηση της μέτρησης υποστήριξης κι επομένως θα γίνει αρνητική).

2. Κανένα υποσύνολο του στοιχειοσυνόλου-k+2 στο οποίο αυξήσαμε την μέτρηση υποστήριξης, δεν πρέπει να έχει μικρότερη μέτρηση υποστήριξης. Αν συμβαίνει κάτι τέτοιο, πρέπει να αυξήσουμε και την δική του μέτρηση υποστήριξης τόσο ώστε οι μετρήσεις υποστήριξης των δύο στοιχειοσυνόλων (στοιχειοσυνόλου-k+2 και υποσυνόλου του, να είναι ίσες).

Με όλη την παραπάνω προσέγγιση, η συνολική μείωση των τιμών της πληθικότητας των στοιχειοσυνόλων-k+1 και άρα η αύξηση της πληθικότητας του στοιχειοσυνόλου στο οποίο είχε συμβεί αποτυχία, είναι όση και η αύξηση της μέτρησης υποστήριξης του στοιχειοσυνόλου-k+2.

Στο παράδειγμα που είχαμε δώσει στην υπο-ενότητα 3.1.3 αλλά και σε άλλες παραγράφους, συνέβη αποτυχία στο στοιχειοσύνολο-2 AB, αφού *f(AB) = -1.* Με βάση όλη την παραπάνω ανάλυση και λαμβάνοντας υπ' όψιν μας και τους περιορισμούς που θέσαμε, θα πρέπει να αυξήσουμε την μέτρηση υποστήριξης του στοιχειοσυνόλου ABCD από 0 που είναι αρχικά, σε 1. Αν λοιπόν γίνει αυτή η αύξηση, θα έχουμε:

$$f(ABCD) = Support\_count(ABCD) = 1$$

$$f(ABC) = Support\_count(ABC) - f(ABCD) = 4 - 1 = 3$$

$$f(ABD) = Support\_count(ABD) - f(ABCD) = 4 - 1 = 3$$

$$f(AB) = Support\_count(AB) - (f(ABC) + f(ABD) + f(ABCD)) = 7 - (3 + 3 + 1) = 7 - 7 = 0$$

Στη συνέχεια δίνουμε την τρίτη τροποποίηση του αλγορίθμου 3.1

Αλγόριθμος 3.4 *– Τρίτη Τροποποίηση Αλγορίθμου reconstruction_by_cardinality*

```
1:  function increase_itemset-k+2_support (itemsets I , itemsets support counts SUP)
2:      SUP ← support_correction(I, SUP)
3:      NSUP ←SUP
4:      D ←∅
5:      Q ←∅
6:      S ←∅
7:      U ←∅
8:      W ←∅
9:      k ←the length of the largest non-empty itemsets
10:     Repeat
11:         Q ←{ I_i | |I_i|= k && support_count (I_i) > 0 }
12:         while non empty (Q)
13:             p ←remove (Q)
14:             duplicate p for new_support_count (p) times and add them into D
15:             S ←{ f_i | f_i⊂ p } ; /* find all subsets of p */
16:             for each q∈S do
17:                 new_support_count (q) ←new_support_count (q) –new_support_count (p)
18:                 if (new_support (q) < 0) then
19:                     U ←insert (q)
20:                 end if
21:             end for
22:         end while
23:         if empty (U) then
24:             k ← k-1
25:         else
26:             U ←subset_elimination (U)
27:             break
28:         end if
29:     until k=0
30:     if empty (U) then
31:         return (D)
32:     else
33:         m←the length of the itemsets in U (those with negative new_support_counts)
34:         W ←{ I_i | |I_i| = m+2 }
35:         while not empty (U)
36:             u ←remove (U)
37:             ns ←abs(new_support_count (u))
38:             for each w ∈W do
39:                 if (u ⊂ w && check_support(u)=1) then
40:                     support_count (w) ←support_count (w)+ns
41:                     for each u1 ∈U
42:                         if (u1 ⊂ w) then
```

```
43:                                    new_support_count (u1)←new_support_count (u1)+ns
44:                                    if (new_support_count (u1)=0) then
45:                                        U ←remove u1 from U
46:                                    end if
47:                                end if
48:                            end for
49:                        end if
50:                    end for
51:                end while
52:                SUP ←support_correction (I, SUP)
53:                increase_itemset-k+2_support(I,SUP)
54:            end if
55:  end function
```

ΠΑΡΑΤΗΡΗΣΕΙΣ – ΕΠΙΣΗΜΑΝΣΕΙΣ

Στον τρίτο αλγόριθμο *increase_itemset-k+2_support* που δίνουμε παραπάνω, έχουμε να κάνουμε τις εξής παρατηρήσεις:

1. Στην εκκίνηση του αλγορίθμου, καλείται μία επιπλέον συνάρτηση (αυτή που ονομάζουμε *support_correction*), η οποία έχει σκοπό να ελέγξει το σύνολο των στοιχειοσυνόλων P(*I*) για το αν ισχύει η Apriori αρχή, σύμφωνα με την οποία κανένα στοιχειοσύνολο δεν μπορεί να έχει μέτρηση υποστήριξης μεγαλύτερη από οποιοδήποτε υποσύνολο του. Σε περίπτωση που ισχύει κάτι τέτοιο, διορθώνει την μέτρηση υποστήριξης του υποσυνόλου θέτοντας την, ίση με την μέτρηση υποστήριξης του υπερσυνόλου του.

   Αυτός ο έλεγχος και ενδεχομένως η διόρθωση, είναι απαραίτητος σ' αυτόν τον αλγόριθμο (σε αντίθεση με τους δύο προηγούμενους), διότι τώρα σε περίπτωση που συμβεί fail σε κάποιο στοιχειοσύνολο-k, αυξάνουμε ανάλογα την μέτρηση υποστήριξης κάποιου υπερσυνόλου-k+2. Η αύξηση αυτή ενδέχεται να είναι παραπάνω από τη μέτρηση υποστήριξης κάποιου στοιχειοσυνόλου-k+1, με αποτέλεσμα την καταστρατήγηση της Apriori αρχής.

2. Η εκτέλεση του αλγορίθμου δεν διακόπτεται αμέσως μόλις συμβεί fail σε κάποιο στοιχειοσύνολο-k, αλλά μόλις ολοκληρωθεί ο υπολογισμός της πληθικότητας για όλα τα στοιχειοσύνολα-k (ίδιας τάξης). Αυτό γιατί:

   2.1. Η πληθικότητα των υπόλοιπων στοιχειοσυνόλων-k δεν εξαρτάται από την πληθικότητα αυτού του στοιχειοσυνόλου (γενικά η πληθικότητα στοιχειοσυνόλου k-τάξης δεν εξαρτάται από την πληθικότητα κανενός άλλου στοιχειοσυνόλου k-τάξης).

Έτσι, η αρνητική πληθικότητα του στοιχειοσυνόλου στο οποίο συνέβη fail, δεν μπορεί να αλλοιώσει και κάποιο ενδεχόμενο fail επόμενου στοιχειοσυνόλου-k.

2.2. Έχοντας όλα τα στοιχειοσύνολα k-τάξης με αρνητικές πληθικότητες (οι οποίες δεν επηρεάστηκαν η μία από την άλλη) μπορούμε να επιλέξουμε να αυξήσουμε την μέτρηση υποστήριξης ενός κατάλληλου υπερσυνόλου k+2 τάξης, δηλαδή κάποιου που να είναι υπερσύνολο δύο «*αποτυχημένων*» στοιχειοσυνόλων. Έτσι με μία μόνο αύξηση της μέτρησης υποστήριξης κάποιου στοιχειοσυνόλου, επιτυγχάνουμε την διόρθωση του fail των δύο υποσυνόλων του.

Για παράδειγμα αν ο αλγόριθμος έφτασε στα στοιχειοσύνολα-1 που είναι κατά σειρά τα A, B, C, D και γίνει fail στο A με f(A)=-1, τότε αφήνουμε τον αλγόριθμο να υπολογίσει τις πληθικότητες και των υπόλοιπων στοιχειοσυνόλων-1 και τότε τον διακόπτουμε. Αν π.χ. υπολογίσει f(B)=1, f(C)=2 και f(D)=-1, τότε μπορούμε να επιλέξουμε να αυξήσουμε την support του ABD κατά 1 έτσι ώστε στην επόμενη εκτέλεση του αλγορίθμου να προκύψουν f(A)=0 και f(D)=0. Αν είχαμε διακόψει τον αλγόριθμο στο fail του A, τότε μπορεί να επιλέγαμε να αυξήσουμε την support του ABC, οπότε στην επόμενη εκτέλεση θα είχαμε ξανά fail στο D, αφού και πάλι f(D)=-1

3. Σε περίπτωση που συμβεί κάποιο fail και επομένως υπάρχει κάποιο σύνολο U στοιχειοσυνόλων με αρνητικές πληθικότητες, καλείται μία επιπλέον συνάρτηση *subset_elimination*. Αυτή η συνάρτηση απαλείφει για κάθε στοιχειοσύνολο του U, *όλα τα υποσύνολά του μέσα στο U*. Με άλλα λόγια, στο σύνολο U θα απομείνουν μόνο τα στοιχειοσύνολα της μεγαλύτερης τάξης.

4. Η συνάρτηση *check_support* που χρησιμοποιούμε, ελέγχει αν κάποιο στοιχειοσύνολο-k+2, και υπερσύνολο κάποιου στοιχειοσυνόλου-k με αρνητική πληθικότητα (έστω *X*), έχει μικρότερη μέτρηση υποστήριξης και από τα δύο στοιχειοσύνολα-k+1 που είναι υπερσύνολα του *X* και προφανώς δικά του υποσύνολα. Αν δεν συμβαίνει κάτι τέτοιο (αν δηλαδή η μέτρηση υποστήριξής του είναι ίση με την μέτρηση υποστήριξης κάποιου από τα δύο στοιχειοσύνολα-k+1), τότε αν αυξήσουμε την μέτρηση υποστήριξής του θα πρέπει λόγω της Apriori αρχής να αυξήσουμε και την μέτρηση υποστήριξης αυτού του στοιχειοσυνόλου-k+1, με αποτέλεσμα να μην έχουμε τελικό όφελος στην μέτρηση υποστήριξης του στοιχειοσυνόλου-k, *X*.

Στη συνέχεια παραθέτουμε τρία παραδείγματα για τα αποτελέσματα που παίρνουμε (τις βάσεις δεδομένων που προκύπτουν) για καθέναν από τους τρεις παραπάνω αλγορίθμους:

ΠΑΡΑΔΕΙΓΜΑ 3.2

Έστω το ακόλουθο σύνολο στοιχειοσυνόλων S1 με τις αντίστοιχες τιμές μέτρησης υποστήριξης: S1 = {$(ABC)_2$, $(AB)_3$, $(AC)_3$, $(BC)_3$, $(A)_3$, $(B)_3$, $(C)_3$}

Εδώ θα έχουμε fail στο A αφού:

$$f(A) = Support\_count(A) - (f(AB) + f(AC) + f(ABC)) = 3 - (1+1+2) = -1$$

Η αναδομημένη βάση D (δηλαδή το σύνολο στοιχειοσυνόλων), που επιστρέφεται από την εκτέλεση καθενός από τους τρεις αλγορίθμους, φαίνεται στον Πίνακα 3.5

Πίνακας 3.5: Αναδομημένη Βάση που επιστρέφουν οι 3 αλγόριθμοι, για το στοιχειοσύνολο S1

S1 = {$(ABC)_2$, $(AB)_3$, $(AC)_3$, $(BC)_3$, $(A)_3$, $(B)_3$, $(C)_3$}		
increase_itemset -k_support	decrease_itemset -k+1_support	increase_itemset -k+2_support
ABC ABC AB AC	ABC ABC AC B	ABC ABC ABC

ΠΑΡΑΔΕΙΓΜΑ 3.3

Έστω τώρα το σύνολο στοιχειοσυνόλων S2, με τις αντίστοιχες τιμές μέτρησης υποστήριξης: S2={$(ABC)_1$, $(AB)_3$, $(AC)_3$, $(BC)_2$, $(A)_4$, $(B)_4$, $(C)_3$ }. Εδώ θα έχουμε fail στο A και στο C αφού:

$$f(A) = Support\_count(A) - (f(AB) + f(AC) + f(ABC)) = 4 - (2+2+1) = -1$$

$$f(C) = Support\_count(C) - (f(AC) + f(BC) + f(ABC)) = 3 - (2+1+1) = -1$$

Η αναδομημένη βάση D που επιστρέφεται από την εκτέλεση καθενός από τους τρεις αλγορίθμους, φαίνεται στον Πίνακα 3.6

Πίνακας 3.6: Αναδομημένη Βάση που επιστρέφουν οι 3 αλγόριθμοι, για το στοιχειοσύνολο S2

$S2 = \{(ABC)_1, (AB)_3, (AC)_3, (BC)_2, (A)_4, (B)_4, (C)_3\}$		
increase_itemset-k_support	decrease_itemset-k+1_support	increase_itemset-k+2_support
ABC	ABC	ABC
AB	AB	ABC
AB	AB	AB
AC	AC	AC
AC	BC	B
BC		

ΠΑΡΑΔΕΙΓΜΑ 3.4

Ας δούμε τώρα την λειτουργία των τριών αλγορίθμων και τα αποτελέσματα που επιστρέφουν, για το σύνολο στοιχειοσυνόλων S3, το οποίο δίνεται ως παράδειγμα στο [03]. Έχουμε λοιπόν: $S3=\{(ABC)_4, (ABD)_4, (AB)_7, (AC)_6, (AD)_6, (BC)_7, (BD)_6, (CD)_4, (A)_{10}, (B)_{11}, (C)_{10}, (D)_8\}$. Όπως αναλύεται και παρουσιάζεται στο [03], έχουμε fail διαδοχικά στα AB, A και D αφού αν εκτελέσουμε τον 1ο βασικό αλγόριθμο, προκύπτουν: $f(AB) = -1$, $f(A) = -1$ και $f(D) = -1$.

Η αναδομημένη βάση που επιστρέφεται από τους τρεις αλγόριθμους φαίνεται στον Πίνακα 3.7 που ακολουθεί (για λόγους οικονομίας χώρου σε κάθε κελί του Πίνακα, μπορεί να υπάρχουν μέχρι και τρεις συναλλαγές):

Πίνακας 3.7: Αναδομημένη Βάση που επιστρέφουν ο reconstruction_by_cardinality και οι τρεις αλγόριθμοι τροποποίησης, για το στοιχειοσύνολο S3

$S3 = \{(ABC)_4, (ABD)_4, (AB)_7, (AC)_6, (AD)_6, (BC)_7, (BD)_6, (CD)_4, (A)_{10}, (B)_{11}, (C)_{10}, (D)_8\}$											
reconstruction_by_cardinality		increase_itemset-k_support			decrease_itemset-k+1_support			increase_itemset-k+2_support	
ABCD	AD	ABC	AD	CD	BC	CD	A	ABCD	BC
ABCD	BC	ABC	AD		BC	CD	A	ABCD	BD
ABCD	BC	ABC	BC		BC	CD	A	ABCD	A
ABC	BC	ABC	BC		BC	A	B	ABCD	B
ABD	CD	ABD	BC		BC	A	B	ABCD	C
AB	BD	ABD	BD		BD	A	C	AB	C
AB	BD	ABD	BD		BD	A		AB	D
AC		ABD	CD		BD	A		AC	
AC		AC	CD		BD	A		AD	
AD		AC	CD		CD	A		BC	

3.1.5 Βελτίωση των Αλγορίθμων increase_itemset-k_support, decrease_itemset-k+1_support, increase_itemset-k+2_support

Το βασικό ερώτημα που προκύπτει σε σχέση με τους τρεις παραπάνω αλγορίθμους που προτείνουμε, είναι: «ποιος από τους τρεις αλγορίθμους είναι ο καλύτερος σε σχέση με την αναδομημένη βάση την οποία επιστρέφει;» ή αλλιώς «ποια είναι η αποδοτικότερη αναδομημένη βάση δεδομένων που επιστρέφεται από τους τρεις παραπάνω αλγορίθμους;». Με τον όρο αποδοτικότερη βάση δεδομένων, προφανώς εννοείται η βάση εκείνη που επιτυγχάνει περισσότερο τους τρεις στόχους της απόκρυψης κανόνων συσχέτισης, όπως αυτοί είχαν τεθεί στην ενότητα 2.2.

Δεν πρέπει να ξεχνάμε όμως, ότι η στρατηγική του Αλγορίθμου reconstruction_by_cardinality και κατ' επέκταση των τρών τροποποιήσεών του που προτείναμε, βασίζεται στην απόκρυψη των συχνών στοιχειοσυνόλων και όχι στην απόκρυψη των κανόνων συσχέτισης. Έτσι, οι *παρενέργειες* (*side effects*) από την αναδομημένη βάση που θα προκύψει, εξαρτώνται σε πολύ μεγάλο βαθμό από το σύνολο των κανόνων (ευαίσθητων και μη) οι οποίοι παράγονται από τα στοιχειοσύνολα που θέλουμε να αποκρύψουμε, καθώς και από το συνολικό αριθμό των ευαίσθητων και το συνολικό αριθμό των μη ευαίσθητων κανόνων. Με άλλα λόγια, μπορεί ένας αλγόριθμος να επιτύχει να επιστρέψει μία αναδομημένη βάση δεδομένων, η οποία να ανταποκρίνεται 100% στους περιορισμούς που θέσαμε (δηλαδή οι μετρήσεις υποστήριξης όλων των στοιχειοσυνόλων να είναι ακριβώς αυτές που δώσαμε), αλλά να παράγονται αρκετές *παρενέργειες* κατά την εξόρυξη των κανόνων συσχέτισης. Επίσης, οι τρεις αλγόριθμοι, δέχονται σαν είσοδο *τις μετρήσεις υποστήριξης των στοιχειοσυνόλων και όχι τις συχνότητες αυτών*. Συνεπώς, και αφού δεν γνωρίζουμε εκ των προτέρων τον αριθμό των συναλλαγών της νέας βάσης δεδομένων που θα προκύψει, δεν μπορούμε να αποφανθούμε πριν την εκτέλεση των τριών αλγορίθμων, ποιος από αυτούς θα μας οδηγήσει στα καλύτερα αποτελέσματα. Προς αποσαφήνιση όλων αυτών, ας δούμε το παρακάτω παράδειγμα:

ΠΑΡΑΔΕΙΓΜΑ 3.5

Έστω το σύνολο 15 συναλλαγών D3={ABCD, ABCD, ABCD, ABC, ABD, ABD, AB, AC, AC, AD, BC, BC, BC, BD, CD}. Στον Πίνακα 3.6 φαίνονται τα συχνά στοιχειοσύνολα με τις αντίστοιχες μετρήσεις υποστήριξης, καθώς και οι παραγόμενοι κανόνες συσχέτισης με την αντίστοιχη εμπιστοσύνη, για κατώφλι υποστήριξης *minsup=0.3* και κατώφλι εμπιστοσύνης *minconf=0.6*.

Πίνακας 3.8: Συχνά στοιχειοσύνολα και κανόνες συσχέτισης για το σύνολο συναλλαγών D3 (minsup=0.3, minconf=0.6)

D3 = {ABCD, ABCD, ABCD, ABC, ABD, ABD, AB, AC, AC, AD, BC, BC, BC, BD, CD)				
Συναλλαγές συνόλου D3	Συχνό Στοιχειοσ	Μέτρηση Υποστήριξης	Κανόνας Συσχέτισης	Εμπιστοσύνη
ABCD ABCD ABCD ABC ABD ABD AB AC AC AD BC BC BC BD CD	ABD	5	A ⇒ B	0.7
	AB	7	B ⇒ A	0.63
	AC	6	B ⇒ C	0.63
	AD	6	C ⇒ B	0.70
	BC	7	D ⇒ A	0.75
	BD	6	D ⇒ B	0.75
	A	10	D ⇒ AB	0.625
	B	11	AB ⇒ D	0.71
	C	10	AD ⇒ B	0.83
	D	8	BD ⇒ A	0.83

Έστω ότι ο κανόνας D ⇒ AB, θεωρείται ευαίσθητος και δεν πρέπει να αποκαλυφθεί. Σε αυτήν την περίπτωση, το στοιχειοσύνολο ABD από το οποίο προκύπτει ο κανόνας, πρέπει να είναι μη συχνό στην αναδομημένη βάση, θα πρέπει δηλαδή η μέτρηση υποστήριξής του να είναι το πολύ 4 (αφού 5/15=0.3=*minsup* και 4/15 =0.266<*minsup*). Μειώνουμε λοιπόν την μέτρηση υποστήριξης του ABD σε 4 και για τα υπόλοιπα στοιχειοσύνολα την αφήνουμε όπως στον Πίνακα 3.6 και έχουμε τελικά το επόμενο σύνολο στοιχειοσυνόλων που θέλουμε να παράγεται από την αναδομημένη βάση: $S4=(ABCD)_3, (ABC)_4, (ABD)_4, (ACD)_3, (BCD)_3, (AB)_7, (AC)_6, (AD)_6, (BC)_7, (BD)_6, (CD)_4, (A)_{10}, (B)_{11}, (C)_{10}, (D)_8\}$.

Στον Πίνακα 3.9 φαίνεται η αναδομημένη βάση που επιστρέφει ο καθένας από τους τρεις αλγορίθμους, καθώς και η μέτρηση υποστήριξης για όλα τα στοιχειοσύνολα.

Πίνακας 3.9: Αναδομημένες βάσεις που επιστρέφονται από τους 3 αλγορίθμους για το S4 και οι μετρήσεις υποστήριξης για όλα τα στοιχειοσύνολα

S4=$(ABCD)_3$, $(ABC)_4$, $(ABD)_4$, $(ACD)_3$, $(BCD)_3$, $(AB)_7$, $(AC)_6$, $(AD)_6$, $(BC)_7$, $(BD)_6$, $(CD)_4$, $(A)_{10}$, $(B)_{11}$, $(C)_{10}$, $(D)_8$}								
increase_itemset-k_support			decrease_itemset-k+1_support			increase_itemset-k+2_support		
Συναλλαγές	Στοιχειοσύνολο	Μέτρηση Υποστήριξης	Συναλλαγές	Στοιχειοσύνολο	Μέτρηση Υποστήριξης	Συναλλαγές	Στοιχειοσύνολο	Μέτρηση Υποστήριξης
ABCD	ABCD	3	ABCD	ABCD	3	ABCD	ABCD	5
ABCD	ABC	4	ABCD	ABC	4	ABCD	ABC	5
ABCD	ABD	4	ABCD	ABD	4	ABCD	ABD	5
ABC	ACD	3	ABC	ACD	3	ABCD	ACD	5
ABD	BCD	3	ABD	BCD	3	ABCD	BCD	5
AB	AB	7	AB	AB	7	AB	AB	7
AB	AC	6	AC	AC	6	AB	AC	6
AC	AD	6	AC	AD	6	AC	AD	6
AC	BC	7	AD	BC	7	AD	BC	7
AD	BD	6	BC	BD	6	BC	BD	6
AD	CD	4	BC	CD	4	BC	CD	5
BC	A	11	BC	A	11	BD	A	10
BC	B	12	BD	B	12	A	B	11
BC	C	10	BD	C	10	B	C	10
BD	D	9	CD	D	9	C	D	8
BD			A			C		
CD						D		

Από τον παραπάνω πίνακα, φαίνεται ότι οι αλγόριθμοι *increase_itemset-k_support* και *decrease_itemset-k+1_support* είναι πιο επιτυχημένοι, αφού επέστρεψαν μία βάση στην οποία όλα σχεδόν τα στοιχειοσύνολα έχουν την επιθυμητή μέτρηση υποστήριξης (όπως φαίνεται από το σύνολο S4 που δώσαμε). Πράγματι, στη βάση δεδομένων που επιστρέφει ο *increase_itemset-k_support*, τα μόνα στοιχειοσύνολα που δεν έχουν την επιθυμητή μέτρηση υποστήριξης, είναι τα A (11 αντί 10), B (12 αντί 11) και D (9 αντί 8). Αν σκεφτούμε επιπλέον, ότι αυτά είναι στοιχειοσύνολα-1 οπότε δεν παράγουν κανόνες, τότε μπορούμε να ισχυρισθούμε ότι ο αλγόριθμος *increase_itemset-k_support*, πέτυχε απόλυτα. Ο αλγόριθμος *decrease_itemset-k+1_support*, επέστρεψε μία βάση στην οποία μόνο δύο στοιχειοσύνολα δεν έχουν την επιθυμητή μέτρηση υποστήριξης και συγκεκριμένα τα AB (6 αντί 7) και AD (5 αντί 6). Και τα δύο όμως αυτά στοιχειοσύνολα παραμένουν συχνά, αφού το σύνολο των συναλλαγών της αναδομημένης βάσης είναι 15 και έτσι η υποστήριξή τους είναι αντίστοιχα 0.4 και 0.33. Έτσι και ο αλγόριθμος *decrease_itemset-k+1_support*, φαίνεται ότι πέτυχε απόλυτα. Σε αντίθεση με τους προηγούμενους αλγόριθμους, ο *increase_itemset-k+2_support*, φαίνεται να μην είναι εξ' ίσου αποτελεσματικός, αφού στην αναδομημένη βάση που επιστρέφει, αρκετά στοιχειοσύνολα (έξι συνολικά) έχουν μέτρηση υποστήριξης διαφορετική από την επιθυμητή και μάλιστα το ευαίσθητο στοιχειοσύνολο ABD, έχει την ίδια μέτρηση υποστήριξης που είχε και στην αρχική βάση D3,

δηλαδή 5. Αν όμως εξετάσουμε σε κάθε μία από τις αναδομημένες βάσεις, το σύνολο των συχνών στοιχειοσυνόλων και τους κανόνες συσχέτισης που παράγονται, τότε προκύπτουν διαφορετικά συμπεράσματα: Στον Πίνακα 3.10 φαίνεται η αναδομημένη βάση που επιστρέφει ο καθένας από τους τρεις αλγορίθμους, τα συχνά στοιχειοσύνολα καθώς και οι κανόνες συσχέτισης.

Πίνακας 3.10: Αναδομημένες βάσεις που επιστρέφονται από τους τρεις αλγορίθμους για το S4, τα συχνά στοιχειοσύνολα και οι κανόνες συσχέτισης για κάθε μία από αυτές

$S4=(ABCD)_3, (ABC)_4, (ABD)_4, (ACD)_3, (BCD)_3, (AB)_7, (AC)_6, (AD)_6, (BC)_7, (BD)_6, (CD)_4, (A)_{10}, (B)_{11}, (C)_{10}, (D)_8\}$								
increase_itemset-k_support			decrease_itemset-k+1_support			increase_itemset-k+2_support		
Συναλλαγές	Συχνά Στοιχειοσύνολα	Κανόνες Συσχέτισης	Συναλλαγές	Συχνά Στοιχειοσύνολα	Κανόνες Συσχέτισης	Συναλλαγές	Συχνά Στοιχειοσύνολα	Κανόνες Συσχέτισης
ABCD ABCD ABCD ABC ABD AB AB AC AC AD AD BC BC BC BD BD CD	AB AC AD BC BD A B C D	$A \Rightarrow B$ $C \Rightarrow B$ $D \Rightarrow A$ $D \Rightarrow B$	ABCD ABCD ABCD ABC ABD AB AC AC AD BC BC BC BD BD CD A	AB AC AD BC BD A B C D	$B \Rightarrow C$ $C \Rightarrow B$ $D \Rightarrow A$ $D \Rightarrow B$	ABCD ABCD ABCD ABCD ABCD AB AB AC AD BC BC BD A B C C D	AB AC AD BC BD A B C D	$A \Rightarrow B$ $B \Rightarrow A$ $B \Rightarrow C$ $C \Rightarrow B$ $D \Rightarrow A$ $D \Rightarrow B$

Από τον παραπάνω Πίνακα, βλέπουμε ότι και οι τεις αλγόριθμοι κατορθώνουν να αποκρύψουν το ευαίσθητο στοιχειοσύνολο ABD (αφού δεν είναι συχνό). Επίσης, όλα τα υπόλοιπα μη ευαίσθητα στοιχειοσύνολα παραμένουν συχνά και στις τρεις αναδομημένες βάσεις. Όμως, σε σχέση με τον τελικό στόχο της απόκρυψης κανόνων, είναι εμφανές ότι ο αλγόριθμος *increase_itemset-k+2_support*, είναι σαφώς πιο επιτυχημένος από τους άλλους δύο, αφού κατορθώνει να μην αποκρύψει κανέναν από τους μη ευαίσθητους κανόνες, παρά μόνον όσους είναι αναπόφευκτο να κρυφθούν, λόγω του ότι παράγονται από το ευαίσθητο στοιχειοσύνολο ABD που θέλαμε και πετύχαμε να κρύψουμε. Από όλα τα παραπάνω, είναι μάλλον εμφανές ότι το ερώτημα «Ποιος αλγόριθμος είναι ο καλύτερος;», πρέπει να διατυπωθεί πιο σωστά στο «Ποιος αλγόριθμος είναι ο καλύτερος για συγκεκριμένες συνθήκες και περιστάσεις;».

Προσπαθώντας λοιπόν να απαντήσουμε σ' αυτό ακριβώς το ερώτημα, δοκιμάσαμε τους τρεις αλγορίθμους σε αρκετά σύνολα συναλλαγών, στα οποία θέταμε κάθε φορά διαφορετικούς

ευαίσθητους κανόνες συσχέτισης οι οποίοι δεν έπρεπε να εξορυχθούν από την αναδομημένη βάση δεδομένων και κάτω από το ίδια κατώφλια υποστήριξης και εμπιστοσύνης. Μετά από όλες αυτές τις δοκιμές, καταλήξαμε στα παρακάτω συμπεράσματα:

1. Με τον πρώτο αλγόριθμο *increase_itemset-k_support*, δημιουργούνται επιπλέον συναλλαγές για κάθε fail που συναντάμε (τόσες όση είναι η απόλυτη τιμή της πληθικότητας f του στοιχειοσυνόλου στο οποίο συμβαίνει fail). Έτσι η αναδομημένη βάση δεδομένων που θα προκύψει στο τέλος, αποτελείται (στις περισσότερες περιπτώσεις) από αρκετά περισσότερες συναλλαγές απ' ότι στους δύο άλλους αλγορίθμους.

   Αυτό έχει σαν αποτέλεσμα να προκύπτουν αρκετά μη συχνά στοιχειοσύνολα από όλα αυτά που δεν είναι ευαίσθητα (αφού η συχνότητά τους μπορεί να πέσει κάτω από το κατώφλι λόγω των αυξημένων συναλλαγών). Συνεπώς, αρκετά από τα μη ευαίσθητα στοιχειοσύνολα μπορεί να μην εμφανίζονται ως συχνά, με αποτέλεσμα να κρύβονται αρκετοί ή πολλοί μη ευαίσθητοι κανόνες.

2. Τα ίδια προβλήματα παρατηρούνται και με το δεύτερο αλγόριθμο *decrease_itemset-k+1_support*, όχι όμως τόσο έντονα. Εδώ έχουμε συνήθως λιγότερες συναλλαγές στην νέα βάση δεδομένων (απ' ότι στην βάση που επιστρέφει ο πρώτος αλγόριθμος), αλλά αφού μειώνουμε την μέτρηση υποστήριξης (*support_count*) κάποιων υπερσυνόλων, ενδέχεται η *συχνότητα* (*frequency)* αυτών να πέσει κάτω από το κατώφλι.

3. Με τον τρίτο αλγόριθμο *increase_itemset-k+2_support*, αυξάνουμε την μέτρηση υποστήριξης ενός στοιχειοσυνόλου το οποίο είναι δύο τάξεις παραπάνω από εκείνο το στοιχειοσύνολο στο οποίο συνέβη fail (αυξάνουμε δηλαδή την μέτρηση υποστήριξης ενός υπερσυνόλου) και επιπλέον ενδέχεται να αυξηθεί και η μέτρηση υποστήριξης κάποιων υποσυνόλων αυτού που του αυξήσαμε την μέτρηση υποστήριξης (για να μην είναι μικρότερη από αυτήν). Έτσι ενδέχεται να προκύψουν στην νέα βάση δεδομένων κάποια συχνά στοιχειοσύνολα από αυτά που δεν ήταν στην αρχική. Συνεπώς, μπορεί από την αναδομημένη βάση δεδομένων να προκύπτουν πολλοί *ghost rules*.

Βλέποντας λοιπόν και πειραματικά, ότι το μεγαλύτερο πρόβλημα είναι η απώλεια αρκετών μη ευαίσθητων στοιχειοσυνόλων και κατ' επέκταση αρκετών μη ευαίσθητων κανόνων συσχέτισης, προτείνουμε την παρακάτω προσέγγιση για τη βελτίωση της αναδομημένης βάσης δεδομένων, η οποία μπορεί να εφαρμοσθεί για οποιανδήποτε αναδομημένη βάση, δηλαδή για αναδομημένη

βάση που προέκυψε από οποιονδήποτε από τους τρεις αλγορίθμους που προτείναμε στην υποενότητα 3.1.4.

Βελτιωση Αναδομημενης Βασης

1. Ταξινομούμε την αναδομημένη βάση με τέτοιον τρόπο ώστε οι συναλλαγές να είναι: i) σε αύξουσα σειρά ως προς τον αριθμό των αντικειμένων (items) που συμμετέχουν σ' αυτές και ii) σε φθίνουσα σειρά ως προς την μέτρηση υποστήριξης. Δηλαδή προς τα επάνω θα είναι οι συναλλαγές με τον μικρότερο αριθμό αντικειμένων και όσο προχωράμε προς τα κάτω θα συναντάμε συναλλαγές με περισσότερα αντικείμενα. Μεταξύ συναλλαγών με ίσο αριθμό αντικειμένων, προς τα επάνω θα γράφονται οι συναλλαγές με την μεγαλύτερη μέτρηση υποστήριξης και προς τα κάτω αυτές με την μικρότερη. Σκοπός της ταξινόμησης αυτής, είναι να έχουμε στο βήμα 6, διαγραφές συναλλαγών (και επομένως στοιχειοσυνόλων) με όσο το δυνατόν λιγότερα αντικείμενα, έτσι ώστε να «χάσουμε» όσο το δυνατόν λιγότερους κανόνες (ο αριθμός παραγόμενων κανόνων από ένα στοιχειοσύνολο, είναι ανάλογος με τον αριθμό αντικειμένων k του στοιχειοσυνόλου. Για την ακρίβεια $rules = 2^k - 2$). Επίσης, προτιμούμε να διαγράφουμε συναλλαγές που εμφανίζονται όσο το δυνατόν περισσότερες φορές (μεγάλη μέτρηση υποστήριξης), έτσι ώστε η διαγραφή τους να μην διαγράφει αναγκαστικά και τα αντικείμενα της συναλλαγής (αφού ακριβώς αυτά θα υπάρχουν και σε άλλες συναλλαγές).

2. Δίνουμε τα ευαίσθητα στοιχειοσύνολα (ή την μέτρηση υποστήριξης αυτών).

3. Βρίσκουμε το ευαίσθητο στοιχειοσύνολο με τη μέγιστη δυνατή μέτρηση υποστήριξης στην αναδομημένη βάση, ώστε να μην βγαίνει συχνό.

4. Βρίσκουμε την αμέσως επόμενη μέτρηση υποστήριξης από την παραπάνω (προσθέτοντας 1), που αντιστοιχεί προφανώς στην ελάχιστη μέτρηση υποστήριξης των μη ευαίσθητων στοιχειοσυνόλων, ώστε αυτά να είναι συχνά.

5. Βρίσκουμε την απαιτούμενη συχνότητα γι αυτήν την μέτρηση υποστήριξης, έτσι ώστε να είναι συχνά αυτά τα στοιχειοσύνολα.

6. Αφαιρούμε από την αναδομημένη βάση τόσες συναλλαγές ώστε να παραμείνουν οι συναλλαγές που απαιτούνται για να προκύπτει η συχνότητα που βρήκαμε στο βήμα 4, εκτελώντας τον Αλγόριθμο που ακολουθεί:

Αλγόριθμος 3.5 *– Βελτίωση Αναδομημένης Βάσης*

```
1:  function improve_reconstruction (Reconstructed database D0 , not sensitive itemsets minimum support
count  min_sup , frequent itemsets minimum frequency freq)
2:        transactions ← min_sup / freq          /* Number of transactions in new improved database */
3:        erased ← transactions(D0) – transactions    /* how many  transactions will be erased from D0 */
4:        while (erased > 0)
5:              D0 ← remove (D0)                  /* erase the 1st transaction from D0 */
6:              erased ← erased - 1
7:        end while
8:        return(D0)
9:  end function
```

3.2 Δεύτερος Βασικός Αλγόριθμος *(hide item)*

3.2.1 Θεωρητική Ανάλυση

Το 2011 στην εργασία που παρουσίασαν οι Yogendra Kumar Jain, Vinod Kumar Yadav, Geetika S. Panday [08], έδωσαν την δική τους εκδοχή (τον δικό τους αλγόριθμο) για την Απόκρυψη Κανόνων Συσχέτισης.

Πριν συνεχίσουμε με τον δεύτερο αλγόριθμο που μελετήσαμε, θα σημειώσουμε ότι επειδή αυτός εστιάζει στην απόκρυψη ευαίσθητου αντικειμένου, τον ονομάζουμε *hide_item*. Στη συνέχεια αυτής της διατριβής λοιπόν, όταν αναφέρουμε *αλγόριθμος hide_item*, αναφερόμαστε στον αλγόριθμο των Jain, Yadav καιPanday.

Στην παρουσίαση του αλγορίθμου τους, οι συγγραφείς αρχικά παρατηρούν τα εξής:

Για να είναι δυνατή η απόκρυψη ενός κανόνα συσχέτισης $X \Rightarrow Y$, μπορούμε να μειώσουμε είτε την υποστήριξή του είτε την εμπιστοσύνη του, ώστε να είναι μικρότερες από τα καθορισμένα κατώφλια υποστήριξης και εμπιστοσύνης (MST και MCT αντίστοιχα).

1. Για να μπορέσουμε να μειώσουμε την εμπιστοσύνη ενός κανόνα μπορούμε:

    1.1. Να αυξήσουμε την μέτρηση υποστήριξης στοιχειοσυνόλου X σ(X), χωρίς όμως να αυξηθεί η μέτρηση υποστήριξης του στοιχειοσυνόλου $X \cup Y$, σ($X \cup Y$).

1.2. Να μειώσουμε την μέτρηση υποστήριξης του στοιχειοσυνόλου $X \cup Y$, $\sigma(X \cup Y)$. Σ' αυτήν την περίπτωση, η μείωση μόνο της μέτρησης υποστήριξης του στοιχειοσυνόλου Y, σ(Y), θα έχει σαν αποτέλεσμα τη μείωση της εμπιστοσύνης, γρηγορότερα απ' ότι αν μειώσουμε την μέτρηση υποστήριξης του $X \cup Y$, $\sigma(X \cup Y)$.

2. Για να μπορέσουμε να μειώσουμε την μέτρηση υποστήριξης ενός στοιχειοσυνόλου σε μία βάση δεδομένων *δυαδικής μορφής*, όπου για ένα σύνολο αντικειμένων (*items*) κάθε συναλλαγή είναι μία ακολουθία από 0 και 1 (για κάθε αντικείμενο, το 0 υποδηλώνει ότι η συναλλαγή δεν περιέχει το συγκεκριμένο αντικείμενο ενώ το 1 σημαίνει ότι η συναλλαγή περιλαμβάνει το συγκεκριμένο αντικείμενο), αρκεί να τροποποιήσουμε την ύπαρξη στη συναλλαγή, ενός αντικειμένου τη φορά. Για να το πετύχουμε αυτό, αλλάζουμε σε μία επιλεγμένη συναλλαγή τα 0 σε 1 και τα 1 σε 0, για κάποιο συγκεκριμένο αντικείμενο.

Βασισμένοι στις παραπάνω αρχές, οι συγγραφείς, προτείνουν έναν νέο αλγόριθμο για την Απόκρυψη Κανόνων Συσχέτισης και πιο συγκεκριμένα για την απόκρυψη ευαίσθητων αντικειμένων (*items*) στους κανόνες συσχέτισης. Στον προτεινόμενο αλγόριθμο, ένας κανόνας συσχέτισης $X \Rightarrow Y$ αποκρύπτεται στην διαδικασία εξόρυξης της αναδομημένης βάσης, μειώνοντας την τιμή της μέτρησης υποστήριξης του στοιχειοσυνόλου $X \cup Y$, $\sigma(X \cup Y)$ και αυξάνοντας την μέτρηση υποστήριξης του στοιχειοσυνόλου X, σ(X). Αυτό θα έχει σαν συνέπεια, την αύξηση της υποστήριξης του LHS (αριστερό μέρος) και μείωση της υποστήριξης του RHS (δεξί μέρος) του κανόνα συσχέτισης. Ο αλγόριθμος αρχικά προσπαθεί να αποκρύψει τους κανόνες στους οποίους το προς απόκρυψη αντικείμενο (π.χ. A) είναι στο δεξί μέρος του κανόνα και στη συνέχεια προσπαθεί να αποκρύψει τους κανόνες στους οποίους το A είναι στο αριστερό μέρος.

3.2.2 Παρουσίαση Αλγορίθμου hide_item

Στον αλγόριθμο Αναδόμησης Βάσης Δεδομένων hide_item, t είναι μία συναλλαγή, T είναι ένα σύνολο συναλλαγών, R χρησιμοποιείται για τον κανόνα, *RHS(R)* είναι το δεξί μέρος του κανόνα R, *LHS(R)* είναι το αριστερό μέρος του κανόνα R, *confidence(R)* είναι η εμπιστοσύνη του κανόνα R και *H* είναι το σύνολο των προς απόκρυψη αντικειμένων (*items*).

Ακολουθεί ο αλγόριθμος όπως ακριβώς παρουσιάζεται στην εργασία των Jain, Yadav και Panday.

Αλγόριθμος 3.6 *- Αλγόριθμος Jain, Yadav, Panday*

INPUT: A source database D, a minimum support min_support (MST), a minimum confidence min_confidence (MCT), a set of hiden items X

OUTPUT: The sanitized database D, where rules containing X on Left Hand Side (LHS) or Right Hand Side (RHS) will be hidden

1. Begin
2. Generate all possible rules from given items X
3. Compute confidence of all the rules for each hidden item H, compute confidence of rule R
4. For each rule R in which H is in RHS
4.1 if confidence(R) < MCT then
4.2 go to next 2-itemset
4.3 else go to step 5
5. Decrease support_count of RHS item H
5.1 Find T=t in D fully support R
5.2 while (T is not empty)
5.3 choose the first transaction t from T
5.4 modify t by putting 0 instead of 1 for RHS item
5.5 remove and save the first transaction t from T; end while
6. compute confidence(R)
7. if T is empty, then H cannot be hidden
8. For each rule R in which H is in LHS
9. increase support of LHS
10. find T=t in D | t does not support R
11. while (T is not empty)
12. modify t by putting 1 instead of 0 for LHS item
13. remove and save the first transaction t from T; end while
14. compute confidence(R)
15. if T is empty then H cannot be hidden
end for;
end else;
end for;
16. Ouput update D, as the transformed D

ΠΑΡΑΔΕΙΓΜΑ 3.6

Έστω ότι έχουμε το παρακάτω σύνολο συναλλαγών: D4 = {ABC, ABC, ABC, AB, A, AC}, Θεωρούμε επίσης ότι κατώφλι υποστήριξης MST = 0.3 και κατώφλι εμπιστοσύνης MCT = 0.7. Το σύνολο συναλλαγών σε δυαδική μορφή φαίνεται στον Πίνακα 3.11. Οι μετρήσεις υποστήριξης για κάθε στοιχειοσύνολο του D4, δίνονται στον Πίνακα 3.12

Πίνακας 3.11: Σύνολο Συναλλαγών D4

Συναλλαγή	ABC
T1	111
T2	111
T3	111
T4	110
T5	100
T6	101

Πίνακας 3.12:Μέτρηση Υποστήριξης Στοιχειοσυνόλων Συνόλου Συναλλαγών D4

Στοιχειοσύνολο	Μέτρηση Υποστήριξης	
ABC	3	
AB	4	
AC	4	
BC	3	
A	6	
B	4	
C	4	

Όλοι οι πιθανοί κανόνες συσχέτισης είναι οι παρακάτω (εντός παρένθεσης είναι οι τιμές της εμπιστοσύνης):

$A \Rightarrow B$ (0.667) $A \Rightarrow C$ (0.667) $B \Rightarrow A$ (1) $B \Rightarrow C$ (0.75) $C \Rightarrow A$ (1) $C \Rightarrow B$ (0.75)

Υποθέτουμε ότι στην αρχή θέλουμε να κρύψουμε το στοιχείο A. Πρώτα παίρνουμε έναν από τους κανόνες στους οποίους το στοιχείο A, είναι στο δεξί μέρος του κανόνα. Αυτοί οι κανόνες είναι οι: $B \Rightarrow A$ και $C \Rightarrow A$, που και οι δύο έχουν εμπιστοσύνη μεγαλύτερη από το MCT. Πρώτα παίρνουμε τον κανόνα $B \Rightarrow A$ και ψάχνουμε για τις συναλλαγές όπου εμφανίζονται και τα δύο στοιχειοσύνολα του κανόνα B και A. Αυτές είναι οι τέσσερις πρώτες συναλλαγές T1, T2, T3, T4. Στη συνέχεια τροποποιούμε και τις τέσσερις προαναφερόμενες συναλλαγές θέτοντας 0 για το στοιχείο A. Το νέο σύνολο συναλλαγών D, που προκύπτει φαίνεται στον Πίνακα 3.13, ενώ οι νέες μετρήσεις υποστήριξης των στοιχειοσυνόλων δίνονται στον Πίνακα 3.14:

Πίνακας 3.13: Σύνολο Συναλλαγών D

Συναλλαγή	ABC
T1	011
T2	011
T3	011
T4	010
T5	100
T6	101

Πίνακας 3.14: Μέτρηση Υποστήριξης Στοιχειοσυνόλων Συνόλου Συναλλαγών D

Στοιχειο-σύνολο	Μέτρηση Υποστήριξης
ABC	0
AB	0
AC	1
BC	3
A	2
B	4
C	4

Η εμπιστοσύνη του κανόνα B ⇒ A, τώρα είναι 0 (κάτω από το MCT) και επομένως ο κανόνας αυτός θα κρυφθεί.

Στη συνέχεια παίρνουμε τον κανόνα C ⇒ A και για τις συναλλαγές όπου εμφανίζονται και τα δύο στοιχειοσύνολα του κανόνα C και A. Από τον τροποποιημένο πίνακα που δίνεται αμέσως πιο πάνω, βλέπουμε ότι μόνο η συναλλαγή T6, περιλαμβάνει και το C και το A. Τροποποιούμε αυτήν τη συναλλαγή βάζοντας 0 αντί για 1, για το στοιχείο A.

Ακολούθως, παίρνουμε τους κανόνες όπου το A είναι στο αριστερό μέρος. Αυτοί είναι οι κανόνες A ⇒ B και A ⇒ C, αλλά και οι δύο έχουν εμπιστοσύνη μικρότερη από MCT και επομένως δεν υπάρχει ανάγκη για απόκρυψη αυτών των κανόνων. Έτσι, ο Πίνακας 3.13 δείχνει την αναδομημένη βάση μετά την απόκρυψη του αντικειμένου A.

3.2.3 Προβλήματα του Αλγορίθμου hide_item

1. Τόσο από τη θεωρητική παρουσίαση του Αλγορίθμου *hide_item*, όσο και από τον ίδιο τον αλγόριθμο και από το Παράδειγμα 3.6, φαίνεται ότι προκειμένου να μειώσει την υποστήριξη ενός στοιχειοσυνόλου Y που βρίσκεται στο δεξί μέρος του κανόνα (RHS), ο αλγόριθμος το διαγράφει από όλες τις συναλλαγές στις οποίες συμμετέχει μαζί με το στοιχειοσύνολο *X* του αριστερού μέρους του κανόνα (LHS) (βλ. βήματα 5.1-5.5). Έτσι, η μέτρηση υποστήριξης $\sigma(X \cup Y)$ του στοιχειοσυνόλου $X \cup Y$ και κατά συνέπεια η υποστήριξή του $s(X \cup Y)$, αλλά και η εμπιστοσύνη $c(X \cup Y)$ του κανόνα $X \Rightarrow Y$, γίνονται ίσες με 0.

   Όμως, για να μείνει κρυμμένος ο κανόνας που θέλουμε, αρκεί η εμπιστοσύνη του να είναι κάτω από το κατώφλι εμπιστοσύνης MCT (και όχι απαραίτητα 0). Επομένως, η υποστήριξη του στοιχειοσυνόλου $X \cup Y$ και προφανώς η μέτρηση υποστήριξής του, αρκεί να μειωθούν μέχρι μία συγκεκριμένη τιμή και όχι μέχρι το 0. Με άλλα λόγια, αρκεί να διαγράψουμε το συγκεκριμένο στοιχειοσύνολο Y από ορισμένες μόνο συναλλαγές που συμμετέχει μαζί με το στοιχειοσύνολο X, και όχι από όλες. Σημειώνουμε ότι η διαγραφή του στοιχειοσυνόλου Y, από συγκεκριμένες συναλλαγές, μειώνει προφανώς όχι μόνο την υποστήριξη του $X \cup Y$ αλλά την υποστήριξη και άλλων στοιχειοσυνόλων (π.χ. $X \cup Y \cup Z$) και αναπόφευκτα αυξάνεται και η πληθικότητα κάποιων στοιχειοσυνόλων (π.χ. $X \cup Z$). Συνεπώς, όσο περισσότερες είναι οι συναλλαγές από τις οποίες διαγράφεται το Y, τόσο περισσότερα θα είναι οι *παρενέργειες* στην αναδομημένη βάση.

Σύμφωνα με τα παραπάνω λοιπόν, κάθε φορά που διαγράφουμε το στοιχειοσύνολο Y από κάποια συναλλαγή, πρέπει να υπολογίζουμε την εμπιστοσύνη του κανόνα $X \Rightarrow Y$ και να ελέγχουμε αν αυτή έχει πέσει κάτω από το κατώφλι εμπιστοσύνης MCT. Αν συμβαίνει κάτι τέτοιο, σταματούμε τη διαγραφή του στοιχειοσυνόλου Y από τις υπόλοιπες συναλλαγές.

Δεν θα πρέπει να ξεχνάμε όμως κι ένα άλλο πολύ σημαντικό γεγονός. Καθώς διαγράφουμε το στοιχειοσύνολο Y από μία συναλλαγή και κατά συνέπεια μειώνεται η μέτρηση υποστήριξης του στοιχειοσυνόλου $X \cup Y$, $\sigma(X \cup Y)$, όπως εξάλλου ήταν ο στόχος μας, ενδέχεται αυτή η μείωση να οδηγήσει σε πτώση της υποστήριξης $s(X \cup Y)$, κάτω από το κατώφλι MST. Σ' αυτήν την περίπτωση ο κανόνας δεν μπορεί να αποκαλυφθεί, ανεξάρτητα από την τιμή της εμπιστοσύνης του. Ενδέχεται μάλιστα, η τιμή της υποστήριξης του κανόνα να είναι κάτω από το κατώφλι υποστήριξης MST, ενώ η εμπιστοσύνη του κανόνα να είναι πάνω από το κατώφλι εμπιστοσύνης MCT (όσο πιο μεγάλο είναι το κατώφλι υποστήριξης MST, τόσο πιο πιθανό είναι να συμβεί κάτι τέτοιο).

Επομένως, κάθε φορά που διαγράφουμε το στοιχειοσύνολο Y από κάποια συναλλαγή, πρέπει να υπολογίζουμε και την υποστήριξη του κανόνα $X \Rightarrow Y$, δηλαδή την υποστήριξη του στοιχειοσυνόλου $X \cup Y$, $s(X \cup Y)$ και να ελέγχουμε αν αυτή έχει πέσει κάτω από το κατώφλι υποστήριξης MST. Αν συμβαίνει κάτι τέτοιο, σταματούμε τη διαγραφή του στοιχειοσυνόλου Y από τις υπόλοιπες συναλλαγές.

2. Το ίδιο ακριβώς πρόβλημα συναντούμε και στο μέρος εκείνο του αλγορίθμου, όπου προσπαθεί να αποκρύψει τους κανόνες στους οποίους το ευαίσθητο στοιχειοσύνολο Y, βρίσκεται στο αριστερό μέρος κάποιου κανόνα (π.χ. $Y \Rightarrow X$) (βλ. βήματα 10-13). Όπως αναφέρθηκε και στην θεωρητική ανάλυση του αλγορίθμου στην υπο-ενότητα 3.2.1, ο αλγόριθμος αυξάνει την υποστήριξη του Y χωρίς να αυξήσει την υποστήριξη του X. Για να το πετύχει όμως αυτό, εισάγει το Y σε όλες τις συναλλαγές στις οποίες δεν συμμετέχει το X. Και εδώ, δεν υπάρχει λόγος για την εισαγωγή του Y σε όλες τις συναλλαγές, αλλά μόνο σε όσες χρειάζεται (μέχρι δηλαδή η εμπιστοσύνη του αντίστοιχου κανόνα να πέσει κάτω από το κατώφλι εμπιστοσύνης MCT).

   Έτσι, κάθε φορά που εισάγουμε το στοιχειοσύνολο Y σε κάποια συναλλαγή, πρέπει να υπολογίζουμε εκ νέου την εμπιστοσύνη του κανόνα $Y \Rightarrow X$, να ελέγχουμε αν έχει πέσει κάτω από το κατώφλι εμπιστοσύνης και ανάλογα να σταματούμε ή να συνεχίζουμε τη διαδικασία.

Στη συνέχεια δίνουμε τον τροποποιημένο Αλγόριθμο.

Αλγόριθμος 3.7 *– Τροποποίηση Αλγορίθμου hide_item*

```
1:  function controlled_hide_item (Original database D, minimum support  min_sup, minimum confidence
min_conf, set of hidden items X)
2:        Generate all possible rules from given items X
3:        Compute confidence of all the rules for each hidden item H, compute confidence of rule R
4:        for each rule R in which H is in RHS do
5:              if (support(R) > min_sup && confidence(R) ≥ min_conf)  then
6:                    T ←{t_i |t_i  fully support R}
7:                    for each t ∈T do
8:                          t ←remove(T)
9 :                         modify t in D, by putting 0 instead of 1 for RHS item
10:                         if (support(R) < min_sup OR confidence(R) < min_conf) then
11:                               break
12:                         end if
13:                   end for
14:             end if
15:       end for
16:       for each rule R in which H is in LHS do
17:             if (confidence(R) ≥ min_conf)  then
18:                   T ←{t_i |t_i does not  support R}
19:                   for each t ∈T do
20:                         t ←remove(T)
21 :                        modify t in D, by putting 1 instead of 0 for LHS item
22:                         if (confidence(R) < min_conf)
23:                               break
24:                         end if
25:                   end for
26:             end if
27        end for
28:       return (D)
29:  end function
```

Παραδειγμα 3.7

Για τα δεδομένα του Παραδείγματος 3.6 (σύνολο συναλλαγών D4 = {ABC, ABC, ABC, AB, A, AC} *minsup* = 0.3, *minconf* = 0.7) και αν επιθυμούμε και τώρα την απόκρυψη του αντικειμένου A, μέχρι ενός σημείου (βήμα 9 του Αλγορίθμου 3.7 *controlled_hide_item*), δουλεύουμε όπως και πριν. Δηλαδή, όπως και στο Παράδειγμα 3.6, πρώτα παίρνουμε τον κανόνα B ⇒ A και ψάχνουμε για τις συναλλαγές όπου εμφανίζονται και τα δύο στοιχειοσύνολα του κανόνα B και A. Αυτές είναι οι τεσσερις πρώτες συναλλαγές T1, T2, T3, T4.

Στη συνέχεια τροποποιούμε στη βάση D4 την συναλλαγή T1, θέτοντας 0 για το στοιχείο A. Υπολογίζουμε την υποστήριξη και την εμπιστοσύνη του κανόνα στην τροποποιημένη πλέον βάση D και βρίσκουμε ότι είναι *support(B ⇒ A) = 3/6 = 0.5* και *confidence(B ⇒ A) = 3/4 = 0.75*. Επειδή ούτε η υποστήριξη ούτε η εμπιστοσύνη, έπεσαν κάτω από τα *minsup* και *minconf* αντίστοιχα, τροποποιούμε στην βάση D4, την επόμενη συναλλαγή T2. Τώρα *support(B ⇒ A) = 2/6 = 0.33* και *confidence(B ⇒ A) = 2/4 = 0.5*. Επειδή η εμπιστοσύνη του κανόνα, είναι μικρότερη από το κατώφλι MCT, σταματούμε την τροποποίηση των συναλλαγών. Σ' αυτό το σημείο, το σύνολο των συναλλαγών D4 όπως έχει διαμορφωθεί δίνονται στον Πίνακα 3.15 και οι νέες μετρήσεις υποστήριξης των στοιχειοσυνόλων, δίνονται στον Πίνακα 3.16.

Πίνακας 3.15: Σύνολο Συναλλαγών D4

Συναλλαγή	ABC
T1	011
T2	011
T3	111
T4	110
T5	100
T6	101

Πίνακας 3.16: Μέτρηση Υποστήριξης Στοιχειοσυνόλων Συνόλου Συναλλαγών D4

Στοιχειο-σύνολο	Μέτρηση Υποστήριξης	
ABC	1	
AB	2	
AC	2	
BC	3	
A	4	
B	4	
C	4	

Στη συνέχεια παίρνουμε τον κανόνα C ⇒ A και για τις συναλλαγές όπου εμφανίζονται και τα δύο στοιχειοσύνολα του κανόνα C και A. Από τον τροποποιημένο πίνακα που δίνεται αμέσως πιο πάνω, βλέπουμε ότι δύο συναλλαγές περιλαμβάνουν και το C και το A και συγκεκριμένα οι T4, T6. Υπολογίζοντας τις νέες τιμές για την υποστήριξη και την εμπιστοσύνη του κανόνα, βρίσκουμε ότι *support(C ⇒ A) = 2/6 = 0.33* και *confidence(C ⇒ A) = 2/4 = 0.5* Επομένως, αφού η εμπιστοσύνη είναι μικρότερη από το κατώφλι MCT, δεν τροποποιούμε καμία από τις συναλλαγές.

Για τους κανόνες όπου το A είναι στο αριστερό μέρος, (A ⇒ B και A ⇒ C), ισχύουν ότι και στο Παράδειγμα 3.6 (και οι δύο έχουν εμπιστοσύνη μικρότερη από MCT και επομένως δεν υπάρχει ανάγκη για απόκρυψη αυτών των κανόνων). Έτσι, ο Πίνακας 3.13 δείχνει την αναδομημένη βάση D4 μετά την απόκρυψη του αντικειμένου A.

Κεφάλαιο 4
Μέτρηση Αποτελεσματικότητας Αλγορίθμων

Σ' αυτό το Κεφάλαιο ασχολούμαστε με την αποτελεσματικότητα των διαφόρων αλγορίθμων. Αφού αναφέρουμε στην ενότητα 4.1 τι σημαίνει και πώς ορίζεται η αποτελεσματικότητα, στην ενότητα 4.2 παρουσιάζουμε έναν αλγόριθμο μέτρησης αυτής της αποτελεσματικότητας και δίνουμε ένα Παράδειγμα. Στην ενότητα 4.3 τέλος, κάνουμε μία συγκριτική μελέτη σχετικά με την επίδοση των τριών αλγορίθμων που προτείναμε και παρουσιάσαμε στο 3.1.4.

4.1 Αποτελεσματικότητα Αλγορίθμων Αναδόμησης Βάσης Δεδομένων

Όπως ήδη έχει αναφερθεί πολλοί αλγόριθμοι έχουν προταθεί για την Απόκρυψη των Κανόνων Συσχέτισης. Κανένας από αυτούς όμως δεν μπορεί να πετύχει για οποιαδήποτε σύνολο δεδομένων, την ολοκληρωτική απόκρυψη των ευαίσθητων κανόνων με ταυτόχρονη απουσία των παρενεργειών (side effects).

Η αποτελεσματικότητα λοιπόν των διαφόρων αλγορίθμων, έγκειται ακριβώς στο κατά πόσο καταφέρνουν να επιτύχουν τους στόχους της Απόκρυψης Κανόνων Συσχέτισης (βλ. ενότητα 2.2). Έτσι, αν *D* μία βάση δεδομένων συναλλαγών, R το σύνολο των κανόνων που παράγονται κάτω από συγκεκριμένες τιμές των minsup και minconf, R_s το σύνολο των ευαίσθητων κανόνων και *D'* η αναδομημένη βάση, ο αλγόριθμος θα είναι απόλυτα επιτυχημένος αν από την βάση *D'* και κάτω από τις ίδιες τιμές των *minsup* και *minconf*:

1. Δεν θα παράγεται κανένας κανόνας από αυτούς που ανήκουν στο σύνολο R_s

2. Θα παράγονται όλοι οι κανόνες που ανήκουν στο σύνολο R-R_s

3. Δεν θα παράγεται κανένας κανόνας που δεν ανήκει στο R

ή με άλλα λόγια, όταν το σύνολο κανόνων *R'* που παραγονται από την *D'*, ταυτίζεται με το σύνολο R-R_s $(R' \equiv R\text{-}R_s)$.

4.2 Αλγόριθμος Μέτρησης Αποτελεσματικότητας

4.2.1 Θεωρητική Ανάλυση

Με βάση αυτά που αναφέρθηκαν παραπάνω, μπορούμε να πούμε ότι η αποτελεσματικότητα των αλγορίθμων Αναδόμησης της Βάσης Δεδομένων, μπορεί να απεικονισθεί ως ποσοστό επιτυχίας των τριών επί μέρους στόχων. Το ιδανικό ποσοστό των ευαίσθητων κανόνων που κρύβονται (δηλαδή δεν παράγονται) είναι 100%, το ιδανικό ποσοστό των μη ευαίσθητων κανόνων που δεν κρύβονται (δηλαδή παράγονται) είναι 100% και το ιδανικό ποσοστό των *ghost rules* που δεν παράγονται είναι 0%.

Έτσι, αν από την αρχική βάση δεδομένων *D* και την αναδομημένη βάση δεδομένων *D'*, πάρουμε τα σύνολα παραγόμενων κανόνων *R* και *R'* αντίστοιχα, εφαρμόζοντας έναν οποιονδήποτε κατάλληλο αλγόριθμο (π.χ. Apriori), μπορούμε στη συνέχεια να υπολογίσουμε πολύ εύκολα τα τρία παραπάνω ποσοστά (ευαίσθητων κανόνων, μη ευαίσθητων κανόνων και *ghost rules*) και να έχουμε την αποτελεσματικότητα του αλγορίθμου.

4.2.2 Παρουσίαση Αλγορίθμου

Αλγόριθμος 4.1 *– Μέτρηση Αποτελεσματικότητας Αλγορίθμων*

```
1:  function algorithm_efficiency (original database total rules R , reconstructed database total rules R', sensitive rules R_S)
2:      R_NS ← R – R_S
3:      sr ← 0
4:      nsr ← 0
5:      for each r_1 ∈ R_s do
6:          for each r_2 ∈ R' do
7:              if (r_2 ≡ r_1) then
8:                  sr ← sr + 1
9 :                 remove r_2 from R'
10:                 break
11:             end if
12:         end for
13:     end for
14:     for each r_1 ∈ R_NS do
15:         for each r_2 ∈ R' do
16:             if (r_2 ≡ r_1) then
17:                 nsr ← nsr + 1
18:                 remove r_2 from R'
19:                 break
20:             end if
21 :        end for
22:     end for
23:     gr ← |R'| - sr - nsr
24:     print (sr / |R_s|)
25:     print (nsr / |R_NS|
26:     print (gr / |R'|
27: end function
```

Ο αλγόριθμος, αφού υπολογίσει το σύνολο των μη ευαίσθητων κανόνων R_{NS}, θέτει 0 στις τιμές των μεταβλητών *sr* και *nsr* που αντιστοιχούν στους ευαίσθητους και μη ευαίσθητους κανόνες που αποκαλύπτονται (παράγονται), στην αναδομημένη βάση *D'*.

Στη συνέχεια αναζητά αν υπάρχουν ευαίσθητοι κανόνες στο σύνολο των παραγομένων κανόνων *R'* της νέας βάσης *D'* (βήματα 5-13). Κάθε φορά που ανακαλύπτει έναν ευαίσθητο κανόνα στο *R'*, αυξάνει την τιμή της μεταβλητής *sr* κατά 1.

Μετά, αναζητά τους μη ευαίσθητους κανόνες στο σύνολο των παραγομένων κανόνων *R'* της νέας βάσης *D'* (βήματα 14-22). Κάθε φορά που ανακαλύπτει έναν μη ευαίσθητο κανόνα στο R', αυξάνει την τιμή της μεταβλητής *nsr* κατά 1.

Τέλος, υπολογίζει τους *ghost rules* που έχουν παραχθεί και εκτυπώνει τα τρία ποσοστά που αναφέραμε πιο πάνω.

ΠΑΡΑΔΕΙΓΜΑ 4.1

Παίρνουμε ξανά τα δεδομένα που είχαμε δώσει Παράδειγμα 3.5. Εκεί είχαμε λοιπόν το σύνολο δεκαπέντε συναλλαγών D3={ABCD, ABCD, ABCD, ABC, ABD, ABD, AB, AC, AC, AD, BC, BC, BC, BD, CD}, minsup = 0.3 και minconf = 0.6. Οι παραγόμενοι κανόνες ήταν συνολικά δέκα:

$A \Rightarrow B$, $B \Rightarrow A$, $B \Rightarrow C$, $C \Rightarrow B$, $D \Rightarrow A$, $D \Rightarrow B$, $D \Rightarrow AB$, $AB \Rightarrow D$, $AD \Rightarrow B$, $BD \Rightarrow A$

Θεωρήσαμε σαν ευαίσθητο τον κανόνα $D \Rightarrow AB$ μειώσαμε την μέτρηση υποστήριξής του από 5 σε 4, και εκτελέσαμε τους τρεις αλγορίθμους *increase_itemset-k_support*, *decrease_itemset-k+1_support*, *increase_itemset-k+2_support*. Από τις βάσεις που μας επέστρεψαν, εξορύχθηκαν οι κανόνες που φαίνονται στον Πίνακα 4.1.

Πίνακας 4.1: Κανόνες συσχέτισης απ' τις αναδομημένες βάσεις που παράγονται από τους τρεις αλγορίθμους

Αλγόριθμος	Κανόνες Συσχέτισης
increase_itemset-k_support	$A \Rightarrow B$ $C \Rightarrow B$ $D \Rightarrow A$ $D \Rightarrow B$
decrease_itemset-k+1_support	$B \Rightarrow C$ $C \Rightarrow B$ $D \Rightarrow A$ $D \Rightarrow B$
increase_itemset-k+2_support	$A \Rightarrow B$ $B \Rightarrow A$ $B \Rightarrow C$ $C \Rightarrow B$ $D \Rightarrow A$ $D \Rightarrow B$

Αν εκτελέσουμε τον Αλγόριθμο μέτρησης αποτελεσματικότητας για τον καθένα από τους τρεις παραπάνω αλγορίθμους , παίρνουμε τα αποτελέσματα που φαίνονται στον Πίνακα 4.2.

Πίνακας 4.2: Αποτελεσματικότητα αλγορίθμων για ένα ευαίσθητο κανόνα του συνόλου συναλλαγών D3

Αλγόριθμος	Senstive Hidden Rules	Not Sensitive Revealed Rules	Ghost Rules
increase_itemset-k_support	100%	44,44%	0%
decrease_itemset-k+1_support	100%	44,44%	0%
increase_itemset-k+2_support	100%	66,67%	0%

4.3 Πειράματα – Συγκριτική Μελέτη των Αλγορίθμων *increase_itemset-k_support, decrease_itemset-k+1_support, increase_itemset-k+2_support*

Στην ενότητα αυτή θα παρουσιάσουμε τα αποτέλεσματα των μετρήσεων της αποτελεσματικότητας των αλγορίθμων *increase_itemset-k_support*, *decrease_itemset-k+1_support* και *increase_itemset-k+2_support.*

Οι μετρήσεις μας αυτές, έγιναν με δύο διαφορετικούς τρόπους: i) με βάση τα ποσοστά των ευαίσθητων κανόνων αλλά και των *hidden rules*, των *lost rules* και των *ghost rules* ii) με βάση τις απόλυτες τιμές των ευαίσθητων κανόνων των hidden rules, των lost rules και των ghost rules.

Με τον όρο *hidden rules* εννοούμε τους ευαίσθητους κανόνες που παραμένουν κρυφοί στην αναδομημένη βάση (το ιδανικό ποσοστό είναι 100% και η ιδανική απόλυτη τιμή είναι ίση με τον αριθμό των ευαίσθητων κανόνων). Με τον όρο *lost rules* εννοούμε τους μη ευαίσθητους κανόνες οι οποίοι δεν αποκαλύπτονται στην αναδομημένη βάση (το ιδανικό ποσοστό είναι 0% και η ιδανικλή απόλυτη τιμή είναι 0) και με τον όρο *ghost rules* δίνονται εκείνοι οι κανόνες που δεν υπήρχαν στην αρχική βάση αλλά εμφανίζονται στην αναδομημένη (το ιδανικό ποσοστό είναι 0% και η ιδανική απόλυτη τιμή 0)).

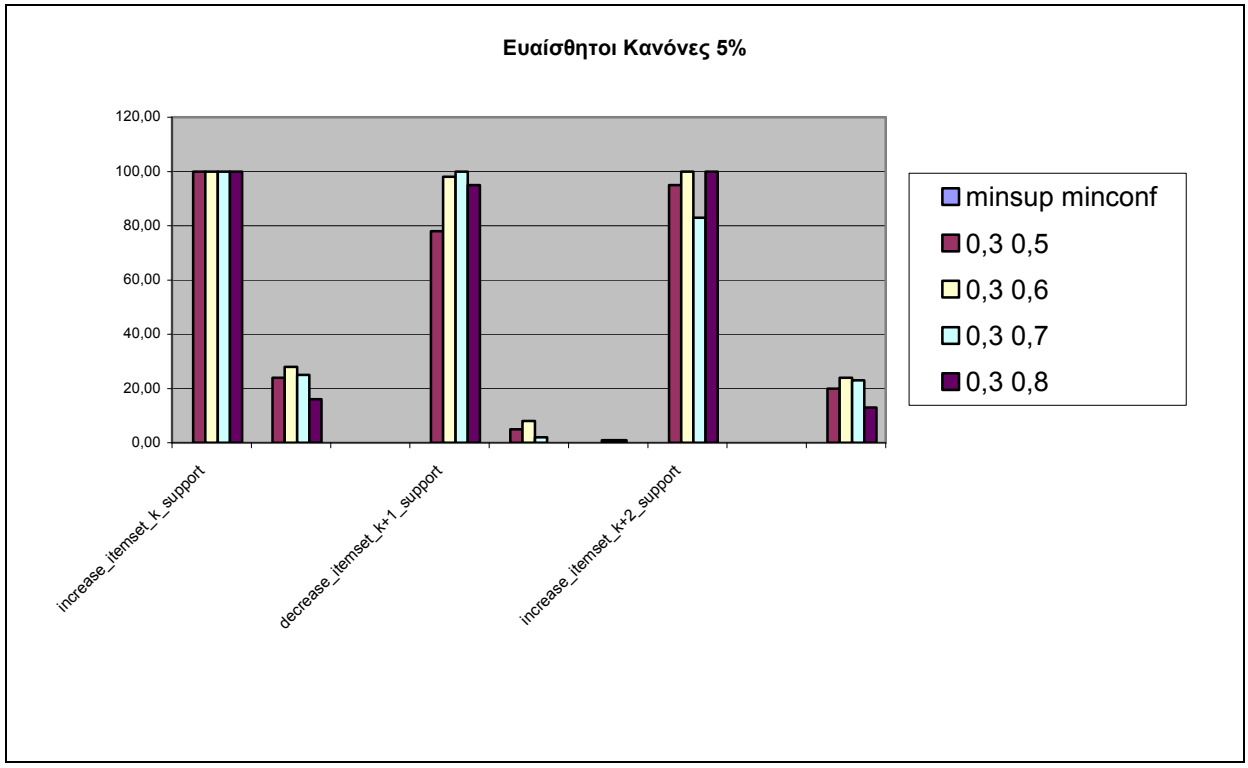
4.3.1 Μέτρηση με βάση τα ποσοστά

Για αυτή τη μέτρηση αποτελεσματικότητας, οι αλγόριθμοι δοκιμάσθηκαν σε μία μικρή βάση δεδομένων 40 συναλλαγών και 10 αντικειμένων, την οποία κατασκευάσαμε για το σκοπό αυτό. Πρόκειται για μία βάση της οποίας τα δεδομένα αναπαρίστανται σε δυαδική μορφή (κάθε αντικείμενο παίρνει σε κάθε συναλλαγή την τιμή 0 ή 1). Τα αποτελέσματα που πήραμε δίνονται συνοπτικά στα επόμενα διαγράμματα. Οι αλγόριθμοι δοκιμάσθηκαν για συγκεκριμένο ποσοστό ευαίσθητων κανόνων 5% οι οποίοι επιλέγονταν τυχαία από το σύνολο των παραγόμενων κανόνων (π.χ. αν από την αρχική βάση είχαν παραχθεί 40 κανόνες συσχέτισης τότε θα είχαμε δύο ευαίσθητους κανόνες που θα επιλέγαμε τυχαία).

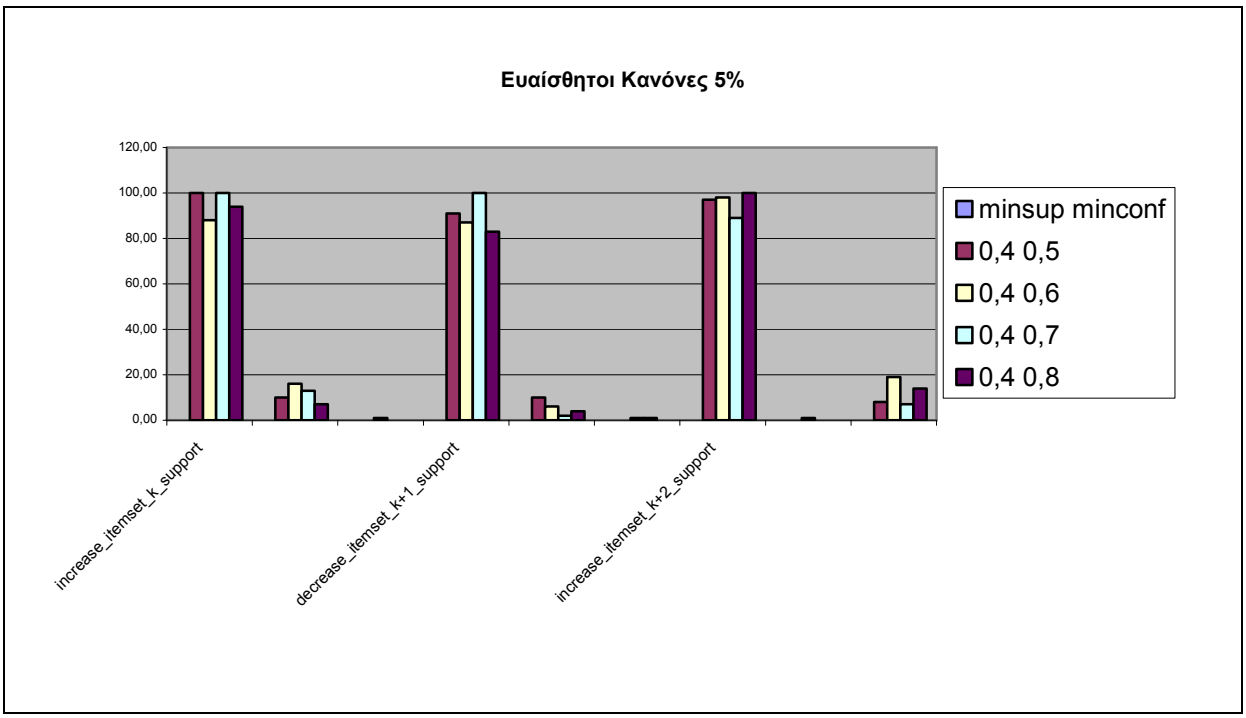
Πρέπει τέλος να σημειωθεί ότι οι μετρήσεις που αποτυπώνονται στα διαγράμματα (για όλους τους προαναφερόμενους κανόνες) είναι ποσοστά και όχι απόλυτοι αριθμοί.

Αποτελέσματα - Διαγράμματα

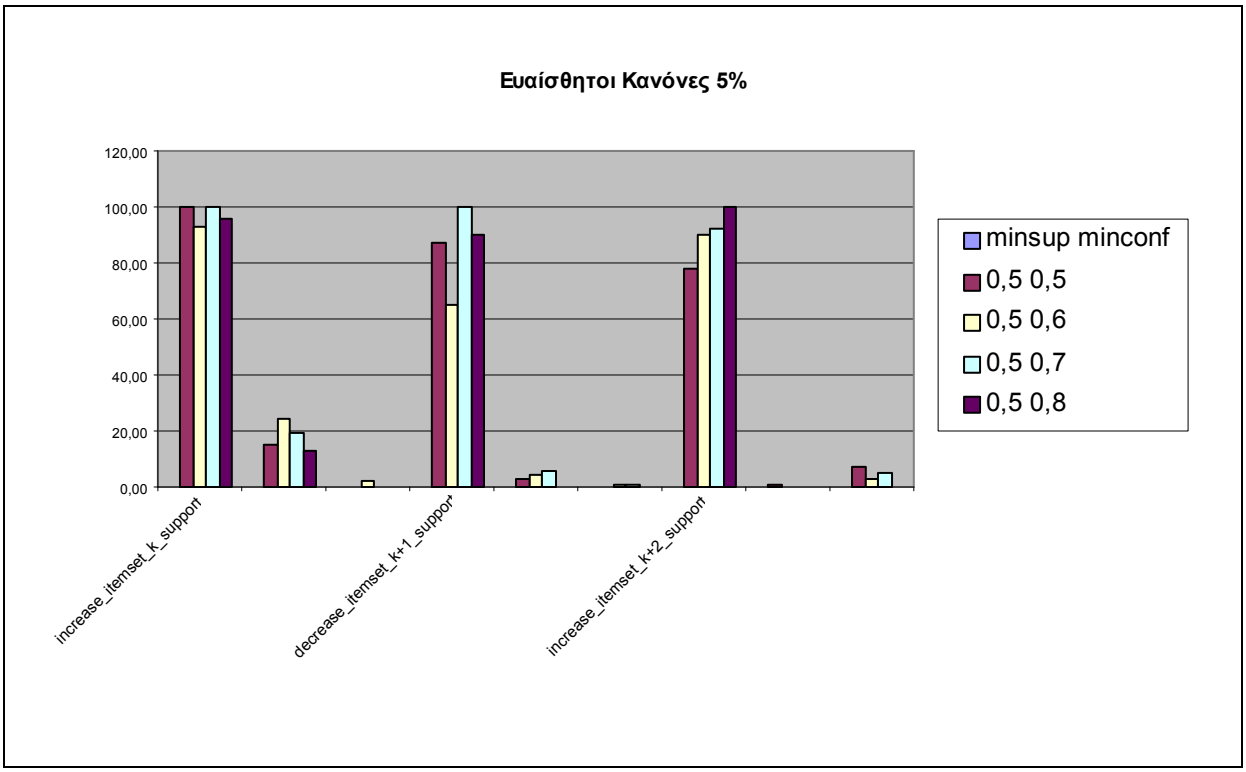
Στο Σχήμα 4.1 φαίνονται τα αποτελέσματα για *minsup = 0.3* και για τέσσερις τιμές της *minconf*(0.5,0.6,0.7,0.8) στο Σχήμα 4.2 *για minsup = 0.4* και τις ίδιες τιμές της *minconf* και στο Σχήμα 4.3 για *minsup=0.5* και ίδιες τιμές της *minconf*.



Σχήμα 4.1: Αποτελεσματικότητα (*hidden rules, lost rules, ghost rules*) σε μορφή ποσοστού των αλγορίθμων *increase_itemset-k_support, decrease_itemset-k+1_support, increase_itemset-k+2_support, για διαφορετικές τιμές minsup και minconf*



Σχήμα 4.2: Αποτελεσματικότητα *(hidden rules, lost rules, ghost rules)* σε μορφή ποσοστού των αλγορίθμων *increase_itemset-k_support, decrease_itemset-k+1_support, increase_itemset-k+2_support, για διαφορετικές τιμές minsup και minconf*



Σχήμα 4.3: Αποτελεσματικότητα *(hidden rules, lost rules, ghost rules)* σε μορφή ποσοστού των αλγορίθμων *increase_itemset-k_support, decrease_itemset-k+1_support, increase_itemset-k+2_support, για διαφορετικές τιμές minsup και minconf*

4.3.2 Μέτρηση με βάση απόλυτους αριθμούς

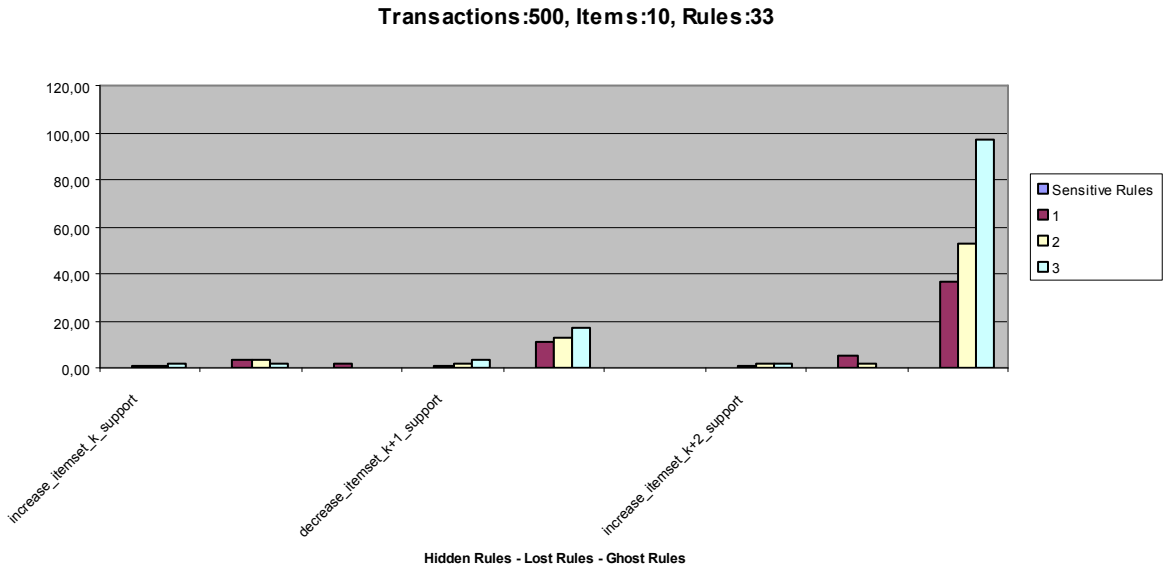
Για τη μέτρηση αυτή, οι αλγόριθμοι δοκιμάσθηκαν σε διαφορετικές βάσεις δεδομένων 500 και 1.000 συναλλαγών και 6, 10 και 13 αντικειμένων τις οποίες κατασκευάσαμε για το σκοπό αυτό. Πρόκειται για βάσεις της οποίας τα δεδομένα αναπαρίστανται σε δυαδική μορφή (κάθε αντικείμενο παίρνει σε κάθε συναλλαγή την τιμή 0 ή 1). Τα αποτελέσματα που πήραμε δίνονται συνοπτικά στα επόμενα διαγράμματα. Οι αλγόριθμοι δοκιμάσθηκαν για αριθμό ευαίσθητων κανόνων από 1 μέχρι και 6 οι οποίοι επιλέγονταν τυχαία από το σύνολο των παραγόμενων κανόνων.

Οι μετρήσεις που αποτυπώνονται στα διαγράμματα (για όλους τους προαναφερόμενους κανόνες) είναι προφανώς απόλυτοι αριθμοί.

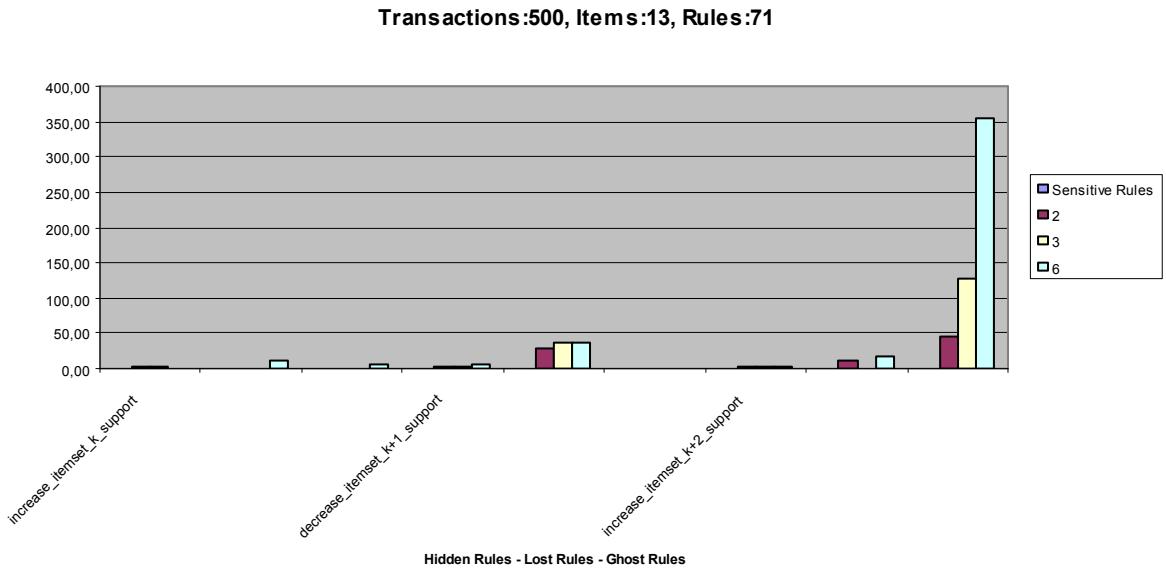
Αποτελέσματα - Διαγράμματα

Στο Σχήμα 4.4, στο Σχήμα 4.5, στο Σχήμα 4.6, στο Σχήμα 4.7, στο Σχήμα 4.8 και στο Σχήμα 4.9 φαίνονται τα αποτελέσματα για διάφορες συναλλαγές, αντικείμενα και ευαίσθητους κανόνες .

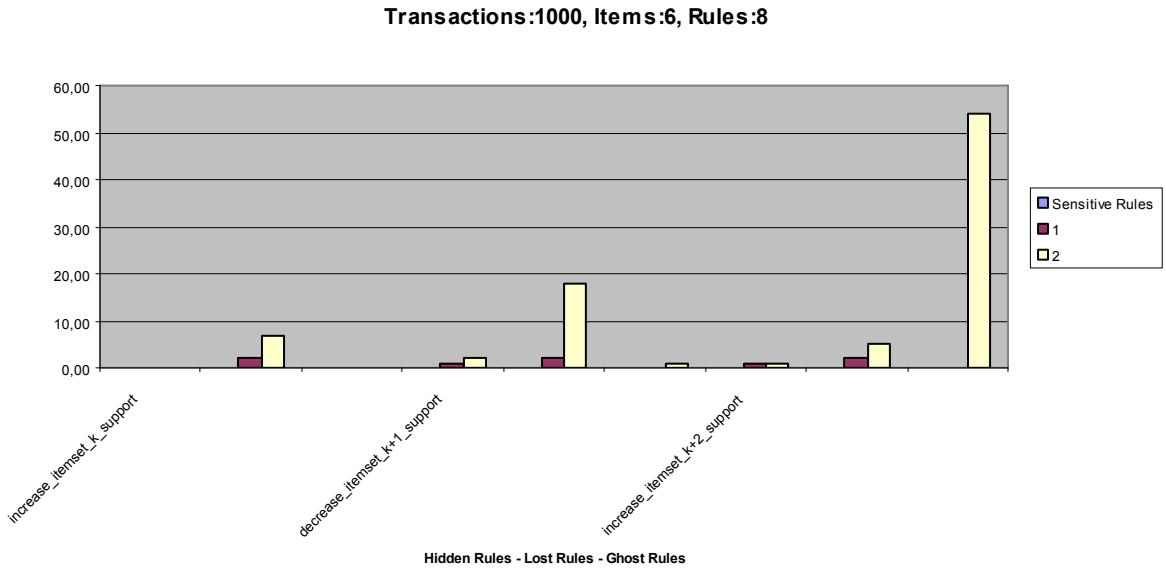
Σχήμα 4.4 Αποτελεσματικότητα *(hidden rules, lost rules, ghost rules)* σε απόλυτους αριθμούς των αλγορίθμων *increase_itemset-k_support, decrease_itemset-k+1_support, increase_itemset-k+2_support* για 500 συναλλαγές και 6 αντικείμενα



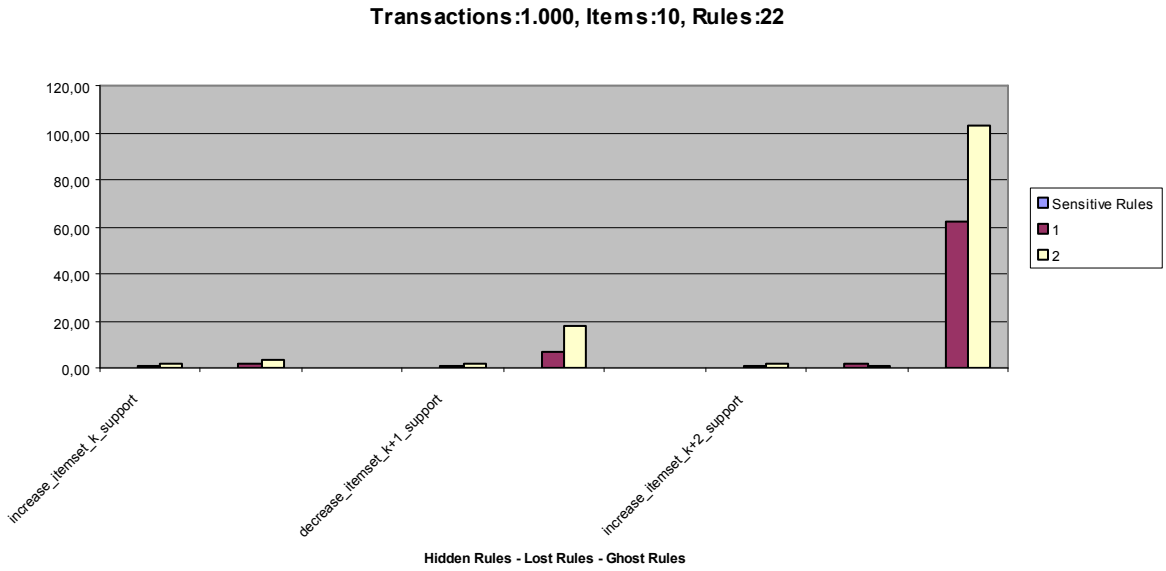
Σχήμα 4.5 Αποτελεσματικότητα *(hidden rules, lost rules, ghost rules)* σε απόλυτους αριθμούς των αλγορίθμων *increase_itemset-k_support, decrease_itemset-k+1_support, increase_itemset-k+2_support* για 500 συναλλαγές και 10 αντικείμενα



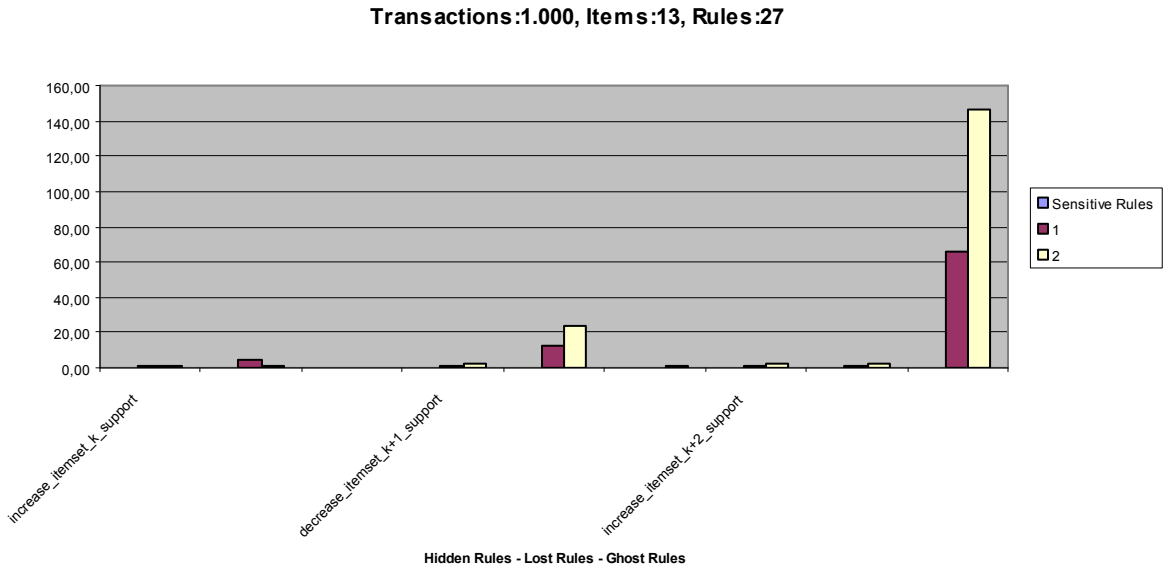
Σχήμα 4.6 Αποτελεσματικότητα *(hidden rules, lost rules, ghost rules)* σε απόλυτους αριθμούς των αλγορίθμων *increase_itemset-k_support, decrease_itemset-k+1_support, increase_itemset-k+2_support* για 500 συναλλαγές και 13 αντικείμενα.



Σχήμα 4.7 Αποτελεσματικότητα *(hidden rules, lost rules, ghost rules)* σε απόλυτους αριθμούς των αλγορίθμων *increase_itemset-k_support, decrease_itemset-k+1_support, increase_itemset-k+2_support* για 1.000 συναλλαγές και 6 αντικείμενα.



Σχήμα 4.8 Αποτελεσματικότητα *(hidden rules, lost rules, ghost rules)* σε απόλυτους αριθμούς των αλγορίθμων *increase_itemset-k_support, decrease_itemset-k+1_support, increase_itemset-k+2_support* για 1.000 συναλλαγές και 10 αντικείμενα.



Σχήμα 4.9 Αποτελεσματικότητα *(hidden rules, lost rules, ghost rules)* σε απόλυτους αριθμούς των αλγορίθμων *increase_itemset-k_support, decrease_itemset-k+1_support, increase_itemset-k+2_support* για 1.000 συναλλαγές και 13 αντικείμενα.

4.3.3 Συμπεράσματα

Από την συγκριτική μελέτη των παραπάνω αποτελεσμάτων μπορούμε να καταλήξουμε στα εξής:

Για την πρώτη σειρά πειραμάτων όπου η βάση δεδομένων ήταν αρκετά μικρή ως προς τον αριθμό συναλλαγών, ισχύουν τα παρακάτω:

Από τους τρεις αλγορίθμους, οι *increase_itemset-k_support, increase_itemset-k+2_support* πετυχαίνουν σχεδόν εξ΄ισου στον βασικό στόχο, που είναι η απόκρυψη των ευαίσθητων κανόνων. Αντίθετα, ο *decrease_itemset-k+1_support* φαίνεται ότι δεν τα καταφέρνει τόσο καλά, αφού ειδικά όταν οι τιμές *minsup* και *minconf* είναι παραπλήσιες, το ποσοστό των ευαίσθητων κανόνων που μένουν κρυφοί, είναι μάλλον χαμηλό.

Σχετικά με τις *παρενέργειες*, ο *increase_itemset-k_support* «χάνει» πολλούς μη ευαίσθητους κανόνες, ενώ δεν παράγει σχεδόν καθόλου *ghost rules*. Ο *decrease_itemset-k+1_support*, «χάνει» λιγότερους μη ευαίσθητους κανόνες, αλλά το ποσοστό δεν είναι αμελητέο και αναφορικά με τους *ghost rules* τα πάει εξ' ίσου καλά με τον *increase_itemset-k_support*. Με τον *increase_itemset-k+2_support* τέλος, δεν χάνεται σχεδόν κανένας μη ευαίσθητος κανόνας, αλλά παράγονται

υπερβολικά πολλοί *ghost rules*. Αυτό είναι κάτι που το περιμέναμε, αφού σε κάθε fail ενός στοιχειοσυνόλου—k, ο *increase_itemset-k+2_support* παράγει και προσθέτει στη βάση στοιχειοσύνολα-k+2 τα οποία προφανώς παράγουν πολλούς νέους κανόνες. Έτσι όσο πιο μεγάλης τάξης είναι το ευαίσθητο στοιχειοσύνολο, η παραγωγή *ghost rules*, θα είναι επίσης πολύ μεγάλη.

Συνοπτικά λοιπόν, μπορούμε να πούμε ότι στην απόκρυψη των ευαίσθητων κανόνων υπερτερούν ο *increase_itemset-k_support* και *increase_itemset-k+2_support*, στην εξόρυξη των μη ευαίσθητων κανόνων υπερτερεί σαφώς ο *increase_itemset-k+2_support*, ενώ στην εμφάνιση των *ghost rules*, ο *increase_itemset-k+2_support* δεν τα πάει καθόλου καλά, ειδικά για χαμηλές τιμές υποστήριξης και εμπιστοσύνης όπου έχουμε μεγαλύτερο αριθμό παραγόμενων κανόνων στην αρχική βάση.

Για την επόμενη σειρά πειραμάτων όπου η βάση δεδομένων ήταν αρκετά μεγαλύτερη ως προς τον αριθμό συναλλαγών, ισχύουν τα παρακάτω:

Ο τρίτος αλγόριθμος *increase_itemset-k+2_support_count*, τα καταφέρνει πολύ καλά στους *lost rules*, παράγει όμως υπερβολικά πολλούς *ghost rules*. Επίσης κάποιες φορές αποτυγχάνει να κρύψει όλους τους ευαίσθητους κανόνες.

Ο δεύτερος αλγόριθμος *decrease_itemset-k+1_support_count,* έχει τα καλύτερα αποτελέσματα σχετικά με τους *ghost rules* (παράγει ελάχιστους έως κανέναν) και την απόκρυψη των ευαίσθητων κανόνων (τους κρύβει σχεδόν όλους). Κρύβει όμως και πολλούς μη ευαίσθητους κανόνες (έχει πολλούς *lost rules*).

Τέλος ο 1ος αλγόριθμος *increase_itemset-k_support_count* έχει καλά αποτελέσματα στους *lost rules* (σχεδόν σαν τον τρίτο αλγόριθμο που είδαμε πιο πάνω) και στους *ghost rules*, αλλά και στις κατηγορίες έρχεται δεύτερος. Το μεγάλο μειονέκτημα, είναι ότι αρκετές φορές δεν καταφέρνει να κρύψει όλους τους ευαίσθητους κανόνες που είναι και ο βασικός μας στόχος

Απ' όλα τα παραπάνω, είναι εμφανές ότι δεν μπορούμε να αποφανθούμε σχετικά με το ποιος αλγόριθμος είναι ο πιο αποτελεσματικός. Ανάλογα με τις περιστάσεις και τις συνεπακόλουθες προτεραιότητες που θέτει ο κάτοχος της βάσης δεδομένων, αλλά και το πλήθος των συναλλαγών, μπορεί να επιλεχθεί διαφορετικός αλγόριθμος κάθε φορά.

Στο σημείο αυτό θα θέλαμε να ξανατονίσουμε αυτό που αναφέραμε στην υπο-ενότητα 3.1.5, ότι οι τρεις αλγόριθμοι που αξιολογήσαμε πιο πάνω σχετικά με το πόσο επιτυγχάνουν τους στόχους της Απόκρυψης των Κανόνων Συσχέτισης, προσπαθούν να επιτύχουν την απόκρυψη συχνών

στοιχειοσυνόλων. Συνεπώς θα ήταν ίσως πιο «δίκαιο» να αξιολογηθούν σε σχέση μ' αυτόν τον στόχο.

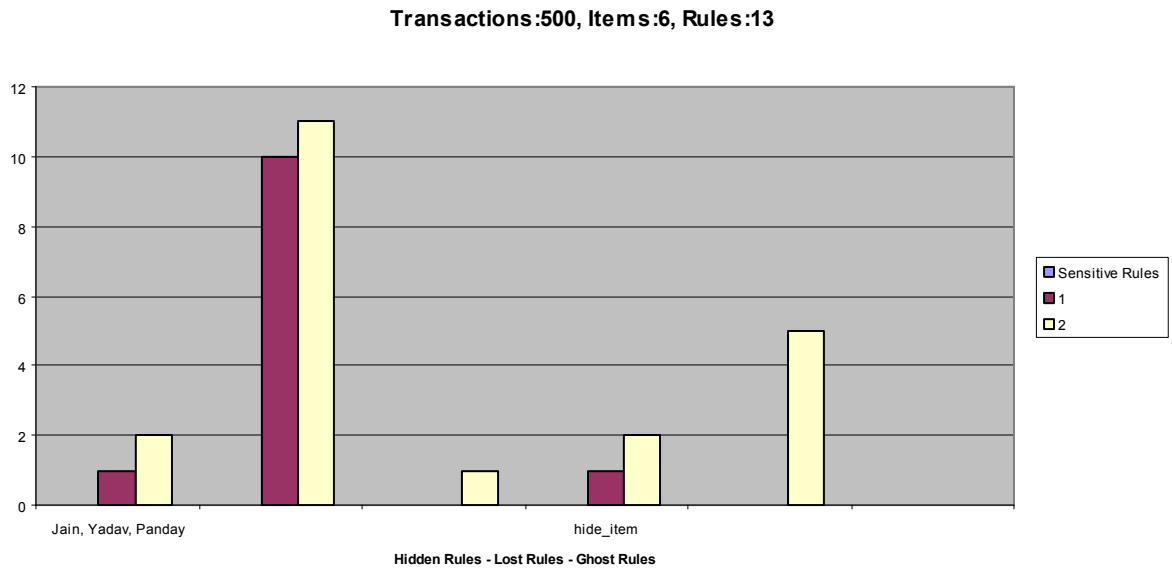
Δεν θα πρέπει να ξεχνάμε επίσης ότι η τελική επίτευξη των τριών στόχων της απόκρυψης κανόνων συσχέτισης, εξαρτάται σε πολύ μεγάλο βαθμό και από τη στρατηγική της απόκρυψης (αν δηλαδή θα μειώσουμε την μέτρηση υποστήριξης του στοιχειοσυνόλου από το οποίο παράγεται ο κανόνας, ή αν θα αυξήσουμε την μέτρηση υποστήριξης του στοιχειοσυνόλου του αριστερού μέρους του κανόνα κλπ). Μπορεί αν αλλάξουμε στρατηγική απόκρυψης να αλλάξει και η αποτελεσματικότητα των αλγορίθμων.

4.4 Πειράματα – Συγκριτική Μελέτη των Αλγορίθμων *hide_item, controlled_hide_item*

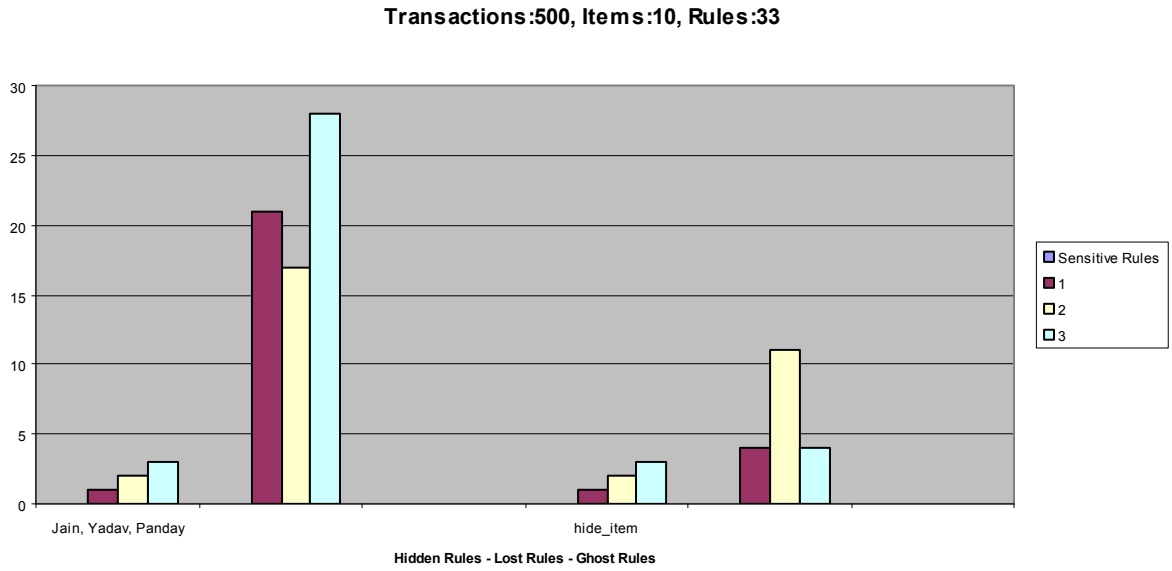
Στην ενότητα αυτή θα παρουσιάσουμε τα αποτέλεσματα των μετρήσεων της αποτελεσματικότητας των αλγορίθμων *hide_item* και *controlled_hide_item*. Για τη μέτρηση της αποτελεσματικότητάς τους οι αλγόριθμοι δοκιμάσθηκαν στις βάσεις δεδομένων που είχαν χρησιμοποιηθεί και για τις μετρήσεις που αναφέρονται στην υπο-ενότητα 4.3.2.

4.4.1 Αποτελέσματα - Διαγράμματα

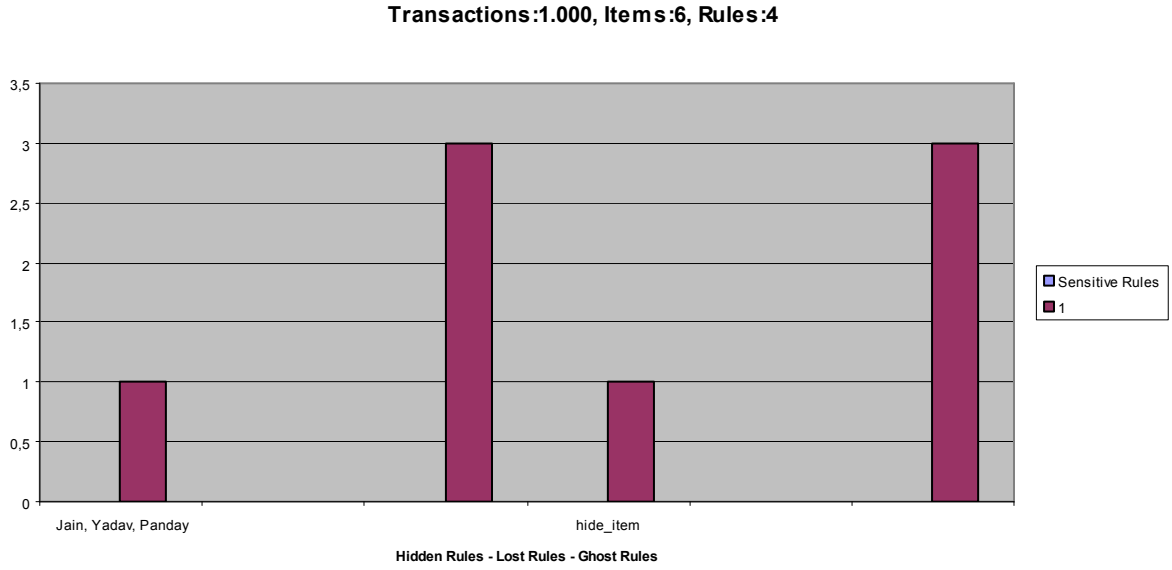
Τα αποτελέσματα που πήραμε από την εκτέλεση των αλγορίθμων, σε διάφορες βάσεις δεδομένων, φαίνονται στα παρακάτω σχήματα: Σχήμα 4.10, Σχήμα 4.11, Σχήμα 4.12, Σχήμα 4.13.



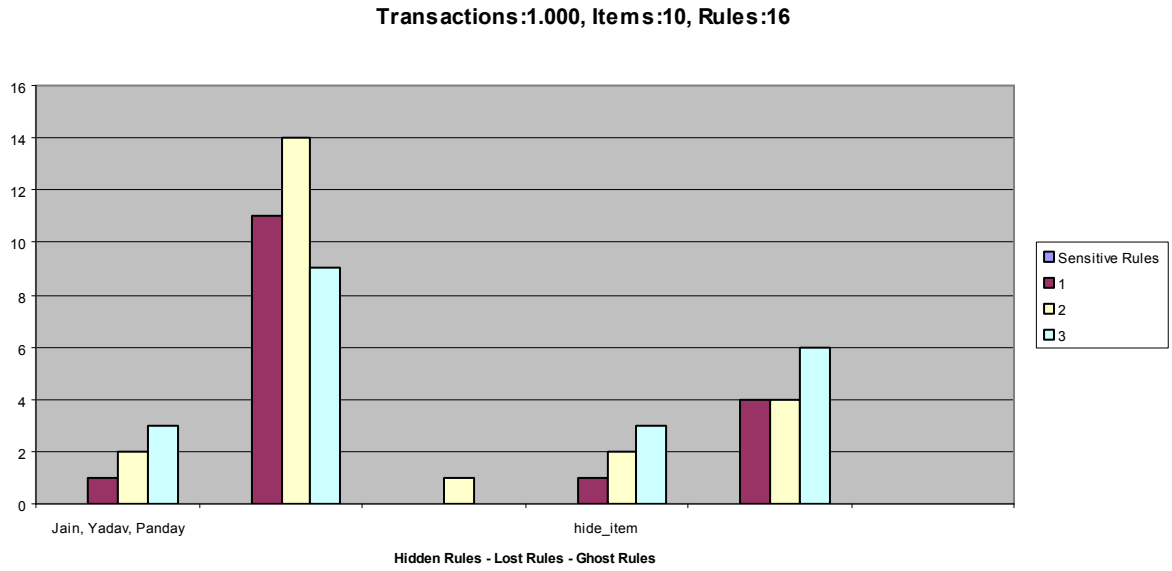
Σχήμα 4.10: Αποτελεσματικότητα *(hidden rules, lost rules, ghost rules)* των αλγορίθμων *hide_item* και *controlled_hide_item* για 500 συναλλαγές και 6 αντικείμενα



Σχήμα 4.11: Αποτελεσματικότητα *(hidden rules, lost rules, ghost rules)* των αλγορίθμων *hide_item* και *controlled_hide_item* για 500 συναλλαγές και 10 αντικείμενα



Σχήμα 4.12: Αποτελεσματικότητα *(hidden rules, lost rules, ghost rules)* των αλγορίθμων *hide_item* και *controlled_hide_item* για 1.000 συναλλαγές και 6 αντικείμενα



Σχήμα 4.13: Αποτελεσματικότητα *(hidden rules, lost rules, ghost rules)* των αλγορίθμων *hide_item* και controlled_hide_item για 1.000 συναλλαγές και 10 αντικείμενα

4.4.2 Συμπεράσματα

Για τους αλγορίθμους *hide_item* και *controlled_hide_item* λοιπόν , μπορούμε να παρατηρήσουμε τα εξής:

1) Και οι δύο αποκρύπτουν όλους (ή σχεδόν όλους) τους ευαίσθητους κανόνες

2) Και οι δύο παράγουν ελάχιστους *ghost rules* (ο *controlled_hide_item* σχεδόν πάντα μηδέν)

3) Ο *controlled_hide_item* υπερτερεί σαφώς στους *lost rules.*

Από τα παραπάνω συνάγεται ότι ο *controlled_hide_item,* είναι εμφανώς πιο αποτελεσματικός αφού αποκρύπτει το ίδιο αποτελεσματικά τους ευαίσθητους κανόνες και έχει λιγότερα *side effects* (κυρίως *lost rules*).

Κεφάλαιο 5
Μελλοντική Έρευνα

Στο παρόν κεφάλαιο δίνουμε κάποιες πιθανές μελλοντικές κατευθύνσεις της έρευνας, σαν αποτέλεσμα της εργασίας μας. Στην ενότητα 5.1 παρουσιάζουμε τα σημεία τα οποία κατά τη γνώμη μας μπορεί να αποτελέσουν αντικείμενο έρευνας στο μέλλον, σε σχέση με τον αλγόριθμο *reconstruction_by_cardinality* και κυρίως με τους τρεις αλγορίθμους *increase_itemset-k_support, decrease_itemset_k+1_support και increase_itemset-k+2_support* που προτείναμε. Αντίστοιχα, στην ενότητα 5.2, αναφέρουμε την ερευνητική περιοχή που μπορεί ν' αναπτυχθεί γύρω από τους αλγορίθμους απόκρυψης αντικειμένων, *hide_item* και *controlled_hide_item*.

6.1 Αλγόριθμος reconstruction_by_cardinality (Περαιτέρω Μελέτη)

- Τόσο ο βασικός αλγόριθμος *reconstruction_by_cardinality,* όσο και οι τροποποιήσεις – βελτιώσεις του, *increase_itemset-k_support, decrease_itemset_k+1_support και increase_itemset-k+2_support*, δέχονται σαν είσοδο το πλέγμα στοιχειοσυνόλων με τις τιμές μέτρησης υποστήριξης. Είναι προφανές λοιπόν, ότι πριν από την εκτέλεση αυτών των αλγορίθμων, πρέπει να υπολογίσουμε το πλέγμα στοιχειοσυνόλων. Το πλέγμα αυτό είναι το δυναμοσύνολο των αντικειμένων που συμμετέχουν στη βάση δεδομένων. Για k

αντικείμενα, το μέγεθος του δυναμοσυνόλου (δηλαδή το πλήθος των δυνατών συνδυασμών των αντικειμένων, οποιουδήποτε μεγέθους από 1 έως και k), είναι 2^k. Επομένως, για τιμές του k από 25-27, η εξεύρεση του πλέγματος είναι πολύ ακριβή υπολογιστικά και για τιμές του k πάνω από 35 η εξεύρεση του πλέγματος είναι αδύνατη.

Για την υπέρβαση αυτού του προβλήματος όταν ο αριθμός των αντικειμένων στη βάση δεδομένων είναι μεγάλος, μπορούμε να εργασθούμε ως εξής:

1) Υπολογίζουμε τον αριθμό των αντικειμένων που συμμετέχουν σε κάθε συναλλαγή

2) Βρίσκουμε τον μέγιστο αριθμό αντικειμένων που συμμετέχουν σε οποιαδήποτε συναλλαγή

3) Υπολογίζουμε το πλέγμα στοιχειοσυνόλων μέχρι αυτόν τον αριθμό

Για παράδειγμα αν έχουμε 25 ατικείμενα, μπορεί ο μέγιστος αριθμός που συμμετέχουν σε οποιαδήποτε συναλλαγή, να είναι 18. Επομένως, αρκεί να βρούμε το πλέμα στοιχειοσυνόλων για μέχρι στοιχειοσύνολα-18.

Θα μπορούσαμε ίσως να περιορίσουμε περαιτέρω το πλέγμα, θέτοντας κάποιους περιορισμούς στις συναλλαγές και στα αντικείμενα που αυτές περιέχουν, βάζοντας ένα αντίστοιχο κατώφλι. Θα υπολογίσουμε δηλαδή, πόσες ακριβώς συναλλαγές περιλαμβάνουν 1 αντικείμενο, πόσες 2 αντικείμενα, πόσες 3 αντικείμενα κλπ (ας τις ονομάσουμε συναλλαγές-1, συναλλαγές-2, συναλλαγές-3 κλπ). Από αυτόν τον πίνακα, θα «κόψουμε» τις συναλλαγές που εμφανίζονται λιγότερες φορές από το κατώφλι που θέσαμε (μπορεί να «κόψουμε» τις συναλλαγές-1, συναλλαγές-7, συναλλαγές-17, συναλλαγές-18). Έτσι τώρα, ο μέγιστος αριθμός αντικειμένων σε οποιαδήποτε συναλλαγή θα είναι 16 και θα υπολογίσουμε το πλέγμα μέχρι στοιχειοσύνολα-16.

Από τα παραπάνω, γίνεται εμφανές ότι ο περιορισμός του αριθμού αντικειμένων για τα οποία πρέπει να βρούμε το πλέγμα στοιχειοσυνόλων, είναι ζωτικής σημασίας για την εξεύρεση του πλέγματος.

Ο περιορισμός λοιπόν του πλέγματος στοιχειοσυνόλων και η εξεύρεση ενός προσεγγιστικού σε σχέση με το πραγματικό, μπορούν ν' αποτελέσουν αντικείμενο μελέτης στο μέλλον

- Ο αλγόριθμος *reconstruction_by_cardinality* δέχεται σαν είσοδο τις μετρήσεις υποστήριξης των στοιχειοσυνόλων. Έτσι, όπως έχουμε ήδη αναφέρει στην ενότητα 3.1.5, η εκτέλεση οποιασδήποτε από τις τρεις τροποποιήσεις που προτείνουμε όταν συμβεί αποτυχία, θα τροποποιήσει κατά πάσα πιθανότητα τον αριθμό συναλλαγών στην αναδομημένη βάση που θα προκύψει. Συνεπώς, ακόμα κι αν στην αναδομημένη βάση ικανοποιούνται οι τιμές μέτρησης υποστήριξης των στοιχειοσυνόλων που δώσαμε σαν είσοδο, ενδέχεται να μην ικανοποιούνται οι τιμές της υποστήριξης των στοιχειοσυνόλων οι οποίες εξαρτώνται και από το πλήθος των συναλλαγών (βλ. σχέση 2.2).

  Αυτό θα έχει σαν συνέπεια, είτε κάποια στοιχειοσύνολα στην αναδομημένη βάση να παραμένουν συχνά ενώ επιθυμούμε να είναι μη συχνά, είτε το αντίθετο, δηλαδή κάποια στοιχειοσύνολα να προκύπτουν μη συχνά, ενώ θα θέλαμε να είναι συχνά. Οι παραπάνω ανεπιθύμητες καταστάσεις οδηγούν σε μη απόκρυψη των ευαίσθητων κανόνων και σε εμφάνιση *lost rules* και *ghost rules,* αντίστοιχα.

  Ορισμένα από τα παραπάνω προβλήματα μπορούν ενδεχομένως να περιοριστούν, αν ο βασικός αλγόριθμος *reconstruction_by_cardinality* και οι τρεις τροποποιήσεις του που προτείναμε, δέχονται σαν είσοδο *την υποστήριξη (συχνότητα)* των στοιχειοσυνόλων και όχι την μέτρηση υποστήριξής τους.

  Στην ίδια ενότητα 3.1.5, προτείναμε κάποια τροποποίηση που κινείται προς αυτήν την κατεύθυνση, αλλά η προσέγγισή μας ήταν ευριστική και οπωσδήποτε η διερεύνηση του παραπάνω προβλήματος, μπορεί να αποτελέσει αντικείμενο μελλοντικής έρευνας.

- Τέλος, για την άρση της αποτυχίας του αλγορίθμου *reconstruction_by_cardinality*, προτείναμε τρεις διαφορετικούς αλγορίθμους των οποίων μετρήσαμε την αποτελεσματικότητα, χωρίς όμως να διερευνήσουμε αν υπάρχει κάποια σχέση μεταξύ του πλέγματος στοιχειοσυνόλων και του αλγορίθμου που πρέπει να ακολουθηθεί για την άρση της αποτυχίας (ποιος δηλαδή είναι πιο κατάλληλος για το συγκεκριμένο πλέγμα και το συγκεκριμένο σημείο στο οποίο συνέβη αποτυχία).

  Έτσι, για διαδοχικές αποτυχίες σε ένα πλέγμα στοιχειοσυνόλων, ακολουθούσαμε πάντα τον ίδιο αλγόριθμο για άρση αυτών των αποτυχιών. Αν όμως υπάρχει κάποια σχέση που συνδέει το πλέγμα στοιχειοσυνόλων, τις μετρήσεις υποστήριξης και τον αλγόριθμο που πρέπει να ακολουθήσουμε για να ξεπεράσουμε την εμφανιζόμενη αποτυχία, τότε είναι

προφανές ότι θα οδηγηθούμε σε μία αναδομημένη βάση η οποία θα επιτυγχάνει καλύτερα τους στόχους της Απόκρυψης Κανόνων Συσχέτισης.

Η ανακάλυψη μιας τέτοιας σχέσης (αν υπάρχει), είναι σίγουρα ένα θέμα που μπορεί να διερευνηθεί στο μέλλον.

6.2 Αλγόριθμος hide_item (Περαιτέρω Μελέτη)

Ο αλγόριθμος *hide_item* και ο βελτιωμένος *controlled_hide_item* που προτείναμε, αποκρύπτουν ευαίσθητα αντικείμενα που υφίστανται μέσα στον κανόνα συσχέτισης. Μπορεί να διερευνηθεί το κατά πόσο οι παραπάνω αλγόριθμοι (με ορισμένες τροποποιήσεις ενδεχομένως) μπορούν να εφαρμοσθούν, ώστε να αποκρύπτουν ευαίσθητα στοιχειοσύνολα μέσα από τον κανόνα συσχέτισης. Και αυτό το ερώτημα λοιπόν, μπορεί να αποτελέσει πεδίο μελλοντικής έρευνας.

Κεφάλαιο 6
Επίλογος

Στην παρούσα διατριβή, ασχοληθήκαμε με το πρόβλημα της Απόκρυψης Κανόνων Συσχέτισης. Διερευνήσαμε την λύση αυτού του προβλήματος, μελετώντας τις μεθοδολογίες που βασίζονται στην αναδόμηση της βάσης δεδομένων. Εστιάσαμε σε δύο συγκεκριμένες τεχνικές, τις αναλύσαμε, εντοπίσαμε τα προβλήματα που παρουσιάζουν και προτείναμε πέντε συνολικά αλγορίθμους (ο τέταρτος προσπαθεί ευριστικά να βελτιώσει τους πρώτους τρεις), με σκοπό την επίλυση των προβλημάτων και τη συνολική βελτίωση αυτών των τεχνικών.

Οι αλγόριθμοι που προτείναμε, παρουσιάζουν κι αυτοί τα δικά τους προβλήματα και αδυναμίες. Κατά τη γνώμη μας, τα προβλήματα αυτά οφείλονται κυρίως σε δύο λόγους: i) στην μη ύπαρξη συγκεκριμένης στρατηγικής για την απόκρυψη συγκεκριμένων στοιχειοσυνόλων και ii) στο γεγονός ότι οι συγκεκριμένοι αλγόριθμοι προσεγγίζουν το πρόβλημα της απόκρυψης, λαμβάνοντας υπ' όψιν τους μόνο την μέτρηση υποστήριξης των στοιχειοσυνόλων και όχι την υποστήριξή τους (συχνότητα). Σημειώνουμε ότι προς αυτήν την κατεύθυνση κινείται ο τέταρτος αλγόριθμος βελτιστοποίησης των τριών πρώτων, που αναφέρουμε πιο πάνω.

Πιστεύουμε ότι αν καταφέρουμε να συνδυάσουμε τους αλγορίθμους που προτείναμε, με μία συγκεκριμένη στρατηγική απόκρυψης η οποία θα βασίζεται και στη συχνότητα των

στοιχειοσυνόλων, η αναδομημένη βάση που θα λαμβάνουμε θα είναι πολύ πιο κοντά στην ιδεατή.

Βιβλιογραφία

[01] R. Agrawal and R. Srikant, «Privacy preserving data mining», SIGMOD Record, 29(2):439–450, 2000

[02] A. Amiri, «Dare to Share: Protecting Sensitive Knowledge with Data Sanitization», Decision Support Systems, 43(1):181–191, 2007

[03] M. Atallah, E. Bertino, A. Elmagarmid, M. Ibrahim, and V. S. Verykios, «Disclosure Limitation of Sensitive Rules», In Proceedings of the 1999 IEEE Knowledge and Data Engineering Exchange Workshop (KDEX), pages 45-52, 1999

[04] X. Chen, M. Orlowska, and X. Li, «A New Framework of Privacy Preserving Data Sharing», In Proceedings of the 4th IEEE ICDM Workshop: Privacy and Security Aspects of Data Mining, IEEE Computer Society, pages 47-56, 2004

[05] E. Dasseni, V. S. Verykios, A. K. Elmagarmid, and E. Bertino, «Hiding Association Rules by Using Confidence and Support», In Proceedings of the 4th International Workshop on Information Hiding, pages 369–383, 2001

[06] A. Gkoulalas-Divanis and V. S. Verykios, «Association Rule Hiding for Data Mining», Springer, 2010

[07] Y. Guo, «Reconstruction-Based Asocciation Rule Hiding», In Proceedings of SIGMOD2007 Ph.D. Workshop on Innovative Database Research, 2007 (IDAR 2007), June 2007

[08] Y. K. Jain, V. K. Yadav, and G. S. Panday, «An Efficient Association Rule Hiding Algorithm for Privacy Preserving Data Mining», International Journal on Computer Science and Engineering (IJCSE), Vol. 3 No. 7 July 2011, pages 2792 – 2798, 2011

[09] A. Katsarou, A. Gkoulalas-Divanis, and V. S. Verykios, «Reconstruction–Based Classification Rule Hiding through Controlled Data Modification», In Proceedings of the 5th IFIP Conference on Artificial Intelligence Applications and Innovations (AIAI), 2009

[10] G. Lee, C. Y. Chang and A. L. P. Chen, «Hiding Sensitive Patterns in Association Rules Mining», In Proceedings of the 28th International Computer Software and Applications Conference (COMPSAC), pages 424–429, 2004

[11] Y. Lindell and B. Pinkas, « Privacy preserving data mining», Journal of Cryptology, 15(3):36–54, 2000

[12] N. Matloff , «The Art of R Programming», No Starch Press, 2011

[13] T. Mielikainen, « On Inverse Frequent Set Mining», In Proceedings of the 2nd Workshop on Privacy Preserving Data Mining, pages 18-23, 2003

[14] G. V. Moustakides and V. S. Verykios, «A max–min approach for hiding frequent itemsets», In Workshops Proceedings of the 6th IEEE International Conference on Data Mining (ICDM), pages 502–506, 2006

[15] G. V. Moustakides and V. S. Verykios, «A maxmin approach for hiding frequent itemsets», Data and Knowledge Engineering, 65(1):75–89, 2008

[16] S. R. M. Oliveira, and O. R. Zaïane, «Protecting Sensitive Knowledge by Data Sanitization», In Proceedings of the 3rd IEEE International Conference on Data Mining (ICDM), pages 211–218, 2003

[17] E. D. Pontikakis, A. A. Tsitsonis, and V. S. Verykios, «An Experimental Study of Distortion–based Techniques for Association Rule Hiding», In Proceedings of the 18th Conference on Database Security (DBSEC), pages 325–339, 2004

[18] E. Pontikakis, Y. Theodoridis, A. Tsitsonis, L. Chang, and V. S. Verykios, «A Quantitative and Qualitative Analysis of Blocking in Association Rule Hiding», In Proceedings of the 2004 ACM Workshop on Privacy in the Electronic Society (WPES), pages 29–30, 2004

[19] Y. Saygin, V. S. Verykios, and C. W. Clifton, «Using Unknowns to Prevent Discovery of Association Rules», ACM SIGMOD Record, 30(4):45–54, 2001

[20] Y. Saygin, V. S. Verykios, and A. K. Elmagarmid, «Privacy Preserving Association Rule Mining», In Proceedings of the 2002 International Workshop on Research Issues in Data Engineering: Engineering E–Commerce/E–Business Systems (RIDE), pages 151–163, 2002

[21] P. N. Tan, M. Steinbach, and V. Kumar (Επιμέλεια Μετάφρασης: Βασίλειος Σ. Βερύκιος, Μετάφραση: Σταύρος Σουραβλάς), «Εισαγωγή στην Εξόρυξη Δεδομένων», Εκδόσεις Τζιόλα, Θεσσαλονίκη 2010

[22] W. N. Venables, D. M. Smith and the R development Core Team, «An Introduction to R», Version 2.15.0, 30-03-2012, πηγή Internet http://cran.r-project.org/doc/manuals/R-intro.pdf

[23] V. S. Verykios, A. K. Emagarmid, E. Bertino, Y. Saygin, and E. Dasseni, «Association Rule Hiding», IEEE Transactions on Knowledge and Data Engineering, 16(4):434–447, 2004

[24] S. L.Wang and A. Jafari, «Using Unknowns for Hiding Sensitive Predictive Association Rules», In Proceedings of the 2005 IEEE International Conference on Information Reuse and Integration (IRI), pages 223–228, 2005

[25] S. L.Wang, B. Parikh, and A. Jafari, «Hiding Informative Association Rule Sets», Expert Systems with Applications, 33(2):316–323, 2007

[26] Y. H. Wu, C. M. Chiang, and A. L. P. Chen, «Hiding Sensitive Association Rules with Limited Side Effects», IEEE Transactions on Knowledge and Data Engineering, 19(1):29–42, 2007

Παράρτημα Α
Κώδικας Αλγορίθμων σε R

Αλγόριθμος 3.2

```
increase_itemset-k_support <- function(dataset,support){
        support1 <- matrix()
        support1 <- support
        D <- matrix()
        Q <- matrix()
        S <- matrix()
        C <- matrix()
        c <- 0

        for (i in 1:length(dataset))
                if (support1[i] > 0){
                        k <- length(dataset[[i]])
                        thesi_k <- i
                        break
                }

        repeat{
                for (i in thesi_k:length(dataset)){
                        if (length(dataset[[i]])==k){
                                if (support1[i]>0){
                                        for (j in 1:support1[i]){
                                                c <- c+1
                                                C[c] <- list(dataset[[i]])
                                        }
                                }
                                if (i < length(dataset))
                                        for (j in (i+1):length(dataset)){
                                                if(length(setdiff(dataset[[i]],dataset[[j]])) ==
                                                length(dataset[[i]])- length(dataset[[j]]))
                                                        support1[j] <- support1[j]-support1[i]
                                                if (support1[j]<0){
```

```
                                                support[j] <- support[j]-support1[j]
                                                break
                                        }
                                }
                        }
                if (support1[j] < 0)
                        break
                }
                if (support1[j] < 0)
                        break
                thesi_k<-thesi_k+1
                k <- k-1
                if (k == 0)
                        break
        }
        if (support1[j] < 0)
                increase_itemset-k_support(dataset,support)
        else
                return(reconstr_trans(dataset,C))
}
```

Αλγόριθμος 3.3

```
decrease_itemset-k+1_support <- function(dataset,sup2){
        supp <- matrix()
        supp <- sup2
        D <- matrix()
        Q <- matrix()
        S <- matrix()
        C <- matrix()
        c <- 0
        d <- 0

        for (i in 1:length(dataset))
                if (supp[i] > 0){
                        k <- length(dataset[[i]])
                        thesi_k <- i
                        break
                }

        repeat{
                for (i in thesi_k:length(dataset)){
                        if (length(dataset[[i]])==k){
                                if (supp[i]>0){
                                        for (j in 1:supp[i]){
                                                c <- c+1
                                                C[c] <- list(dataset[[i]])
                                        }
                                }
                                if (i < length(dataset))
                                        for (j in (i+1):length(dataset)){
                                                if(length(setdiff(dataset[[i]],dataset[[j]])) ==
                                                length(dataset[[i]])- length(dataset[[j]]))
                                                        supp[j] <- supp[j]-supp[i]
                                        }
                        }
                        else
                                break
                }
                if (any(supp<0)){
                        for (l in i:length(dataset)){
                                if(supp[l]<0){
                                        d<-d+1
                                        D[d]<-list(dataset[[l]])
                                        S[d]<-supp[l]
                                }
                        }
                        S<-matrix(S,ncol=1)
                        break
                }
                thesi_k <- i
                k <- k-1
                if (k==0)
                        break   #βγαίνω από το repeat
        }

        if (all(supp>=0))
                return(reconstr_trans(dataset,C))
        else{
                for (m in thesi_k:(i-1)){
                        for (n in 1:length(D)){
                                if (length(setdiff(dataset[[m]],D[[n]]))==length(dataset[[m]])-
                                length(D[[n]]) && S[[n]]<0){
                                        meiosi_support <- min(sup2[m],abs(S[n]))
```

```
                                        sup2[m] <- sup2[m]-meiosi_support
                                        S[n] <- S[n]+meiosi_support
                                        if (n<length(D)-1){
                                        for (p in (n+1):length(D))
                                        if (length(setdiff(dataset[[m]],D[[p]]))==length(dataset[[m]])-
                                        length(D[[p]]))
                                        S[p]<-S[p]+meiosi_support
                                        }
                                }
                        }
                        if (all(S>=0))
                                break
                }
                decrease_itemset-k+1_support(dataset,sup2)
        }
}
```

Αλγόριθμος 3.4

```
increase_itemset-k+2_support <- function(dataset,support){
        support <- support_correction(dataset,1,support)
        support1 <- matrix()
        support1 <- support
        D <- matrix()
        Q <- matrix()
        S <- matrix()
        C <- matrix()
        c <- 0
        d <- 0

        for (i in 1:length(dataset))
                if (support1[i] > 0){
                        k <- length(dataset[[i]])
                        thesi_k <- i
                        break
                }

        repeat{
                for (i in thesi_k:length(dataset)){
                        if (length(dataset[[i]])==k){
                                if (support1[i]>0){
                                        for (j in 1:support1[i]){
                                                c <- c+1
                                                C[c] <- list(dataset[[i]])
                                        }
                                }
                                if (i < length(dataset))
                                        for (j in (i+1):length(dataset)){
                                        if(length(setdiff(dataset[[i]],dataset[[j]])) == length(dataset[[i]])-
                                        length(dataset[[j]]))
                                                support1[j] <- support1[j]-support1[i]
                                        }
                        }
                        else
                                break
                }
                if (any(support1<0)){
                        for (l in i:length(dataset)){
                                if(support1[l]<0){
                                        d<-d+1
                                        D[d]<-list(dataset[[l]])
                                        S[d]<-support1[l]
                                }
                        }
                        D <- subset_elimination(D)
                        S <- matrix(S,ncol=1)
                        break
                }
                thesi_k <- i
                k <- k-1
                if (k==0)
                        break
        }

        if (all(support1>=0))
                return(reconstr_trans(dataset,C))
        else{
                point <- find_thesi_k(dataset,i-1,D[[1]])
                for (m in point:1){
                        for (n in length(D):1){
```

```
if (length(setdiff(dataset[[m]],D[[n]]))==length(dataset[[m]])-length(D[[n]]) && S[[n]]<0 &&
check_support(dataset[[m]],D[[n]],dataset,support)==1){
        support[m]<-support[m]+abs(S[n])
        if (n>1){
        for (p in (n-1):1)
                if (length(setdiff(dataset[[m]],D[[p]]))==length(dataset[[m]])-length(D[[p]]))
                        S[p]<-S[p]+abs(S[n])
                                        }
                                        S[n]<-0
                                }
                        }
                        if (all(S>=0))
                                break
                }
                support<-support_correction(dataset,1,support)
                increase_itemset-k+2_support(dataset,support)
        }
}
```

Συνάρτηση support_correction – Ελέγχει αν ισχύει ηApriori αρχή μεταξύ όλων των υποσυνόλων

```
support_correction<-function(dataset,pointer,support){
        for (i in pointer:(length(dataset)-1)){
                for (j in (i+1):length(dataset)){
                        if (length(setdiff(dataset[[i]],dataset[[j]]))==length(dataset[[i]])-
                        length(dataset[[j]]))
                                if (support[j]<support[i])
                                        support[j]<-support[i]
                        }
        }
        return(support)
}
```

Συνάρτηση check_support – Ελέγχει αν το στοιχειοσύνολο-k+2, έχει μικρότερη support_count και από τα 2 στοιχειοσύνολα-k (υπερσύνολα του στοιχειοσυνόλου-k στο οποίο έγινε fail

```
check_support <- function(set1,set2,dataset,support){
        set3 <- setdiff(set1,set2)
        set4 <- c(set2,set3[1])
        set5 <- c(set2,set3[2])
        s1 <- set_position(set1,dataset)
        s4 <- set_position(set4,dataset)
        s5 <- set_position(set5,dataset)
        if (support[s1]<support[s4] && support[s1]<support[s5])
                return (1)
        else
                return (0)
}
```

Συνάρτηση set_position – Βρίσκει τη θέση ενός στοιχειοσυνόλου μέσα σ' ένα σύνολο στοιχειοσυνόλων

```
set_position <- function(set,dataset){
        for (i in 1:length(dataset))
                if (setequal(set,dataset[[i]])==TRUE)
                        break
        return(i)
}
```

Συνάρτηση subset_elimination – Για ένα σύνολο στοιχειοσυνόλων απαλείφει όλα τα υποσύνολα κάθε στοιχειοσυνόλου

```
subset_elimination <- function(dataset){
        dataset_n <- matrix()
        d <- 1
        if (length(dataset)==1)
                return(dataset)
        dataset_n[1]<-dataset[1]
        for (i in 2:(length(dataset))){
                if (length(dataset[[i]])==length(dataset[[1]])){
                        d <- d+1
                        dataset_n[d] <- dataset[i]
                }
        }
        return(dataset_n)
}
```

Αλγόριθμος 3.5

```
improve_reconstruction <- function(rec_db, min_sup, freq){
        total_trans<-min_sup/freq
        erase_trans <- nrow(rec_db)-total_trans
        for (i in 1:nrow(rec_db)){
                if (erase_trans>0){
                        rec_db<-rec_db[-nrow(rec_db),]
                        erase_trans<-erase_trans-1
                }
                else
                        break
                }
        return(rec_db)
}
```

Αλγόριθμος 3.7

```
controlled_hide_item<-function(transactions,minsup,minconf,hidden_items){
        T1<-matrix()
        T2<-matrix()
        T <- matrix()
        set1<-0
        set2<-0

        items <- names(transactions)
        pairs<-combn(items,2)
        for(i in 1:ncol(pairs)){
                set1<-set1+1
                set2<-set2+1
                T1[set1]<-list(c(pairs[1,i],pairs[2,i]))
                T2[set2]<-list(c(pairs[2,i],pairs[1,i]))
        }

        T<-union(T1,T2)
        for (i in 1:length(T)){
                if (any(hidden_items==T[[i]][2]))
                        if((supports(T[[i]],transactions)/nrow(transactions)>=minsup) &&
                        (supports(T[[i]],transactions)/supports(T[[i]][1],transactions)>=minconf))
                                transactions<-delete_R(transactions,minsup,minconf,T[[i]],T[[i]][2])

                }
                for (i in 1:length(T)){
                        if (any(hidden_items==T[[i]][1]))
                                if(supports(T[[i]],transactions)/supports(T[[i]][1],transactions)>=minconf)
                                        transactions<-increase_L(transactions,minconf,T[[i]],T[[i]][1])

                }
        return(transactions)
}
```

Συνάρτηση delete_R – Διαγράφει από τις συναλλαγές που έχουν και τα δύο στοιχειοσύνολα του κανόνα, το RHS ελέγχοντας παράλληλα την support & confidence

```
delete_R <- function(transactions,minsup,minconf,itemset,item1){
        transactions1<-transactions

        columns <- find_column(itemset,transactions)
        column1 <- find_column(item1,transactions)
        item2<-setdiff(itemset,item1)
        for (i in 1:nrow(transactions1)){
                if (sum(transactions1[i,columns])==2)
                        if ((supports(itemset,transactions)/nrow(transactions)<minsup) ||
                        (supports(itemset,transactions)/supports(item2,transactions)<minconf))
                                break
                        else
                                transactions[i,column1]<-0
        }
        return(transactions)
}
```

Συνάρτηση increase_L – Εισάγει στις συναλλαγές που δ εν έχουν και τα δύο στοιχειοσύνολα του κανόνα, το LHS ελέγχοντας παράλληλα την support

```
increase_L <- function(transactions,minconf,itemset,item1){
        transactions1<-transactions

        columns <- find_column(itemset,transactions)
        column1 <- find_column(item1,transactions)
        for (i in 1:nrow(transactions1)){
                if (sum(transactions1[i,columns])==0)
```

```
                        if (supports(itemset,transactions)/supports(item1,transactions)<minconf)
                                break
                        else
                                transactions[i,column1]<-1
        }
        return(transactions)
}
```

Αλγόριθμος 3.8

```
compare_algorithms <- function(an1,de1,an2,de2,sensitives){
        totalrules <- length(an2)
        notsensitives <- c(1:length(an1))[-sensitives]
        sensnothidden <- 0
        notsensrevealed <- 0
        lost_rules <- length(notsensitives)
        for (i in 1:length(sensitives)){
                a <- an1[sensitives[i]]
                b <- de1[sensitives[i]]
                for (j in 1:length(an2)){
                        if (setequal(an2[j],a))
                                if (setequal(de2[j],b)){
                                        sensnothidden <- sensnothidden+1
                                        an2 <- an2[-j]
                                        de2 <- de2[-j]
                                        break
                                }
                }
        }
        srh <- 1-(sensnothidden/length(sensitives))
        cat("Sensitive Rules Hidden:",srh*100,"%\n")

        for (i in 1:length(notsensitives)){
                a <- an1[notsensitives[i]]
                b <- de1[notsensitives[i]]
                for (j in 1:length(an2)){
                        if (setequal(an2[j],a))
                                if (setequal(de2[j],b)){
                                        lost_rules <- lost_rules-1
                                        notsensrevealed <- notsensrevealed +1
                                        an2 <- an2[-j]
                                        de2 <- de2[-j]
                                        break
                                }
                }
        }
        lrr<-lost_rules/length(notsensitives)
        cat("\nLost Rules:",lrr*100,"%\n")
        if (length(de2)!=0)
                gr <- totalrules-(sensnothidden+notsensrevealed)/length(de2)
        else
                gr<-totalrules-(sensnothidden+notsensrevealed)
        cat("\nGhost Rules:",gr*100,"%\n")
}
```