Εθνικο Μετσοβιο Πολυτεχνειο

Σχολη Ηλεκτρολογων Μηχανικων και
Μηχανικων Υπολογιστων

Τομεας Τεχνολογιας Πληροφορικης και Υπολογιστων

Εξατομικευμένα Συστήματα Διαχείρισης Δεδομένων

Διδακτορική Διατριβή

του

Αναστάσιου Αρβανίτη

Διπλωματούχου Ηλεκτρολόγου Μηχανικού και

Μηχανικού Υπολογιστών Ε.Μ.Π. (2005)

Αθήνα, Ιανουάριος 2013

Η παρούσα έρευνα έχει συγχρηματοδοτηθεί από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο - ΕΚΤ) και από εθνικούς πόρους μέσω του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» του Εθνικού Στρατηγικού Πλαισίου Αναφοράς (ΕΣΠΑ) – Ερευνητικό Χρηματοδοτούμενο Έργο: Ηράκλειτος ΙΙ. Επένδυση στην κοινωνία της γνώσης μέσω του Ευρωπαϊκού Κοινωνικού Ταμείου.

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

Εξατομικευμένα Συστήματα Διαχείρισης Δεδομένων

Διδακτορική Διατριβή

του

Αναστάσιου Αρβανίτη

Διπλωματούχου Ηλεκτρολόγου Μηχανικού και

Μηχανικού Υπολογιστών Ε.Μ.Π. (2005)

Συμβουλευτική Επιτροπή: Ι. Βασιλείου
Τ. Σελλής
Ι. Σταύρακας

Εγκρίθηκε από την επταμελή εξεταστική επιτροπή την 4η Ιανουαρίου 2013.

...	...	...
Ι. Βασιλείου	Τ. Σελλής	Ι. Σταύρακας
Καθ. ΕΜΠ	Καθ. ΕΜΠ	Ερευνητής Β΄ ΙΠΣΥ/ΕΚ Αθηνά
...	...	...
Φ. Αφράτη	Α. Γ. Σταφυλοπάτης	Κ. Κοντογιάννης
Καθ. ΕΜΠ	Καθ. ΕΜΠ	Αναπλ. Καθ. ΕΜΠ
	...	
	Α. Δεληγιαννάκης	
	Επικ. Καθ. Πολυτεχνείου Κρήτης	

Αθήνα, Ιανουάριος 2013

ΠΡΟΛΟΓΟΣ

Η διατριβή αυτή είναι μέρος των απαιτήσεων του τίτλου του Διδάκτορα Μηχανικού στη Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του Εθνικού Μετσόβιου Πολυτεχνείου. Το αντικείμενο της διατριβής αφορά στα εξατομικευμένα συστήματα διαχείρισης δεδομένων και έχει πραγματοποιηθεί τα τελευταία πέντε χρόνια στο Εργαστήριο Συστημάτων Βάσεων Γνώσεων και Δεδομένων (ΕΣΒΓΔ) του ΕΜΠ.

Στο σημείο αυτό θα ήθελα να ευχαριστήσω όσους συνέβαλαν στην ολοκλήρωση της διατριβής αυτής. Πρώτα απ΄ όλα θα ήθελα να εκφράσω τις θερμότερες ευχαριστίες μου στον Καθ. Τίμο Σελλή και στον Καθ. Γιάννη Βασιλείου για την πολύτιμη καθοδήγηση, τις συμβουλές και την υποστήριξή τους σε θέματα όχι μόνο της διατριβής αλλά και καθ' όλη τη διάρκεια της παρουσίας μου στο ΕΜΠ ως προπτυχιακός και μεταπτυχιακός φοιτητής.

Επιπλέον, θέλω να ευχαριστήσω ιδιαιτέρως τη Δρ. Γεωργία Κούτρικα, Ερευνήτρια στα HP Labs, Palo Alto, California για την ιδιαίτερα εποικοδομητική συνεργασία που είχαμε όλα αυτά τα χρόνια, την υπομονή και την αταλάντευτη υποστήριξή της σε όλες τις δύσκολες στιγμές καθ' όλη τη διάρκεια της εκπόνησης της διατριβής μου. Οι συμβουλές της και η καθοδήγησή της με βοήθησαν σημαντικά να βελτιώσω την ποιότητα της διατριβής, να κατανοήσω σε βάθος τη συγκεκριμένη περιοχή έρευνας και να αποκτήσω δεξιότητες καθοριστικές στην μέχρι τώρα πορεία μου ως ερευνητής.

Επίσης θέλω να ευχαριστήσω ξεχωριστά όλους όσους συνεργάστηκα κατά περιόδους σε διαφορετικές φάσης της διατριβής μου. Τον Αντώνη Δεληγιαννάκη, Επ. Καθ. στο Πολυτεχνείο Κρήτης, τον Δρ. Δημήτρη Σαχαρίδη, τον Δημήτρη Παπαδιά, Καθ. στο Hong Kong University of Science and Technology, τον Βαγγέλη Χρηστίδη, Αν. Καθ. στο University of California, Riverside, τον Δρ. Γιάννη Σταύρακα, Ερευνητή στο Ινστιτούτο Πληροφοριακών Συστημάτων του Ερευνητικού Κέντρου 'Αθηνά' και τον Γιάννη Ρούσσο, διδακτορικό φοιτητή στο Εργαστήριο Συστημάτων Βάσεων Γνώσεων και Δεδομένων του ΕΜΠ. Η συγκεκριμένη διατριβή θα ήταν εξαιρετικά δύσκολο να έρθει σε πέρας χωρίς τη βοήθεια, τις ιδέες τους και τα πολύτιμα σχόλιά τους.

Επιπρόσθετα οφείλω να ευχαριστήσω όλα τα μέλη του Εργαστηρίου Συστημάτων Βάσεων Γνώσεων και Δεδομένων του ΕΜΠ και του Ινστιτούτου Πληροφοριακών Συστημάτων του Ερευνητικού Κέντρου 'Αθηνά' με τα οποία είχα τη χαρά να δουλέψω όλα αυτά τα χρόνια. Ιδιαιτέρως δε, τονίζω την καθοριστική βοήθεια όσων συναδέλφων είχαν επιφορτιστεί με την τεχνική υποστήριξη του ΕΣΒΓΔ. Παράλληλα θέλω να ευχαριστήσω την Καθ. Φώτω Αφράτη, τον Καθ. Ανδρέα-Γεώργιο Σταφυλοπάτη και τον Αν. Καθ. Κώστα Κοντογιάννη από το Εθνικό Μετσόβιο Πολυτεχνείο που με προθυμία

δέχτηκαν να συμμετέχουν στη διαδικασία κρίσης της διατριβής, διατελώντας μέλη της επταμελούς εξεταστικής επιτροπής. Επίσης θα ήταν παράλειψη να μην αναφερθώ στους δεκάδες ανώνυμους κριτές των εργασιών μου. Οι διορθώσεις τους, οι ενδιαφέρουσες οπτικές τους και οι παρατηρήσεις τους βελτίωσαν σε σημαντικό βαθμό την ποιοτική στάθμη της διατριβής.

Τέλος, θέλω να εκφράσω την ευγνωμοσύνη μου στους γονείς μου για τη ακούραστη και συνεχή υποστήριξη και ενθάρρυνση τους όλα αυτά τα χρόνια. Ήταν πάντα δίπλα μου και τους ευχαριστώ με όλη μου την καρδιά.

Αναστάσιος Αρβανίτης
Αθήνα, Ιανουάριος 2013

ΠΕΡΙΛΗΨΗ

Η αλματώδης ανάπτυξη του Παγκόσμιου Ιστού τον έχει καταστήσει μία τεράστια δεξαμενή πληροφοριών για την ανθρωπότητα. Βασικά χαρακτηριστικά της ανάπτυξης αυτής είναι: (α) η υπερπροσφορά πληροφοριών και υπηρεσιών, (β) η κάλυψη των αναγκών χρηστών με διαφορετικά ενδιαφέροντα και (γ) η ποικιλία μεθόδων πρόσβασης στις πληροφορίες. Παράλληλα, η ανάπτυξη του Web 2.0 και των κοινωνικών δικτύων καθώς και η εξέλιξη των ασύρματων δικτύων, των κινητών συσκευών (π.χ. smartphones, tablets) και των αισθητήρων χαμηλού κόστους (π.χ. δέκτες GPS) κάνουν εφικτή τη συλλογή τεράστιων όγκων δεδομένων για τα χαρακτηριστικά, τα ενδιαφέροντα, τις προτιμήσεις και την τρέχουσα κατάσταση των χρηστών. Σε αυτό το πλαίσιο, δημιουργούνται προβλήματα τόσο για τους χρήστες όσο και για τους παρόχους πληροφοριών, καθώς οι πρώτοι αδυνατούν να ξεχωρίσουν τις πληροφορίες που είναι πιο σχετικές με τα ενδιαφέροντα και τις προτιμήσεις τους, ενώ οι δεύτεροι αναζητούν τρόπους πιο αποτελεσματικής και στοχευμένης προώθησης των υπηρεσιών τους προς καταναλωτές που είναι πιο πιθανό να ενδιαφερθούν γι΄ αυτές. Ως εκ τούτου η ανάγκη για συστήματα και τεχνικές που να παρέχουν εξατομικευμένες υπηρεσίες σε χρήστες και παρόχους γίνεται ολοένα και πιο επιτακτική. Η συγκεκριμένη διατριβή πραγματεύεται ζητήματα εξατομίκευσης στη διαχείριση δεδομένων, τόσο σε επίπεδο συστήματος όσο και σε επίπεδο εφαρμογών. Συγκεκριμένα, εστιάζουμε: (α) σε συστήματα διαχείρισης προτιμήσεων χρηστών για σχεσιακά δεδομένα, και (β) σε τεχνικές αποτίμησης ερωτημάτων προτιμήσεων (preference queries). Για το πρώτο κομμάτι, προτείνεται ένα νέο μοντέλο αναπαράστασης προτιμήσεων για σχεσιακά δεδομένα και επεκτείνονται οι υπάρχοντες τελεστές της σχεσιακής άλγεβρας ώστε να λαμβάνουν υπόψη τους τις προτιμήσεις κάθε χρήστη. Επιπλέον, έχει υλοποιηθεί ένα πρωτότυπο σύστημα διαχείρισης προτιμήσεων ενσωματωμένο σε ένα τυπικό σχεσιακό σύστημα βάσεων δεδομένων. Σε σχέση με τις τεχνικές αποτίμησης ερωτημάτων προτιμήσεων προσεγγίζουμε το πρόβλημα: (α) από την πλευρά των χρηστών όπου προτείνουμε αλγορίθμους αποτίμησης ερωτημάτων κορυφογραμμής (skyline queries) όταν οι προτιμήσεις των χρηστών μεταβάλλονται σε σχέση με το περιβάλλον χρήσης (context), π.χ. τη γεωγραφική θέση, χρόνο, καιρικές συνθήκες, κοινωνικό περιβάλλον, και (β) από την πλευρά των παρόχων προτείνοντας τεχνικές αποτίμησης ερωτημάτων έρευνας αγοράς. Πιο συγκεκριμένα, προτείνονται αλγόριθμοι: (α) για την εύρεση των χρηστών οι οποίοι είναι πιο πιθανό να αγοράσουν ένα προϊόν με σκοπό την στοχευμένη προώθησή του (personalized advertising), και (β) για τη βέλτιστη διαμόρφωση των χαρακτηριστικών ενός νέου προϊόντος (product positioning) ώστε να μεγιστοποιείται το όφελος μετρούμενο ως το εκτιμώμενο πλήθος των πιθανών αγοραστών.

ABSTRACT

The exponential growth of the Web has turned it into a huge repository of information for humanity. Key features of this development are: (a) the oversupply of information and services, (b) the coverage of the demands for users with various interests and characteristics, and (c) the variety of information accessing means. Furthermore, the evolution of social networks and Web 2.0, and the growing internet access via fast wireless networks and mobile devices (e.g. smartphones, tablets) have enabled the collection of vast amounts of data regarding the characteristics, interests and preferences of users and the corresponding usage scenarios. In this context, several challenges emerge for both users and information providers, since for the former the task of discovering the portions of information that are the most relevant to their interests and preferences is becoming harder, while the latter seek ways for more effective and targeted promotion of their services to the users who are more likely to be interested in them. Hence, the need for systems and methods that provide personalized services to users and providers is becoming urgent. This thesis deals with personalization issues in data management at both system and application levels. Specifically, we focus on: (a) user preference management for relational data, and (b) preference query evaluation techniques. For the first part, we propose a new model for representing preferences over relational data and we extend the existing relational algebra operators, such that they take into account the user preferences. Moreover, we have implemented a prototype preference-aware data management framework embedded in a standard relational database system. With regards to preference queries evaluation we approach the problem: (a) from a user's perspective where we propose algorithms for processing skyline queries when user preferences depend on the usage context, e.g., location, time, weather, social environment, and (b) from the view of data providers, where we propose methods for efficient processing of market research queries. In particular, we propose algorithms: (a) for identifying those users that are more likely to buy a product, with applications in personalized advertising, and (b) for optimal product positioning, where we aim to maximize the profit measured as the estimated number of potential product buyers.

Περιεχόμενα

Κατάλογος Σχημάτων

Κατάλογος Πινάκων

Κεφάλαιο 1

Εισαγωγή

1.1 Συστήματα εξατομίκευσης

Καθημερινά κατακλυζόμαστε από ένα πλήθος πληροφοριών και προσφερόμενων υπηρεσιών. Ολοένα αυξανόμενοι όγκοι πληροφοριών είναι πλέον άμεσα διαθέσιμοι σε δισεκατομμύρια ανθρώπων, κάθε ηλικίας, εθνικότητας, μορφωτικού επιπέδου και κουλτούρας. Βασικά χαρακτηριστικά αυτής της εξέλιξης είναι: (α) η υπερπροσφορά πληροφοριών (information overload), (β) η ανάγκη για κάλυψη των αναγκών χρηστών με διαφορετικά ενδιαφέροντα (user heterogeneity) και (γ) η ποικιλία μεθόδων πρόσβασης στις πληροφορίες (από διαφορετικές συσκευές, υπό διαφορετικές συνθήκες χρήσης κλπ.). Σε αυτό το πλαίσιο είναι πολύ συχνό το φαινόμενο οι χρήστες εφαρμογών διαχείρισης δεδομένων να δυσκολεύονται να βρουν χρήσιμες πληροφορίες που σχετίζονται με τις επιδιώξεις και τα ενδιαφέροντά τους. Έτσι πολύ σύντομα έγινε επιτακτική η ανάγκη για συστήματα και τεχνικές που να παρέχουν στοχευμένες πληροφορίες και υπηρεσίες που να καλύπτουν τα ενδιαφέροντα κάθε ανθρώπου. Αναφερόμαστε συλλογικά σε τέτοιου είδους συστήματα χρησιμοποιώντας τον όρο *συστήματα εξατομίκευσης*.

Κοινό χαρακτηριστικό των συστημάτων εξατομίκευσης είναι ότι προσαρμόζουν τη λειτουργικότητά τους στα ενδιαφέροντα, τις προτιμήσεις, τη συμπεριφορά ή την τρέχουσα κατάσταση των χρηστών. Η προσαρμογή της λειτουργικότητας (*εξατομίκευση*) σχετίζεται με ένα ή περισσότερα από τα παρακάτω:

- *Εξατομίκευση στην προσπέλαση πληροφοριών.* Για παράδειγμα, έστω δύο χρήστες που επισκέπτονται ένα ηλεκτρονικό βιβλιοπωλείο. Το σύστημα έχει συλλέξει πληροφορίες από προηγούμενες αγόρες ή αναζητήσεις κάθε χρήστη. Έστω ότι ο χρήστης Α έχει αγοράσει προηγούμενα βιβλία του Dan Brown, ενώ ο Β ενδιαφέρθηκε για βιβλία της σειράς Harry Potter. Σε αυτή την περίπτωση ένα σύστημα εξατομίκευσης αναμένεται να προτείνει στον χρήστη Α βιβλία μυστηρίου και στον Β παιδικά μυθιστορήματα.

- *Εξατομίκευση στην παρουσίαση των πληροφοριών ή στη διεπαφή ενός συστήματος.* Για παράδειγμα, το ίδιο σύστημα μπορεί να παρουσιάζει τα διαθέσιμα βιβλία προς πώληση με αρκετές λεπτομέρειες (π.χ. κριτικές άλλων χρηστών, πίνακα περιεχομένων, φωτογραφία εξωφύλλου κλπ.) για ένα χρήστη που έχει πρόσβα-

ση στο ηλεκτρονικό κατάστημα μέσω του ηλεκτρονικού του υπολογιστή, ή πιο περιληπτικά για ένα χρήστη που το επισκέπτεται μέσω μιας κινητής συσκευής.

- *Εξατομίκευση υπηρεσιών.* Για παράδειγμα, ένα ηλεκτρονικό κατάστημα μπορεί να προσφέρει ειδικές εκπτώσεις σε πιο τακτικούς του πελάτες, ή να παρέχει άλλες εξειδικευμένες προσφορές και υπηρεσίες.

Οι βασικοί παράγοντες που λαμβάνονται υπόψη για τη διαμόρφωση της συμπεριφοράς ενός συστήματος εξατομίκευσης είναι:

- *τα δημογραφικά χαρακτηριστικά ενός χρήστη (user profile)* όπως η ηλικία, το φύλο, ο τόπος διαμονής, η οικογενειακή κατάσταση, το μορφωτικό επίπεδο, η απασχόληση, η γλώσσα κ.α. Για παράδειγμα αν ένας χρήστης έχει παιδιά είναι πολύ πιθανό να τον ενδιαφέρει η παιδική λογοτεχνία.

- *οι προτιμήσεις ενός χρήστη εκφρασμένες σε σχέση με τα χαρακτηριστικά των παρεχόμενων υπηρεσιών (user preferences)* όπως για παράδειγμα η προτίμηση κάποιου στα αστυνομικά μυθιστορήματα.

- *το περιβάλλον χρήσης (context)* όπως για παράδειγμα η τρέχουσα γεωγραφική θέση ενός χρήστη, ο χρόνος, ο τρόπος πρόσβασης στα δεδομένα, το κοινωνικό περιβάλλον κ.α. Για παράδειγμα, κατά την περίοδο των καλοκαιρινών διακοπών είναι αρκετά πιθανό να αγοράσει κάποιος έναν ταξιδιωτικό οδηγό για τα νησιά του Αιγαίου.

Σε πολλές περιπτώσεις μάλιστα, οι παράγοντες αυτοί μπορεί να είναι αλληλεξαρτώμενοι. Για παράδειγμα την περίοδο των Χριστουγέννων, κάποιος χρήστης που έχει παιδιά είναι περισσότερο πιθανό να προτιμήσει παιδικά παιχνίδια ή βιβλία, ενώ μια γυναίκα Χριστουγεννιάτικα διακοσμητικά για το σπίτι.

1.1.1 Η εξέλιξη των συστημάτων εξατομίκευσης

Τα τελευταία χρόνια η ανάπτυξη εμπορικών συστημάτων εξατομίκευσης υπήρξε ραγδαία. Η εξατομίκευση έγινε βασικό συστατικό της λειτουργικότητας μιας μεγάλης ποικιλίας εφαρμογών όπως κοινωνικά δίκτυα, ηλεκτρονικά καταστήματα, εφαρμογές πολυμέσων, ειδησεογραφικές ιστοσελίδες, διαφημιστικές πλατφόρμες κ.ο.κ. Παρακάτω αναλύουμε τους βασικούς παράγοντες που συνέβαλαν σε αυτή την εξέλιξη.

Ίσως ο πιο καθοριστικός λόγος που οδήγησε στην ανάπτυξη συστημάτων εξατομίκευσης έχει να κάνει με την ποσότητα πληροφοριών και το εύρος επιλογών που έχουν σήμερα οι χρήστες των διαφόρων συστημάτων και εφαρμογών. Η ροή των πληροφοριών και η ποικιλία των υπηρεσιών που προσφέρονται ιδιαιτέρως μέσω του διαδικτύου υπερβαίνει κατά πολύ αυτή που ο μέσος άνθρωπος μπορεί να διαχειριστεί και να καταναλώσει. Για κάθε πιθανή ανάγκη (π.χ. ενημέρωση, ψυχαγωγία, επικοινωνία, αγορές προϊόντων, κ.α.) υπάρχουν χιλιάδες διαφορετικές επιλογές που ανταγωνίζονται για το ενδιαφέρον και την προσοχή των χρηστών. Αυτό δημιουργεί σημαντική δυσκολία στους χρήστες ώστε να επιλέξουν αυτό το οποίο ταιριάζει περισσότερο με τα ενδιαφέροντα

και τις προτιμήσεις τους. Παράλληλα, αυτή η υπερπροσφορά δεδομένων και υπηρεσιών αποτελεί μεγάλη πρόκληση και για τους παρόχους περιεχομένου, καθώς προσπαθούν να βελτιώσουν την εμπειρία των χρηστών, να καλύψουν διαφορετικές πιθανές ανάγκες τους και να προσελκύσουν περισσότερους καταναλωτές. Συνεπώς, η ανάγκη για συστήματα που αντιλαμβάνονται τις διαφορετικές προτιμήσεις, ανάγκες και ενδιαφέροντα των χρηστών έχει πλέον αποκτήσει τεράστια σημασία τόσο για την πλευρά των καταναλωτών/χρηστών τέτοιων συστημάτων όσο και για την πλευρά των παρόχων περιεχομένου και υπηρεσιών.

Ένας δεύτερος σημαντικός λόγος που διευκόλυνε την ανάπτυξη συστημάτων εξατομίκευσης είναι ότι η συλλογή, καταγραφή και επεξεργασία τεράστιων όγκων δεδομένων για τα χαρακτηριστικά, τα ενδιαφέροντα και τις προτιμήσεις των χρηστών καθώς και για το τρέχον περιβάλλον χρήσης έχει γίνει πλέον αρκετά ευκολότερη. Δύο βασικοί τρόποι που συλλέγονται τέτοια δεδομένα είναι μέσω των κοινωνικών δικτύων καθώς και μέσω της πρόσβασης στο διαδίκτυο από κινητές συσκευές. Και στις δύο περιπτώσεις οι ίδιοι οι χρήστες δέχονται να διαθέσουν τα δεδομένα αυτά στις αντίστοιχες εφαρμογές είτε ευθέως (δημιουργώντας έναν λογαριασμό/προφίλ χρήστη) είτε έμμεσα κατά τη χρήση της εφαρμογής. Τα δεδομένα που χρησιμοποιούνται για εξατομίκευση περιλαμβάνουν δημογραφικά χαρακτηριστικά των χρηστών (ηλικία, φύλο, τόπος διαμονής κλπ.), προτιμήσεις που δηλώνουν οι χρήστες απαντώντας σε σχετικές ερωτήσεις ή βαθμολογώντας ενδιαφέρουσες πληροφορίες, ή δεδομένα που έχουν συλλεχθεί παρακολουθώντας τη συμπεριφορά κάθε χρήστη, για παράδειγμα ποιες σελίδες επισκέφτηκε, πόσο χρόνο έμεινε σε κάθε σελίδα, σελίδες, σχόλια ή αξιολογήσεις που έκανε κ.α. Πολύ συχνά η ενσωμάτωση πληροφοριών που σχετίζονται με τα ενδιαφέροντα και τις προτιμήσεις ενός χρήστη είναι πολύ απλή, ή μπορεί να γίνεται ακόμα και χωρίς την άμεση συμμετοχή του χρήστη. Για παράδειγμα πολλοί χρήστες κοινωνικών δικτύων μοιράζονται πληροφορίες που τους ενδιαφέρουν (π.χ. μουσική, ειδήσεις, φωτογραφίες, βίντεο) ή την τρέχουσα τοποθεσία ή κατάστασή τους.

Τέλος, μια άλλη σημαντική παράμετρος που συνέβαλε στην εξέλιξη των συστημάτων εξατομίκευσης έχει να κάνει με το είδος των υπηρεσιών που προσφέρονται μέσω του διαδικτύου και των κινητών συσκευών (smartphones, tablets). Ενώ αρχικά οι υπηρεσίες που αναπτύχθηκαν αφορούσαν κύριως τη διεκπεραίωση συναλλαγών (transactions), σήμερα το διαδίκτυο έχει αποκτήσει μια πιο κοινωνική υπόσταση (social web). Μάλιστα αξίζει να σημειωθεί ότι οι περισσότερες από τις πιο δημοφιλείς εφαρμογές που χρησιμοποιούνται σε καθημερινή βάση από δισεκατομμύρια ανθρώπους είναι τέτοιας μορφής. Ως παράδειγμα μπορούμε να αναφέρουμε εφαρμογές του κοινωνικού ιστού (π.χ. facebook, twitter, linkedin), εφαρμογές πολυμέσων (π.χ. youtube, flickr, pandora, last.fm, netflix), αναζήτησης τοπικών σημείων ενδιαφέροντος (π.χ. foursquare, yelp), ημερήσιων προσφορών (π.χ. groupon, livingsocial), αναζήτησης ή συστάσεων για νέα ή blogs (π.χ. stumbleupon, delicious, reddit), ηλεκτρονικών αγοραπωλησιών (π.χ. amazon, ebay), κ.ο.κ.

Μεταξύ των δύο τύπων εφαρμογών υπάρχουν αρκετές και σημαντικές διαφορές. Όπως θα περιγράψουμε με περισσότερη λεπτομέρεια και στην Ενότητα 1.2.1, οι διαφορές αυτές επηρεάζουν τον τρόπο λειτουργίας ενός συστήματος βάσεων δεδομένων. Πιο συγκεκριμένα, οι εφαρμογές διεκπεραίωσης συναλλαγών θέτουν πολύ αυστηρές

και σαφείς απαιτήσεις από τη βάση δεδομένων. Για παράδειγμα έστω ένας πελάτης μιας τράπεζας που θέλει να ελέγξει το ύψος του υπολοίπου του τραπεζικού του λογαριασμού. Σε αυτή την περίπτωση, το σύστημα θα πρέπει να δίνει πάντοτε την ίδια απάντηση με ακρίβεια (εφόσον εντωμεταξύ δεν έχουν εκτελεστεί συναλλαγές), ανεξαρτήτως του χρόνου που έγινε η ερώτηση υπολοίπου ή αν η ερώτηση έγινε από τον κάτοχο του λογαριασμού ή κάποιον υπάλληλο της τράπεζας ή αν ο ερωτών υπέβαλε το ερώτημα μέσω ενός ATM ή μέσω του διαδικτύου ή άλλου εναλλακτικού δικτύου της τράπεζας. Αντιθέτως, για πολλές νέου τύπου εφαρμογές οι απαιτήσεις από τη βάση δεδομένων είναι λιγότερο αυστηρές ή ξεκάθαρες ή ακόμα η ακρίβεια της απάντησης δεν είναι το κύριο ζητούμενο. Οι χρήστες συνήθως αναμένουν από το σύστημα να τους παρέχει χρήσιμες ή ενδιαφέρουσες πληροφορίες χωρίς όμως εγγύηση ότι αυτές οι πληροφορίες είναι επίκαιρες ή ακριβείς, ενώ επιπλέον κάθε χρήστης επιθυμεί να προσλαμβάνει εξατομικευμένο περιεχόμενο σύμφωνα με τα ενδιαφέροντα και την προηγούμενη συμπεριφορά του στο σύστημα. Για τις εφαρμογές αυτές, ο τρόπος που κάθε σύστημα αξιολογεί τι είναι 'σχετικό' για κάθε χρήστη δεν είναι προκαθορισμένος και απόλυτος ενώ πολλές φορές και τα αποτελέσματα που προσφέρονται δεν είναι συνεπή. Συνεπώς για αυτές τις νέου τύπου εφαρμογές, η ανάγκη ενσωμάτωσης λειτουργιών εξατομίκευσης στον τρόπο που διαχειρίζονται το περιεχόμενό τους έρχεται στο προσκήνιο με επιτακτικό τρόπο.

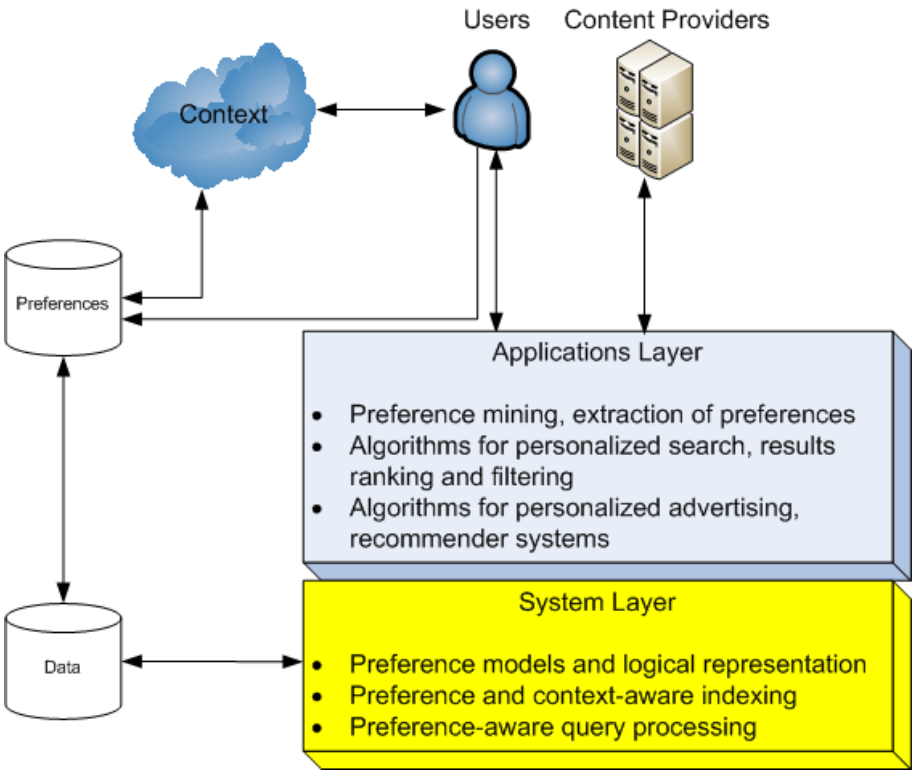
1.1.2 Λειτουργίες ενός συστήματος εξατομίκευσης

Στην πράξη ένα σύστημα εξατομίκευσης ενσωματώνει ένα σύνολο λειτουργιών και υποσυστημάτων. Σε υψηλό επίπεδο, η αρχιτεκτονική ενός τέτοιου συστήματος φαίνεται στο σχήμα 1.1. Στη συνέχεια παρουσιάζουμε επιγραμματικά τις βασικές λειτουργίες που επιτελούνται σε ένα σύστημα εξατομίκευσης:

- *συλλογή της απαραίτητης πληροφορίας για τις προτιμήσεις, ή τα χαρακτηριστικά των χρηστών καθώς και το περιβάλλον χρήσης (preference extraction).* Η διαδικασία αυτή μπορεί να γίνει είτε άμεσα από τον ίδιο τον χρήστη, είτε έμμεσα με παρακολούθηση της συμπεριφοράς του (web usage/log mining).

- *εξαγωγή/εκμάθηση κανόνων προτίμησης (preference elicitation/learning).* Πολλές φορές τα δεδομένα που συλλέγονται στο προηγούμενο βήμα είναι ελλιπή ή αντιφατικά. Για να εξαχθούν πιο καθαροί ή εφαρμόσιμοι κανόνες για τις προτιμήσεις των χρηστών συνήθως χρησιμοποιούνται τεχνικές εξόρυξης δεδομένων (Data Mining) ή μηχανικής μάθησης (Machine Learning). Ο σκοπός είναι το σύστημα να μάθει ένα μοντέλο συμπεριφοράς ή προτιμήσεων του χρήστη που ταιριάζει όσο το δυνατόν στις προτιμήσεις που έχουν συλλεχθεί σε πρωτογενές επίπεδο.

- *διαχείριση δεδομένων που εξαρτώνται από προτιμήσεις (preference-aware data management).* Το βήμα αυτό περιλαμβάνει την λογική αναπαράσταση, αποθήκευση, δεικτοδότηση κλπ. των προτιμήσεων καθώς και των δεδομένων που εξαρτώνται από αυτές.



Σχήμα 1.1: *Αρχιτεκτονική Συστήματος Εξατομίκευσης*

- *ανάκτηση, ταξινόμηση, φιλτράρισμα ή συνδυασμό των δεδομένων που ικανοποιούν τις προτιμήσεις* με βάση κάποιον αλγόριθμο προτίμησης (preference algorithm), όπως για παράδειγμα ερωτήματα κατάταξης, κορυφογραμμής κλπ.

- *κατάλληλη παρουσίαση των δεδομένων στον τελικό χρήστη* με προσαρμογή της διεπαφής του συστήματος.

1.1.3 Ερευνητικά θέματα εξατομίκευσης

Στην πράξη, το πρόβλημα της εξατομίκευσης είναι αρκετά πολυσύνθετο. Ως εκ τούτου, παράλληλα με την ανάπτυξη εμπορικών συστημάτων, τα ζητήματα εξατομίκευσης προσέλκυσαν το έντονο ενδιαφέρον της ερευνητικής κοινότητας. Είναι χαρακτηριστικό ότι πολλές διαφορετικές επιστήμες ασχολούνται με το συγκεκριμένο αντικείμενο, όπως για παράδειγμα οι Κοινωνικές Επιστήμες (π.χ. Ψυχολογία), η Μηχανική Μάθηση (Machine Learning), η Εξόρυξη Δεδομένων (Data Mining), η Τεχνητή Νοημοσύνη (Artificial Intelligence), οι Βάσεις Δεδομένων (Databases), η Ανάκτηση Πληροφοριών (Information Retrieval) και η Αλληλεπίδραση Ανθρώπου Μηχανής (Human-Computer Interaction). Κάθε τομέας έχει σημαντική συνεισφορά σε επιμέρους λειτουργίες ενός συστήματος εξατομίκευσης. Η Ψυχολογία, η Μηχανική Μάθηση, η Εξόρυξη Δεδομένων και η Τεχνική Νοημοσύνη εστιάζουν κυρίως στη συλλογή προτιμήσεων, στην εξαγωγή κανόνων και στη δημιουργία προφίλ χρηστών και μοντέλων συμπεριφοράς. Οι Βάσεις Δεδομένων και η Ανάκτηση Πληροφοριών ασχολούνται με συστήματα διαχείρισης δεδομένων που εξαρτώνται από τις προτιμήσεις και με τεχνικές αποτίμησης ερωτημάτων με προτιμήσεις, για δομημένα (structured) και αδόμητα (unstructured) δεδομένα αντί-

στοιχα. Η Αλληλεπίδραση Ανθρώπου-Μηχανής επικεντρώνεται στην προσαρμογή της διεπαφής στις προτιμήσεις και το περιβάλλον χρήσης.

Εστιάζοντας στις Βάσεις Δεδομένων τα ερευνητικά θέματα που προκύπτουν είναι αρκετά και ενδιαφέροντα. Ενδεικτικά αναφέρουμε κάποια προβλήματα που παραμένουν ανοικτά όπως η γενικευμένη αναπαράσταση των προτιμήσεων για δομημένα δεδομένα, η κατάλληλη δεικτοδότηση των δεδομένων και των προτιμήσεων, η επιλογή των κατάλληλων προτιμήσεων για ένα ερώτημα, ο τρόπος ενσωμάτωσής τους σε ένα ερώτημα, η μέθοδος επιλογής των πιο σχετικών αποτελεσμάτων, αλγόριθμοι κατάταξης, κ.α.

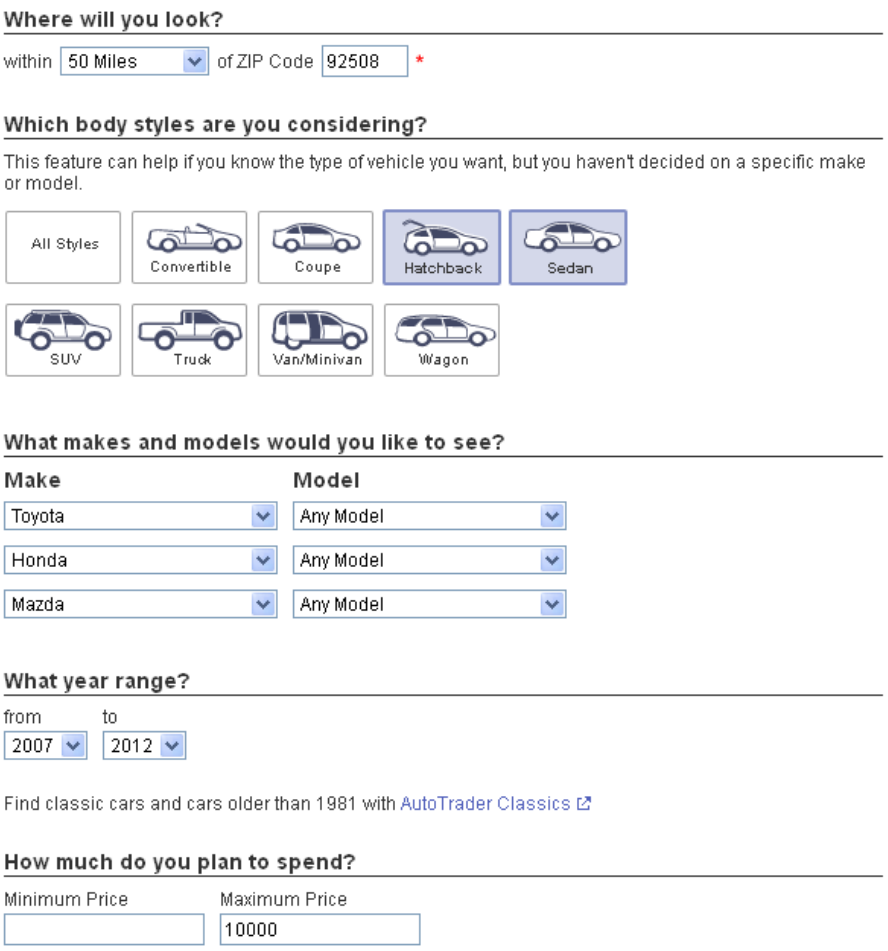
Αντικείμενο της παρούσας διατριβής είναι η μελέτη τεχνικών και συστημάτων εξατομίκευσης με έμφαση στην εξατομικευμένη διαχείριση δεδομένων. Συγκεκριμένα, εστιάζουμε: (α) σε σχεσιακά συστήματα διαχείρισης δεδομένων που εξαρτώνται από τις προτιμήσεις χρηστών, και (β) σε τεχνικές αποτίμησης ερωτημάτων προτιμήσεων (preference queries). Στις ακόλουθες ενότητες σκιαγραφούμε την τρέχουσα έρευνα που έχει γίνει στα συγκεκριμένα θέματα και επικεντρωνόμαστε στα ερευνητικά και πρακτικά προβλήματα που προκύπτουν κατά τη χρήση συστημάτων και τεχνικών εξατομίκευσης. Πιο αναλυτικά, στην ενότητα 1.2 περιγράφουμε τα συστήματα εξατομίκευσης που έχουν προταθεί για σχεσιακά δεδομένα και εστιάζουμε στις ανεπάρκειες και περιορισμούς που αυτά επιβάλλουν. Στις ενότητες 1.3 και 1.4 επικεντρωνόμαστε στους αλγορίθμους εξατομίκευσης που έχουν προταθεί από την πλευρά των χρηστών και των παρόχων περιεχομένου αντιστοίχως και επιχειρούμε να αναδείξουμε κάποια από τα προβλήματα που μένουν ανοικτά και με τα οποία η παρούσα διατριβή καταπιάνεται.

1.2 Διαχείριση Δεδομένων Εξαρτώμενων από Προτιμήσεις

1.2.1 Προβλήματα και περιορισμοί

Παράδειγμα. Θα περιγράψουμε τα προβλήματα που προκύπτουν από την αποτίμηση ερωτημάτων με προτιμήσεις κάνοντας χρήση του υφιστάμενου μοντέλου ερωτημάτων σε σχεσιακά συστήματα βάσεων δεδομένων με τη βοήθεια ενός παραδείγματος. Το Σχήμα 1.2 δείχνει μια φόρμα αναζήτησης για μια ιστοσελίδα αναζήτησης αγγελιών μεταχειρισμένων αυτοκινήτων.

Κάνοντας χρήση της συγκεκριμένης φόρμας οι χρήστες ενός τέτοιου συστήματος καλούνται να αποτυπώσουν τα κριτήρια αναζήτησής τους. Ας φανταστούμε λοιπόν κάποιον χρήστη μιας τέτοιας εφαρμογής ο οποίος ενδιαφέρεται να αγοράσει ένα μεταχειρισμένο αυτοκίνητο. Ο χρήστης αυτός συνήθως έχει στο μυαλό του μια γενική εικόνα του τι θέλει να αγοράσει, για παράδειγμα θα προτιμούσε να μην πληρώσει πάνω από 10000 ευρώ, για ένα αυτοκίνητο που είναι νεότερο του 2007, έχει μεσαίο κυβισμό και δεν έχει διανύσει παραπάνω από 80000 km. Συνεπώς ο χρήστης αυτός θα επιλέξει από μια φόρμα σαν αυτή του Σχήματος 1.2 τα κριτήρια αναζήτησής του έστω: τιμή $\leq$ 10000, έτος κατασκευής $\geq$ 2007, 1400 $\leq$ κυβισμός $\leq$ 1600 και km $\leq$ 80000. Επίσης για τις ανάγκες του παραδείγματος έστω ότι η βάση δεδομένων περιέχει τα αυτοκίνητα που φαίνονται στον Πίνακα 1.1.



Σχήμα 1.2: *Εφαρμογή αναζήτησης μεταχειρισμένων αυτοκινήτων*

car_id	Μάρκα	Μοντέλο	Έτος	Κυβισμός	km	Τιμή
c_1	Toyota	Corolla	2006	1400	80000	8000
c_2	Honda	Civic	2008	1600	100000	9000
c_3	Opel	Corsa	2011	1400	50000	11000
c_4	Ford	Focus	2007	1800	90000	10000
c_5	Mazda	2	2009	1500	30000	10500

Πίνακας 1.1: *Πίνακας μιας Σχεσιακής Βάσης Δεδομένων με μεταχειρισμένα αυτοκίνητα*

Δυστυχώς κανένα από τα προσφερόμενα αυτοκίνητα δεν ικανοποιεί απόλυτα όλα τα κριτήρια αναζήτησης και συνεπώς η βάση δεν θα επιστρέψει κανένα αποτέλεσμα. Συνεπώς στην περίπτωση αυτή ο χρήστης θα αναγκαστεί να αναπροσαρμόσει κατάλληλα τα κριτήρια αναζήτησης που θέτει μέχρι να λάβει ικανοποιητικά αποτελέσματα (εφόσον υπάρχουν). Για παράδειγμα αποδεχόμενος να πληρώσει λίγο παραπάνω (π.χ. 10500 ευρώ) το σύστημα θα του επιστρέψει το αυτοκίνητο c_5, ενώ αλλάζοντας το έτος κατασκευής σε 2006, το σύστημα θα βρει επίσης το αυτοκίνητο c_1. Γενικά κοιτώντας τα χαρακτηριστικά των διαθέσιμων αυτοκινήτων, είναι εύκολο να παρατηρήσει κανείς ότι υπάρχουν αρκετά διαθέσιμα αυτοκίνητα με χαρακτηριστικά πολύ κοντά σε αυτά που προσδιορίστηκαν στο αρχικό ερώτημα. Συνεπώς, θα ήταν επιθυμητό για ένα σύστημα να μπορεί να αντιμετωπίζει τέτοιες καταστάσεις και να επιστρέφει κοντινά αποτελέσματα τα οποία σε μεγάλο αν και όχι απόλυτο βαθμό ικανοποιούν τις προτιμήσεις ενός

χρήστη, χωρίς ο χρήστης να χρειάζεται να καταφύγει σε πολλαπλές αναζητήσεις.

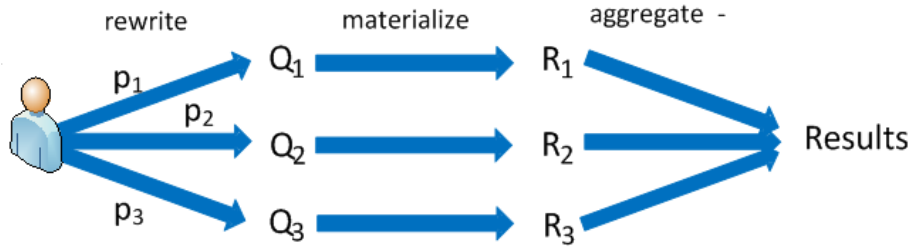
Αντίστροφα, προκύπτουν καταστάσεις όπου ένας χρήστης βρίσκεται αντιμέτωπος με πληθώρα αποτελεσμάτων, ιδίως αν τα κριτήρια που έχει θέσει είναι αρκετά χαλαρά. Για παράδειγμα αν στην περίπτωσή μας ο χρήστης θέσει ως αποκλειστικό κριτήριο τη συνθήκη $\mathrm{km} \leq 100000$, τότε όλα τα αυτοκίνητα του Πίνακα 1.1 ικανοποιούν το συγκεκριμένο κρίτηριο. Σε αυτή την περίπτωση, το σύστημα θα ήταν καλό να μπορεί να φιλτράρει κάποια αποτελέσματα που δεν είναι ιδιαίτερα καλά, ή να παρουσιάζει τα αποτελέσματα ταξινομημένα με κάποιο βαθμό σχετικότητας προς τις προτιμήσεις του χρήστη. Για παράδειγμα, τα αυτοκίνητα θα μπορούσαν να ταξινομηθούν με κριτήριο την τιμή τους ξεκινώντας από το πιο φτηνό.

Μια άλλη ενδιαφέρουσα παρατήρηση που μπορούμε να κάνουμε για τέτοιου τύπου εφαρμογές είναι ότι τα κριτήρια που συχνά θέτει ένας χρήστης δεν είναι απόλυτα, αλλά είναι ορθότερο να θεωρηθούν ως προαιρετικά κριτήρια που θα ήταν καλό να ικανοποιούνται όσο το δυνατόν περισσότερο, σε πλήρη αντιδιαστολή με το παραδοσιακά αυστηρό μοντέλο αποτίμησης ερωτήσεων που θεωρεί όλα τα κριτήρια ως υποχρεωτικά. Ακολουθώντας έναν τέτοιο τρόπο προσέγγισης θα μπορούσε ένα σύστημα να ταξινομεί τα αποτελέσματα σε σχέση με τον αριθμό των κριτηρίων (προτιμήσεων) που ικανοποιεί. Επιπλέον, διαφορετικές προτιμήσεις είναι δυνατόν να έχουν διαφορετική βαρύτητα ή προτεραιότητα. Για παράδειγμα, αν για κάποιον χρήστη το βασικότερο κριτήριο είναι η τιμή θα ήταν καλό τα επιστρεφόμενα αποτελέσματα να επιστρέφονται ταξινομημένα ως προς αυτήν, ενώ αν για κάποιον άλλον είναι η παλαιότητα, τότε αντίστοιχα τα αποτελέσματα θα πρέπει να επιστρέφονται ταξινομημένα ως προς το έτος κατασκευής. Είναι δηλαδή φανερό ότι διαφορετικές προτιμήσεις είναι δυνατό να έχουν μεγαλύτερη ή μικρότερη σημασία στην τελική βαθμολογία.

Τέλος, πολλές φορές είναι χρήσιμο ένα τέτοιο σύστημα να είναι σε θέση να προτείνει αποτελέσματα με παρόμοια χαρακτηριστικά. Για παράδειγμα, με βάση το Σχήμα 1.2 θα μπορούσαμε να βγάλουμε το σύμπερασμα ότι ο χρήστης προτιμά τα ιαπωνικά αυτοκίνητα. Σε αυτή την περίπτωση το σύστημα ενδεχομένως θα μπορούσε να προτείνει και άλλες ιαπωνικές μάρκες αυτοκινήτων.

Η ανάγκη για πιο εύελικτη αποτίμηση ερωτημάτων. Σύμφωνα με το τυπικό μοντέλο ερωτημάτων σε μια βάση, όλες οι συνθήκες ενός ερωτήματος αντιμετωπίζονται ως ισοδύναμες και τα αποτελέσματα είναι όλα το ίδιο (απόλυτα) σχετικά με το αρχικό ερώτημα. Κάθε αποτέλεσμα πρέπει απαραίτητα να ικανοποιεί όλες τις συνθήκες του ερωτήματος, ειδάλλως απορρίπτεται.

Δυστυχώς, όπως είδαμε και παραπάνω, αυτό το μοντέλο δημιουργεί αρκετά προβλήματα για πολλές σύγχρονες εφαρμογές που έχουν πιο χαλαρές ή συγκεχυμένες απαιτήσεις από τα δεδομένα. Ένας βασικός λόγος είναι ότι συχνά ο τρόπος που πρέπει να διαμορφωθεί μια ερώτηση καθώς και το επιδιωκόμενο αποτέλεσμα δεν είναι εύκολο να προσδιοριστούν, είτε γιατί οι χρήστες των συστημάτων δεν έχουν μια αυστηρά καθορισμένη εικόνα για αυτό που αναζητούν, είτε γιατί δεν έχουν επίγνωση του σχήματος ή της κατανομής των δεδομένων, είτε για όλους τους παραπάνω λόγους μαζί. Σε εφαρμογές όπως η παραπάνω, η ενσωμάτωση προτιμήσεων στην αποτίμηση των ερωτημάτων σε μια βάση δείχνει να είναι μια φιλόδοξη και αρκετά ελπιδοφόρα προσέγγιση. Οι προτιμήσεις χαλαρώνουν το παραδοσιακά αυστηρό μοντέλο ερωτημάτων σε μια βάση



Σχήμα 1.3: *Σχηματική απεικόνιση μιας plug-in μεθόδου*

δεδομένων επιτρέποντας την ταξινόμηση των αποτελεσμάτων με διάφορα κριτήρια, την απόρριψη κάποιων αποτελεσμάτων, ή ακόμα και την εύρεση αποτελεσμάτων που δεν είναι απόλυτα συμβατά με τα δοθέντα κριτήρια. Επιπλέον, δίνουν τη δυνατότητα στις εφαρμογές να παρέχουν διαφορετική οπτική των δεδομένων σε κάθε χρήστη, ανάλογα με τα ενδιαφέροντα και τις προτιμήσεις του.

Μέθοδοι και συστήματα εξατομίκευσης. Με βάση τα παραπάνω είναι φανερό ότι η ενσωμάτωση των προτιμήσεων κατά την αποτίμηση ερωτημάτων σε μια βάση δεν είναι εύκολο να μετασχηματιστεί σε ένα κλασικό τύπο ερωτήματος. Παράλληλα, θέτει πλήθος περιορισμών που αντίκειται στον τρόπο που οι άνθρωποι αντιλαμβάνονται την έννοια της προτίμησης, ως μια ευχή ή μια χαλαρή συνθήκη που θα ήθελαν να ικανοποιείται, αν και όχι απαραίτητα.

Με σκοπό να αντιμετωπιστούν τα παραπάνω προβλήματα, στη βιβλιογραφία έχουν προταθεί διάφορες προσεγγίσεις. Σε αδρές γραμμές οι μέθοδοι αυτοί μπορούν να διακριθούν στις παρακάτω δύο κατηγορίες: (α) μέθοδοι αποτίμησης ερωτημάτων σε επίπεδο εφαρμογής (plug-in methods), και (β) μέθοδοι σε επίπεδο συστήματος (built-in ή native methods). Παρακάτω παρουσιάζουμε περιληπτικά κάποιες από τις προταθείσες μεθόδους και τους περιορισμούς τους.

Οι μέθοδοι σε επίπεδο εφαρμογής (πχ. [4, 36, 20, 40]) παρέχουν ένα επίπεδο (layer) εκτός της βάσης δεδομένων που επιτρέπει την έκφραση και την εκτέλεση ερωτημάτων με προτιμήσεις μεταφράζοντάς κάθε προτίμηση ως συμβατική συνθήκη πάνω σε μια σχεσιακή βάση. Το βασικό πλεονέκτημα αυτών των μεθόδων είναι η απλότητά τους και η ευκολία υλοποίησής τους για συνηθισμένες (συνήθως απλές) εφαρμογές. Σχηματικά, ο τρόπος με τον οποίο μια plug-in μέθοδος εκτελείται φαίνεται στο Σχήμα 1.3. Αρχικά επιλέγονται οι κατάλληλες προτιμήσεις που πρέπει να ενσωματωθούν στο ερώτημα και σχηματίζονται τα αντίστοιχα υποερωτήματα (φάση rewrite). Στη συνέχεια κάθε υποερώτημα κατευθύνεται προς εκτέλεση προς τη βάση (φάση materialize). Τα αποτελέσματα κάθε υποερωτήματος λαμβάνουν μια βαθμολογία που αντικατοπτρίζει σε ποιο βαθμό ικανοποιούν την αντίστοιχη προτίμηση. Τελικά, τα επιμέρους αποτελέσματα συναθροίζονται και παράγεται το τελικό αποτέλεσμα του ερωτήματος (φάση aggregate).

Στην πράξη η ενσωμάτωση των προτιμήσεων στο μοντέλο ερωτημάτων μπορεί να γίνει με πολλούς τρόπους και σχετίζεται με την εφαρμογή, το ερώτημα και τον χρήστη. Για παράδειγμα, κάποιες εργασίες [40, 41] απαιτούν τουλάχιστον k από τις n προτιμήσεις να ικανοποιούνται ώστε κάποιο αποτέλεσμα να επιστραφεί και συνεπώς χτίζουν υποσύνολα k υποερωτημάτων με προτιμήσεις. Η εργασία [43] επεκτείνει τα ερωτήματα

ενός χρήστη με επιπλέον συνθήκες προτίμησης οι οποίες χρησιμοποιούνται με σκοπό τη μείωση του αριθμού των επιστρεφόμενων αποτελεσμάτων. Αντιστρόφως η εργασία [52] προτείνει τεχνικές με βάση τις οποίες αν ένα ερώτημα δεν επιστρέφει αρκετά αποτελέσματα, οι συνθήκες του ερωτήματος χαλαρώνουν μέχρι να βρεθεί ικανοποιητικός αριθμός αποτελεσμάτων.

Σε σχέση με τη συνάθροιση των αποτελεσμάτων, επίσης μπορούν να ακολουθηθούν διάφορες προσεγγίσεις. Για παράδειγμα τα αποτελέσματα μπορούν να ταξινομηθούν με βάση μια συνάρτηση συνάθροισης και να επιλεγούν τα top-k κορυφαία [27, 28, 5], ή θεωρώντας όλες τις προτιμήσεις ισοδύναμες να επιστραφούν τα βέλτιστα αποτελέσματα κατά Pareto όπως στην περίπτωση των ερωτημάτων κορυφογραμμής (skyline queries) [14, 20, 55]. Σε άλλες περιπτώσεις [40], τα αποτελέσματα μπορούν να ταξινομηθούν με κριτήριο το πόσες ή πόσο σημαντικές προτιμήσεις ικανοποιούν. Γενικότερος σκοπός όλων των προτεινόμενων μεθόδων είναι να επιλέξουν τα 'καλύτερα' ή 'πιο ενδιαφέροντα' αποτελέσματα για κάθε χρήστη με βάση τις προτιμήσεις που έχει δηλώσει στο σύστημα.

Όπως προκύπτει από τα παραπάνω, ο τρόπος που οι προτιμήσεις πρέπει να μεταφραστούν σε στοιχεία του ερωτήματος, η ροή εκτέλεσης του ερωτήματος καθώς και η συναρτήσεις βαθμολόγησης και συνάθροισης είναι κωδικοποιημένα στη λογική της εφαρμογής, πράγμα που έχει ως συνέπεια οι plug-in μέθοδοι να επιτρέπουν περιορισμένο έλεγχο για τους προγραμματιστές εφαρμογών. Το γεγονός αυτό δυσχεραίνει την ανάπτυξη και συντήρηση εφαρμογών που χρησιμοποιούν προτιμήσεις ακολουθώντας αυτή την προσέγγιση. Επιπλέον, η υλοποίηση κάθε ενός από τα τρία στάδια εκτέλεσης μια plug-in μεθόδου σε προγραμματιστικό επίπεδο μπορεί να είναι αυθαίρετα πολύπλοκη και σύνθετη. Το ιδανικό για μια εφαρμογή θα ήταν να μπορεί να προσδιορίζει το ερώτημα ή/και τις προτιμήσεις ενός χρήστη με δηλωτικό τρόπο, αφήνοντας στο σύστημα τη βελτιστοποίηση και εκτέλεση του ερωτήματος.

Ένα άλλο σημαντικό πρόβλημα των plug-in μεθόδων είναι ότι για να εκφραστεί ένα ερώτημα προτίμησης συνήθως απαιτούνται πολλές ερωτήσεις, ενδεχομένως με επαναλαμβανόμενα κοινά υποερωτήματα, επηρεάζοντας έτσι τον τρόπο που εκτελείται το ερώτημα και την απόδοσή του από πλευράς κόστους εκτέλεσης. Γενικότερα, οι μέθοδοι αυτές ανήκουν καθαρά στο επίπεδο εφαρμογής (βλέπε σχήμα 1.1) καθώς αντιμετωπίζουν το σύστημα διαχείρισης δεδομένων ως μαύρο κουτί, με δεδομένο ότι η προτεινόμενη μέθοδος αποτίμησης ερωτημάτων με προτιμήσεις είναι εντελώς αποκομμένη από τη βάση δεδομένων του συστήματος. Συνεπώς, οι μέθοδοι αυτοί δεν μπορούν να εκμεταλλευτούν τις όποιες βελτιστοποιήσεις παρέχονται από τη δεικτοδότηση των δεδομένων ή τον τρόπο εκτέλεσης ενός ερωτήματος από έναν 'παραδοσιακό' βελτιστοποιητή ερωτημάτων (query optimizer).

Με βασικό στόχο να αντιμετωπίσουν τα προβλήματα αποδοτικότητας των plug-in μεθόδων, έχουν επίσης προταθεί μέθοδοι σε επίπεδο συστήματος. Οι μέθοδοι αυτοί (native methods) εισάγουν νέους τελεστές προτίμησης μέσα στον πυρήνα της βάσης δεδομένων. Για παράδειγμα, η RankSQL [46] επεκτείνει τη σχεσιακή άλγεβρα με ένα νέο τελεστή (rank operator) που επιτρέπει το pipelining κατά τη διάρκεια της εκτέλεσης ενός ερωτήματος, ενώ παράλληλα μπορεί να συνδυαστεί υπό προϋποθέσεις με τους υπόλοιπους σχεσιακούς τελεστές. Με αυτό τον τρόπο επιτυγχάνεται η βελτιστοποίηση στην εκτέλεση top-k ερωτημάτων. Παρομοίως, στην εργασία [20] προτείνεται ένας

νέος τελεστής με την ονομασία winnow ο οποίος επιλέγει τις εγγραφές (tuples) που αντιστοιχούν στο βέλτιστο σύνολο κατά Pareto.

Προφανώς κάθε μία από τις παραπάνω προσεγγίσεις επιτυγχάνει πολύ καλύτερη απόδοση σε σχέση με μια plug-in μέθοδο για το συγκεκριμένο τύπο ερωτήματος με προτιμήσεις στον οποίο επικεντρώνεται. Όμως σε αυτή την περίπτωση, ο εκάστοτε προτεινόμενος τελεστής αντιμετωπίζει δύο σαφώς διαφορετικές λειτουργίες, την αποτίμηση της προτίμησης - βαθμολόγηση των αποτελεσμάτων και την επιλογή των πιο 'ενδιαφέροντων' αποτελεσμάτων ως μία ενιαία λειτουργία. Για παράδειγμα ο rank operator υπολογίζει τα scores κάθε εγγραφής και επιλέγει τις καλύτερες στο ίδιο βήμα εκτέλεσης. Συνεπώς, οι μέθοδοι αυτές έχουν περιορισμένη ευελιξία, καθώς η λογική εκτέλεσης της προτίμησης είναι στενά δεμένη με τον τελεστή προτίμησης, και συνεπώς μπορούν να εφαρμοστούν μόνο για συγκεκριμένους τύπους ερωτημάτων. Επίσης, οι native μέθοδοι απαιτούν την αλλαγή του πυρήνα της βάσης δεδομένων, κάτι που σε πολλές περιπτώσεις δεν είναι εφικτό ή επιθυμητό.

Αναγνωρίζοντας το πρόβλημα αυτό, μια πρόσφατη εργασία [45] επιχειρεί να διευκολύνει τους προγραμματιστές εφαρμογών ώστε να μπορούν να δηλώνουν διαφορετικούς αλγορίθμους προτίμησης (πχ. top-k ή skyline ερωτήματα) πάνω από μία σχεσιακή βάση. Οι συγγραφείς προτείνουν ένα νέο πλαίσιο με την ονομασία FlexPref το οποίο εισάγει πιο περιορισμένες αλλαγές στον πυρήνα της βάσης δεδομένων, ενώ παράλληλα επιτρέπει την εκτέλεση διαφορετικών τύπων ερωτημάτων με προτιμήσεις, έχοντας απόδοση που κυμαίνεται ανάμεσα σε αυτή των plug-in και των native μεθόδων.

1.2.2 Συνεισφορά της διατριβής

Όπως προκύπτει από τα παραπάνω, οι υπάρχουσες προσεγγίσεις αντιμετωπίζουν το πρόβλημα μάλλον αποσπασματικά, χωρίς να προτείνουν μια πιο γενικευμένη μεθοδολογία ενσωμάτωσης των προτιμήσεων στο σχεσιακό μοντέλο με ευέλικτο και επεκτάσιμο τρόπο. Με βάση αυτή την παρατήρηση, στην παρούσα διατριβή προτείνουμε ένα νέο σύστημα εξατομίκευσης για σχεσιακά δεδομένα, με την ονομασία *PrefDB*. Παρακάτω περιγράφουμε εν συντομία τα βασικά χαρακτηριστικά του προτεινόμενου συστήματος.

Μοντέλο Δεδομένων και Αναπαράστασης Προτιμήσεων. Το σύστημα *PrefDB* ακολουθεί ένα νέο μοντέλο δεδομένων και σχεσιακή άλγεβρα τα οποία λαμβάνουν υπόψη τους τις προτιμήσεις των χρηστών αντιμετωπίζοντας την προτίμηση ως 'πρώτο' πολίτη μιας βάσης δεδομένων. Προτείνουμε ένα νέο μοντέλο αναπαράστασης προτιμήσεων για σχεσιακά δεδομένα, το οποίο επιτρέπει τον ορισμό προτιμήσεων σε τρεις άξονες: (α) μια συνθήκη προτίμησης (condition) με την οποία καθορίζονται οι εγγραφές που επηρεάζονται από μια προτίμηση, (β) μια συνάρτηση βαθμολόγησης των εγγραφών (ranking), και (γ) ένα βαθμό εμπιστοσύνης (confidence) που καθορίζει πόσο ισχυρή και βέβαιη είναι η συγκεκριμένη προτίμηση.

Αξιζεί να σημειωθεί ότι ενώ προηγούμενα μοντέλα αναπαράστασης προτιμήσεων [3, 40, 46, 45] έχουν μελετήσει τις πρώτες δύο διαστάσεις, η προτεινόμενη προσέγγιση είναι η πρώτη που τα χειρίζεται με ενιαίο τρόπο και συνδέει με το ποσοτικό μοντέλο προτιμήσεων (βλέπε Κεφάλαιο 2.1). Παράλληλα, το προτεινόμενο μοντέλο προτιμήσεων επιτρέπει για παράδειγμα να συνδυάσει κάποιος μια συνθήκη σε ένα γνώρισμα (π.χ. έτος

κατασκευής ≥ 2005), με μια συνάρτηση βαθμολόγησης σε κάποιο άλλο γνώρισμα (π.χ. τα φθηνότερα αυτοκίνητα είναι προτιμότερα). Επιπλέον, ο βαθμός εμπιστοσύνης, μία εγγενής ιδιότητα των προτιμήσεων, εισάγεται για πρώτη φορά στην παρούσα εργασία. Για παράδειγμα, επιστρέφοντας στο παράδειγμα της προηγούμενης ενότητας, ο βαθμός εμπιστοσύνης μας επιτρέπει να διαχωρίσουμε μια προτίμηση για μια συγκεκριμένη μάρκα αυτοκινήτων που ο ίδιος ο χρήστης έχει δηλώσει ρητά, από μια προτίμηση που εξήγαγε το σύστημα βασισμένο για παράδειγμα στο ιστορικό αναζήτησης του χρήστη ή στον χρόνο που δαπάνησε καθώς εξέταζε κάθε διαθέσιμο αυτοκίνητο. Είναι φανερό ότι στη δεύτερη περίπτωση η προτίμηση αυτή έχει ένα βαθμό αβεβαιότητας που σχετίζεται με τον τρόπο που εξήχθη η συγκεκριμένη προτίμηση.

Σε σχέση με το μοντέλο δεδομένων, σε κάθε εγγραφή ανατίθενται δύο επιπλέον γνωρίσματα: ένα σκορ και ο αντίστοιχος βαθμός εμπιστοσύνης. Επιπλέον επεκτείνουμε τους κλασικούς σχεσιακούς τελεστές ώστε να διαχειρίζονται σκορ και βαθμούς εμπιστοσύνης. Για παράδειγμα, ο τελεστής σύζευξης (join operator), συνενώνει δύο εγγραφές και υπολογίζει ένα νέο ζεύγος σκορ-βεβαιότητας, συνδυάζοντας τα σκορ-τιμές εμπιστοσύνης των επιμέρους εγγραφών. Παράλληλα, η νέα άλγεβρα που προτείνουμε εισάγει έναν νέο τελεστή προτίμησης (prefer operator) ο οποίος αποτιμά μια προτίμηση πάνω σε μία σχέση, με άλλα λόγια λαμβάνοντας ως είσοδο μια σχέση και μια προτίμηση, ο τελεστής προτίμησης δίνει ως έξοδο τη σχέση με ενημερωμένα σκορ και τιμές εμπιστοσύνης.

Πρότυπο σύστημα εξατομίκευσης. Τα προτεινόμενα μοντέλα δεδομένων και αναπαράστασης προτιμήσεων καθώς και η νέοι σχεσιακοί τελεστές έχουν υλοποιηθεί σε ένα πρωτότυπο σύστημα. Το σύστημα *PrefDB* παρέχει ένα πλαίσιο εξατομίκευσης το οποίο διευκολύνει τον εμπλουτισμό των ερωτημάτων με προτιμήσεις, έτσι ώστε τα αποτελέσματα ενός ερωτήματος να ταιριάζουν κατά το δυνατόν με τις καθορισμένες προτιμήσεις. Παράλληλα, επιτρέπει πιο απλοποιημένη ανάπτυξη εφαρμογών που απαιτούν επεξεργασία προτιμήσεων πάνω από σχεσιακά δεδομένα. Αντί να συνδέει την ενσωμάτωση και τον τρόπο αποτίμησης των προτιμήσεων με τη λογική της εφαρμογής όπως κάνουν οι μέθοδοι plug-in, το σύστημα *PrefDB* υποστηρίζει τον ορισμό διαφορετικών τύπων ερωτημάτων με προτιμήσεις με καθαρά δηλωτικό τρόπο, με τρόπο παρόμοιο προς αυτόν που η γλώσσα SQL επιτρέπει για παραδοσιακά ερωτήματα. Το σύστημα αναλαμβάνει να εκτελέσει τα ερωτήματα αυτά και να επιστρέψει τα πιο σχετικά αποτελέσματα, χωρίς καμία ανάμειξη της λογικής της εφαρμογής. Παράλληλα, όπως θα φανεί και παρακάτω, η υβριδική υλοποίηση του συστήματος μεταφέρει την αποτίμηση προτιμήσεων πιο κοντά στον πυρήνα της βάσης δεδομένων, επιτρέποντας έτσι βελτιστοποιήσεις σε επίπεδο τελεστή, χωρίς όμως να απαιτεί καμία απολύτως αλλαγή στον πηγαίο κώδικα του συστήματος. Με αυτό τον τρόπο, το σύστημα *PrefDB* είναι πλήρως επεκτάσιμο και συμβατό με οποιοδήποτε σχεσιακό σύστημα διαχείρισης βάσεων δεδομένων.

Εκτέλεση ερωτημάτων με προτιμήσεις. Κατά την εκτέλεση ενός ερωτήματος χρησιμοποιούνται τόσο η συνθήκη όσο και η συνάρτηση βαθμολόγησης μιας προτίμησης. Η συνθήκη προτίμηση δρα ως μια χαλαρή συνθήκη που καθορίζει ποιες εγγραφές θα βαθμολογηθούν. Σε αντίθεση όμως με προηγούμενες εργασίες, δεν απορρίπτει καμία εγγραφή από το αποτέλεσμα του ερωτήματος. Κατ΄ αυτό τον τρόπο,

το *PrefDB* διαχωρίζει την αποτίμηση μιας προτίμησης από το φιλτράρισμα των εγγραφών. Αυτός ο διαχωρισμός είναι μια σημαντική διαφορά σε σχέση με προηγούμενες προσεγγίσεις, καθώς μας επιτρέπει να ορίσουμε αλγεβρικές ιδιότητες για τον τελεστή προτίμησης και να ακολουθήσουμε γενικής χρήσης στρατηγικές εκτέλεσης και βελτιστοποίησης, οι οποίες είναι ανεξάρτητες από τον τύπο προτίμησης ή ερωτήματος ή από το είδος του αναμενόμενου αποτελέσματος.

Το σύστημα μας ακολουθεί μία υβριδική προσέγγιση κατά την επεξεργασία ενός ερωτήματος, σε σχέση με τις plug-in και native μεθόδους. Αρχικά, συνθέτει ένα εκτεταμένο πλάνο εκτέλεσης που περιέχει τους νέους τελεστές και τον τελεστή προτίμησης. Στη συνέχεια ακολουθεί η βελτιστοποίηση του πλάνου εκτέλεσης και τελικά το βελτιστοποιημένο πλάνο εκτελείται από τη μηχανή εκτέλεσης. Σε αυτό το βήμα, η γενική στρατηγική εκτέλεσης που ακολουθούμε επιχειρεί να αναμείξει την εκτέλεση του ερωτήματος με την αποτίμηση των προτιμήσεων, ενώ παράλληλα χρησιμοποιεί κατά το δυνατό την υποκείμενη μηχανή εκτέλεσης της βάσης δεδομένων για να εκτελέσει κομμάτια της ερώτησης που δεν σχετίζονται με προτιμήσεις.

Βελτιστοποίηση πλάνων εκτέλεσης. Δοθέντος ενός ερωτήματος με προτιμήσεις, ο σκοπός της βελτιστοποίησης είναι να ελαχιστοποιήσει το κόστος που σχετίζεται με την αποτίμηση προτιμήσεων. Βασισμένοι στις αλγεβρικές ιδιότητες του τελεστή προτίμησης, αρχικά εφαρμόζουμε ένα σύνολο ευριστικών κανόνων με στόχο να ελαχιστοποιήσουμε τον αριθμό εγγραφών που επηρεάζονται από τους τελεστές προτίμησης. Στη συνέχεια προτείνουμε μια μεθοδολογία βελτιστοποίησης βασισμένης στο κόστος. Χρησιμοποιώντας το πλάνο εκτέλεσης που παράγει το πρώτο βήμα βελτιστοποίησης ως είσοδο και ένα μοντέλο κόστους για την αποτίμηση προτιμήσεων, ο βελτιστοποιητής κάνει εκτίμηση του κόστους που θα έχει η αποτίμηση διαφορετικών εναλλακτικών πλάνων εκτέλεσης. Δύο αλγόριθμοι βελτιστοποίησης προτείνονται: ένας βασισμένος στη λογική του δυναμικού προγραμματισμού και ένας άπληστος αλγόριθμος.

Πειραματική αξιολόγηση. Έχουμε διεξαγάγει διεξοδική πειραματική αξιολόγηση των προτεινόμενων τεχνικών βελτιστοποίησης και εκτέλεσης ερωτημάτων σε δύο πραγματικά σύνολα δεδομένων. Συγκεκριμένα, αξιολογούμε τους δύο αλγορίθμους βελτιστοποίησης και συγκρίνουμε την απόδοσή τους εναντίον δύο plug-in μεθόδων που έχουμε υλοποιήσει. Επιπλέον, συγκρίνουμε την αποτελεσματικότητα των τεχνικών βελτιστοποίησης με το βέλτιστο πλάνο εκτέλεσης όπως αυτό θα προέκυπτε από μια εξαντλητική εκτίμηση κόστους σε όλο τον χώρο των πιθανών πλάνων εκτέλεσης. Τέλος, έχουμε διεξαγάγει πειραματική ανάλυση ευαισθησίας (sensitivity analysis) σε σχέση με ένα πλήθος παραμέτρων όπως ο αριθμός joins, το πλήθος των αποτελεσμάτων ενός ερωτήματος, ο αριθμός, η κατανομή και η επιλεκτικότητα (selectivity) των προτιμήσεων.

1.3 Αλγόριθμοι Εξατομίκευσης από την πλευρά των χρηστών

1.3.1 Προβλήματα και περιορισμοί

Σε πολλές εφαρμογές, οι προτιμήσεις ενός χρήστη δεν ισχύουν καθολικά, αλλά μόνο υπό κάποιες συνθήκες που εξαρτώνται από το τρέχον περιβάλλον χρήσης (context). Για παράδειγμα ένας χρήστης που είναι πεζός μπορεί να αναζητά εστιατόρια ανεξαρτήτως κουζίνας σε ακτίνα 1 km, αλλά να τον ενδιαφέρουν μόνο εστιατόρια ασιατικής κουζίνας ανεξαρτήτως απόστασης αν διαθέτει μεταφορικό μέσο. Σε αυτή την περίπτωση δεν υπάρχει ξεκάθαρη προτίμηση μεταξύ των γνωρισμάτων απόσταση και κουζίνα, καθώς το ένα μπορεί να υπερισχύει έναντι του άλλου σε διαφορετικές καταστάσεις χρήσης.

Το γεγονός αυτό έχει καίρια σημασία για εφαρμογές που βασίζονται σε υπηρεσίες βασισμένες στη θέση του χρήστη (location-based services - LBS), όπως για παράδειγμα τα συστήματα πλοήγησης GPS ή εφαρμογές όπως το Yelp[1] και το Foursquare[2]. Η χρήση τέτοιων συστημάτων γίνεται ολοένα και πιο ευρεία, καθώς η διαθεσιμότητα έξυπνων κινητών συσκευών (smartphones) και ασυρμάτων δικτύων υψηλής ταχύτητας επιτρέπει τη συλλογή και επεξεργασία δεδομένων που περιλαμβάνουν τη γεωγραφική θέση των χρηστών μιας υπηρεσίας, την ώρα/ημέρα, τις καιρικές συνθήκες και το κοινωνικό του περιβάλλον. Μία τυπική τέτοια εφαρμογή, χτίζει σταδιακά ένα προφίλ για κάθε χρήστη, το οποίο περιλαμβάνει για παράδειγμα τα εστιατόρια ή τα μπαρς στα οποία συχνάζει αυτός και οι φίλοι του. Στη συνέχεια, ο χρήστης υποδηλώνει μέσω του κινητού του τηλεφώνου την τρέχουσα γεωγραφική του θέση και το σύστημα εξατομίκευσης, βασισμένο στις προτιμήσεις του, το τρέχον context (π.χ. τη γεωγραφική του θέση) και άλλα γνωρίσματα (π.χ. αξιολογήσεις χρηστών, δεδομένα κυκλοφορίας στους δρόμους) ή περιορισμούς (π.χ. εγγύτητα σε σταθμό του μετρό) μπορεί να του προτείνει κοντινά σημεία ενδιαφέροντος.

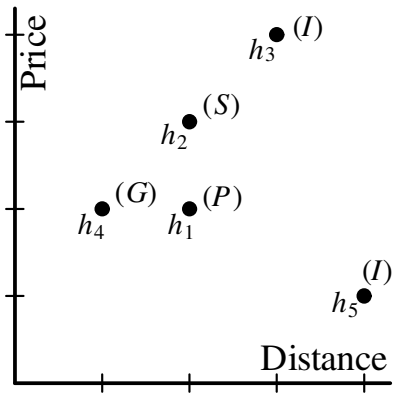
Στη βιβλιογραφία έχουν παρουσιαστεί συστήματα εξατομίκευσης που εξαρτώνται από το περιβάλλον χρήσης (context). Στην εργασία [3], δοθέντος ενός συνόλου προτιμήσεων ορισμένων από μια ομάδα χρηστών όπου οι προτιμήσεις εξαρτώνται από το περιβάλλον χρήσης (contextual preferences), τα αποτελέσματα ενός ερωτήματος αναδιατάσσονται ώστε να συμφωνούν κατά το δυνατό με τις προτιμήσεις που ισχύουν στο τρέχον περιβάλλον χρήσης. Η εργασία [66] αντιμετωπίζει το ίδιο πρόβλημα ακολουθώντας διαφορετικό μοντέλο αναπαράστασης προτιμήσεων. Και οι δύο παραπάνω εργασίες επιστρέφουν τα αποτελέσματα εκείνα που έχουν την υψηλότερη βαθμολογία για το τρέχον περιβάλλον χρήσης, ακολουθώντας το παράδειγμα των ερωτημάτων top-k.

Στην παρούσα διατριβή εστιάζουμε στο πρόβλημα της αποτίμησης ερωτημάτων με προτιμήσεις που εξαρτώνται από το περιβάλλον χρήσης ακολουθώντας το παράδειγμα των ερωτημάτων κορυφογραμμής. Τα ερωτήματα κορυφογραμμής χρησιμοποιούνται πολύ συχνά σε εφαρμογές που απαιτούν τη λήψη αποφάσεων με βάση πολλαπλά κριτήρια (multi-criteria decision making). Στις περιπτώσεις αυτές δεν είναι εύκολο ή επιθυμητό να οριστεί μία συναθροιστική συνάρτηση που θα επιστρέψει τα καλύτερα

[1] http://www.yelp.com
[2] http://www.foursquare.com

Hotel	Price	Distance	Amenity
h_1	200	10	Pool (P)
h_2	300	10	Spa (S)
h_3	400	15	Internet (I)
h_4	200	5	Gym (G)
h_5	100	20	Internet (I)

(α′) Βάση δεδομένων



(β′) Απεικόνιση σε 2 διαστάσεις

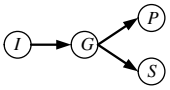
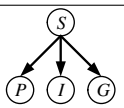
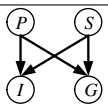
Σχήμα 1.4: *Βάση δεδομένων με πληροφορίες για ξενοδοχεία*

αποτελέσματα. Αντιθέτως, μπορεί να θεωρήσει κανείς ότι κάθε χρήστης έχει μια σειρά από προτιμήσεις πάνω σε γνωρίσματα των δεδομένων οι οποίες ορίζουν επιμέρους κατατάξεις των αποτελεσμάτων. Επιστρέφοντας στο παράδειγμα της προηγούμενης ενότητας, θα μπορούσαμε να κατατάξουμε τα αποτελέσματα του ερωτήματος (μεταχειρισμένα αυτοκίνητα) με κριτήριο την τιμή τους (από το πιο φτηνό στο πιο ακριβό), το έτος κατασκευής τους (από το νεότερο στο παλαιότερο) ή τα χιλιόμετρα που έχουν διανύσει (ξεκινώντας από αυτό με τα λιγότερα και καταλήγοντας σε αυτό με τα περισσότερα).

Σε τέτοιες περιπτώσεις είναι πιο λογικό να θεωρήσει κανείς ότι όλες οι προτιμήσεις (κατατάξεις) είναι ισοδύναμες μεταξύ τους. Ακολουθώντας αυτό το σκεπτικό μπορούμε να απορρίψουμε αποτελέσματα τα οποία είναι μη βέλτιστα, για παράδειγμα αν υπάρχει κάποιο αυτοκίνητο A που είναι φτηνότερο και νεότερο και έχει διανύσει λιγότερα χιλιόμετρα από το B, τότε το B μπορεί να απορριφθεί. Με αυτό τον τρόπο μπορεί να προκύψει ένα μικρό υποσύνολο που περιέχει τα πιο 'ενδιαφέροντα' αποτελέσματα σύμφωνα με διάφορα κριτήρια. Το υποσύνολο αυτό ονομάζεται βέλτιστο σύνολο κατά Pareto, η αλλιώς κορυφογραμμή (skyline set).

Πιο τυπικά, μια εγγραφή A ανήκει στην κορυφογραμμή αν και μόνο αν δεν υπάρχει καμία άλλη εγγραφή B που να είναι καλύτερη από ή ισοδύναμη με την A για κάθε διαθέσιμη προτίμηση και αυστηρά καλύτερη για τουλάχιστον μια προτίμηση, ή όπως συνήθως λέγεται δεν υπάρχει άλλη εγγραφή B η οποία να κυριαρχεί (dominates) επί της A. Τα ερωτήματα που επιστρέφουν τις εγγραφές που ανήκουν στην κορυφογραμμή ονομάζονται ερωτήματα κορυφογραμμής. Αξίζει να σημειωθεί ότι τα αποτελέσματα ενός ερωτήματος κορυφογραμμής είναι μη συγκρίσιμα μεταξύ τους. Παρακάτω δίνουμε ένα παράδειγμα για να γίνει πιο κατανοητή η έννοια ενός ερωτήματος κορυφογραμμής.

Έστω μια βάση δεδομένων που περιέχει πληροφορίες για ξενοδοχεία στο Σχήμα 1.4(α′). Για κάποιον ταξιδιώτη που ενδιαφέρεται να επισκεφτεί έναν παραλιακό τουριστικό προορισμό, ένα ερώτημα κορυφογραμμής θα επιστρέψει όλα τα ξενοδοχεία για τα οποία δεν υπάρχει άλλο διαθέσιμο που να είναι φτηνότερο και την ίδια στιγμή πιο κοντά στην θάλασσα. Για τη συγκεκριμένη βάση δεδομένων το αποτέλεσμα θα περιέχει τα ξενοδοχεία h_4 και h_5.

Context	Προτιμήσεις	Κορυφογραμμή
C_1: Business, June		h_3, h_4, h_5
C_2: Vacation		h_2, h_4, h_5
C_3: Summer		h_1, h_2, h_4, h_5
C_q: Business, Summer	—	;

Πίνακας 1.2: *Contexts, προτιμήσεις και ερωτήματα κορυφογραμμής εξαρτώμενα από το τρέχον context*

Τώρα ας υποθέσουμε ότι υπάρχουν κάποια γνωρίσματα της βάσης για τα οποία η προτίμηση ενός χρήστη δεν είναι σταθερή και δεδομένη αλλά εξαρτάται από το περιβάλλον χρήσης. Με σκοπό να δώσουμε έμφαση στο γεγονός ότι οι προτιμήσεις (και συνεπώς οι σχέσεις κυριαρχίας μεταξύ των εγγραφών) εξαρτώνται από το περιβάλλον χρήσης (context), θα υιοθετήσουμε τον όρο *contextual skyline queries - CSQ*. Επιστρέφοντας στο προηγούμενο παράδειγμα, κάποιος που ταξιδεύει για επαγγελματικό σκοπό είναι πιο πιθανό να προτιμήσει ξενοδοχεία που είναι κοντά στο αεροδρόμιο. Αντίθετα, κάποιος ο οποίος ταξιδεύει για διακοπές μάλλον θα προτιμήσει ξενοδοχεία που είναι πιο κοντά στη θάλασσα ή διαθέτουν πισίνα.

Όλες οι προηγούμενες εργασίες στο αντικείμενο των ερωτημάτων κορυφογραμμής θεωρούν ως δεδομένο ότι το σύστημα έχει διαθέσιμο ένα σύνολο προτιμήσεων (προφίλ) για κάθε χρήστη. Στην παρούσα εργασία, χαλαρώνουμε αυτή την προϋπόθεση επιτρέποντας στους χρήστες να θέτουν ερωτήματα κορυφογραμμής χωρίς να καθορίσουν ρητά τις προτιμήσεις τους.

Ως παράδειγμα, θα χρησιμοποιήσουμε τη βάση δεδομένων ξενοδοχείων του σχήματος 1.4(α′). Ο πίνακας περιέχει για κάθε ξενοδοχείο πληροφορίες για την τιμή του, την απόσταση από τη θάλασσα και τις προσφερόμενες υπηρεσίες (Amenities). Να σημειωθεί ότι το τελευταίο γνώρισμα λαμβάνει τιμές από ένα σύνολο (set-valued domain), καθώς κάθε ξενοδοχείο συνήθως προσφέρει ένα σύνολο υπηρεσιών. Στο παράδειγμα, χάριν ευκολίας θα θεωρήσουμε ότι κάθε ξενοδοχείο παρέχει μία μοναδική υπηρεσία, παρόλαυτά η μεθοδολογία που θα παρουσιάσουμε στη συνέχεια μπορεί να εφαρμοστεί και στη γενική περίπτωση. Για τα πρώτα δύο γνωρίσματα (τιμή και απόσταση από τη θάλασσα), οι μικρότερες τιμές είναι προτιμητέες, ενώ για το γνώρισμα υπηρεσία η προτιμότερη τιμή εξαρτάται από το περιβάλλον χρήσης. Το Σχήμα 1.4(β′) απεικονίζει τα ξενοδοχεία σε ένα 2-διάστατο επίπεδο για τις διαστάσεις τιμή και απόσταση. Η τιμή της υπηρεσίας που παρέχεται από κάθε ξενοδοχείο, φαίνεται δίπλα σε κάθε εγγραφή.

Ο Πίνακας 1.2 δείχνει κάποια παραδείγματα ερωτημάτων κορυφογραμμής που εξαρτώνται από το περιβάλλον χρήσης. Τα περιβάλλοντα χρήσης C_1–C_3, φαίνονται στην πρώτη στήλη. Η δεύτερη στήλη περιέχει έναν κατευθυνόμενο γράφο που καθορίζει τις σχέσεις προτίμησης που ισχύουν για το κάθε context, ενώ η τρίτη στήλη δείχνει τα

αποτελέσματα του αντίστοιχου ερωτήματος κορυφογραμμής.

Αρχικά, ας θεωρήσουμε έναν χρήστη ο οποίος ταξιδεύει για επαγγελματικούς σκοπούς το μήνα Ιούνιο (context C_1) και προτιμάει ένα ξενοδοχείο που διαθέτει δωρεάν ίντερνετ (I) σε σχέση με το να διαθέτει γυμναστήριο, και τα δύο παραπάνω σε σχέση με οποιαδήποτε άλλη υπηρεσία. Με βάση αυτές τις προτιμήσεις, το αποτέλεσμα ενός ερωτήματος κορυφογραμμής θα περιέχει τα ξενοδοχεία h_3, h_4, h_5 όπως δείχνει και η τρίτη στήλη του Πίνακα 1.2. Παρόλο που το ξενοδοχείο h_3 είναι πιο ακριβό και πιο μακριά από τη θάλασσα σε σχέση με τα h_1, h_2, h_4, παρέχει μια πιο επιθυμητή υπηρεσία I, και επομένως δεν μπορεί να απορριφθεί από την κορυφογραμμή. Παρομοίως, ο χρήστης έχει καθορίσει ένα σύνολο προτιμήσεων για τα contexts C_2 (ταξίδι διακοπών) και C_3 (καλοκαιρινό ταξίδι), με τα αντίστοιχα αποτελέσματα που δείχνει ο Πίνακας 1.2.

Ας εξετάσουμε τώρα το context C_q (τέταρτη σειρά του Πίνακα 1.2), όπου ο χρήστης σχεδιάζει ένα επαγγελματικό ταξίδι το καλοκαίρι, αλλά για την περίπτωση αυτή οι προτιμήσεις του δεν είναι γνωστές στο σύστημα. Πώς θα πρέπει ένα τέτοιο ερώτημα να απαντηθεί· (α) Πρέπει το ίντερνετ I να προτιμηθεί σε σχέση με την ύπαρξη πισίνας P όπως στο context C_1·, (β) πρέπει η πισίνα P να προτιμηθεί σε σχέση με το ίντερνετ I όπως στο C_3, ή (γ) το ίντερνετ I και η πισίνα P είναι το ίδιο προτιμητέα όπως στο C_2· Στην πραγματικότητα, όλες οι παραπάνω εναλλακτικές περιπτώσεις ισχύουν με μια πιθανότητα η οποία εξαρτάται από την ομοιότητα του context C_q με τα contexts C_1, C_2, C_3. Επιπλέον, η αβεβαιότητα αυτή μεταφέρεται και στις σχέσεις κυριαρχίας: κάθε ξενοδοχείο κυριαρχεί έναντι κάποιου άλλου με κάποια πιθανότητα που εξαρτάται από το context.

Μέρος της παρούσας διατριβής είναι η αντιμετώπιση του προβλήματος που περιγράψαμε παραπάνω. Με αυτό το στόχο προτείνουμε μία μέθοδο η οποία επιχειρεί να αντισταθμίσει την έλλειψη γνώσης των προτιμήσεων ενός χρήστη. Συγκεκριμένα, επιλύουμε δυο υποπροβλήματα: (α) προσδιορίζουμε ένα σύνολο αβέβαιων προτιμήσεων που είναι σχετικές με το τρέχον περιβάλλον χρήσης βασισμένοι στις προτιμήσεις που είναι γνωστές για προηγούμενα περιβάλλοντα χρήσης, και (β) αντιμετωπίζουμε την αβεβαιότητα στις ισχύουσες σχέσεις κυριαρχίας προτείνοντας μια πιθανοτική εκδοχή των ερωτημάτων κορυφογραμμής *probabilistic contextual skyline query* p-CSQ όπου επιστρέφονται όλες οι εγγραφές που δεν κυριαρχούνται με μεγάλη πιθανότητα.

Για την επίλυση του πρώτου υποπροβλήματος δανειζόμαστε ιδέες από τις μεθόδους που έχουν προταθεί στη βιβλιογραφία για ερωτήματα προτιμήσεων που εξαρτώνται από το περιβάλλον χρήσης, και πιο συγκεκριμένα από τη μέθοδο εναρμόνισης προτιμήσεων που προτείνεται στην εργασία [3] και τη μέθοδο προσδιορισμού του περιβάλλοντος χρήσης που προτείνεται στην εργασία [66].

Πιο αναλυτικά, έστω ότι το σύστημα έχει συλλέξει κάποιες προτιμήσεις για κάποιον χρήστη για ένα σύνολο χαρακτηριστικών περιβαλλόντων χρήσης C_i, πχ. 'επαγγελματικό ταξίδι', 'καλοκαιρινές διακοπές ' κλπ. Με βάση αυτή την πληροφορία, υπολογίζουμε την ομοιότητα του τρέχοντος περιβάλλοντος χρήσης C_q με κάθε C_i, και χρησιμοποιούμε αυτή την ομοιότητα με σκοπό να αναθέσουμε μια πιθανότητα να ισχύει κάθε προτίμηση για το συγκεκριμένο context. Η αβεβαιότητα σχετικά με το ποιες προτιμήσεις ισχύουν επηρεάζει τις σχέσεις κυριαρχίας μεταξύ των εγγραφών της βάσης.

Για το δεύτερο υποπρόβλημα, η πιθανότητα μια εγγραφή να ανήκει στην κορυφο-

γραμμή είναι ίση με την πιθανότητα να μην κυριαρχείται από καμιά άλλη εγγραφή. Επομένως ένα πιθανοτικό ερώτημα κορυφογραμμής p-CSQ επιστρέφει όλες τις εγγραφές για τις οποίες η παραπάνω πιθανότητα είναι υψηλότερη από κάποια ελάχιστη τιμή. Να σημειωθεί ότι αν οι προτιμήσεις είναι βέβαιες, δηλαδή αν το τρέχον περιβάλλον χρήσης ταιριάζει απόλυτα με κάποιο από τα περιβάλλοντα χρήσης για τα οποία το σύστημα γνωρίζει με ακρίβεια τις προτιμήσεις ενός χρήστη, τότε το ερώτημα p-CSQ μετατρέπεται σε ένα συμβατικό ερώτημα κορυφογραμμής.

Στο παρελθόν έχουν προταθεί κάποιες τεχνικές για την επίλυση παραλλαγών του πρώτου υποπροβλήματος. Παρόλαυτά, δεν υπάρχει προηγούμενη βιβλιογραφία που να αναφέρεται στο δεύτερο υποπρόβλημα. Αξίζει να σημειωθεί ότι πιθανοτικές εκδοχές των ερωτημάτων κορυφογραμμής έχουν προταθεί στο παρελθόν με σκοπό τον χειρισμό περιπτώσεων όπου η αβεβαιότητα έγκειται στις τιμές των γνωρισμάτων μιας εγγραφής [56, 48]. Στο συγκεκριμένο πρόβλημα όμως στο οποίο επικεντρωνόμαστε, η αβεβαιότητα έγκειται στην ισχύ των προτιμήσεων ενός χρήστη. Επιπλέον, οι αλγόριθμοι που έχουν προταθεί για την αποτίμηση συμβατικών ερωτημάτων κορυφογραμμής δεν μπορούν να εφαρμοστούν στην περίπτωση μας, καθώς οι περισσότεροι από αυτούς (με την εξαίρεση της μεθόδου εμφωλευμένων βρόχων), επεξεργάζονται τις εγγραφές ακολουθώντας μια μονότονη διάταξη, ξεκινώντας από τις εγγραφές που έχουν τις πιο επιθυμητές τιμές γνωρισμάτων και καταλήγοντας στις λιγότερο επιθυμητές. Τέτοια διάταξη μπορεί να προκύψει στην περίπτωση όπου η αβεβαιότητα έγκειται στις τιμές των εγγραφών [56, 48] υπό την προϋπόθεση ότι το εύρος τιμών είναι φραγμένο. Η ύπαρξη αυτής της ιδιότητας περιορίζει τον μέσο αριθμό ελέγχων κυριαρχίας που απαιτούνται, ενώ παράλληλα επιτρέπει την προοδευτική παραγωγή αποτελεσμάτων από το σύστημα. Παρολαυτά, όταν η αβεβαιότητα σχετίζεται με την ισχύ των προτιμήσεων των χρηστών δεν υπάρχει τέτοια διάταξη, καθώς σε αυτή την περίπτωση δεν ισχύει η μεταβατική ιδιότητα (βλέπε Κεφάλαιο 4.2.1). Συνεπώς απαιτούνται νέες μέθοδοι για την επεξεργασία τέτοιων ερωτημάτων.

1.3.2 Συνεισφορά της διατριβής

Οι κυριότερες συνεισφορές αυτής της εργασίας παρουσιάζονται επιγραμματικά παρακάτω:

- Προτείνουμε μια μεθοδολογία η οποία επιχειρεί να αντιμετωπίσει το πρόβλημα της απουσίας γνώσης των προτιμήσεων ενός χρήστη για το τρέχον περιβάλλον χρήσης. Για αυτό το σκοπό, εισάγουμε την έννοια των αβέβαιων προτιμήσεων και ορίζουμε τα πιθανοτικά ερωτήματα κορυφογραμμής που εξαρτώνται από το περιβάλλον χρήσης (Probabilistic Contextual Skylines - p-CSQ).

- Δοθέντος ενός συνόλου προτιμήσεων που ισχύουν για ένα σύνολο contexts και του τρέχοντος context, προτείνουμε μια μέθοδο για την εξαγωγή πιθανοτήτων με τις οποίες ισχύει κάθε προτίμηση στο τρέχον context.

- Προτείνουμε αλγορίθμους για την αποτίμηση p-CSQ οι οποίοι βασίζονται στην ύπαρξη ή όχι ευρετηρίων στα δεδομένα και είναι σημαντικά πιο αποδοτικοί από μια προσαρμοσμένη εκδοχή της μεθόδου εμφωλευμένων βρόχων. Η εκτενής

πειραματική αξιολόγηση των μεθόδων αποδεικνύει την εγκυρότητα και την αποδοτικότητα των προτεινόμενων αλγορίθμων.

1.4 Αλγόριθμοι Εξατομίκευσης από την πλευρά των παρόχων

1.4.1 Προβλήματα και περιορισμοί

Η ανάπτυξη μεθόδων για την αποτίμηση ερωτημάτων με προτιμήσεις παρουσιάζει ιδιαίτερο ενδιαφέρον και από την πλευρά των παρόχων υπηρεσιών. Και αυτό γιατί στο σύγχρονο περιβάλλον, οι πάροχοι-επιχειρήσεις καλούνται να προσελκύσουν καταναλωτές με διαφορετικά χαρακτηριστικά και καταναλωτικές συνήθειες, ακολουθώντας όσο το δυνατόν πιο εξατομικευμένες στρατηγικές. Σε αυτό το πλαίσιο, η αποτύπωση των προτιμήσεων των πελατών μιας επιχείρησης παίζει σημαντικό ρόλο στη λήψη αποφάσεων για τη σχεδίαση και προώθηση νέων προϊόντων στην αγορά. Παραδείγματα σημαντικών εφαρμογών που σχετίζονται με την ανάλυση των προτιμήσεων των καταναλωτών είναι η στοχευμένη διαφήμιση (personalized advertising), η κατάτμηση της αγοράς (market segmentation), η τοποθέτηση ενός προϊόντος στην αγορά (product positioning) κ.α.

Για παράδειγμα έστω ότι το τμήμα Μάρκετινγκ μιας εταιρίας πληροφορικής, έχει διεξαγάγει μια έρευνα αγοράς για να συγκεντρώσει τις προτιμήσεις των καταναλωτών όσον αφορά τα επιθυμητά χαρακτηριστικά ενός φορητού υπολογιστή. Χρησιμοποιώντας τα δεδομένα που έχει συγκεντρώσει, η εταιρία θέλει να εκτιμήσει ποιοι καταναλωτές είναι πιο πιθανό να αγοράσουν κάποιο από τα μοντέλα υπολογιστών που προσφέρει. Με αυτό τον τρόπο, η εταιρία θα μπορούσε να προσαρμόσει την διαφημιστική της στρατηγική προς αυτούς τους χρήστες (για παράδειγμα στέλνοντάς τους εξατομικευμένα e-mails ή ειδικές προσφορές). Μια άλλη ενδιαφέρουσα εφαρμογή είναι ο προσδιορισμός των χαρακτηριστικών που θα πρέπει να έχει ένα νέο προϊόν ώστε να καταστεί όσο το δυνατόν πιο δημοφιλές μεταξύ των καταναλωτών. Σε αυτή την περίπτωση, η εταιρία θα επέλεγε να προσαρμόσει ανάλογα τις προδιαγραφές που θέτει το τμήμα σχεδιασμού.

Η επιστήμη που κατά κύριο λόγο ασχολείται με την υποστήριξη της λήψης τέτοιου είδους αποφάσεων είναι η Επιχειρησιακή Έρευνα (Operational Research). Το αντικείμενο της Επιχειρησιακής Έρευνας συνηθίζεται να περιγράφεται ως εξής: Ζητείται η βελτιστοποίηση μιας σειράς παραμέτρων με στόχο τη μεγιστοποίηση μιας συνάρτησης ωφέλειας (utility function), όπως για παράδειγμα το εκτιμώμενο κέρδος, η απόδοση μιας επένδυσης, το ρίσκο κ.α. Διάφορες υποπεριοχές της Επιχειρησιακής Έρευνας εστιάζουν στην εφαρμογή αναλυτικών μεθόδων (στατιστικής, εξόρυξης δεδομένων κλπ.) για την υποστήριξη της λήψης αποφάσεων. Δεδομένου ότι πολλά από τα ερευνητικά προβλήματα της Επιχειρησιακής Έρευνας έχουν να κάνουν με τη ανάλυση μεγάλων όγκων δεδομένων, το τελευταίο διάστημα έχει αναπτυχθεί έντονο ενδιαφέρον για την εφαρμογή μεθόδων διαχείρισης δεδομένων στην επίλυση αυτών των προβλημάτων.

Μέρος αυτής της διατριβής είναι η ανάπτυξη νέων αλγορίθμων για δύο προβλήμα-

τα που σχετίζονται με την ανάλυση μεγάλων όγκων καταναλωτικών προτιμήσεων, με πρακτικές εφαρμογές στην έρευνα αγοράς. Στη συνέχεια παρουσιάζουμε εν συντομία τα συγκεκριμένα προβλήματα στα οποία επικεντρωθήκαμε.

Το πρώτο πρόβλημα που εξετάζουμε είναι η εύρεση των πιθανών αγοραστών ενός προϊόντος (potential customers identification). Σχηματοποιούμε το πρόβλημα αυτό ως ένα αντίστροφο ερώτημα κορυφογραμμής και προτείνουμε έναν νέο αλγόριθμο που ονομάζουμε RSA ο οποίος υπερτερεί σε απόδοση συγκρινόμενος με τον πιο αποδοτικό αλγόριθμο που έχει προταθεί στην έως τώρα βιβλιογραφία BRS [81], σε σχέση τόσο με το υπολογιστικό κόστος, όσο και με το κόστος προσπελάσεων στον δίσκο. Σε αντίθεση με τον αλγόριθμο BRS, ο αλγόριθμος RSA που προτείνουμε βασίζεται σε διαφορετική σειρά επεξεργασίας των δεδομένων, γεγονός που επιτρέπει σημαντικές βελτιώσεις σε σχέση με την ταχύτητα εκτέλεσης (performance), τις δυνατότητες κλιμάκωσης (scalability), και την προοδευτική παραγωγή αποτελεσμάτων (progressiveness).

Στην πράξη οι εφαρμογές συχνά απαιτούν ταυτόχρονη επεξεργασία πολλών ερωτημάτων. Για παράδειγμα, έστω μια εταιρία κινητής τηλεφωνίας η οποία διατηρεί μια βάση δεδομένων των συνδρομητών της, όπου διατηρεί πληροφορίες για το τρέχον συνδρομητικό πρόγραμμα του κάθε πελάτη, και στατιστικά χρήσης όπως τη μέση μηνιαία διάρκεια κλήσεων, τον αριθμό των απεσταλμένων μηνυμάτων κειμένου, τον όγκο δεδομένων που καταναλώθηκε κλπ. Επιπλέον, έστω ότι το τμήμα πωλήσεων έχει προτείνει την έναρξη μιας καινούριας σειράς συνδρομητικών προγραμμάτων. Η εταιρία θα ήθελε να προβλέψει τα προγράμματα αυτά που, συλλογικά, είναι πιο πιθανό να προσελκύσουν το μέγιστο αριθμό συνδρομητών.

Για να επιλύσουμε αυτό το πρόβλημα, το σχηματοποιούμε ως ένα νέο τύπο ερωτήματος, στο οποίο θα αναφερόμαστε εφεξής ως *Ερώτημα Εύρεσης των k πιο Ελκυστικών Υποψηφίων* (k-Most Attractive Candidates (k-MAC) query). Δοθέντος ενός συνόλου υπαρχόντων προϊόντων στην αγορά P, ενός συνόλου προτιμήσεων καταναλωτών C και ενός συνόλου υποψηφίων νέων προϊόντων Q, ένα ερώτημα k-MAC επιστρέφει ένα σύνολο k υποψηφίων νέων προϊόντων από το αρχικό σύνολο Q, έτσι ώστε τα επιλεγμένα νέα προϊόντα να έχουν συλλογικά τον μέγιστο συνολικό αριθμό εκτιμώμενων αγοραστών. Σύμφωνα με τη σχηματοποίηση του προβλήματος που προτείνουμε, ένας καταναλωτής θεωρείται ως πιθανός αγοραστής ενός προϊόντος, αν και μόνο αν το προϊόν ανήκει στο αποτέλεσμα ενός ερωτήματος κορυφογραμμής (reverse skyline set or influence set), θεωρώντας για κάθε καταναλωτή ως βέλτιστες τις τιμές των γνωρισμάτων που σχετίζονται με τις προτιμήσεις του.

Δύο πρόσφατες εργασίες [49, 57] εξετάζουν ένα παρόμοιο πρόβλημα, περιοριζόμενοι όμως μόνο σε γνωρίσματα των δεδομένων που έχουν μια καθολικά αποδεκτή ολική διάταξη, όπως η τιμή ή το βάρος ενός φορητού υπολογιστή (και στις δύο περιπτώσεις οι χαμηλότερες τιμές είναι πάντοτε προτιμότερες). Αν ισχύει αυτή η προϋπόθεση, οι καλύτερες εγγραφές για έναν χρήστη είναι εύκολο να προκύψουν εκτελώντας ένα συμβατικό ερώτημα κορυφογραμμής. Συνεπώς, οι παραπάνω εργασίες επικεντρώνονται στο να προτείνουν άπληστους αλγορίθμους που αναζητούν την πιο προσοδοφόρα λύση, συνδυάζοντας τα σύνολα πιθανών αγοραστών κάθε νέου προϊόντος.

Σε αυτή την εργασία, γενικεύουμε την έννοια των προτιμήσεων ενός καταναλωτή ώστε να καλύπτει επίσης γνωρίσματα για τα οποία δεν υπάρχει μια αντικειμενικά

βέλτιστη τιμή (δηλαδή δεν προκύπτει μια ολική διάταξη τιμών) όπως είναι το μέγεθος της οθόνης, ο τύπος επεξεργαστή, το λειτουργικό σύστημα κλπ. Για παράδειγμα, ένας καταναλωτής μπορεί να ενδιαφέρεται να αγοράσει ένα όσο το δυνατό πιο εύκολα μεταφέρσιμο φορητό υπολογιστή, ενώ κάποιος άλλος έναν φορητό υπολογιστή με σκοπό να αντικαταστήσει τον σταθερό προσωπικό του υπολογιστή. Σε μια τέτοια περίπτωση, αναμένεται ο πρώτος καταναλωτής να προτιμήσει έναν φορητό υπολογιστή με μικρότερη οθόνη (κάνοντας κάποιες θυσίες στη χρηστικότητα και τις εφαρμογές που μπορεί να εκτελέσει), ενώ ο δεύτερος έναν με μεγαλύτερη οθόνη (θυσιάζοντας ίσως την ευκολία μεταφοράς του).

Για τέτοιου είδους γνωρίσματα, η εφαρμογή των τεχνικών που προτείνεται στις εργασίες [49, 57] απαιτεί να έχουν εξαχθεί προηγουμένως οι σχέσεις κυριαρχίας που ισχύουν μεταξύ των εγγραφών για κάθε καταναλωτή, εφόσον αυτές εξαρτώνται από τις προτιμήσεις. Συνεπώς, αυτή η μέθοδος θα απαιτούσε την εκτέλεση ενός δυναμικού ερωτήματος κορυφογραμμής [55] για κάθε καταναλωτή, κάτι που είναι απαγορευτικά ακριβό σε σχέση με τον χρόνο εκτέλεσης που θα χρειαζόταν. Ταυτόχρονα, αν επιλέγαμε να εκτελέσουμε κάποιον από τους αλγορίθμους που έχουν προταθεί για (απλά) αντίστροφα ερωτήματα κορυφογραμμής για να αποτιμήσουμε ένα k-MAC ερώτημα, θα χρειαζόταν να υπολογίσουμε το αποτέλεσμα ενός αντίστροφου ερωτήματος κορυφογραμμής μια φορά για κάθε υποψήφιο νέο προϊόν, κάτι που είναι επίσης πολύ δαπανηρό από πλευράς χρόνου εκτέλεσης, ιδιαιτέρως για μεγαλύτερα σύνολα δεδομένων. Με αυτό το σκεπτικό, προσαρμόζουμε τον αλγόριθμο RSA που προτείναμε για απλά αντίστροφα ερωτήματα κορυφογραμμής, ώστε να μπορεί να χειριστεί ταυτόχρονα ομάδες ερωτημάτων. Ο αλγόριθμος που προτείνουμε γι'αυτό το πρόβλημα μειώνει αισθητά τον απαιτούμενο χρόνο εκτέλεσης σε σχέση με το να επεξεργαζόμασταν κάθε επιμέρους ερώτημα ξεχωριστά, ομαδοποιώντας κατάλληλα παρόμοια ερωτήματα, εκτελώντας κοινές προσπελάσεις στον δίσκο, και επιτρέποντας την ταυτόχρονη επεξεργασία πολλών ερωτημάτων. Τέλος, αφού εξαχθεί το σύνολο αντίστροφης κορυφογραμμής για κάθε ερώτημα, προτείνουμε έναν άπληστο αλγόριθμο που υπολογίζει την τελική λύση για ένα ερώτημα k-MAC, συνδυάζοντας τα επιμέρους σύνολα αντίστροφης κορυφογραμμής.

1.4.2 Συνεισφορά της διατριβής

Εν συντομία, οι συνεισφορές αυτού του κομματιού της διατριβής είναι οι εξής:

- Αναπτύσσουμε νέους αλγορίθμους για δύο προβλήματα που σχετίζονται με την ανάλυση μεγάλων όγκων καταναλωτικών προτιμήσεων, με πρακτικές εφαρμογές στην έρευνα αγοράς. Σχηματοποιούμε τα δύο προβλήματα ως παραλλαγές ενός και πολλαπλών αντίστροφων ερωτημάτων κορυφογραμμής αντιστοίχως.

- Προτείνουμε έναν νέο αλγόριθμο, ονόματι RSA για την αποτίμηση αντίστροφων ερωτημάτων κορυφογραμμής. Ο αλγόριθμος RSA παρουσιάζει καλύτερη κλιμάκωση σε σύνολα δεδομένων που περιέχουν μεγάλο αριθμό αποτελεσμάτων που ανήκουν στην κορυφογραμμή (όπως π.χ. πολυδιάστατα δεδομένα), ενώ ταυτόχρονα παράγει τα πρώτα αποτελέσματα σημαντικά πιο γρήγορα από τον καλύτερο αλγόριθμο που έχει προταθεί στην έως σήμερα βιβλιογραφία.

- Αναπτύσσουμε μια παραλλαγή του αλγορίθμου RSA για ομάδες ερωτημάτων ο οποίος μειώνει αισθητά τον απαιτούμενο χρόνο εκτέλεσης σε σχέση με το να επεξεργαζόμασταν κάθε επιμέρους ερώτημα ξεχωριστά, ομαδοποιώντας κατάλληλα παρόμοια υποψήφια προϊόντα, εκτελώντας κοινές προσπελάσεις στον δίσκο, και επιτρέποντας την ταυτόχρονη επεξεργασία πολλών ερωτημάτων. Στη συνέχεια εφαρμόζουμε τον νέο αυτό αλγόριθμο για την αποτίμηση ερωτημάτων k-MAC. Το ερώτημα k-MAC γενικεύει παρόμοια ερωτήματα που έχουν προταθεί σε προηγούμενες εργασίες [49, 57] για περιπτώσεις όπου οι προτιμήσεις των καταναλωτών συμπεριλαμβάνουν γνωρίσματα χωρίς αντικειμενικά βέλτιστη τιμή.

- Διεξάγουμε εκτενή πειραματική αξιολόγηση των προτεινόμενων αλγορίθμων τόσο σε πραγματικά δεδομένα όσο και σε δεδομένα που έχουν παραχθεί συνθετικά. Η πειραματική μελέτη μας καταδεικνύει ότι (α) ο αλγόριθμος RSA υπερτερεί αισθητά του αλγορίθμου BRS για την περίπτωση ενός αντίστροφου ερωτήματος κορυφογραμμής σε σχέση με την ταχύτητα εκτέλεσης (performance), τις δυνατότητες κλιμάκωσης (scalability), και την προοδευτική παραγωγή αποτελεσμάτων (progressiveness), ιδιαιτέρως για πολυδιάστατα δεδομένα ή όταν το μέγεθος του συνόλου προϊόντων είναι μεγαλύτερο από το μέγεθος του συνόλου των προτιμήσεων των καταναλωτών, και (β) ο αλγόριθμος που προτείνουμε για την ταυτόχρονη εκτέλεση πολλαπλών ερωτημάτων υπερτερεί έναντι μεθόδων που επεξεργάζονται κάθε ερώτημα ξεχωριστά.

1.5 Δομή της έκθεσης

Η συνέχεια της έκθεσης διαρθρώνεται ως εξής: στο Κεφάλαιο 2 γίνεται μία επισκόπηση των κυριότερων εργασιών που έχουν γίνει στην περιοχή της εξατομίκευσης σε συστήματα διαχείρισης δεδομένων. Στο Κεφάλαιο 3 περιγράφουμε αναλυτικά το σύστημα *PrefDB* όπως αυτό προτείνεται στις εργασίες [9, 6, 8]. Το Κεφάλαιο 4 διαπραγματεύεται την αποτίμηση ερωτημάτων κορυφογραμμής για προτιμήσεις που εξαρτώνται από το περιβάλλον χρήσης, όπως παρουσιάστηκε στην εργασία [60]. Στο Κεφάλαιο 5 επικεντρωνόμαστε στην αποτίμηση ερωτημάτων έρευνας αγοράς που επεξεργάζονται μεγάλους όγκους προτιμήσεων καταναλωτών [7]. Τέλος, στο Κεφάλαιο 6 παρέχουμε μια σύνοψη των εργασιών που έχουν γίνει στα πλαίσια της διατριβής και καταγράφουμε ενδεικτικά κάποια θέματα και ιδέες προς μελλοντική διερεύνηση.

Κεφάλαιο 2

Σχετική Βιβλιογραφία

Στην ενότητα αυτή παρουσιάζουμε τις σημαντικότερες εργασίες που έχουν γίνει στην περιοχή της εξατομίκευσης σε συστήματα διαχείρισης δεδομένων. Στην ενότητα 2.1 παρουσιάζουμε μοντέλα και συστήματα που έχουν προταθεί για την αναπαράσταση, διαχείριση και αποτίμηση ερωτημάτων με προτιμήσεις σε σχεσιακά δεδομένα. Στην ενότητα 2.2 εστιάζουμε σε τεχνικές και αλγορίθμους για την αποτίμηση ερωτημάτων με προτιμήσεις σε επίπεδο εφαρμογής, ενώ στην ενότητα 2.3 παρουσιάζουμε τις εργασίες που έχουν γίνει στην περιοχή των ερωτημάτων έρευνας αγοράς.

2.1 Συστήματα Εξατομίκευσης

Η έννοια της επεξεργασίας ερωτημάτων με προτιμήσεις εμφανίζεται σε πολλές εφαρμογές όπου οι χρήστες μπορούν να επιλέξουν ανάμεσα σε διάφορες εναλλακτικές, συμπεριλαμβανομένων των εξατομικευμένων βάσεων δεδομένων [20], [36], [40], των συστημάτων συστάσεων [2], της λήψης αποφάσεων με πολλαπλά κριτήρια [14, 28] και της αναζήτησης με λέξεις κλειδιά [65]. Παρακάτω επιχειρούμε να παρουσιάσουμε την σχετική βιβλιογραφία σε δύο άξονες: αρχικά σε σχέση με την αναπαράσταση προτιμήσεων σε σχεσιακά δεδομένα και στη συνέχεια σε σχέση με τον τρόπο που αυτές ενσωματώνονται και χρησιμοποιούνται σε ερωτήματα.

Στη βιβλιογραφία έχουν προταθεί δύο γενικές μέθοδοι αναπαράστασης προτιμήσεων σε σχεσιακά δεδομένα. Οι εργασίες που ακολουθούν την *ποσοτική* (quantitative) προσέγγιση [4, 33, 40, 46] αναθέτουν αριθμητικά σκορ σε γνωρίσματα μιας εγγραφής ή σε επιμέρους συνθήκες μιας σχέσης κάνοντας χρήση συναρτήσεων βαθμολόγησης (scoring functions). Στη συνέχεια μια συναθροιστική συνάρτηση (aggregate function) συναθροίζει τα επιμέρους σκορ που ισχύουν για κάθε εγγραφή, σταθμίζοντας το κάθε σκορ με το κατάλληλο βάρος. Με αυτό τον τρόπο για κάθε εγγραφή υπολογίζεται ένα συνολικό σκορ και προκύπτει μια ολική διάταξη των εγγραφών με κριτήριο το σκορ που τους αντιστοιχεί. Αντιθέτως, στις εργασίες που ακολουθούν την *ποιοτική* (qualitative) προσέγγιση, όπως οι [36, 20], οι προτιμήσεις υποδηλώνονται χρησιμοποιώντας δυαδικές συγκρίσεις ανάμεσα στις τιμές των γνωρισμάτων μιας σχέσης. Για παράδειγμα, έστω ότι για ένα γνώρισμα η τιμή a είναι προτιμητέα σε σχέση με τις τιμές b και c. Σε αυτή την περίπτωση μπορούν να υπάρχουν και τιμές που δεν είναι συγκρίσιμες μεταξύ τους,

π.χ. οι τιμές b και c, και επομένως προκύπτει μερική μόνο διάταξη των εγγραφών.

Σε νεότερες εργασίες [3, 66] παρουσιάζονται συστήματα εξατομίκευσης που εξαρτώνται από το περιβάλλον χρήσης (context). Σύμφωνα με αυτή την προσέγγιση, οι προτιμήσεις ενός χρήστη δεν ισχύουν καθολικά, αλλά υπό συνθήκη. Οι συνθήκες αυτές μπορεί να σχετίζονται με τα δεδομένα [3] (για παράδειγμα, μιλώντας για κωμωδίες προτιμώ αυτές που γυρίστηκαν παλιότερα) ή να είναι εξωτερικές συνθήκες ως προς τη βάση δεδομένων [66] (για παράδειγμα, προτιμώ να βλέπω ταινίες θρίλερ μόνο όταν είμαι μαζί με φίλους). Στη δεύτερη περίπτωση πιθανές παράμετροι που λαμβάνονται υπόψη μπορεί να είναι η τρέχουσα γεωγραφική θέση ενός χρήστη, η ώρα/ημέρα, οι καιρικές συνθήκες ή το κοινωνικό περιβάλλον. Στην εργασία [3], δίνεται ένα συνόλο προτιμήσεων που ισχύουν υπό συνθήκη (contextual preferences) για μια ομάδα χρηστών όπου οι προτιμήσεις αναπαρίστανται ακολουθώντας την 'ποιοτική' προσέγγιση. Η συγκεκριμένη εργασία προτείνει μεθόδους για την αναδιάταξη των αποτελεσμάτων ενός ερωτήματος ώστε να συμφωνούν κατά το δυνατό με τις προτιμήσεις που ισχύουν στο τρέχον περιβάλλον χρήσης. Η εργασία [66] αντιμετωπίζει το ίδιο πρόβλημα στην περίπτωση ενός χρήστη όπου οι προτιμήσεις έχουν οριστεί με ποσοτικό τρόπο και είναι εξωτερικές ως προς τα δεδομένα.

Σε σχέση με τον τρόπο που γίνεται η ενσωμάτωση των προτιμήσεων σε ένα ερώτημα, μια συνηθισμένη προσέγγιση είναι ο μετασχηματισμός των προτιμήσεων σε συμβατικά υποερωτήματα τα οποία στη συνέχεια εκτελούνται από τη βάση δεδομένων κατά τα γνωστά. Για παράδειγμα, τα παρακάτω συστήματα εξατομίκευσης [43, 4, 36, 20, 40] παρέχουν ταξινομημένα αποτελέσματα στα ερωτήματα των χρηστών συνδυάζοντας τις προτιμήσεις τους ακολουθώντας μια plug-in προσέγγιση. Η εργασία [43] επεκτείνει τα ερωτήματα ενός χρήστη με επιπλέον συνθήκες προτίμησης που δρουν ως 'χαλαροί' περιορισμοί (soft constraints). Σε αντίθεση με τους αυστηρούς περιορισμούς (hard constraints) ενός SQL ερωτήματος που πρέπει να ικανοποιηθούν απαραίτητα, αν δεν υπάρχουν εγγραφές που ικανοποιούν μια συνθήκη προτίμησης τότε η συνθήκη χαλαρώνει μέχρι να βρεθούν εγγραφές που να την ικανοποιούν. Στην εργασία [4] προτείνεται η χρήση γενικευμένων συναρτήσεων που αναμειγνύουν προτιμήσεις ορισμένες με 'ποσοτικό' τρόπο. Στην περίπτωση που οι προτιμήσεις έχουν εκφραστεί με 'ποιοτικό' τρόπο, στις εργασίες [36, 20] προτείνονται πλαίσια (frameworks) που επιτρέπουν τη σύνθεση και συνδυασμό των προτιμήσεων αυτών. Ανάμεσα στις προτεινόμενες μεθόδους είναι και η σύνθεση προτιμήσεων κατά Pareto που ισοδυναμεί με την εκτέλεση ενός ερωτήματος κορυφογραμμής (skyline query). Τέλος, στην εργασία [40] παρέχονται τεχνικές αποτίμησης ερωτημάτων εξατομίκευσης θεωρώντας προτιμήσεις που έχουν οριστεί σε τιμές γνωρισμάτων (attribute values) και σε συνθήκες συνένωσης (join conditions) μεταξύ δύο σχέσεων.

Στην εργασία [46] οι συγγραφείς παρουσιάζουν ένα πλαίσιο για την υποστήριξη ερωτημάτων κατάταξης σε σχεσιακά δεδομένα επεκτείνοντας τη σχεσιακή άλγεβρα με ένα νέο τελεστή κατάταξης (rank operator). Ο τελεστής κατάταξης μπορεί να συνδυαστεί με τους υπόλοιπους τελεστές της σχεσιακής άλγεβρας, επιτρέποντας μετασχηματισμούς ισοδυναμίας στο πλάνο εκτέλεσης ενός ερωτήματος που οδηγούν σε σημαντική βελτίωση της απόδοσης. Παρομοίως, στην εργασία [20] προτείνεται ο τελεστής winnow ο οποίος επιλέγει τις εγγραφές που αντιστοιχούν στο βέλτιστο σύνολο κατά Pareto.

Επιπλέον, οι εργασίες [19, 83, 85, 84, 88] εισάγουν τεχνικές δεικτοδότησης (indexing) για να επιταχύνουν την αποτίμηση ερωτημάτων με προτιμήσεις, ενώ στις εργασίες [33, 23] προτείνεται η χρήση υλοποιημένων όψεων (materialized views) για την κατάταξη των εγγραφών μιας σχέσης.

Μια διαφορετική προσέγγιση που έχει ως βασικό στόχο την πιο ευέλικτη επεξεργασία ερωτημάτων με προτιμήσεις προτείνεται στην εργασία [45]. Οι συγγραφείς εισάγουν ένα νέο πλαίσιο εξατομίκευσης με την ονομασία FlexPref το οποίο επιτρέπει τη χρήση διαφορετικών αλγορίθμων εξατομίκευσης (όπως για παράδειγμα top-k ερωτήματα ή ερωτήματα κορυφογραμμής), απαιτώντας λιγοστές αλλαγές στον πηγαίο κώδικα του πυρήνα της βάσης αφήνοντας στους προγραμματιστές την αρμοδιότητα να ορίσουν απλούς κανόνες που καθορίζουν με ποιον τρόπο το σύστημα θα επιλέξει τις προτιμότερες εγγραφές.

2.2 Αλγόριθμοι Αποτίμησης Ερωτημάτων Προτιμήσεων

Εκτενής έρευνα έχει γίνει στην περιοχή των αλγορίθμων αποτίμησης ερωτημάτων προτιμήσεων και σε επίπεδο εφαρμογής. Οι αλγόριθμοι που έχουν προταθεί χωρίζονται σε δύο βασικές κατηγορίες: στα ερωτήματα ανάκτησης των k κορυφαίων αποτελεσμάτων (top-k queries) και στα ερωτήματα κορυφογραμμής (skyline queries). Στις παρακάτω υποενότητες παρουσιάζουμε συνοπτικά τις βασικότερες εργασίες για κάθε τύπο ερωτήματος.

2.2.1 Ερωτήματα top-k

Ένα ερώτημα top-k επιστρέφει τα κορυφαία k αποτελέσματα με το υψηλότερο σκορ, ταξινομημένα με τη χρήση μιας συναθροιστικής συνάρτησης. Τα ερωτήματα top-k έχουν το πλεονέκτημα ότι επιστρέφουν ένα περιορισμένο πλήθος αποτελεσμάτων για ένα ερώτημα, αντιμετωπίζοντας έτσι το πρόβλημα της υπερπληθώρας αποτελεσμάτων (information overload), απαιτούν όμως την ακριβή γνώση μιας συναθροιστικής συνάρτησης η οποία στις περισσότερες περιπτώσεις δεν είναι εύκολο να προσδιοριστεί με ακρίβεια για κάθε χρήστη.

Τα ερωτήματα top-k εισάγονται για πρώτη φορά στην εργασία [27] με στόχο να μειώσουν το κόστος επικοινωνίας που απαιτείται σε κατανεμημένα συστήματα και εφαρμογές ενδιάμεσου λογισμικού (middleware). Σε τέτοιες εφαρμογές απαιτείται πολλές φορές η συνάθροιση ταξινομημένων αποτελεσμάτων που προέρχονται από διάφορες πηγές κάνοντας χρήση μιας συναθροιστικής συνάρτησης. Έστω ότι υπάρχουν m πηγές, κάθε μια από τις οποίες αναθέτει ένα σκορ για κάθε εγγραφή. Επιπλέον, σε κάθε μία πηγή υπάρχει μια ταξινομημένη λίστα των εγγραφών με βάση το σκορ, έστω L_i. Υπάρχουν δύο τρόποι προσπέλασης στα δεδομένα κάθε ταξινομημένης λίστας, η σειριακή προσπέλαση (sorted access) και η τυχαία προσπέλαση (random access). Στη σειριακή προσπέλαση για να ανακτηθεί το αντικείμενο με το i-όστο μεγαλύτερο σκορ απαιτούνται i προσπελάσεις. Αντιθέτως, η τυχαία προσπέλαση επιτρέπει την άμεση ανάκτηση

ενός αντικειμένου.

Για την αποδοτική ανάκτηση των k αντικειμένων με τα υψηλότερα σκορ, έχουν προταθεί μια σειρά από αλγόριθμοι. Αρχικά, ο Fagin [27, 28] πρότεινε τον αλγόριθμο FA (Fagin's Algorithm). Ο FA προσπελαύνει σειριακά κάθε ταξινομημένη λίστα παράλληλα (στην πρώτη επανάληψη ανακτά το καλύτερο αντικείμενο κάθε λίστας, στη δεύτερη το επόμενο καλύτερο κ.ο.κ). Οι επαναλήψεις συνεχίζονται μέχρι να βρεθούν k αντικείμενα, τέτοια ώστε και τα k να έχουν ανακτηθεί σε όλες τις m ταξινομημένες λίστες. Στη συνέχεια, για τα υπόλοιπα αντικείμενα που έχουν ήδη ανακτηθεί, εκτελεί τυχαίες προσπελάσεις για να ανακτήσει όσα επιμέρους σκορ του λείπουν. Τελικά, ο αλγόριθμος FA υπολογίζει για όλα τα αντικείμενα το συνολικό τους σκορ και επιστρέφει τα k με τα υψηλότερα σκορ.

Ο αλγόριθμος κατωφλίου (Threshold Algorithm (TA)) [28] αποτελεί βελτίωση του FA. Ο TA εκτελεί σειριακή προσπέλαση σε κάθε ταξινομημένη λίστα όπως και ο TA, αλλά για κάθε νέο αντικείμενο που ανακτά εκτελεί τυχαίες προσπελάσεις και στις υπόλοιπες λίστες και έτσι υπολογίζει άμεσα το συνολικό σκορ για κάθε εγγραφή. Ο TA διατηρεί την τελευταία τιμή που ανέκτησε σε κάθε λίστα, έστω $\overline{x_i}$ και υπολογίζει ένα κατώφλι $t = f(\overline{x_1}, ..., \overline{x_m})$, όπου f είναι η συναθροιστική συνάρτηση. Ο αλγόριθμος τερματίζει όταν ανακτηθούν k αντικείμενα με τιμή μεγαλύτερη ή ίση του t, τα οποία και επιστρέφονται ως αποτέλεσμα του ερωτήματος. Παρόμοιοι με τον αλγόριθμο TA είναι επίσης οι αλγόριθμοι Multistep [54] και Quick-Combine [30], ενώ ο αλγόριθμος BPA που προτείνεται στην εργασία [5] βελτιώνει τον TA ως προς το κόστος επεξεργασίας.

Στην εργασία [28] προτείνεται επίσης ο αλγόριθμος NRA (No Random Access Algorithm) για περιπτώσεις όπου η τυχαία προσπέλαση στις ταξινομημένες λίστες δεν είναι εφικτή. Ο NRA προσπελαύνει σειριακά όλες τις ταξινομημένες λίστες παράλληλα. Σε βάθος d, δηλαδή όταν τα πρώτα d αντικείμενα κάθε λίστας έχουν ανακτηθεί, έστω ότι $\overline{x_i}^{(d)}$ είναι η τελευταία τιμή που ανακτήθηκε για τη λίστα L_i. Για κάθε αντικείμενο o, ο NRA υπολογίζει ένα κατώτατο και ένα ανώτατο όριο, $L(o)^{(d)}, U(o)^{(d)}$ αντιστοίχως. Για τον υπολογισμό του κατώτατου ορίου για κάθε αντικείμενο o, τα σκορ για τα άγνωστα γνωρίσματα $\overline{x_1}$ τίθενται ίσα με μηδέν, ενώ για το ανώτατο όριο ως ίσα με το $\overline{x_i}^{(d)}$. Ο αλγόριθμος διατηρεί μόνο τα k αντικείμενα με τα μεγαλύτερα κατώτατα όρια. Ο NRA τερματίζει όταν όλα τα αντικείμενα εκτός των top-k έχουν ανώτατα όρια μικρότερα του κατώτατου ορίου του k-στου αντικειμένου. Παρόμοιοι με τον αλγόριθμο NRA είναι επίσης οι αλγόριθμοι Stream-Combine [70] και SR-Combine [10]. Τέλος, στο [15], οι συγγραφείς προτείνουν τον αλγόριθμο Upper and Pick για την αποτίμηση ερωτημάτων top-k για πηγές προσβάσιμες μέσω του web, θεωρώντας ότι η τυχαία προσπέλαση επιτρέπεται μόνο για ένα υποσύνολο των πηγών. Παρομοίως, ο αλγόριθμος MPro [18], εστιάζει στην ελαχιστοποίηση των τυχαίων προσπελάσεων για την ανάκτηση των επιμέρους σκορ.

Στις εργασίες [34, 53] προτείνονται οι αλγόριθμοι Rank Join και J* αντιστοίχως για την αποτίμηση ερωτημάτων top-k κατά τη σύζευξη (join) δύο σχέσεων.

2.2.2 Ερωτήματα Κορυφογραμμής

Τα ερωτήματα κορυφογραμμής χρησιμοποιούνται πολύ συχνά σε εφαρμογές που απαιτούν τη λήψη αποφάσεων με βάση πολλαπλά κριτήρια (multi-criteria decision making). Στις περιπτώσεις αυτές δεν είναι εύκολο ή επιθυμητό να οριστεί μία συναθροιστική συνάρτηση που θα παράξει τα καλύτερα αποτελέσματα. Αντιθέτως, μπορεί να θεωρήσει κανείς ότι κάθε χρήστης έχει μια σειρά από προτιμήσεις πάνω σε γνωρίσματα των δεδομένων οι οποίες ορίζουν επιμέρους κατατάξεις των αποτελεσμάτων. Σε τέτοιες περιπτώσεις είναι πιο λογικό να θεωρήσει κανείς ότι όλες οι προτιμήσεις (κατατάξεις) είναι ισοδύναμες μεταξύ τους. Ακολουθώντας αυτό το σκεπτικό μπορούμε να απορρίψουμε αποτελέσματα τα οποία είναι μη βέλτιστα. Με αυτό τον τρόπο μπορεί να προκύψει ένα μικρό υποσύνολο που περιέχει τα πιο 'ενδιαφέροντα' αποτελέσματα σύμφωνα με διάφορα κριτήρια. Το υποσύνολο αυτό ονομάζεται βέλτιστο σύνολο κατά Pareto, η αλλιώς κορυφογραμμή (skyline set).

Πιο τυπικά, μια εγγραφή A ανήκει στην κορυφογραμμή αν και μόνο αν δεν υπάρχει καμία άλλη εγγραφή B που να είναι καλύτερη από ή ισοδύναμη με την A για κάθε διαθέσιμη προτίμηση και αυστηρά καλύτερη για τουλάχιστον μια προτίμηση, ή όπως συνήθως λέγεται δεν υπάρχει άλλη εγγραφή B η οποία να κυριαρχεί (dominates) επί της A. Τα ερωτήματα που επιστρέφουν τις εγγραφές που ανήκουν στην κορυφογραμμή ονομάζονται ερωτήματα κορυφογραμμής. Αξίζει να σημειωθεί ότι τα αποτελέσματα ενός ερωτήματος κορυφογραμμής είναι μη συγκρίσιμα μεταξύ τους. Επιπλέον μπορεί να αποδειχθεί ότι η εγγραφή με την υψηλότερη κατάταξη (top-1) για κάθε συνάρτηση βαθμολόγησης που είναι γνησίως μονότονη στα γνωρίσματα της σχέσης περιέχεται οπωσδήποτε στην κορυφογραμμή. Με άλλα λόγια, η κορυφογραμμή περιέχει αρκετά διαφοροποιημένα αποτελέσματα που ικανοποιούν πολλές διαφορετικές προτιμήσεις.

Στη βιβλιογραφία, το πρόβλημα της εύρεσης του βέλτιστου συνόλου κατά Pareto παρουσιάζεται για πρώτη φορά σε εργασίες υπολογιστικής γεωμετρίας όπως η [42]. Η εργασία [14] εισάγει το πρόβλημα στο πεδίο των βάσεων δεδομένων και προτείνει μια σειρά αλγορίθμων που εκτελούνται σε εξωτερική μνήμη. Η περισσότερο γνωστή μέθοδος είναι η μέθοδος εμφωλευμένων βρόχων (Block Nested Loops (BNL)), η οποία ελέγχει για κάθε εγγραφή A αν υπάρχει κάποια άλλη εγγραφή B στο σύνολο δεδομένων (dataset) τέτοια ώστε η B να κυριαρχεί επί της A. Επιπλέον, οι συγγραφείς περιγράφουν μια προσέγγιση βασισμένη σε δεικτοδότηση των δεδομένων με τη βοήθεια ενός B-tree, όπως επίσης και μια επέκταση του αλγορίθμου διαίρει και βασίλευε (divide and conquer (DC)) που είχε προταθεί στο [58]. Η εργασία [67] εισάγει τεχνικές με τις οποίες είναι εφικτή η προοδευτική παραγωγή κάποιων αποτελεσμάτων καθώς εκτελείται ο αλγόριθμος χωρίς να απαιτείται η πλήρης σάρωση ολόκληρου του συνόλου δεδομένων. Η εργασία [21] παρατηρεί ότι αν τα σημεία εξεταστούν ταξινομημένα σύμφωνα με μια οποιαδήποτε συναθροιστική συνάρτηση που είναι μονότονη στα γνωρίσματα μιας σχέσης, τότε ο μέσος αριθμός ελέγχων κυριαρχίας (dominance checks) που απαιτούνται ελαχιστοποιείται. Βασισμένοι σε αυτή την παρατήρηση οι συγγραφείς προτείνουν τον αλγόριθμο SFS (Sort-first Skyline), οποίος είναι παρόμοιος με τον BNL με τη διαφορά ότι περιλαμβάνει ένα επιπλέον βήμα προεπεξεργασίας όπου τα δεδομένα ταξινομούνται σύμφωνα με μια μονότονη συναθροιστική συνάρτηση. Διάφορες βελτιώσεις του SFS

όπως οι [29, 11] αυξάνουν την αποδοτικότητά του.

Οι παραπάνω μέθοδοι δεν μπορούν να εφαρμοστούν σε κατηγορικά, ιεραρχικά, κλπ. γνωρίσματα εφόσον δεν μπορεί να οριστεί μια ολική διάταξη των τιμών τους από την περισσότερο προς την λιγότερο επιθυμητή. Για τον χειρισμό τέτοιων τύπων γνωρισμάτων, η εργασία [16] προτείνει ένα αλγόριθμο βασισμένο σε έναν πιο ισχυρό ορισμό της έννοιας της κυριαρχίας (dominance). Αυτή η τροποποίηση έχει ως μειονέκτημα ότι μπορεί να παράγει ορισμένα αποτελέσματα που δεν ανήκουν στην κορυφογραμμή σύμφωνα με τον κλασικό ορισμό (false positives) και επομένως απαιτείται και ένα επιπλέον βήμα φιλτραρίσματος (μέσω ελέγχων κυριαρχίας) μεταξύ των αρχικών αποτελεσμάτων. Η εργασία [78] προσδιορίζει το ελάχιστο σύνολο 'ποιοτικά' ορισμένων προτιμήσεων που έχουν ως αποτέλεσμα τον αποκλεισμό ενός σημείου από την κορυφογραμμή.

Μια άλλη κατηγορία εργασιών εκμεταλλεύονται την ύπαρξη δεικτών στα δεδομένα για να καθοδηγήσουν την αναζήτηση των σημείων που ανήκουν στην κορυφογραμμή, κλαδεύοντας μεγάλα τμήματα του χώρου αναζήτησης. Μια εγγραφή μπορεί να θεωρηθεί ως σημείο ενός πολυδιάστατου χώρου. Συνεπώς, για τη δεικτοδότηση των δεδομένων μπορούν να χρησιμοποιηθούν δομές που έχουν προταθεί για πολυδιάστατα ή χωρικά δεδομένα. Ο αλγόριθμος εγγύτερου γείτονα (Nearest Neighbor (NN)) (εισήχθη στην εργασία [59]) που παρουσιάζεται στο [39] χρησιμοποιεί R-trees για να δεικτοδοτήσει τις εγγραφές και σε κάθε επανάληψη αναζητά τον κοντινότερο γείτονα μόνο σε μη κυριαρχημένες (non-dominated) περιοχές του χώρου. Υποθέτοντας (χάριν ευκολίας) ότι για όλα τα γνωρίσματα οι μικρότερες τιμές είναι προτιμητέες, η βασική ιδέα του αλγορίθμου NN είναι ότι όσο πιο κοντά είναι ένα σημείο στην αρχή των αξόνων, τόσο μεγαλύτερη πιθανότητα έχει να βρίσκεται τελικά στην κορυφογραμμή. Ο αλγόριθμος αυτός έχει το μειονέκτημα ότι χρειάζεται να εκτελέσει πολλαπλούς ελέγχους επειδή οι περιοχές του χώρου που έχουν ήδη κυριαρχηθεί (dominated areas) από ενδιάμεσα αποτελέσματα έχουν μεγάλο βαθμό επικάλυψης. Αντιθέτως, ο αλγόριθμος 'κλαδέματος - περίφραξης ' (Branch and Bound Skyline (BBS)) που προτείνεται στο [55] και χρησιμοποιεί επίσης R-trees αποδεικνύεται ότι είναι βέλτιστος όσον αφορά το μέγιστο πλήθος λειτουργιών ανάγνωσης και εγγραφής στο δίσκο που απαιτούνται. Ο BBS διατηρεί (α) μία στοίβα κόμβων του R-tree ταξινομημένους κατ' αύξουσα ελάχιστη Ευκλείδεια απόσταση από την αρχή των αξόνων, και (β) μία λίστα των σημείων που έχει ήδη βρεθεί ότι ανήκουν στην κορυφογραμμή. Κατά το άνοιγμα του πρώτου αντικειμένου της στοίβας, η κάτω αριστερή γωνία του ελάχιστου περιέχοντος κύβου (minimum bounding box (MBB)) ελέγχεται για κυριαρχία με τα σημεία της κορυφογραμμής. Αν κυριαρχείται τότε ολόκληρο το υπόδεντρο του συγκεκριμένου κόμβου μπορεί να κλαδευτεί. Αλλιώς οι απόγονοί του θα πρέπει να ελεγχτούν επίσης και πρέπει να εισαχθούν στη στοίβα. Η εκτέλεση σταματά όταν δεν έχουν μείνει προς εξέταση άλλα αντικείμενα στη στοίβα και τότε η τρέχουσα λίστα αποτελεσμάτων αποτελεί την κορυφογραμμή. Παρομοίως με τον αλγόριθμο SFS ο BBS εξετάζει τα σημεία σύμφωνα με μια μονότονη συναθροιστική συνάρτηση (Ευκλείδεια απόσταση από την αρχή των αξόνων), επιπλέον όμως κλαδεύει ολόκληρα τμήματα του χώρου αναζήτησης χωρίς να χρειαστεί να τα προσπελάσει. Ανάλογα καλά πειραματικά αποτελέσματα ισχύουν και αν τα σημεία προς εξέταση ταξινομηθούν σύμφωνα με άλλες καμπύλες γεμίσματος του χώρου (space filling curves) όπως η z-order [44].

Καθώς ο αριθμός των διαστάσεων-γνωρισμάτων αυξάνεται, το πλήθος των αποτελεσμάτων ενός ερωτήματος κορυφογραμμής μπορεί να γίνει εξαιρετικά μεγάλο, γεγονός που ουσιαστικά αχρηστεύει το ερώτημα. Δύο είναι οι κυριότερες προσεγγίσεις για την αντιμετώπιση του προβλήματος αυτού, γνωστού και ως curse of dimensionality. Μια κατηγορία εργασιών [55, 86, 17, 50] προσπαθεί να συνδυάσει τα θετικά στοιχεία των ερωτημάτων top-k και κορυφογραμμής, ενώ μια άλλη κατηγορία εργασιών [87, 69] εστιάζει σε ερωτήματα κορυφογραμμής σε υποχώρους (subspaces) του προβλήματος.

Στο [55], εισάγονται τα ερωτήματα *k-skyband* τα οποία επιστρέφουν τις εγγραφές που κυριαρχούνται από λιγότερες από k άλλες εγγραφές, ενώ η κορυφογραμμή συμπίπτει με το 1-skyband. Στη εργασία [86] χρησιμοποιούνται συναθροιστικά R-trees (aggregate R-trees) για να κατατάξουν τις εγγραφές με βάση των αριθμό των εγγραφών επί των οποίων κυριαρχούν, ενώ στο [17] η έννοια της κυριαρχίας χαλαρώνει σε k-κυριαρχία (k-dominance) ώστε περισσότερα σημεία να μπορούν να αποκλειστούν από το αποτέλεσμα. Μια παραλλαγή του ερωτήματος που επιστρέφει τα k πιο αντιπροσωπευτικά αποτελέσματα (k most representative skyline) συζητείται στο [50]. Το ερώτημα αυτό επιλέγει ένα υποσύνολο k σημείων που ανήκουν στην κορυφογραμμή έτσι ώστε ο αριθμός των σημείων που κυριαρχούνται από τα k σημεία να μεγιστοποιείται. Το ερώτημα *skycube* [87] επιστρέφει τις εγγραφές που δεν κυριαρχούνται σε ένα προκαθορισμένο υποσύνολο των διαστάσεων, ενώ στο [69] προτείνονται τεχνικές δεικτοδότησης με χρήση B-trees για την πιο αποδοτική αποτίμηση ερωτημάτων κορυφογραμμής σε υποχώρους.

Διάφορες άλλες επεκτάσεις και παραλλαγές του προβλήματος έχουν επίσης μελετηθεί στη βιβλιογραφία. Στα δυναμικά ερωτήματα κορυφογραμμής (dynamic skyline query) [55], οι προτιμήσεις μεταξύ των τιμών των γνωρισμάτων δεν είναι εκ των προτέρων γνωστές, αλλά προσδιορίζονται κατά τον χρόνο εκτέλεσης του ερωτήματος. Δύο νεότερες εργασίες [76, 62] ασχολούνται με δυναμικά ερωτήματα κορυφογραμμής για κατηγορικά γνωρίσματα. Στην απλούστερη μορφή ενός δυναμικού ερωτήματος κορυφογραμμής μια 'ιδανική' εγγραφή q δίνεται ως είσοδος στο ερώτημα. Σε αυτή την περίπτωση, οι σχέσεις κυριαρχίας μεταξύ των υπόλοιπων πλειάδων δεν προσδιορίζονται πλέον με βάση την αρχή των αξόνων αλλά σε σχέση με την εγγραφή q. Η εργασία [61] παρουσιάζει τεχνικές προσωρινής αποθήκευσης (caching) των αποτελεσμάτων προηγούμενων ερωτημάτων με σκοπό την επιτάχυνση της αποτίμησης νέων ερωτημάτων κορυφογραμμής. Οι μέθοδοι που προτείνονται στις εργασίες [25, 64] ασχολούνται με την εύρεση των εγγραφών που δεν κυριαρχούνται σε σχέση με πολλαπλά ιδανικά σημεία.

Η έννοια των πιθανοτικών ερωτημάτων κορυφογραμμής εισάγεται στην εργασία [56] για να αντιμετωπίσει το πρόβλημα στην περίπτωση που κάθε εγγραφή αναπαρίσταται από ένα σύνολο αβέβαιων στιγμιότυπων. Κάθε αβέβαιη εγγραφή αναπαρίσταται με τη βοήθεια ενός ελάχιστου περιέχοντος κύβου (MBB) που περικλείει όλα τα πιθανά στιγμιότυπα της συγκεκριμένης εγγραφής. Σε αυτή την περίπτωση, τα MBBs κάθε εγγραφής είναι πιθανό να επικαλύπτονται και δεν μπορεί να προσδιοριστεί με βεβαιότητα αν μια εγγραφή B κυριαρχεί επί της A. Εφόσον οι σχέσεις κυριαρχίας ανάμεσα στις εγγραφές ισχύουν με κάποια πιθανότητα, η πιθανότητα η εγγραφή A να ανήκει στην κορυφογραμμή είναι ίση με την πιθανότητα καμία άλλη εγγραφή B να μην κυριαρχεί επί

της Α. Ένα πιθανοτικό ερώτημα p-skyline επιστρέφει τις εγγραφές με πιθανότητα να ανήκουν στην κορυφογραμμή μεγαλύτερη ή ίση από ένα κατώφλι p.

2.3 Ερωτήματα Έρευνας Αγοράς

Στη βιβλιογραφία έχουν δημοσιευθεί αρκετές εργασίες που προσεγγίζουν το πρόβλημα της εξατομίκευσης από την πλευρά μιας επιχείρησης. Και αυτό γιατί στο σύγχρονο περιβάλλον, μια επιχείρηση καλείται να προσελκύσει καταναλωτές με διαφορετικά χαρακτηριστικά και καταναλωτικές συμπεριφορές, ακολουθώντας κατά το δυνατόν εξατομικευμένες στρατηγικές. Σε αυτό το πλαίσιο, η αποτύπωση των προτιμήσεων των πελατών μιας επιχείρησης παίζει σημαντικό ρόλο στη λήψη αποφάσεων για τη σχεδίαση και προώθηση νέων προϊόντων στην αγορά.

Η επιστήμη που κατά κύριο λόγο ασχολείται με την υποστήριξη της λήψης τέτοιου είδους αποφάσεων είναι η Επιχειρησιακή Έρευνα (Operational Research). Διάφορες υποπεριοχές της Επιχειρησιακής Έρευνας εστιάζουν στην εφαρμογή αναλυτικών μεθόδων (στατιστικής, εξόρυξης δεδομένων κλπ.) για την υποστήριξη της λήψης αποφάσεων. Δεδομένου ότι πολλά από τα ερευνητικά προβλήματα της Επιχειρησιακής Έρευνας έχουν να κάνουν με τη ανάλυση μεγάλων όγκων δεδομένων, το τελευταίο διάστημα έχει αναπτυχθεί έντονο ενδιαφέρον για την εφαρμογή μεθόδων διαχείρισης δεδομένων στην επίλυση αυτών των προβλημάτων. Το πεδίο της έρευνας αγοράς ιδιαιτέρως, περιλαμβάνει τη συστηματική συλλογή, καταγραφή και ανάλυση δεδομένων που σχετίζονται με μια συγκεκριμένη αγορά λαμβάνοντας υπόψη παράγοντες όπως τα διαθέσιμα προϊόντα και τα χαρακτηριστικά τους, τον ανταγωνισμό και την καταναλωτική συμπεριφορά.

Στην εργασία [37] προτείνεται για πρώτη φορά η σχηματοποίηση αρκετών προβλημάτων της επιχειρησιακής έρευνας, όπως για παράδειγμα η εύρεση πιθανών αγοραστών (potential customers identification), η προώθηση των ιδιαίτερων χαρακτηριστικών (product feature promotion) και η τοποθέτηση ενός προϊόντος στην αγορά (product positioning), ως προβλήματα βελτιστοποίησης από τη σκοπιά της επιστήμης της εξόρυξης δεδομένων.

Στο χώρο των βάσεων δεδομένων η εργασία [47] προτείνει διάφορα είδη ερωτημάτων ανάλυσης αγοράς, βασισμένη στην ανάλυση των σχέσεων κυριαρχίας (dominance relationships) μεταξύ των ανταγωνιστικών προϊόντων αλλά και στις προτιμήσεις των καταναλωτών. Στόχος είναι να βοηθηθεί μια επιχείρηση να τοποθετήσει τα προϊόντα της όσο το δυνατόν καλύτερα στην αγορά ώστε να προσελκύσει όσο το δυνατόν περισσότερους πελάτες.

Αρκετές εργασίες [24, 48, 81, 71, 26, 13] εστιάζουν στο πρόβλημα της εύρεσης πιθανών αγοραστών για ένα προϊόν. Με στόχο την ανάδειξη των χαρακτηριστικών ενός προϊόντος σε σχέση με τον ανταγωνισμό, η εργασία [51] επικεντρώνεται στο πρόβλημα του προσδιορισμού και της προώθησης των χαρακτηριστικών ενός προϊόντος, τα οποία το καθιστούν ανταγωνιστικό σε σχέση με άλλα προϊόντα. Στην εργασία [80] το πρόβλημα της προώθησης ενός προϊόντος μετασχηματίζεται σε πρόβλημα εύρεσης 'ελκυστικών' υποχωρών, δηλαδή υποσυνόλων διαστάσεων για τις οποίες το προϊόν

έχει καλή κατάταξη. Ένα αντίστοιχο πρόβλημα είναι η εύρεση των top-k καλύτερων περιοχών για την προώθηση ενός προϊόντος [79].

Μια άλλη πρακτική εφαρμογή σχετίζεται με τη σχεδίαση νέων προϊόντων με σκοπό τη μεγιστοποίηση του εκτιμώμενου οφέλους (utility function), ένα πρόβλημα ευρύτερα γνωστό ως βέλτιστη τοποθέτηση ενός προϊόντος [73, 72, 74, 49, 57]. Η συνάρτηση ωφέλειας μπορεί να περιλαμβάνει διάφορους παράγοντες όπως ο αριθμός εκτιμώμενων αγοραστών [72, 74, 49, 57], το τελικό κέρδος (τιμή προϊόντος μείον κόστος παραγωγής) [47, 73, 74, 57] ή ο ανταγωνισμός [47, 49]. Σε άλλες εργασίες [73, 74] εισάγεται το πρόβλημα της εύρεσης ανταγωνιστικών πακέτων συνθέτοντας επιμέρους προσφορές προϊόντων, για παράδειγμα τιμές πτήσεων με τιμές ξενοδοχείων. Ένα πακέτο είναι ανταγωνιστικό αν δεν κυριαρχείται από άλλα πακέτα.

Εστιάζοντας σε συναρτήσεις ωφέλειας που σχετίζονται με τον αριθμό των αναμενόμενων αγοραστών, ένα ζητούμενο είναι ο τρόπος αναπαράστασης των προτιμήσεων των καταναλωτών. Μια δημοφιλής μέθοδος υποθέτει την ύπαρξη μιας συναθροιστικής συνάρτησης με βάρη τα οποία αντικατοπτρίζουν τη σχετική σημασία που έχει κάθε χαρακτηριστικό γνώρισμα για έναν συγκεκριμένο καταναλωτή. Με βάση αυτή την παραδοχή, σε κάθε προϊόν ανατίθεται ένα σκορ εφαρμόζοντας την αντίστοιχη συναθροιστική συνάρτηση προτίμησης στις τιμές του προϊόντος. Τα προϊόντα που λαμβάνουν τα υψηλότερα σκορ θεωρούνται ως τα πιο ελκυστικά για τον συγκεκριμένο χρήστη. Οι εργασίες [71, 72] ακολουθούν αυτή την προσέγγιση εισάγοντας την έννοια των αντίστροφων top-k ερωτημάτων. Ένα αντίστροφο top-k ερώτημα επιστρέφει τον γεωμετρικό τόπο των διανυσμάτων (συναρτήσεων συνάθροισης) για τα οποία (διανύσματα) ένα προϊόν q ανήκει στα top-k.

Παρολαυτά, στην πράξη είναι συχνά αρκετά δύσκολο να εξαχθεί με ακρίβεια η συνάρτηση συνάθροισης για κάθε χρήστη του συστήματος [68]. Αντιθέτως, ένας πιο φυσικός τρόπος αναπαράστασης προτιμήσεων μπορεί να προκύψει επιτρέποντας στους χρήστες να καθορίσουν ευθέως τα ιδανικά γι'αυτούς χαρακτηριστικά για ένα προϊόν. Ακολουθώντας αυτή την προσέγγιση, τόσο τα προϊόντα όσο και οι προτιμήσεις των καταναλωτών μπορούν να αναπαρασταθούν ως σημεία στον ίδιο πολυδιάστατο χώρο. Σε μια τέτοια περίπτωση, έχουν προταθεί διαφορετικοί τρόποι μέτρησης της ικανοποίησης ενός καταναλωτή από ένα προϊόν. Μια επιλογή είναι να επιτρέπεται στους χρήστες να καθορίσουν μια ελάχιστα αποδεκτή τιμή σε κάθε διάσταση-γνώρισμα [57]. Με βάση αυτό τον τρόπο προσέγγισης, όλα τα προϊόντα που έχουν καλύτερα χαρακτηριστικά από τα ελάχιστα αποδεκτά μπορούν να θεωρηθούν ως ικανοποιητικά. Ένας σημαντικός περιορισμός μιας τέτοιας σχηματοποίησης είναι ότι δεν μπορεί να εφαρμοστεί για τύπους γνωρισμάτων που δεν έχουν ολική κατάταξη, αλλά η βέλτιστη τιμή τους είναι υποκειμενική ανά χρήστη. Επιπλέον, δεν δίνει αίσθηση του βαθμού ικανοποίησης ενός καταναλωτή από το προϊόν.

Συνεπώς, μια άλλη επιλογή για τη μέτρηση του βαθμού ικανοποίησης είναι με βάση το πόσο κοντά βρίσκονται οι τιμές του προϊόντος στις ιδανικές για τον χρήστη. Με βάση αυτή τη λογική, μπορεί κάποιος να προσδιορίσει τα top-k πιο ελκυστικά προϊόντα για έναν καταναλωτή, με τη βοήθεια ενός ερωτήματος εγγύτερου γείτονα. Δοθέντος ενός συνόλου σημείων P και ενός σημείου q, ένα ερώτημα εγγύτερου γείτονα (Nearest Neighbor (NN) query) [59] επιστρέφει το σημείο $p \in P$ που έχει τη μικρότερη απόστα-

ση από το q. Ακόμα όμως και σε αυτή την περίπτωση, στην πράξη είναι αρκετές φορές δύσκολο να προσδιοριστεί μια κατάλληλη συνάρτηση απόστασης, κυρίως επειδή η κάθε διάσταση έχει διαφορετική βαρύτητα που εξαρτάται από τις προτιμήσεις ενός χρήστη, ή επειδή κάθε διάσταση συνήθως έχει τη δική της μονάδα μέτρησης.

Με στόχο να αντιμετωπίσουν κάποιους από τους προαναφερόμενους περιορισμούς, τα ερωτήματα κορυφογραμμής έχουν χρησιμοποιηθεί ευρέως και για εφαρμογές έρευνας αγοράς. Έστω ότι οι προτιμήσεις των χρηστών και τα χαρακτηριστικά των προϊόντων ανήκουν στα σύνολα C και P αντίστοιχα. Σε αυτή την περίπτωση η έννοια του αντίστροφου ερωτήματος κορυφογραμμής (*reverse skyline query*) ως προς ένα σημείο q που εισάγεται στην εργασία [24] χρησιμοποιείται με σκοπό την εύρεση όλων των σημείων $p \in P$ για τα οποία το q ανήκει στη δυναμική κορυφογραμμή τους. Με άλλα λόγια ένα τέτοιο ερώτημα επιστρέφει όλους τους πελάτες $c \in C$ για τους οποίους ένα συγκεκριμένο προϊόν q είναι ελκυστικό. Στη συγκεκριμένη εργασία προτείνεται μια παραλλαγή του αλγορίθμου BBS [55] που χρησιμοποιείται για συμβατικά ερωτήματα κορυφογραμμής, με στόχο το κλάδεμα μεγάλου μέρους του χώρου αναζήτησης.

Η εργασία [48] βελτιώνει τη μέθοδο της [24] εξετάζοντας το πρόβλημα στην περίπτωση που οι προτιμήσεις και τα χαρακτηριστικά των προϊόντων είναι αβέβαια, προτείνοντας έναν αλγόριθμο που επιλύει και το αρχικό πρόβλημα ελαχιστοποιώντας τις λειτουργίες εγγραφής - ανάγνωσης από το δίσκο. Ο αλγόριθμος BRS που παρουσιάζεται στην εργασία [81] προτείνει κάποιες επιμέρους βελτιστοποιήσεις της μέθοδο που είχε προταθεί στο [48] για την περίπτωση βέβαιων δεδομένων. Η εργασία [26] εξετάζει το πρόβλημα για την περίπτωση μη μετρικών χώρων και προτείνει αλγορίθμους που υπολογίζουν αποδοτικά την αντίστροφη κορυφογραμμή σε αυτή την περίπτωση. Επιπλέον, η εργασία [75] μελετά το πρόβλημα της επεξεργασίας αντίστροφων ερωτημάτων κορυφογραμμής με ενεργειακά αποδοτικό τρόπο σε ένα ασύρματο δίκτυο αισθητήρων.

Τέλος, μπορεί να υποστηριχτεί ότι υπάρχει συνάφεια ανάμεσα στα ερωτήματα ανάλυσης αγοράς και σε εργασίες που έχουν γίνει σχετικά με το πρόβλημα της βέλτιστης χωροθέτησης (facility location planning). Στα αντίστροφα ερωτήματα εγγύτερου γείτονα (Reverse Nearest Neighbor queries (RNN)) [38], για ένα σύνολο σημείων Q, επιστρέφονται τα σημεία $q \in Q$, για τα οποία ένα σημείο $s \in S$ είναι ο εγγύτερος γείτονάς τους. Η εύρεση των k καλύτερων σημείων s σε μια προκαθορισμένη περιοχή του χώρου ώστε να μεγιστοποιείται ο αριθμός των σημείων q που τα έχουν ως εγγύτερο γείτονα αντιμετωπίζεται στην εργασία [82]. Το πρόβλημα αυτό είναι ακριβώς ανάλογο με το πρόβλημα που αντιμετωπίζεται και στην έρευνά μας. Σε νεότερη εργασία [77] εξετάζεται το πρόβλημα της εύρεσης του σημείου που μεγιστοποιεί τον αριθμό των αντίστροφων εγγύτερων γειτόνων του χωρίς περιορισμό σε συγκεκριμένη περιοχή του χώρου.

Κεφάλαιο 3

Διαχείριση Δεδομένων Εξαρτώμενων από Προτιμήσεις

Στην ενότητα 3.1 εισάγουμε το προτεινόμενο μοντέλο αναπαράστασης προτιμήσεων και στην ενότητα 3.2 παρουσιάζουμε το επεκτεταμένο μοντέλο δεδομένων και ερωτημάτων. Η ενότητα 3.3 περιλαμβάνει μια σύνοψη του συστήματος *PrefDB*. Στην ενότητα 3.4 ασχολούμαστε με τη βελτιστοποίηση και εκτέλεση ερωτημάτων με προτιμήσεις στο σύστημα *PrefDB*. Η ενότητα 3.5 παρουσιάζει την πειραματική αξιολόγηση του συστήματος.

3.1 Μοντέλο Αναπαράστασης Προτιμήσεων

Έστω μια σχεσιακή βάση δεδομένων. Κάθε σχέση R_B αποτελείται από ένα σύνολο γνωρισμάτων $\mathbf{A} = \{A_1, \ldots, A_d\}$ και ας θεωρήσουμε μια εγγραφή t που ανήκει σε ένα στιγμιότυπο της σχέσης R_B. Για κάθε γνώρισμα A_i, $1 \leq i \leq d$, θα χρησιμοποιούμε το συμβολισμό $t.A_i$ για να αναφερθούμε στην τιμή της t για το γνώρισμα A_i, και το $dom(A_i)$ για το πεδίο τιμών του γνωρίσματος A_i.

Μία προτίμηση p ορίζεται πάνω σε μία σχέση R_B ως μία τριπλέτα που αποτελείται από (α) μία συνθήκη προτίμησης (condition) με την οποία καθορίζονται οι εγγραφές που επηρεάζονται από την προτίμηση, (β) μία συνάρτηση βαθμολόγησης (scoring function) που χρησιμοποιείται για τη βαθμολόγηση αυτών των εγγραφών, και (γ) ένα βαθμό εμπιστοσύνης (confidence) που καθορίζει πόσο ισχυρή και βέβαιη είναι η συγκεκριμένη προτίμηση. Πιο αναλυτικά:

Ορισμός 3.1. *(Προτίμηση). Μια προτίμηση p ορισμένη πάνω στη σχέση R_B είναι μια τριπλέτα (σ_ϕ, S, C), όπου σ_ϕ συμβολίζει μια συνθήκη προτίμησης που εμπλέκει ένα σύνολο γνωρισμάτων $\mathbf{A}_\phi \subseteq \mathbf{A}$ από τη σχέση R_B, S είναι μια συνάρτηση βαθμολόγησης ορισμένη στο καρτεσιανό γινόμενο ενός συνόλου γνωρισμάτων $\mathbf{A_s} \subseteq \mathbf{A}$ της R_B, έτσι ώστε S: $\prod_{A_i \in \mathbf{A_s}} dom(A_i) \rightarrow [0,1] \cup \{\perp\}$, και C είναι μια σταθερά στο διάστημα $[0,1]$.*

Με άλλα λόγια μια προτίμηση p αναθέτει σε κάθε εγγραφή t που ανήκει στο $\sigma_\phi(R_B)$ ένα αριθμητικό σκορ εφαρμόζοντας τη συνάρτηση βαθμολόγησης S με βαθμό βεβαιότητας C. Ακολουθώντας αυτή τη σημασιολογία, μια εγγραφή t προτιμάται σε σχέση

με μια εγγραφή t', αν η t έχει υψηλότερο σκορ από την t'. Η τιμή $\perp$ για ένα σκορ υποδηλώνει απουσία οποιασδήποτε προτίμησης για τη συγκεκριμένη εγγραφή και χρησιμοποιείται ως το προκαθορισμένο σκορ. Επίσης η τιμή 1 για ένα σκορ εκφράζει απόλυτη προτίμηση, ενώ η τιμή 0 εκφράζει απόλυτη αντιπάθεια για την συγκεκριμένη εγγραφή.

Αξίζει να σημειωθεί ότι τα πεδία τιμών για το σκορ και τον βαθμό βεβαιότητας θα μπορούσαν να είναι διαφορετικά, για παράδειγμα θα μπορούσαμε να χρησιμοποιούμε θετικά σκορ για προτίμηση και αρνητικά σκορ για απέχθεια. Η επιλογή του πεδίου τιμών είναι ορθογώνια με το προτεινόμενο πλαίσιο. Επιλέξαμε το διάστημα $[0, 1] \cup \{\perp\}$ για τα σκορ, επειδή στις περισσότερες εφαρμογές οι χρήστες είναι πιο συνηθισμένοι να βαθμολογούν αντικείμενα σε μια θετική παρά σε μια αρνητική κλίμακα, για παράδειγμα αναθέτοντας 1-5 αστέρια ή βαθμολογίες στο διάστημα 1-10, όπου οι μικρότερες τιμές υποδηλώνουν απέχθεια.

Η συνάρτηση βαθμολόγησης S είναι δυνατόν να αναθέτει το ίδιο σταθερό σκορ σε όλες τις εγγραφές στο $\sigma_\phi(R_B)$ ή να αναθέτει διαφορετικά σκορ σε κάθε εγγραφή ανάλογα με τις τιμές τους για τα γνωρίσματα $\mathbf{A_s}$. Στην βιβλιογραφία έχουν προταθεί αρκετές μέθοδοι εξαγωγής μιας συνάρτησης βαθμολόγησης S, ώστε αυτή να αντικατοπτρίζει κατά το δυνατόν τις προτιμήσεις ενός χρήστη, για παράδειγμα χρησιμοποιώντας τεχνικής μηχανικής μάθησης [22, 35], εξόρυξη προτιμήσεων με βάση προηγούμενες αναζητήσεις (query log mining [32], κ.α. Εφεξής, θα κάνουμε την παραδοχή ότι η κατάλληλη συνάρτηση βαθμολόγησης και συνθήκες προτιμήσεις έχουν εξαχθεί για κάθε χρήστη.

Ο βαθμός βεβαιότητας αντικατοπτρίζει την αβεβαιότητα που δημιουργεί η μέθοδος εκμάθησης των προτιμήσεων ενός χρήστη. Διαισθητικά, μεγαλύτερος βαθμός βεβαιότητας υποδεικνύει πιο ισχυρή ένδειξη για μια προτίμηση. Για παράδειγμα, μια προτίμηση που έχει δηλωθεί ρητά από τον ίδιο τον χρήστη θα έχει τον μέγιστο βαθμό βεβαιότητας που είναι ίσος με 1. Αντιθέτως, για προτιμήσεις που δεν έχουν εκφραστεί από τους χρήστες αλλά έχουν εξαχθεί από το σύστημα, ο βαθμός βεβαιότητάς τους εξαρτάται από την ακολουθούμενη μέθοδο εκμάθησης, και είναι χαμηλότερος του 1. Για παράδειγμα, αν ένας χρήστης έχει παρακολουθήσει πολλές κωμωδίες, τότε μια μέθοδος εκμάθησης μπορεί να συμπεράνει ότι ο χρήστης αγαπά τις κωμωδίες. Σε αυτή την περίπτωση, ο βαθμός βεβαιότητας που θα ανατεθεί στη συγκεκριμένη προτίμηση θα πρέπει να λάβει υπόψη το μέγεθος του δείγματος, δηλαδή τον αριθμό κωμωδιών που ο χρήστης παρακολούθησε σε σχέση με το συνολικό πλήθος ταινιών της βάσης δεδομένων. Ο βαθμός εμπιστοσύνης μιας προτίμησης χρησιμοποιείται ως βάρος που επηρεάζει το συνολικό σκορ μιας εγγραφής (ενότητα 3.2.1).

Όπως θα δούμε αργότερα στην ενότητα 3.2.4, διαφορετικά είδη ερωτημάτων με προτιμήσεις μπορούν να σχηματιστούν, όπου η αναμενόμενη απάντηση καθορίζεται με βάση έναν οποιονδήποτε συνδυασμό σκορ, βαθμών βεβαιότητας και συνθηκών επιλογής. Για παράδειγμα, μια εφαρμογή μπορεί να αναζητά τις εγγραφές που είναι πολύ πιθανό (δηλαδή με μεγάλο βαθμό εμπιστοσύνης) να είναι αρεστές σε κάποιον χρήστη (με υψηλά σκορ). Μια άλλη εφαρμογή είναι δυνατόν να επιλέξει να επιστρέψει πιο τυχαία αποτελέσματα που ο χρήστης είναι λιγότερο πιθανό να θεωρήσει ενδιαφέροντα (χαμηλός βαθμός εμπιστοσύνης).

MOVIES(m_id, title, year, duration, movie_type, d_id),
DIRECTORS(d_id, director), GENRES(m_id, genre),
ACTORS(a_id, actor), CAST(m_id, a_id, role),
RATINGS(m_id, rating, votes), AWARDS(m_id, award, year)

Πίνακας 3.1: *Σχήμα δεδομένων μιας βάσης ταινιών*

p_1 : Alice liked the movie 'Scoop'
p_2 : She didn't like 'Gran Torino'
p_3 : She is a fan of Ben Stiller as an actor
p_4 : She loves comedies
p_5 : She prefers higher-rated films, if voted by many users
p_6 : She likes recent movies that last around 2 hours
p_7 : She prefers the most recent movies by Woody Allen
p_8 : She would like to watch a recent romantic movie
p_9 : Award-winning movies are preferred

Πίνακας 3.2: *Σύνολο προτιμήσεων ενός χρήστη*

Στη συνέχεια περιγράφουμε διάφορα είδη προτιμήσεων που είναι δυνατόν να αναπαρασταθούν με βάση το προτεινόμενο μοντέλο.

Παράδειγμα 3.1. *Έστω η βάση δεδομένων με ταινίες που φαίνεται στον Πίνακα 3.1. Επίσης, στον Πίνακα 3.1 περιέχεται ένα ενδεικτικό σύνολο προτιμήσεων που έχουν εξαχθεί για έναν χρήστη της βάσης. Έστω ότι ο συγκεκριμένος χρήστης έχει βαθμολογήσει τις ταινίες 'Scoop' και 'Gran Torino' με 8/10 και 3/10 αντίστοιχα. Τότε, η προτίμηση p_1 για την ταινία 'Scoop' μπορεί να εκφραστεί ως $p_1[MOVIES] = (\sigma_{m\_id=m_5}, 0.8, 1)$ ενώ η προτίμηση p_2 για την ταινία 'Gran Torino' μπορεί να εκφραστεί ως $p_2[MOVIES] = (\sigma_{m\_id=m_1}, 0.3, 1)$. Παρατηρώντας βλέπουμε ότι η διαφορά μεταξύ των δύο προτιμήσεων έχει να κάνει με το σκορ, όπου στην δεύτερη περίπτωση το χαμηλότερο σκορ υποδηλώνει χαμηλό ενδιαφέρον για τη συγκεκριμένη ταινία. Και οι δύο προτιμήσεις μπορούν να χαρακτηριστούν ως ατομικές καθώς εμπλέκουν μια εγγραφή και ουσιαστικά αντικατοπτρίζουν τις βαθμολογίες που έδωσαν οι ίδιοι οι χρήστες. Για τον ίδιο λόγο και στις δύο περιπτώσεις ο βαθμός εμπιστοσύνης είναι ο μέγιστος δυνατός, ίσος με 1. Παρομοίως, η προτίμηση p_3 μπορεί να εκφραστεί με βάση το προτεινόμενο μοντέλο ως $p_3[ACTORS] = (\sigma_{a\_id=a_1}, 1, 1)$.*

Αντιθέτως, οι γενικευμένες προτιμήσεις αφορούν ένα σύνολο εγγραφών οι οποίες ικανοποιούν τη συνθήκη της προτίμησης. Οι προτιμήσεις $p_4 - p_9$ του Πίνακα 3.2 είναι γενικευμένες προτιμήσεις που έχουν εξαχθεί με βάση το ιστορικό αναζητήσεων του χρήστη καθώς και τις βαθμολογήσεις που έχει δώσει. Επειδή αυτές οι προτιμήσεις δεν έχουν δηλωθεί ρητά από τον χρήστη, ο βαθμός εμπιστοσύνης τους είναι χαμηλότερος του 1. Η προτίμηση p_4 μπορεί να εκφραστεί ως $p_4[GENRES] = (\sigma_{genre='Comedy'}, 1, 0.8)$. □

Για ευκολία στην παρουσίαση, στα επόμενα παραδείγματα θα χρησιμοποιήσουμε τις παρακάτω συναρτήσεις βαθμολόγησης:

- $S_r(rating) = 0.1 * rating$

- $S_m(year, x) = year/x$
- $S_d(duration, x) = 1 - |duration - x|/x$

Παράδειγμα 3.2. *Έστω η προτίμηση p_5 (Πίνακας 3.2) που μπορεί να αναπαρασταθεί ως $p_5[RATINGS] = (\sigma_{votes \geq 100}, S_r(rating), 0.9)$. Για τη συγκεκριμένη προτίμηση, η συνάρτηση βαθμολόγησης S_r χρησιμοποιείται με παράμετρο εισόδου τη βαθμολογία μίας ταινίας η οποία λαμβάνει τιμές στο διάστημα [0, 10] και αναθέτει υψηλότερα σκορ στις ταινίες με τις καλύτερες βαθμολογίες. Επίσης, η προτίμηση p_6 εμπλέκει πολλαπλά γνωρίσματα της σχέσης MOVIES και μπορεί να εκφραστεί ως $p_6[MOVIES] = (-, 0.5 * S_m(year, 2012) + 0.5 * S_d(duration, 120), 0.6)$. Συγκεκριμένα, η προτίμηση p_6 εφαρμόζει τις συναρτήσεις βαθμολόγησης S_m και S_d αναθέτοντας σκορ σε όλες τις ταινίες με βάση τόσο το έτος παραγωγής τους όσο και τη διάρκειά τους. Η συνάρτηση S_m αναθέτει υψηλότερα σκορ στις πιο πρόσφατες ταινίες, ενώ η S_d σε ταινίες με διάρκεια περίπου 2 ώρες.*

Οι προτιμήσεις p_7 και p_8 του Πίνακα 3.2 εμπλέκουν πολλαπλές σχέσεις καθώς ορίζονται στο καρτεσιανό γινόμενο των σχέσεων MOVIES $\bowtie$ DIRECTORS και MOVIES $\bowtie$ GENRES αντιστοίχως. Πιο αναλυτικά η προτίμηση p_7 μπορεί να οριστεί ως $p_7[MOVIES \bowtie DIRECTORS] = (\sigma_{director='W.Allen'}, S_m(year, 2012), 0.7)$ ενώ η προτίμηση p_8 ως $p_8[MOVIES \bowtie GENRES] = (\sigma_{genre='Romance'}, S_m(year, 2012), 0.8)$. Τέλος, η προτίμηση p_9 είναι μια προτίμηση 'συμμετοχής', δηλαδή ορίζει μια προτίμηση σε εγγραφές οι οποίες έχουν join με μια άλλη σχέση, και μπορεί να εκφραστεί ως $p_9[MOVIES \bowtie AWARDS] = (-, 1, 0.9)$. □

3.2 Επεκταμένο Σχεσιακό Μοντέλο

3.2.1 Επεκταμένες Σχέσεις

Με στόχο να εμπλουτίσουμε τις εγγραφές μιας βάσης με επιπλέον τιμές για το σκορ και το αντίστοιχο βαθμό εμπιστοσύνης, ορίζουμε μια *επεκταμένη σχέση*:

Ορισμός 3.2. *(Επεκταμένη Σχέση). Δοθείσης μιας βασικής σχέσης $R_B(A_1, ..., A_d)$, μια επεκταμένη σχέση R ορίζεται επεκτείνοντας τη βασική σχέση R_B με (α) ένα γνώρισμα σκορ S με πεδίο τιμών $dom(S) = \Re^+ \cup \{\perp\}$, και (β) ένα γνώρισμα εμπιστοσύνης C με πεδίο τιμών $dom(C) = \Re^+$. Η προκαθορισμένη τιμή για τα σκορ είναι $\perp$ και για τον βαθμό εμπιστοσύνης είναι 0, δηλαδή $R = \{(t, S_t, C_t) | t \in R_B, S_t = \perp, C_t = 0\}$. Κάθε εγγραφή t σχετίζεται με ένα μοναδικό ζεύγος τιμών $\langle S_t, C_t \rangle$.*

Σημειώνουμε ότι παρόλο που η μέγιστη τιμή για το σκορ και το βαθμό εμπιστοσύνης που μπορεί να ανατεθεί σε μια προτίμηση είναι 1, μια εγγραφή είναι δυνατό να έχει υψηλότερα σκορ ή τιμές εμπιστοσύνης ως αποτέλεσμα του συνδυασμού περισσότερων του ενός σκορ ή βαθμών εμπιστοσύνης που ισχύουν για τη συγκεκριμένη εγγραφή.

Υπάρχουν δύο τρόποι με τους οποίους μπορεί να ανατεθούν σε μια εγγραφή t ένα ζεύγος σκορ και βαθμού εμπιστοσύνης. Ο πρώτος τρόπος είναι αποτιμώντας μια προτίμηση πάνω στη σχέση, όπως περιγράφουμε στην ενότητα 3.2.3. Ο δεύτερος

τρόπος είναι μεταφέροντας στην εγγραφή αυτή τιμές από άλλες σχέσεις με τη βοήθεια ενός τελεστή. Στην πρώτη περίπτωση, τη στιγμή που αποτιμούμε μια προτίμηση στη σχέση, μπορεί η εγγραφή t να έχει ήδη κάποιο σκορ η βαθμό εμπιστοσύνης. Επομένως, χρειάζεται να συνδυάσουμε το παλιό ζεύγος σκορ και βαθμού εμπιστοσύνης με το νέο και να αναθέσουμε το τελικό ζεύγος τιμών στην t. Στη δεύτερη περίπτωση, όταν μια νέα εγγραφή σχηματίζεται ως αποτέλεσμα ενός σχεσιακού τελεστή, οι τιμές για το σκορ και τον βαθμό εμπιστοσύνης παράγονται από τα σκορ και βαθμούς εμπιστοσύνης των αρχικών εγγραφών. Για παράδειγμα, κατά την αποτίμηση ενός τελεστή σύζευξης (join), το ζεύγος τιμών σκορ-βαθμού εμπιστοσύνης που θα ανατεθεί στην παραγόμενη εγγραφή, προκύπτει από τις τιμές των εγγραφών στις αρχικές σχέσεις.

Και για τις δύο περιπτώσεις, ο συνδυασμός δύο ζευγών σκορ και βαθμού εμπιστοσύνης επιτυγχάνεται με τη βοήθεια μια συναθροιστικής συνάρτησης. Σύμφωνα με το προτεινόμενο μοντέλο ένα σκορ είναι πάντοτε συνδεδεμένο με έναν βαθμό εμπιστοσύνης. Εφεξής θα συμβολίσουμε ένα ζεύγος σκορ-βαθμού εμπιστοσύνης ως $\langle S, C\rangle$. Επίσης, είναι προφανές ότι μια συναθροιστική συνάρτηση πρέπει να μπορεί να χειρίζεται ζεύγη αντί για μεμονωμένες τιμές σκορ-βαθμών εμπιστοσύνης. Παρακάτω ακολουθεί ο ορισμός μια συναθροιστικής συνάρτησης.

Ορισμός 3.3. *(Συναθροιστική Συνάρτηση). Μία συναθροιστική συνάρτηση* $F : \langle S, C\rangle \times \langle S, C\rangle \rightarrow \langle S, C\rangle$ *συνδυάζει δύο ζεύγη σκορ-βαθμού εμπιστοσύνης σε ένα τελικό ζεύγος τιμών.*

Διαισθητικά, μια αλλαγή στη σειρά αποτίμησης των προτιμήσεων δεν θα πρέπει να επηρεάζει το τελικό ζεύγος σκορ-βαθμού εμπιστοσύνης. Επομένως, αναμένουμε να ισχύουν η προσεταιριστική και η αντιμεταθετική ιδιότητα για τις συναθροιστικές συναρτήσεις. Επιπλέον, οι συναθροιστικές επιβάλλεται να ικανοποιούν τις ακόλουθες ιδιότητες:

- $F(\langle\bot, 0\rangle, \langle\bot, 0\rangle) = \langle\bot, 0\rangle$
- $F(\langle\bot, 0\rangle, \langle S, C\rangle) = \langle S, C\rangle$

Οι συναθροιστικές συναρτήσεις μπορούν να χρησιμοποιηθούν για να υλοποιήσουν διαφορετικές φιλοσοφίες ή στρατηγικές για το πώς πρέπει να συνδυάζονται οι προτιμήσεις σε μια εφαρμογή. Η επιλογή της κατάλληλης συναθροιστικής συνάρτησης σε κάθε περίπτωση εξαρτάται από το ποια φιλοσοφία αντικατοπτρίζει με μεγαλύτερη ακρίβεια τον τρόπο που το σύστημα πρέπει να συμπεριφέρεται στην πράξη. Παρακάτω δίνουμε κάποια παράδειγμα για να φανεί καλύτερα ο πλούτος διαφορετικών προσεγγίσεων.

Παράδειγμα 3.3. *Μια πιθανή επιλογή που μπορούμε να ακολουθήσουμε κατά τον συνδυασμό δύο ζευγών τιμών σκορ-βαθμού εμπιστοσύνης, είναι να χρησιμοποιήσουμε τους επιμέρους βαθμούς εμπιστοσύνης ως βάρη, με τέτοιο τρόπο ώστε τα σκορ με υψηλότερο βαθμό εμπιστοσύνης να συνεισφέρουν περισσότερο στο τελικό ζεύγος τιμών. Επιπλέον, ο τελικός βαθμός εμπιστοσύνης θα πρέπει να αντικατοπτρίζει κατά το δυνατόν την συνολική 'αξιοπιστία' των προτιμήσεων που έχουν αποτιμηθεί. Με βάση αυτή τη λογική, η συναθροιστική συνάρτηση* F_S *υπολογίζει το παραγόμενο ζεύγος σκορ-βαθμού εμπιστοσύνης, ώστε το τελικό σκορ να ισούται με τον σταθμισμένο μέσο όρο*

των αρχικών σκορ χρησιμοποιώντας τους αντίστοιχους βαθμούς εμπιστοσύνης ως βάρη, και ο τελικός βαθμός εμπιστοσύνης να είναι ίσος με το άθροισμα των επιμέρους βαθμών εμπιστοσύνης.

$$F_S(\langle S_1, C_1\rangle, \langle S_2, C_2\rangle) = \langle C_1S_1 + C_2S_2, C_1 + C_2\rangle$$

Μια άλλη εναλλακτική προσέγγιση θα μπορούσε να υπολογίζει για παράδειγμα τον μέσο όρο των σκορ ή βαθμών εμπιστοσύνης (ή και τα δύο μαζί). Το άθροισμα γενικότερα αντικατοπτρίζει καλύτερα το πλήθος των προτιμήσεων που κάθε εγγραφή ικανοποιεί. Αντιθέτως, ο μέσος όρος είναι πιο κατάλληλος για να εξομαλύνει τις διαφορές στα σκορ που έχουν ανατεθεί σε μια εγγραφή από αντιφατικές προτιμήσεις. Γενικότερα, το ποια στρατηγική είναι πιο κατάλληλη σε κάθε περίπτωση εξαρτάται κυρίως από τη ζητούμενη συμπεριφορά που θα θέλαμε να έχει η εκάστοτε εφαρμογή. □

Παράδειγμα 3.4. *Ακολουθώντας μια διαφορετική φιλοσοφία θα μπορούσε το τελικό σκορ να καθορίζεται από την προτίμηση με τον υψηλότερο βαθμό εμπιστοσύνης, δηλαδή:*

$$F_{\max}(\langle S_1, C_1\rangle, \langle S_2, C_2\rangle) = \begin{cases} \langle S_j, C_j\rangle, j = \arg\max\limits_{i\in\{1,2\}}(C_i), C_1 \neq C_2 \\ \langle S_j, C_j\rangle, j = \arg\max\limits_{i\in\{1,2\}}(S_i), else \end{cases}$$

Η συνάρτηση F_{max} λαμβάνει ως είσοδο δύο ζεύγη τιμών σκορ-βαθμού εμπιστοσύνης $\langle S_1, C_1\rangle$, $\langle S_2, C_2\rangle$ και παράγει ως έξοδο το ζεύγος με τον υψηλότερο βαθμό εμπιστοσύνης. Αν και τα δύο ζεύγη έχουν τον ίδιο βαθμό εμπιστοσύνης, η συνάρτηση F_{max} θα επιστρέψει το ζεύγος με το υψηλότερο σκορ. □

Είναι σχετικά απλό να αποδειχτεί ότι οι παραπάνω συναρτήσεις συνάθροισης είναι αντιμεταθετικές και προσεταιριστικές. Χάριν απλότητας (και χωρίς βλάβη της γενικότητας του προτεινόμενου πλαισίου), στη συνέχεια θα χρησιμοποιήσουμε τη συνάρτηση F_S.

3.2.2 Βασικοί Τελεστές της Επεκτεταμένης Σχεσιακής Άλγεβρας

Στην ενότητα αυτή επεκτείνουμε τους κλασικούς τελεστές της σχεσιακής άλγεβρας έτσι ώστε να είναι σε θέση να χειρίζονται επεκτεταμένες σχέσεις.

- *Τελεστής Επιλογής*, $\sigma_\phi(R)$, επιλέγει τις εγγραφές εκείνες από μια επεκτεταμένη σχέση R οι οποίες ικανοποιούν τη συνθήκη ϕ, δηλαδή, $\sigma_\phi(R) = \{t | t \in R \wedge \sigma_\phi(t)\}$. Η συνθήκη επιλογής μπορεί να εμπλέκει επίσης τα νέα γνωρίσματα σκορ και βαθμού εμπιστοσύνης.

- *Τελεστής Προβολής*, $\pi_{A_1,A_2,...,A_k}(R)$, προβάλλει ένα υποσύνολο γνωρισμάτων της επεκτεταμένης σχέσης R, δηλαδή

$$\pi_{A_1,...,A_k}(R) = \{(t.A_1, ..., t.A_k, S_t, C_t) | t \in R\}$$

Επιπλέον των προβαλλόμενων γνωρισμάτων, ο τελεστής προβολής διατηρεί πάντα τα γνωρίσματα σκορ και βαθμού εμπιστοσύνης, έτσι ώστε η τελική σχέση που προκύπτει να είναι επεκτεταμένη.

- *Τελεστής Τομής*, $R_i \cap_F R_j$, λαμβάνει ως είσοδο δύο επεκτεταμένες σχέσεις R_i και R_j και παράγει μια επεκτεταμένη σχέση που περιλαμβάνει όλες τις εγγραφές που ανήκουν τόσο στην R_i όσο και στην R_j με τελικά σκορ και βαθμούς εμπιστοσύνης που προκύπτουν συνδυάζοντας τα σκορ και βαθμούς εμπιστοσύνης των εγγραφών που βρίσκονται στις αρχικές σχέσεις R_i και R_j, δηλαδή,

$$R_i \cap_F R_j = \{(t, S_t, C_t) | (t, S_i, C_i) \in R_i \wedge (t, S_j, C_j) \in R_j, \\ \langle S_t, C_t \rangle := F(\langle S_i, C_i \rangle, \langle S_j, C_j \rangle)\}$$

- *Τελεστής Ένωσης*, $R_i \cup_F R_j$, λαμβάνει ως είσοδο δύο επεκτεταμένες σχέσεις R_i και R_j και παράγει μια επεκτεταμένη σχέση που περιλαμβάνει όλες τις εγγραφές που ανήκουν είτε στην R_i είτε στην R_j είτε και στις δύο (οι διπλές εγγραφές παραλείπονται) με τελικά σκορ και βαθμούς εμπιστοσύνης που προκύπτουν συνδυάζοντας τα σκορ και βαθμούς εμπιστοσύνης των εγγραφών που βρίσκονται στις αρχικές σχέσεις R_i και R_j, δηλαδή,

$$R_i \cup_F R_j = \{(t, S_t, C_t) | \ ((t, S_i, C_i) \in R_i \wedge (t, *, *) \notin R_j, \langle S_t, C_t \rangle := \langle S_i, C_i \rangle) \vee \\ ((t, S_j, C_j) \in R_j \wedge (t, *, *) \notin R_i, \langle S_t, C_t \rangle := \langle S_j, C_j \rangle) \vee \\ ((t, S_i, C_i) \in R_i \wedge (t, S_j, C_j) \in R_j, \langle S_t, C_t \rangle := F(\langle S_i, C_i \rangle, \langle S_j, C_j \rangle))\}$$

Παράδειγμα 3.5. *Έστω δύο επεκτεταμένες σχέσεις R_A και R_B που ή κάθε μια περιέχει ένα σύνολο ταινιών που είναι ενδιαφέρουσες για δύο χρήστες A και B αντιστοίχως. Αν θέλουμε να προσδιορίσουμε τις ταινίες που θα μπορούσαν να δουν παρέα, τότε πρέπει να βρούμε τις εγγραφές που ανήκουν στην τομή των δύο επεκτεταμένων σχέσεων $R_A \cap_F R_B$. Σε αυτή την περίπτωση, οι τελικές τιμές σκορ και βαθμού εμπιστοσύνης μπορούν να υπολογιστούν εκτελώντας μια συναθροιστική συνάρτηση F στα επιμέρους ζεύγη σκορ-βαθμών εμπιστοσύνης.* □

- *Τελεστής Σύζευξης*, $R_i \bowtie_{\phi,F} R_j$ λαμβάνει ως είσοδο δύο επεκτεταμένες σχέσεις R_i και R_j και παράγει μια επεκτεταμένη σχέση ως:

$$R_i \bowtie_{\phi,F} R_j = \{(t, S_t, C_t) | \ t = t_i \bowtie_\phi t_j, (t_i, S_i, C_i) \in R_i, (t_j, S_j, C_j) \in R_j \\ \langle S_t, C_t \rangle := F(\langle S_i, C_i \rangle, \langle S_j, C_j \rangle)\}$$

Παράδειγμα 3.6. *Έστω οι επεκτεταμένες σχέσεις $MOVIES$ και $DIRECTORS$ που απεικονίζονται στα Σχήματα 3.1(α′), 3.1(β′). Το Σχήμα 3.1(γ′) δείχνει το αποτέλεσμα του $MOVIES \bowtie DIRECTORS$ (για λόγους εξοικονόμησης χώρου στο σχήμα δείχνουμε μόνο ορισμένα από τα γνωρίσματα της παραγόμενης σχέσης).* □

Σημειώνουμε ότι σε όλους τους συνολοθεωρητικούς τελεστές καθώς και στον τελεστή σύζευξης, χρησιμοποιείται μια κοινή συναθροιστική συνάρτηση F η οποία συνδυάζει τα σκορ που προέρχονται από τις επιμέρους σχέσεις. Χάριν απλότητας εφεξής θα γράφουμε $R_i \ \theta \ R_j$ εννοώντας $R_i \ \theta_F \ R_j$ όπου $\theta \in \{\cup, \cap, \bowtie_\phi\}$.

m_id	title	year	duration	d_id	score	conf
m_1	Gran Torino	2008	116	d_1	0.77	1.00
m_2	Wall Street	1987	126	d_3	0.83	1.00
m_3	Million Dollar Baby	2004	132	d_1	0.85	1.00
m_4	Match Point	2005	124	d_2	$\perp$	0.00
m_5	Scoop	2006	96	d_2	0.78	1.00

(α′) $MOVIES$

d_id	director	score	conf
d_1	C. Eastwood	0.80	1.00
d_2	W. Allen	0.90	0.90
d_3	O. Stone	$\perp$	0.00

(β′) $DIRECTORS$

m_id	d_id	score	conf
m_1	d_1	1.57	2.00
m_2	d_3	0.83	1.00
m_3	d_1	1.65	2.00
m_4	d_2	0.90	0.90
m_5	d_2	1.59	1.90

(γ′) $MOVIES \bowtie DIRECTORS$

Σχήμα 3.1: *Παράδειγμα αποτίμησης τελεστή σύζευξης σε επεκταμένες σχέσεις*

m_id	score	conf
m_1	1.77	2.00
m_2	0.83	1.00
m_3	1.85	2.00
m_4	1.00	1.00
m_5	1.78	2.00

(α′) $\lambda_{p_a}(MOVIES)$

m_id	score	conf
m_1	1.77	2.00
m_2	1.31	1.50
m_3	2.30	2.50
m_4	1.48	1.50
m_5	1.78	2.00

(β′) $\lambda_{p_b}(\lambda_{p_a}(MOVIES))$

Σχήμα 3.2: *Παραδείγματα χρήσης του τελεστή προτίμησης*

3.2.3 Τελεστής Προτίμησης

Στην ενότητα αυτή επεκτείνουμε τη σχεσιακή άλγεβρα με έναν νέο *τελεστή προτίμησης* $\lambda_{p,F}(R)$. Ο τελεστής προτίμησης αποτιμά μια προτίμηση p σε μια επεκτεταμένη σχέση R χρησιμοποιώντας την προκαθορισμένη συνάρτηση συνάθροισης F για να συνδυάσει τα προηγούμενα σκορ-βαθμούς εμπιστοσύνης με τα νέα που προκύπτουν από την τρέχουσα προτίμηση. Πιο συγκεκριμένα, ο τελεστής προτίμησης $\lambda_{p,F}(R)$ εκτελεί μια προτίμηση $p := (\sigma_\phi, S, C)$ σε μία επεκτεταμένη σχέση R και παράγει μια νέα επεκτεταμένη σχέση ως εξής:

$$\lambda_{p,F}(R) = \{(t, S'_t, C'_t) | (t, S_t, C_t) \in R \;\; and$$
$$\langle S'_t, C'_t \rangle := \left\{ \begin{array}{ll} F(\langle S_t, C_t \rangle, \langle S(t), C \rangle), & if \;\; t \in \sigma_\phi(R) \\ \langle S_t, C_t \rangle, & else \end{array} \right\}$$

Στη συνέχεια, χάριν ευκολίας θα χρησιμοποιούμε το σύμβολο λ_p αντί του $\lambda_{p,F}$.

Παράδειγμα 3.7. *Έστω η επεκταμένη σχέση MOVIES του Σχήματος 3.1(α′) και δύο προτιμήσεις:* $p_a[MOVIES] = (\sigma_{year \geq 2000}, S_m(year, 2012), 1)$ *και* $p_b[MOVIES] =$

$(\sigma_{duration \geq 120}, S_d(duration, 120), 0.5)$. *Τα Σχήματα 3.2(α΄), 3.2(β΄) δείχνουν τις επεκτεταμένες σχέσεις που προκύπτουν αποτιμώντας διαδοχικά τις προτιμήσεις* $\lambda_{p_a}(MOVIES)$ *και* $\lambda_{p_b}(\lambda_{p_a}(MOVIES))$. □

Παρακάτω παρουσιάζουμε ένα σύνολο ιδιοτήτων που ισχύουν για τον τελεστή προτίμησης. Όπως θα περιγράψουμε στην ενότητα 3.4.2, οι συγκεκριμένες ιδιότητες εφαρμόζονται κατά το στάδιο της βελτιστοποίησης ενός ερωτήματος με σκοπό να προκύψει ένα πιο αποδοτικό πλάνο εκτέλεσης.

Ιδιότητα 1. *Αν ο τελεστής επιλογής* $\sigma_{\phi'}$ *εμπλέκει οποιοδήποτε γνώρισμα εκτός του σκορ και του βαθμού εμπιστοσύνης, τότε ο τελεστής προτίμησης και ο τελεστής επιλογής μπορούν να αντιμετατεθούν μεταξύ τους δηλαδή,* $\sigma_{\phi'}\lambda_p(R) = \lambda_p\sigma_{\phi'}(R)$.

Απόδειξη. Έστω μια επεκτεταμένη σχέση R και μια τυχαία εγγραφή $(t, S_t, C_t) \in R$ με τιμές S_t και C_t για το σκορ και το βαθμό εμπιστοσύνης αντιστοίχως. Επίσης, έστω μια προτίμηση $p[R] = (\sigma_\phi, S, C)$. Βάσει ορισμού, η συνθήκη προτίμησης σ_ϕ εμπλέκει αποκλειστικά γνωρίσματα εκτός του σκορ και του βαθμού εμπιστοσύνης. Επιπλέον, η συνθήκη του τελεστή επιλογής $\sigma_{\phi'}$ επίσης εμπλέκει μόνο γνωρίσματα εκτός του σκορ και του βαθμού εμπιστοσύνης. Επομένως ισχύει ότι η εγγραφή (t, S_t, C_t) ικανοποιεί την συνθήκη σ_ϕ (αντιστοίχως $\sigma_{\phi'}$) αν και μόνο αν ισχύει $\sigma_\phi(t)$ (αντιστοίχως $\sigma_{\phi'}(t)$). Επομένως:

$$
\begin{aligned}
&(t,\, S', C') \in \sigma_{\phi'}\lambda_p(R)\\
&\quad \Leftrightarrow (t, S_t, C_t) \in R \wedge \sigma_{\phi'}(t) \wedge \lambda_p(t, S_t, C_t)\\
&\quad \Leftrightarrow (t, S_t, C_t) \in R \wedge \sigma_{\phi'}(t) \wedge ((\langle S', C'\rangle = \langle S_t, C_t\rangle \wedge \neg\sigma_\phi(t))\\
&\qquad \vee \quad (\langle S', C'\rangle = F(\langle S_t, C_t\rangle, \langle S(t), C\rangle) \wedge \sigma_\phi(t)))\\
&\quad \Leftrightarrow ((t, S_t, C_t) \in R \wedge \sigma_{\phi'}(t) \wedge \neg\sigma_\phi(t) \wedge \langle S', C'\rangle = \langle S_t, C_t\rangle)\\
&\qquad \vee \quad ((t, S_t, C_t) \in R \wedge \sigma_{\phi'}(t) \wedge \sigma_\phi(t) \wedge \langle S', C'\rangle = F(\langle S_t, C_t\rangle, \langle S(t), C\rangle))
\end{aligned}
$$

Στο άλλο σκέλος της εξίσωσης έχουμε:

$$
\begin{aligned}
&(t,\, S', C') \in \lambda_p\sigma_{\phi'}(R)\\
&\quad \Leftrightarrow (t, S_t, C_t) \in R \wedge \sigma_{\phi'}(t) \wedge ((\langle S', C'\rangle = \langle S_t, C_t\rangle \wedge \neg\sigma_\phi(\sigma_{\phi'}(t)))\\
&\qquad \vee \quad (\langle S', C'\rangle = F(\langle S_t, C_t\rangle, \langle S(t), C\rangle) \wedge \sigma_\phi(\sigma_{\phi'}(t))))\\
&\quad \Leftrightarrow (t, S_t, C_t) \in R \wedge \sigma_{\phi'}(t) \wedge ((\langle S', C'\rangle = \langle S_t, C_t\rangle \wedge \neg\sigma_\phi(t) \wedge \sigma_{\phi'}(t))\\
&\qquad \vee \quad (\langle S', C'\rangle = F(\langle S_t, C_t\rangle, \langle S(t), C\rangle) \wedge \sigma_\phi(t) \wedge \sigma_{\phi'}(t)))\\
&\quad \Leftrightarrow ((t, S_t, C_t) \in R \wedge \sigma_{\phi'}(t) \wedge \neg\sigma_\phi(t) \wedge \langle S', C'\rangle = \langle S_t, C_t\rangle)\\
&\qquad \vee \quad ((t, S_t, C_t) \in R \wedge \sigma_{\phi'}(t) \wedge \sigma_\phi(t) \wedge \langle S', C'\rangle = F(\langle S_t, C_t\rangle, \langle S(t), C\rangle))
\end{aligned}
$$

Συνεπώς από τα παραπάνω προκύπτει ότι $\sigma_{\phi'}\lambda_p(R) = \lambda_p\sigma_{\phi'}(R)$. □

Όπως έχουμε αναφέρει: (α) η συναθροιστική συνάρτηση F είναι αντιμεταθετική και προσεταιριστική, και (β) η ίδια συνάρτηση F χρησιμοποιείται από όλους τους τελεστές του ίδιου ερωτήματος. Συνεπώς ισχύουν οι παρακάτω ιδιότητες:

Ιδιότητα 2. *Ο τελεστής προτίμησης είναι αντιμεταθετικός, δηλαδή ισχύει:* $\lambda_{p_1}(\lambda_{p_2}(R)) = \lambda_{p_2}(\lambda_{p_1}(R))$.

Απόδειξη. Έστω μια επεκτεταμένη σχέση R και $(t, S_t, C_t) \in R$ μια τυχαία εγγραφή αυτής. Επίσης έστω οι προτιμήσεις $p_1[R] = (\sigma_{\phi_1}, S_1, C_1)$ και $p_2[R] = (\sigma_{\phi_2}, S_2, C_2)$.

Βάσει ορισμού οι συνθήκες προτίμησης σ_{ϕ_1} και σ_{ϕ_2} εμπλέκουν αποκλειστικά γνωρίσματα εκτός του σκορ και του βαθμού εμπιστοσύνης. Επομένως ισχύει ότι η εγγραφή (t, S_t, C_t) ικανοποιεί τη συνθήκη σ_{ϕ_1} (αντιστοίχως σ_{ϕ_2}) αν και μόνο αν ισχύει $\sigma_{\phi_1}(t)$ (αντιστοίχως $\sigma_{\phi_2}(t)$). Επιπλέον, εφόσον ένας τελεστής προτίμησης δεν απορρίπτει καμία εγγραφή, κάθε εγγραφή (t, S_t, C_t) της R θα περιέχεται επίσης και στο αποτέλεσμα τόσο του $\lambda_{p_1}(\lambda_{p_2}(R))$ όσο και του $\lambda_{p_2}(\lambda_{p_1}(R))$. Έστω $(t, S_{12}, C_{12}) \in \lambda_{p_1}(\lambda_{p_2}(R))$ και $(t, S_{21}, C_{21}) \in \lambda_{p_2}(\lambda_{p_1}(R))$. Θα δείξουμε ότι: $\langle S_{12}, C_{12}\rangle = \langle S_{21}, C_{21}\rangle$. Έχουμε:

$$
\begin{aligned}
&(t,\, S_{t1}, C_{t1}) \in \lambda_{p_1}(R)\\
&\quad \Leftrightarrow ((t, S_t, C_t) \in R \wedge (\langle S_{t1}, C_{t1}\rangle = \langle S_t, C_t\rangle \wedge \neg\sigma_{\phi_1}(t))\\
&\qquad \vee \quad (\langle S_{t1}, C_{t1}\rangle = F(\langle S_t, C_t\rangle, \langle S_1(t), C_1\rangle) \wedge \sigma_{\phi_1}(t)))
\end{aligned}
$$

$$
\begin{aligned}
&(t,\, S_{t2}, C_{t2}) \in \lambda_{p_2}(R)\\
&\quad \Leftrightarrow ((t, S_t, C_t) \in R \wedge (\langle S_{t2}, C_{t2}\rangle = \langle S_t, C_t\rangle \wedge \neg\sigma_{\phi_2}(t))\\
&\qquad \vee \quad (\langle S_{t2}, C_{t2}\rangle = F(\langle S_t, C_t\rangle, \langle S_2(t), C_2\rangle) \wedge \sigma_{\phi_2}(t)))
\end{aligned}
$$

$$
\begin{aligned}
&(t,\, S_{12}, C_{12}) \in \lambda_{p_1}(\lambda_{p_2}(R))\\
&\quad \Leftrightarrow (t, S_{12}, C_{12}) = \lambda_{p_1}(t, S_{t2}, C_{t2})\\
&\quad \Leftrightarrow (t, S_t, C_t) \in R \wedge ((\langle S_{12}, C_{12}\rangle = \langle S_t, C_t\rangle \wedge \neg\sigma_{\phi_2}(t) \wedge \neg\sigma_{\phi_1}(t))\\
&\qquad \vee \quad (\langle S_{12}, C_{12}\rangle = F(\langle S_t, C_t\rangle, \langle S_1(t), C_1\rangle) \wedge \neg\sigma_{\phi_2}(t) \wedge \sigma_{\phi_1}(t))\\
&\qquad \vee \quad (\langle S_{12}, C_{12}\rangle = F(\langle S_t, C_t\rangle, \langle S_2(t), C_2\rangle) \wedge \sigma_{\phi_2}(t) \wedge \neg\sigma_{\phi_1}(t))\\
&\qquad \vee \quad (\langle S_{12}, C_{12}\rangle = F(F(\langle S_t, C_t\rangle, \langle S_2(t), C_2\rangle), \langle S_1(t), C_1\rangle) \wedge \sigma_{\phi_2}(t) \wedge \sigma_{\phi_1}(t)))
\end{aligned}
$$

$$
\begin{aligned}
&(t,\, S_{21}, C_{21}) \in \lambda_{p_2}(\lambda_{p_1}(R))\\
&\quad \Leftrightarrow (t, S_{21}, C_{21}) = \lambda_{p_2}(t, S_{t1}, C_{t1})\\
&\quad \Leftrightarrow (t, S_t, C_t) \in R \wedge ((\langle S_{21}, C_{21}\rangle = \langle S_t, C_t\rangle \wedge \neg\sigma_{\phi_1}(t) \wedge \neg\sigma_{\phi_2}(t))\\
&\qquad \vee \quad (\langle S_{21}, C_{21}\rangle = F(\langle S_t, C_t\rangle, \langle S_2(t), C_2\rangle) \wedge \neg\sigma_{\phi_1}(t) \wedge \sigma_{\phi_2}(t))\\
&\qquad \vee \quad (\langle S_{21}, C_{21}\rangle = F(\langle S_t, C_t\rangle, \langle S_1(t), C_1\rangle) \wedge \sigma_{\phi_1}(t) \wedge \neg\sigma_{\phi_2}(t))\\
&\qquad \vee \quad (\langle S_{21}, C_{21}\rangle = F(F(\langle S_t, C_t\rangle, \langle S_1(t), C_1\rangle), \langle S_2(t), C_2\rangle) \wedge \sigma_{\phi_1}(t) \wedge \sigma_{\phi_2}(t)))
\end{aligned}
$$

Γνωρίζουμε ότι η F είναι αντιμεταθετική και προσεταιριστική, επομένως:

$$F(F(\langle S_t, C_t\rangle, \langle S_2(t), C_2\rangle), \langle S_1(t), C_1\rangle) = F(\langle S_t, C_t\rangle, F(\langle S_2(t), C_2\rangle, \langle S_1(t), C_1\rangle)) =$$
$$F(\langle S_t, C_t\rangle, F(\langle S_1(t), C_1\rangle, \langle S_2(t), C_2\rangle)) = F(F(\langle S_t, C_t\rangle, \langle S_1(t), C_1\rangle), \langle S_2(t), C_2\rangle)$$

Από τα παραπάνω προκύπτει το ζητούμενο. □

Ιδιότητα 3. *Αν μια προτίμηση p εμπλέκει γνωρίσματα μόνο από μια σχέση R_i, τότε ισχύει $\lambda_p(R_i\ \theta\ R_j) = \lambda_p(R_i)\ \theta\ R_j$ όπου $\theta \in \{\cup, \cap, \bowtie_\phi\}$.*

Απόδειξη. Θα αποδείξουμε την παραπάνω ιδιότητα για την περίπτωση ενός τελεστή σύζευξης. Με παρόμοιο τρόπο είναι δυνατό να αποδειχθεί η ισχύς της ιδιότητας και για τους υπόλοιπους δυαδικούς τελεστές. Έστω μια προτίμηση $p[R_i] = (\sigma_\phi, S, C)$. Έχουμε:

$$
\begin{aligned}
&(t,\, S', C') \in \lambda_p(R_i \bowtie_F R_j)\\
&\quad \Leftrightarrow (t, S'', C'') \in R_i \bowtie_F R_j \wedge ((\langle S', C'\rangle = \langle S'', C''\rangle \wedge \neg\sigma_\phi(t))\\
&\qquad \vee \quad (\langle S', C'\rangle = F(\langle S'', C''\rangle, \langle S(t), C\rangle) \wedge \sigma_\phi(t)))\\
&\quad \Leftrightarrow t = t_i \bowtie t_j, (t_i, S_i, C_i) \in R_i, (t_j, S_j, C_j) \in R_j \wedge\\
&\qquad \langle S'', C''\rangle := F(\langle S_i, C_i\rangle, \langle S_j, C_j\rangle) \wedge ((\langle S', C'\rangle = \langle S'', C''\rangle \wedge \neg\sigma_\phi(t))\\
&\qquad \vee \quad (\langle S', C'\rangle = F(\langle S'', C''\rangle, \langle S(t), C\rangle) \wedge \sigma_\phi(t)))\\
&\quad \Leftrightarrow t = t_i \bowtie t_j, (t_i, S_i, C_i) \in R_i, (t_j, S_j, C_j) \in R_j \wedge ((\langle S', C'\rangle = F(\langle S_i, C_i\rangle, \langle S_j, C_j\rangle) \wedge \neg\sigma_\phi(t))\\
&\qquad \vee \quad (\langle S', C'\rangle = F(F(\langle S_i, C_i\rangle, \langle S_j, C_j\rangle), \langle S(t), C\rangle) \wedge \sigma_\phi(t)))
\end{aligned}
$$

Θα συμβολίσουμε με t_i το κομμάτι της t που προέρχεται από την R_i. Επειδή η συνθήκη σ_ϕ ισχύει για την t, συνεπάγεται ότι η συνθήκη σ_ϕ θα ισχύει επίσης και για την t_i. Παρομοίως, επειδή η συνάρτηση βαθμολόγησης S εμπλέκει μόνο γνωρίσματα που εμφανίζονται στην t_i, είναι ισοδύναμο να εφαρμόσουμε τη συνάρτηση βαθμολόγησης S μόνο στην t_i αντί της t. Επομένως έχουμε:

$$
\begin{aligned}
&(t,\, S', C') \in \lambda_p(R_i \ltimes_F R_j) \\
&\quad \Leftrightarrow t = t_i \ltimes t_j, (t_i, S_i, C_i) \in R_i, (t_j, S_j, C_j) \in R_j \wedge ((\langle S', C'\rangle = F(\langle S_i, C_i\rangle, \langle S_j, C_j\rangle) \wedge \neg\sigma_\phi(t_i)) \\
&\qquad \vee \quad (\langle S', C'\rangle = F(F(\langle S_i, C_i\rangle, \langle S_j, C_j\rangle), \langle S(t_i), C\rangle) \wedge \sigma_\phi(t_i)))
\end{aligned}
$$

Επίσης ισχύει:

$$
\begin{aligned}
&(t,\, S', C') \in \lambda_p(R_i) \ltimes_F R_j \\
&\quad \Leftrightarrow t = t_i \ltimes t_j, (t_i, S_i', C_i') \in \lambda_p(R_i), (t_i, S_i, C_i) \in R_i \wedge (t_j, S_j, C_j) \in R_j \wedge \\
&\qquad\quad (\langle S', C'\rangle = F(\langle S_i', C_i'\rangle, \langle S_j, C_j\rangle) \wedge ((\langle S_i', C_i'\rangle = \langle S_i, C_i\rangle \wedge \neg\sigma_\phi(t_i)) \\
&\qquad \vee \quad (\langle S_i', C_i'\rangle = F(\langle S_i, C_i\rangle, \langle S(t_i), C\rangle) \wedge \sigma_\phi(t_i))) \\
&\quad \Leftrightarrow t = t_i \ltimes t_j, (t_i, S_i, C_i) \in R_i \wedge (t_j, S_j, C_j) \in R_j \wedge ((\langle S', C'\rangle = F(\langle S_i, C_i\rangle, \langle S_j, C_j\rangle) \wedge \neg\sigma_\phi(t_i)) \\
&\qquad \vee \quad (\langle S', C'\rangle = F(F(\langle S_i, C_i\rangle, \langle S(t_i), C\rangle), \langle S_j, C_j\rangle) \wedge \sigma_\phi(t_i)))
\end{aligned}
$$

Εφόσον η F είναι αντιμεταθετική και προσεταιριστική, προκύπτει ότι ισχύει η ιδιότητα 3. □

Ιδιότητα 4. *Έστω η προτίμηση $p[R] = (\sigma_\phi, S, C)$ και ένας τελεστής επιλογής $\sigma_{\phi'}$ που εμπλέκει αποκλειστικά γνωρίσματα εκτός του σκορ και του βαθμού εμπιστοσύνης. Τότε ίσχει $\sigma_{\phi'}\lambda_p(R) = \sigma_{\phi'}\lambda_{p'}(R)$ όπου $p'[R] = (\sigma_{\phi\wedge\phi'}, S, C)$.*

Απόδειξη. Έστω (t, S_t, C_t) μια τυχαία εγγραφή της R που ικανοποιεί τη συνθήκη προτίμησης σ_ϕ. Αφού η συνθήκη επιλογής $\sigma_{\phi'}$ εμπλέκει αποκλειστικά γνωρίσματα εκτός του σκορ και του βαθμού εμπιστοσύνης, τότε ισχύει επίσης $\sigma_{\phi'}(t)$. Επιπλέον, αφού ο τελεστής προτίμησης δεν απορρίπτει εγγραφές, τότε κάθε εγγραφή $(t, S_t, C_t) \in \sigma_{\phi'}(R)$ περιέχεται επίσης στο αποτέλεσμα τόσο του $\sigma_{\phi'}\lambda_p(R)$ όσο και του $\sigma_{\phi'}\lambda_{p'}(R)$. Συνεπώς έχουμε:

$$
\begin{aligned}
&(t,\, S', C') \in \sigma_{\phi'}\lambda_{p'}(R) \\
&\quad \Leftrightarrow (t, S_t, C_t) \in R \wedge \sigma_{\phi'}(t) \wedge ((\langle S', C'\rangle = \langle S_t, C_t\rangle \wedge \neg\sigma_{\phi\wedge\phi'}(t)) \\
&\qquad \vee \quad (\langle S', C'\rangle = F(\langle S_t, C_t\rangle, \langle S(t), C\rangle) \wedge \sigma_{\phi\wedge\phi'}(t))) \\
&\quad \Leftrightarrow (t, S_t, C_t) \in R \wedge \sigma_{\phi'}(t) \wedge ((\langle S', C'\rangle = \langle S_t, C_t\rangle \wedge (\neg\sigma_\phi(t) \vee\ \neg\sigma_{\phi'}(t))) \\
&\qquad \vee \quad (\langle S', C'\rangle = F(\langle S_t, C_t\rangle, \langle S(t), C\rangle) \wedge \sigma_\phi(t) \wedge \sigma_{\phi'}(t))) \\
&\quad \Leftrightarrow ((t, S_t, C_t) \in R \wedge \sigma_{\phi'}(t) \wedge \langle S', C'\rangle = \langle S_t, C_t\rangle \wedge \neg\sigma_\phi(t)) \\
&\qquad \vee \quad ((t, S_t, C_t) \in R \wedge \sigma_{\phi'}(t) \wedge \langle S', C'\rangle = \langle S_t, C_t\rangle \wedge \neg\sigma_{\phi'}(t)) \\
&\qquad \vee \quad ((t, S_t, C_t) \in R \wedge \sigma_{\phi'}(t) \wedge \langle S', C'\rangle = F(\langle S_t, C_t\rangle, \langle S(t), C\rangle) \wedge \sigma_\phi(t) \wedge \sigma_{\phi'}(t)) \\
&\quad \Leftrightarrow ((t, S_t, C_t) \in R \wedge \sigma_{\phi'}(t) \wedge \neg\sigma_\phi(t)) \wedge \langle S', C'\rangle = \langle S_t, C_t\rangle) \\
&\qquad \vee \quad ((t, S_t, C_t) \in R \wedge \sigma_{\phi'}(t) \wedge \sigma_\phi(t) \wedge \langle S', C'\rangle = F(\langle S_t, C_t\rangle, \langle S(t), C\rangle))
\end{aligned}
$$

Επιπλέον έχουμε:

$$
\begin{aligned}
&(t,\, S', C') \in \sigma_{\phi'}\lambda_p(R) \\
&\quad \Leftrightarrow (t, S_t, C_t) \in R \wedge \sigma_{\phi'}(t) \wedge ((\langle S', C'\rangle = \langle S_t, C_t\rangle \wedge \neg\sigma_\phi(t)) \\
&\qquad \vee \quad (\langle S', C'\rangle = F(\langle S_t, C_t\rangle, \langle S(t), C\rangle) \wedge \sigma_\phi(t))) \\
&\quad \Leftrightarrow ((t, S_t, C_t) \in R \wedge \sigma_{\phi'}(t) \wedge \neg\sigma_\phi(t) \wedge \langle S', C'\rangle = \langle S_t, C_t\rangle) \\
&\qquad \vee \quad ((t, S_t, C_t) \in R \wedge \sigma_{\phi'}(t) \wedge \sigma_\phi(t) \wedge \langle S', C'\rangle = F(\langle S_t, C_t\rangle, \langle S(t), C\rangle))
\end{aligned}
$$

Από τα παραπάνω προκύπτει ότι ισχύει το ζητούμενο. □

m_id	Προτιμήσεις
m_{11}	p_4, p_5, p_6, p_7
m_{12}	$-$, p_5, p_6, p_7
m_{13}	p_4, $-$, $-$, p_7
m_{14}	p_4, p_5, $-$, $-$
m_{15}	p_4, p_5, $-$, p_7

(α′) Προτιμήσεις που ικανοποιούνται

m_id	score	sum(conf)	max(conf)	#prefs
m_{11}	1.5	3	0.9	4
m_{12}	1.8	2.2	0.9	3
m_{13}	1	1.5	0.8	2
m_{14}	1.7	1.7	0.9	2
m_{15}	1.9	2.4	0.9	3

(β′) Τελικές τιμές

Πίνακας 3.3: *Παραδείγματα ερωτημάτων με προτιμήσεις*

3.2.4 Ερωτήματα Προτίμησης

Οι επεκτεταμένοι σχεσιακοί τελεστές και ο τελεστής προτίμησης αποτελούν τον πυρήνα της επεκτεταμένης σχεσιακής άλγεβρας που προτείνουμε. Σε αυτή την ενότητα παρουσιάζουμε το μοντέλο ερωτημάτων που ακολουθούμε και δίνουμε κάποια παραδείγματα με σκοπό να αναδείξουμε την εκφραστικότητα και την ευελιξία του προτεινόμενου μοντέλου.

Ένα ερώτημα προτίμησης συνδυάζει επεκτεταμένες σχέσεις και αλγεβρικούς τελεστές και επιστρέφει ένα σύνολο εγγραφών οι οποίες ικανοποιούν τις συνθήκες επιλογής του ερωτήματος και έχουν σκορ και βαθμούς εμπιστοσύνης που έχουν προκύψει αφού έχουν εκτελεστεί όλοι τελεστές προτίμησης στις αντίστοιχες σχέσεις. Διαισθητικά, όσο περισσότερο ταιριάζει μια εγγραφή στις προτιμήσεις και όσο περισσότερες ή πιο βέβαιες προτιμήσεις ικανοποιεί, τόσο υψηλότερο τελικό σκορ και βαθμό εμπιστοσύνης θα έχει αντιστοίχως.

Ας θεωρήσουμε μια υπηρεσία θέασης βίντεο μέσω διαδικτύου η οποία χρησιμοποιεί τη βάση δεδομένων του Πίνακα 3.1. Η διεπαφή της εφαρμογής προσφέρει τη δυνατότητα αναζήτησης των διαθέσιμων ταινιών, την επισκόπηση κριτικών και συστάσεων κλπ. Επιπλέον, το σύστημα συλλέγει τις προτιμήσεις των συνδρομητών παρακολουθώντας το ιστορικό των αναζητήσεών τους, τις προηγούμενες ταινίες που επέλεξαν να δουν καθώς και τις βαθμολογίες που έδωσαν. Σκόπος της εφαρμογής είναι να προτείνει στους χρήστες ταινίες που είναι αρκετά πιθανό να ταιριάζουν με τις προτιμήσεις τους.

Παράδειγμα 3.8. *(Επιλέγοντας τα αποτελέσματα με τις υψηλότερες βαθμολογίες). Ένας χρήστης ονόματι Alice πλοηγείται στο σύστημα αναζητώντας μια πρόσφατη ενδιαφέρουσα ταινία. Το σύστημα επισημαίνει τους τίτλους κάποιων ταινιών που είναι πιθανό να αρέσουν στην Alice βασισμένο στις προτιμήσεις που έχει συλλέξει γι'αυτήν. Έστω ότι οι προτιμήσεις για τον συγκεκριμένο χρήστη είναι οι p_4 - p_7 από τον Πίνακα 3.2. Γι'αυτό το λόγο, η εφαρμογή κατασκευάζει και εκτελεί το ακόλουθο ερώτημα το οποίο θα επιστρέφει πρόσφατες ταινίες που ταιριάζουν με τις προτιμήσεις της Alice.*

$$Q_1 : \pi_{title,director}\lambda_{p_7}\{\lambda_{p_6}\sigma_{year=2012}(MOVIES) \bowtie \lambda_{p_5}(RATINGS) \bowtie DIRECTORS\}$$

Έστω ότι τα αποτελέσματα του ερωτήματος μαζί με τα σκορ και τους αντίστοιχους βαθμούς εμπιστοσύνης απεικονίζονται στον Πίνακα 3.3(β′). Τα συγκεκριμένα αποτελέσματα ικανοποιούν τις προτιμήσεις της Alice όπως δείχνει ο Πίνακας 3.3(α′). Με στόχο να δείξουμε καλύτερα διαφορετικούς τύπους ερωτημάτων, οι στήλες 3 έως 5 του

Πίνακα 3.3(β′) αναπαριστούν τις τελικές τιμές για τον βαθμό εμπιστοσύνης που προκύπτει εφαρμόζοντας διαφορετικές συναθροιστικές συναρτήσεις όπως το άθροισμα, το μέγιστο και μια συνάρτηση που μετράει το πλήθος των ικανοποιούμενων προτιμήσεων αντιστοίχως.

Η επιλογή των τελικών αποτελεσμάτων ενός ερωτήματος προτίμησης μπορεί να γίνει με διάφορους τρόπους. Μπορούμε να φιλτράρουμε ή να κατατάξουμε τα αποτελέσματα με βάση το σκορ, το βαθμό εμπιστοσύνης ή και τα δύο συγχρόνως. Για παράδειγμα, αν θέλουμε να επιλέξουμε τα $k = 3$ αποτελέσματα με τις υψηλότερες βαθμολογίες, θα παίρναμε τα αποτελέσματα $\{m_{15}, m_{12}, m_{14}\}$. Εναλλακτικά, μπορούμε να καθορίσουμε ένα ελάχιστο σκορ-κατώφλι τ_s. Για παράδειγμα αν $\tau_s = 1.8$, τότε το αποτέλεσμα θα περιέχει τις εγγραφές $\{m_{12}, m_{15}\}$. □

Παράδειγμα 3.9. *(Επιλέγοντας τα πιο 'σίγουρα' αποτελέσματα). Συνεχίζοντας το προηγούμενο παράδειγμα, η εφαρμογή θα μπορούσαμε να επιλέξει να επιστρέψει τις πιο 'σίγουρες ' προτάσεις, δηλαδή ταινίες οι οποίες ικανοποιούν αρκετές από τις προτιμήσεις ενός χρήστη. Για τις ανάγκες αυτού του παραδείγματος θα χρησιμοποιούμε την τελευταία στήλη του Πίνακα 3.3(β′) η οποία περιέχει τον τελικό βαθμό εμπιστοσύνης των αποτελεσμάτων ακολουθώντας την παραπάνω λογική. Σε αυτή την περίπτωση οι εγγραφές οι οποίες δεν ικανοποιούν αρκετές προτιμήσεις πρέπει να απορριφθούν από τα αποτελέσματα. Για παράδειγμα, αν θέσουμε ένα ελάχιστο κατώφλι για τον βαθμό εμπιστοσύνης $\tau_c = 3$, τότε το τελικό αποτέλεσμα θα περιέχει τις εγγραφές $\{m_{11}, m_{12}, m_{15}\}$. Εναλλακτικά θα μπορούσαμε να λάβουμε υπόψη μόνο την ισχυρότερη προτίμηση, χρησιμοποιώντας την F_{max} ως συναθροιστική συνάρτηση. Σε αυτή την περίπτωση το αποτέλεσμα θα περιέχει τις εγγραφές $\{m_{11}, m_{12}, m_{14}, m_{15}\}$ οι οποίες ικανοποιούν την πιο ισχυρή προτίμηση p_5.* □

Παράδειγμα 3.10. *Το σύστημα παρέχει μια διεπαφή μέσω της οποίας η Alice και ένας φίλος της ονόματι Bob μπορούν από κοινού να δηλώσουν τις προτιμήσεις τους και το σύστημα θα προσπαθήσει να βρει ταινίες που οι δυο τους θα μπορούσαν να δουν παρέα. Για παράδειγμα, έστω ότι η Alice έχει όρεξη να δει μια ρομαντική κομεντί (p_8) και ο Bob είναι θαυμαστής του ηθοποιού Steve Martin (p_{10}). Τότε το παρακάτω ερώτημα προτίμησης θα επιλέξει τις ταινίες οι οποίες ταιριάζουν στις προτιμήσεις και των δύο συνδρομητών:*

$Q_2 : \{\pi_{title}\{\lambda_{p_8}\sigma_{year=2012}(MOVIES) \bowtie GENRES\}\} \cap \{\pi_{title}\{MOVIES \bowtie CAST \bowtie \lambda_{p_{10}}(ACTORS)\}\}$ □

Παράδειγμα 3.11. *(Εμπλουτίζοντας προτιμήσεις με συστάσεις). Σε μια προσπάθεια να προτείνει μια μεγαλύτερη ποικιλία πιο διαφοροποιημένων προτάσεων στους χρήστες της, η εφαρμογή θα μπορούσε να επιλέξει να εμπλουτίσει τις προτιμήσεις ενός χρήστη με προτιμήσεις που έχει συλλέξει για άλλους χρήστες του κοινωνικού του περίγυρου. Για παράδειγμα, το σύστημα θα μπορούσε να αναμίξει τις ταινίες που ταιριάζουν με τις προτιμήσεις της Alice (p_4, p_6, p_7) με ταινίες οι οποίες είναι πιθανόν να αρέσουν στο φίλο της Bob. Έστω ότι για τον Bob το σύστημα έχει συλλέξει προτιμήσεις που εμπλέκουν τη σχέση $MOVIES$ (p_{11}) και τη σχέση $ACTORS$ (p_{12}). Το παρακάτω ερώτημα επιλέγει τα πιο 'σίγουρα' αποτελέσματα που σχετίζονται με τις προτιμήσεις της Alice*

και τα αποτελέσματα που έχουν τις υψηλότερες βαθμολογίες με βάση τις προτιμήσεις του Bob, συνδυάζει τα δύο σύνολα και παράγει μια τελική λίστα με προτεινόμενες ταινίες για την Alice.

$Q_3 : \{\pi_{title}\sigma_{conf>\tau_c}\lambda_{p_7}\{\lambda_{p_6}(MOVIES) \ltimes \lambda_{p_4}(GENRES) \ltimes DIRECTORS\}\} \cup$
$\{\pi_{title}\sigma_{score>\tau_s}\lambda_{p_{11}}(MOVIES) \ltimes CAST \ltimes \lambda_{p_{12}}(ACTORS)\}$ $\square$

3.3 Το σύστημα PrefDB

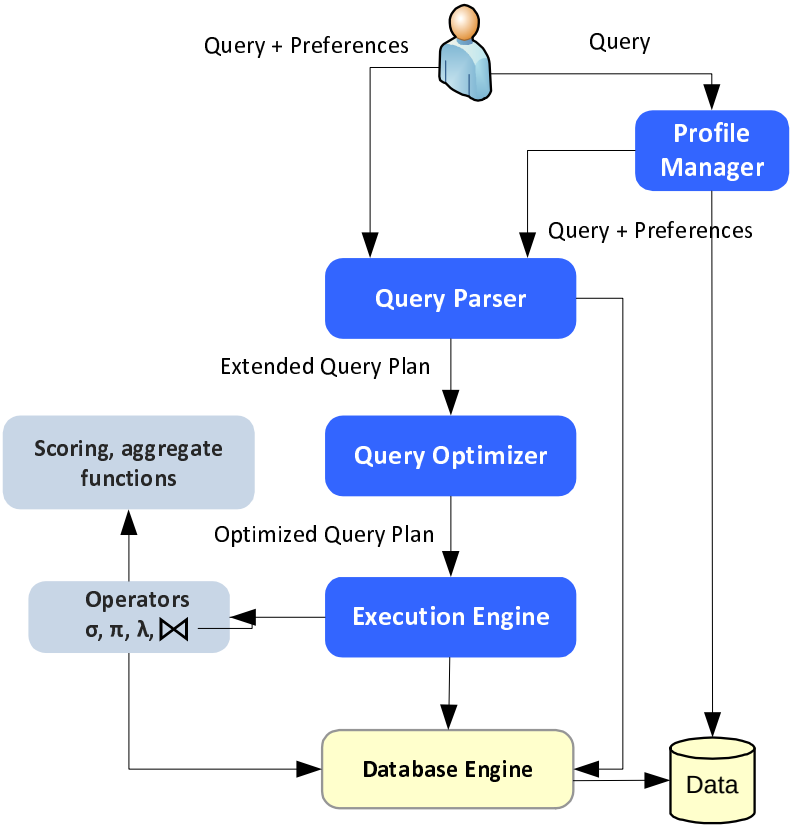
Το PrefDB είναι ένα πρωτότυπο σύστημα που έχουμε υλοποιήσει και βασίζεται στο επεκτεταμένο σχεσιακό μοντέλο και μοντέλο ερωτημάτων που παρουσιάσαμε νωρίτερα. Η ενότητα 3.3.1 παρέχει μια επισκόπηση της αρχιτεκτονικής του συστήματος και της λειτουργικότητας που αυτό προσφέρει. Στην ενότητα 3.3.2 παρουσιάζουμε ένα γραφικό περιβάλλον διαχείρισης του συστήματος. Τέλος, η ενότητα 3.3.3 περιγράφει λεπτομέρειες της υλοποίησης των επεκτεταμένων σχέσεων και των τελεστών.

3.3.1 Επισκόπηση του συστήματος

Το Σχήμα 3.3 απεικονίζει την αρχιτεκτονική του συστήματος PrefDB σε υψηλό επίπεδο. Τα υποσυστήματα που φαίνονται με κίτρινο χρώμα παρέχονται από τη βάση δεδομένων, ενώ με μπλε χρώμα διακρίνονται τα υποσυστήματα που έχουν αναπτυχθεί για το σύστημα PrefDB. Όπως δείχνει το σχήμα, το PrefDB παρέχει δύο εναλλακτικές επιλογές για την εισαγωγή ερωτημάτων με προτιμήσεις. Πιο συγκεκριμένα, οι προτιμήσεις ενός χρήστη μπορούν να δηλωθούν μαζί με το ερώτημα που τίθεται ή το σύστημα μπορεί αυτόματα να εμπλουτίσει ένα συμβατικό ερώτημα με σχετικές προτιμήσεις. Ακολουθώντας την πρώτη επιλογή, οι προτιμήσεις δίνονται με δηλωτικό τρόπο επιπλέον του υπόλοιπου ερωτήματος. Στη δεύτερη περίπτωση οι σχετικές προτιμήσεις παρέχονται από το υποσύστημα Διαχείρισης Προτιμήσεων (*profile manager module*), το οποίο προσπελαύνει προτιμήσεις που έχουν ήδη συλλεχθεί από τη βάση δεδομένων.

Οι αποθηκευμένες προτιμήσεις είναι δυνατό να έχουν συλλεχθεί από προηγούμενα ερωτήματα προτίμησης που έχει θέσει ο χρήστης, ή αναλύοντας τις βαθμολογήσεις που έχει δώσει ο χρήστης στο παρελθόν, ή τη συμπεριφορά του στο σύστημα. Επισημαίνουμε ότι ο τρόπος συλλογής και εκμάθησης των προτιμήσεων ενός χρήστη είναι ορθόγωνος προς την επεξεργασία ενός ερωτήματος προτίμησης, η οποία είναι η πρωταρχική λειτουργικότητα που παρέχει το σύστημα PrefDB. Συνεπώς, για τις ανάγκες του πρωτότυπου συστήματος που υλοποιήσαμε, αποθηκεύουμε (α) προτιμήσεις που έχουν δηλωθεί από τους ίδιους τους χρήστες μέσω ενός γραφικού περιβάλλοντος διεπαφής που έχουμε αναπτύξει (βλέπε ενότητα 3.3.2), και (β) προτιμήσεις που έχουν δηλωθεί σε προηγούμενα ερωτήματα των χρηστών.

Ανεξάρτητα από τον τρόπο που οι προτιμήσεις δίνονται ως είσοδος στο σύστημα, το ερώτημα μαζί με τις προτιμήσεις προωθείται στο υποσύστημα Ανάλυσης Ερωτημάτων (query parser). Επιπλέον, εκτός από τις μεθόδους επεξεργασίας ερωτημάτων που βασίζονται στα προτεινόμενα μοντέλα προτιμήσεων και την εκτεταμένη σχεσιακή άλγεβρα, το σύστημα PrefDB υποστηρίζει επιπλέον μια σειρά μεθόδων plug-in όπως



Σχήμα 3.3: *Αρχιτεκτονική συστήματος PrefDB*

περιγράφηκαν στο κεφάλαιο 1.2 και παρουσιάζονται αναλυτικά στην εργασία [8]. Παρακάτω δίνεται μια επισκόπηση των βασικών υποσυστημάτων του συστήματος PrefDB, τον τρόπο λειτουργίας των οποίων περιγράφουμε αναλυτικά στην ενότητα 3.4.

- Το *υποσύστημα διαχείρισης προτιμήσεων* (profile manager) επιλέγει από τη βάση τις προτιμήσεις που μπορούν να συνδυαστούν με τις συνθήκες του ερωτήματος. Για το λόγο αυτό, χρησιμοποιούμε τον αλγόριθμο επιλογής προτιμήσεων που προτείνεται στην εργασία [40]

- Το *υποσύστημα ανάλυσης ερωτημάτων* (query parser) (ενότητα 3.4.1) λαμβάνει ως είσοδο ένα ερώτημα και ένα σύνολο προτιμήσεων και κατασκευάζει ένα επεκτεταμένο πλάνο εκτέλεσης το οποίο προωθεί στο *υποσύστημα βελτιστοποίησης ερωτημάτων*.

- Το *υποσύστημα βελτιστοποίησης ερωτημάτων* (query optimizer) (ενότητα 3.4.2) βελτιώνει το πλάνο εκτέλεσης που λαμβάνει ως είσοδο εφαρμόζοντας ένα σύνολο ευριστικών κανόνων που βασίζονται στις ιδιότητες του τελεστή προτίμησης που παρουσιάσαμε στην ενότητα 3.2.3. Το βελτιωμένο πλάνο χρησιμοποιείται ως βάση για το επόμενο στάδιο βελτιστοποίησης που ακολουθεί ένα μοντέλο κόστους που έχουμε αναπτύξει. Στόχος της βελτιστοποίησης είναι να επιλέξει ανάμεσα σε διάφορα εναλλακτικά πλάνα, εκείνο με το ελάχιστο κόστος εκτέλεσης.

- Το *υποσύστημα εκτέλεσης ερωτημάτων* (execution engine) (ενότητα 3.4.3) λαμβάνει ως είσοδο το επιλεγμένο πλάνο εκτέλεσης και το εκτελεί ακολουθώντας μία από τις προτεινόμενες μεθόδους εκτέλεσης.

3.3.2 Αλληλεπιδρώντας με το σύστημα PrefDB

Μια τυπική βάση δεδομένων είναι συνήθως εξοπλισμένη με ένα σύνολο εργαλείων τα οποία διευκολύνουν τους διαχειριστές συστημάτων (database administrators) και τους σχεδιαστές εφαρμογών (application designers) σε ένα πλήθος εργασιών, περιλαμβανομένης της υλοποίησης, εκτέλεσης, ελέγχου και εκσφαλμάτωσης ερωτημάτων, της βελτιστοποίησης των παραμέτρων του συστήματος και διαφόρων άλλων διαχειριστικών εργασιών. Δυστυχώς, τέτοια εργαλεία δεν παρέχουν υποστήριξη για ενσωμάτωση προτιμήσεων στη βάση δεδομένων και στα ερωτήματα των εφαρμογών. Γι'αυτό το σκοπό και με στόχο να διευκολύνουμε την αλληλεπίδραση των χρηστών με το σύστημα PrefDB, έχουμε υλοποιήσει ένα πρότυπο γραφικό εργαλείο διαχείρισης, το PrefDBAdmin. Επιγραμματικά, το PrefDBAdmin επιτρέπει στους χρήστες του την εκτέλεση των παρακάτω εργασιών:

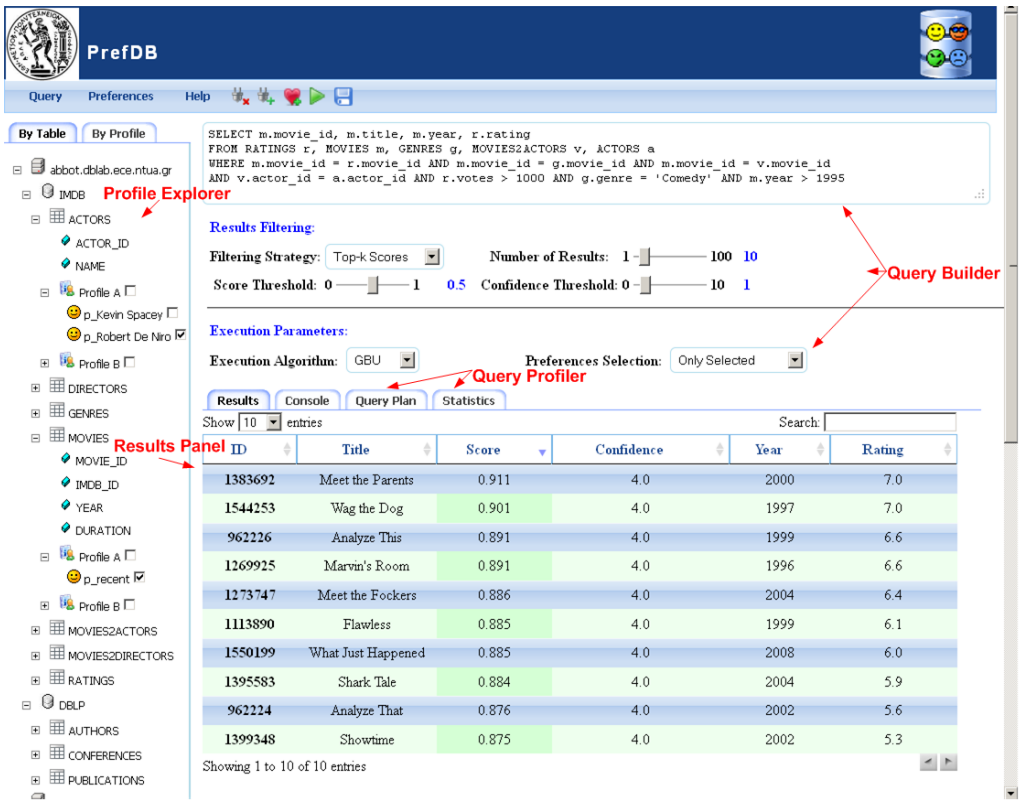
- Διαχείριση προτιμήσεων χρηστών, ομαδοποίησή τους σε σύνθετα προφίλ και επιλογή προτιμήσεων ή προφίλ καθώς και του τρόπου που θα ενσωματωθούν σε ένα ερώτημα προς τη βάση δεδομένων.

- Κατασκευή και εκτέλεση ερωτημάτων με προτιμήσεις, επιλογή ανάμεσα σε ένα σύνολο διαθέσιμων τεχνικών εκτέλεσης και ρύθμιση πλήθους παραμέτρων του ερωτήματος όπως για παράδειγμα ο αριθμός αποτελεσμάτων, ο αλγόριθμος επιλογής των αποτελεσμάτων (π.χ. top-k ή skyline queries), κατώτατα όρια τιμών για το σκορ και τον βαθμό εμπιστοσύνης κ.α.

- Παρακολούθηση του τρόπου εκτέλεσης ενός ερωτήματος μέσω μιας ενσωματωμένης κονσόλας, επιθεώρηση του επεκτεταμένου πλάνου εκτέλεσης που ακολουθείται, παραγωγή στατιστικών που αφορούν την εκτέλεση του ερωτήματος και δεδομένων profiling για τα εκτελούμενα ερωτήματα.

Παρακάτω περιγράφουμε εν συντομία τα βασικά υποσυστήματα του PrefDBAdmin, τα οποία μπορεί να διακρίνει κάποιος και στο Σχήμα 3.4.

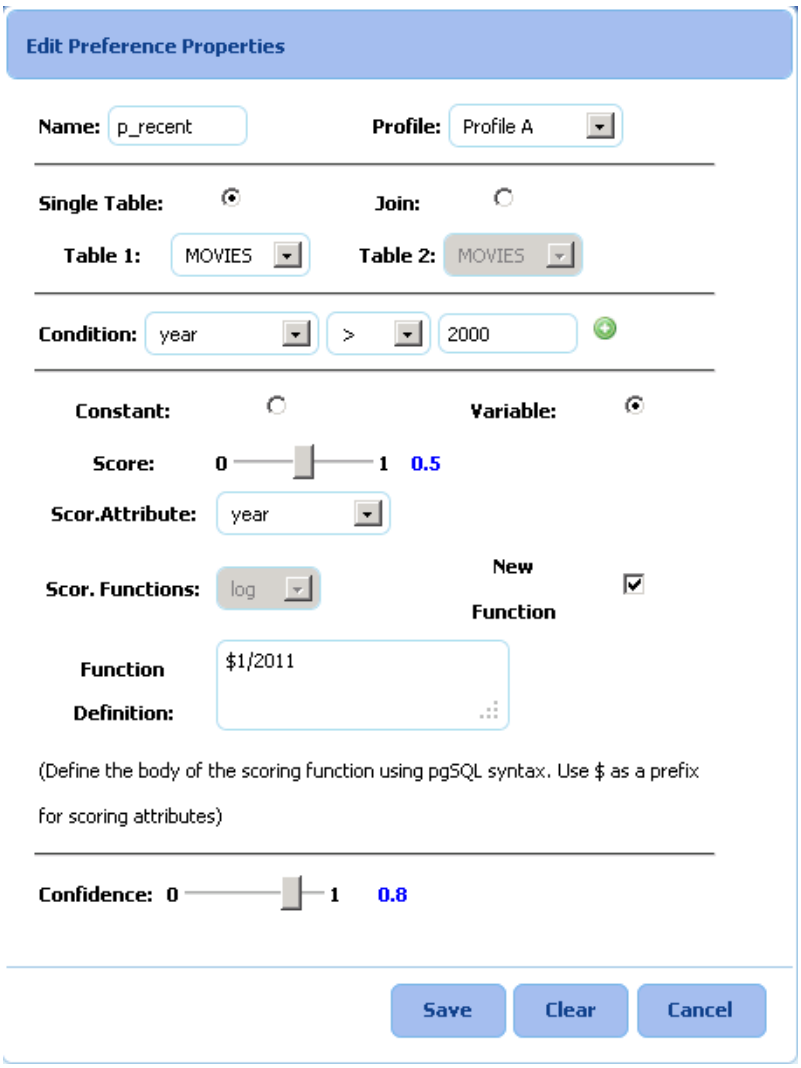
Περιηγητής Προφίλ. Ο περιηγητής προφίλ (profile explorer) (αριστερό μέρος του Σχήματος 3.4) αποτελεί μια δενδρική αναπαράσταση των βάσεων δεδομένων υπό διαχείριση. Για κάθε βάση δεδομένων, ο χρήστης του PrefDBAdmin μπορεί να δει τους πίνακες της, και για κάθε πίνακα, τα γνωρίσματα και τις σχετιζόμενες προτιμήσεις. Για παράδειγμα, στο Σχήμα 3.4 ο πίνακας *ACTORS* έχει δύο προτιμήσεις συνδεδεμένες με αυτόν. Οι προτιμήσεις οργανώνονται σε σύνολα που ονομάζουμε προφίλ. Οι χρήστες μπορούν να επιλέξουν να περιηγηθούν στις προτιμήσεις είτε ανά πίνακα είτε με βάση το προφίλ στο οποίο ανήκουν. Διακρίνουμε δύο κατηγορίες προφίλ: user profiles τα οποία παράγονται αυτόματα ενσωματώνοντας όλες τις προτιμήσεις που είναι διαθέσιμες ανά χρήστη, και test profiles τα οποία κατασκευάζονται χειροκίνητα και χρησιμοποιούνται αρχικά για πειραματισμό ή ενδεχομένως μαζί με ερωτήματα στη βάση. Τα user profiles μπορούν να χρησιμοποιηθούν μόνο χωρίς καμία τροποποίηση στα ερωτήματα και ανωνυμοποιούνται ώστε να αποφευχθεί η συσχέτιση προτιμήσεων με συγκεκριμένους χρήστες της εφαρμογής. Το εργαλείο PrefDBAdmin επιτρέπει στους χρήστες τη δημιουργία νέων προτιμήσεων ή την τροποποίηση υπαρχόντων μέσα σε test profiles με τη βοήθεια ενός συντάκτη προτιμήσεων (preference editor).



Σχήμα 3.4: *PrefDBAdmin*

Συντάκτης Προτιμήσεων. Ο συντάκτης προτιμήσεων (preference editor) (Σχήμα 3.5) επιτρέπει στους χρήστες τον ορισμό και την τροποποίηση προτιμήσεων. Οι χρήστες μπορούν να τροποποιήσουν κατάλληλα ένα σύνολο ιδιοτήτων μιας προτίμησης όπως η συνθήκη προτίμησης, η συνάρτηση βαθμολόγησης και ο βαθμός εμπιστοσύνης. Οι χρήστες μπορούν να ορίσουν ένα σταθερό σκορ για όλες τις εγγραφές που επηρεάζονται από την προτίμηση, είτε να επιλέξουν από ένα σύνολο προκαθορισμένων συναρτήσεων βαθμολογίας, είτε να τροποποιήσουν τον ορισμό της συνάρτησης βαθμολογίας που χρησιμοποιείται.

Κατασκευαστής Ερωτημάτων. Οι χρήστες μπορούν να κατασκευάσουν ερωτήματα με προτιμήσεις που εκτελούνται από το σύστημα PrefDB. Ξεκινώντας από τον περιηγητή προφίλ, οι χρήστες μπορούν να επιλέξουν ποιες προτιμήσεις ή προφίλ θα χρησιμοποιηθούν μαζί με ένα ερώτημα. Το κομμάτι του ερωτήματος που δεν σχετίζεται με προτιμήσεις μπορεί να οριστεί κατά τα γνωστά με βάση τη γλώσσα SQL. Οι χρήστες έχουν τη δυνατότητα να ρυθμίσουν ένα πλήθος παραμέτρων που σχετίζονται με το ερώτημα όπως τον επιθυμητό αριθμό αποτελεσμάτων, το ελάχιστο σκορ ή βαθμό εμπιστοσύνης, τον τρόπο επιλογής των τελικών αποτελεσμάτων (π.χ. top-k ή skyline queries) (επάνω μέρος του Σχήματος 3.4). Επιπλέον οι χρήστες μπορούν να επιλέξουν τον αλγόριθμο εκτέλεσης του ερωτήματος ανάμεσα σε μια σειρά αλγορίθμων που παρουσιάζουμε στην ενότητα 3.4.3 και στην εργασία [8]. Με αυτό τον τρόπο, οι χρήστες μπορούν να δοκιμάσουν την επίδοση διαφορετικών μεθόδων εκτέλεσης ενός ερωτήματος και να επιλέξουν αυτόν με το μικρότερο κόστος εκτέλεσης. Τέλος οι χρήστες

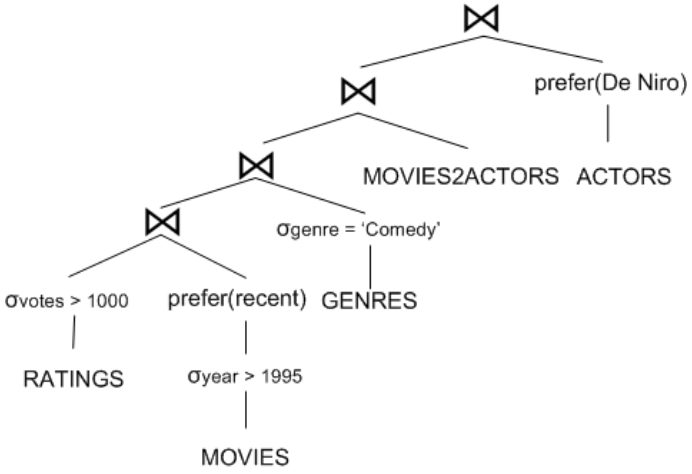


Σχήμα 3.5: *Preference Editor*

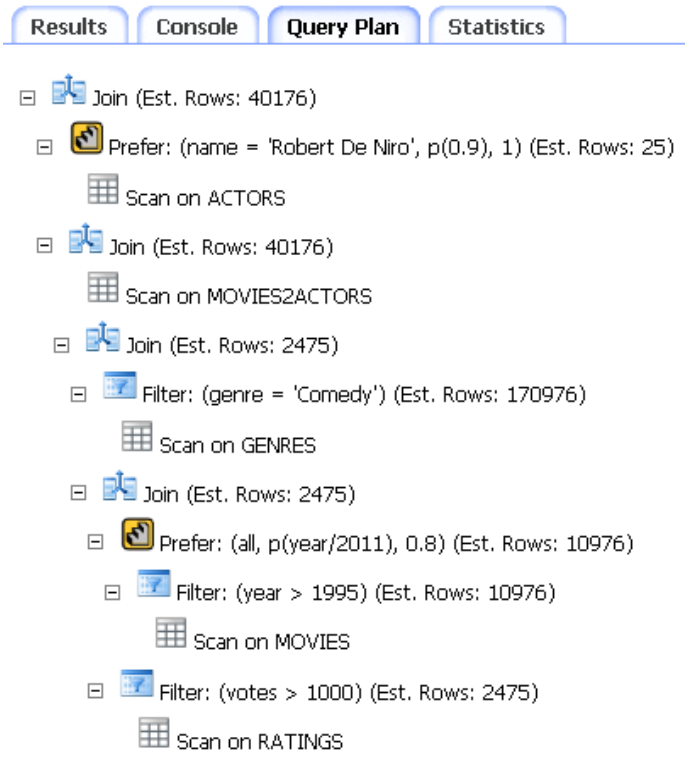
μπορούν να εκτελέσουν το ερώτημα ή να το αποθηκεύσουν μαζί με τις ρυθμίσεις που περιγράψαμε παραπάνω. Χρησιμοποιώντας το μενού, ένας χρήστης μπορεί αργότερα να φορτώσει ένα αποθηκευμένο ερώτημα.

Πίνακας Αποτελεσμάτων. Μετά την ολοκλήρωση της εκτέλεσης ενός ερωτήματος, οι χρήστες μπορούν να περιηγηθούν σε διαφορετικά υποσύνολα των αποτελεσμάτων μέσω του πίνακα αποτελεσμάτων (results panel). Για παράδειγμα, μπορούν να ταξινομήσουν τα αποτελέσματα με διάφορους τρόπους, να εκτελέσουν αναζητήσεις στα αποτελέσματα ή να αλλάξουν το πλήθος των εμφανιζόμενων αποτελεσμάτων ή τα κριτήρια εμφάνισης (π.χ. ελάχιστο σκορ).

Επιθεωρητής Εκτέλεσης Ερωτημάτων. Το σύστημα επιτρέπει στους χρήστες να περιηγηθούν στο επεκτεταμένο πλάνο εκτέλεσης και να εξετάσουν την εκτίμηση κόστους που κάνει το σύστημα για επιμέρους τελεστές μέσω του επιθεωρητή εκτέλεσης ερωτημάτων (query profiler) (Σχήμα 3.6(β′)). Οι χρήστες μπορούν να παρακολουθήσουν την εκτέλεση ενός ερωτήματος μέσω μιας ενσωματωμένης κονσόλας. Μετά την ολοκλήρωση της εκτέλεσης ενός ερωτήματος, οι χρήστες μπορούν να δουν χρήσιμα στατιστικά που σχετίζονται με το ερώτημα ή επιμέρους βήματα εκτέλεσης καθώς και πληροφορίες profiling όπως ο χρόνος που δαπανήθηκε από τη διεργασία στον επεξερ-



(α′) Extended Query Plan



(β′) Query Profiler

Σχήμα 3.6: *Επεκτεταμένο πλάνο εκτέλεσης*

γαστή για την εκτέλεση επιμέρους τελεστών ή τα μεγέθη των ενδιάμεσων πινάκων που παρήχθησαν, καθώς και να τα συγκρίνει με τις προβλεπόμενες τιμές. Τέλος, οι χρήστες μπορούν να δουν ιστορικά στατιστικά για προηγούμενα ερωτήματα που εκτελέστηκαν, όπως ο χρόνος που δαπανήθηκε για την αποτίμηση προτιμήσεων, για την εκτέλεση joins κ.ο.κ.

3.3.3 Υλοποίηση του συστήματος

Παρακάτω περιγράφουμε την υλοποίηση των επεκτεταμένων σχέσεων και των τελεστών στο σύστημα PrefDB.

Υλοποίηση επεκτεταμένων σχέσεων. Οι τιμές για τα σκορ και τους βαθμούς εμπιστοσύνης μιας επεκτεταμένης σχέσης υπολογίζονται κατά την αποτίμηση ενός ερωτήματος με προτιμήσεις. Επιπλέον, συνήθως οι περισσότερες εγγραφές μένουν ανεπηρέαστες από τις προτιμήσεις που δηλώνονται μαζί με ένα ερώτημα. Συνεπώς, δεν έχει νόημα η μόνιμη αποθήκευση των σκορ και βαθμών εμπιστοσύνης στη βάση δεδομένων. Αντιθέτως ακολουθούμε την παρακάτω στρατηγική.

Για κάθε βασικό πίνακα R_B που συμμετέχει σε ένα ερώτημα με προτιμήσεις, κατασκευάζουμε έναν αντίστοιχο πίνακα σκορ R_S(pk, score, conf), όπου με pk συμβολίζουμε το βασικό (πιθανά σύνθετο) κλειδί του βασικού πίνακα R_B. Για να εξοικονομήσουμε επιπλέον χώρο, οι πίνακες σκορ περιέχουν μόνο εγγραφές με μη μηδενικά σκορ, δηλαδή όσες επηρεάζονται από τις προτιμήσεις του ερωτήματος. Συνεπώς αναμένεται να ισχύει ότι $|R_S| \ll |R_B|$. Ο πίνακας R_S ενημερώνεται με νέες τιμές για τα σκορ και τους βαθμούς εμπιστοσύνης κάθε φορά που ένας τελεστής εκτελείται πάνω στη συγκεκριμένη σχέση.

Υλοποίηση Τελεστών. Όλοι οι επεκτεταμένοι τελεστές οι οποίοι εμπλέκουν σχέσεις που δεν περιέχουν εγγραφές με μη μηδενικά σκορ, μπορούν να μετασχηματιστούν σε κλασικούς σχεσιακούς τελεστές. Πριν εκτελέσει κάποιον τελεστή, το σύστημα PrefDB ελέγχει αν ο αντίστοιχος πίνακας σκορ είναι κενός. Αν είναι κενός, τότε η εκτέλεση του τελεστή ανατίθεται στην υποκείμενη μηχανή εκτέλεσης της βάσης δεδομένων. Διαφορετικά, το σύστημα PrefDB κάνει χρήση των υλοποιήσεων των τελεστών όπως περιγράφεται παρακάτω.

Εφόσον έχουμε υλοποιήσει κάθε επεκταταμένη σχέση ως ζεύγος ενός βασικού και ενός πίνακα σκορ, κάθε τελεστής οφείλει να εκτελεστεί και στους δύο πίνακες. Για παράδειγμα, όταν εκτελείται ένας τελεστής επιλογής, οι εγγραφές που δεν ικανοποιούν τις συνθήκες απορρίπτονται και από τους δύο πίνακες. Οι τελεστές προβολείς είναι απλούστεροι: όλα τα γνωρίσματα του πίνακα σκορ πρέπει απαραιτήτως να συμπεριλαμβάνονται στην προβολή, επομένως η προβολή ουσιαστικά αφορά μόνο τον βασικό πίνακα. Οι τελεστές σύζευξης καθώς και οι συνολοθεωρητικοί τελεστές εκτελούνται σε δύο βήματα. Πρώτα, μια συμβατική σύζευξη (αντίστοιχα ένωση ή τομή) εκτελείται στους δύο πίνακες βάσης. Έπειτα, για τις εγγραφές που συμμετέχουν στην παραγόμενη σχέση, υπολογίζονται τα σκορ και οι βαθμοί εμπιστοσύνης τους εφαρμόζοντας την αντίστοιχη συναθροιστική συνάρτηση στους αρχικούς πίνακες σκορ.

Η εκτέλεση ενός τελεστή προτίμησης είναι κάπως πιο περίπλοκη. Συγκεκριμένα, αρχικά η συνθήκη προτίμησης εκτελείται και στους δύο πίνακες R_B και R_S. Όλες οι εγγραφές του πίνακα σκορ που ικανοποιούν τη συνθήκη προτίμησης έχουν ήδη μη μηδενικά σκορ και βαθμούς εμπιστοσύνης, συνεπώς θα πρέπει να τους ανατεθούν νέες τιμές όπως περιγράψαμε στην ενότητα 3.2.3. Επιπλέον, για τις εγγραφές του βασικού πίνακα R_B που δεν εμφανίζονται στον πίνακα σκορ R_S, θα πρέπει να υπολογιστούν οι τιμές σκορ και βαθμών εμπιστοσύνης. Για τον υπολογισμό των σκορ και βαθμών εμπιστοσύνης και για τις δύο κατηγορίες εγγραφών, η συνάρτηση βαθμολόγησης της

προτίμησης εφαρμόζεται στις εγγραφές που ικανοποιούν τη συνθήκη προτίμησης. Στη συνέχεια, καλείται η αντίστοιχη συναθροιστική συνάρτηση για να υπολογίσει τις τελικές τιμές.

3.4 Επεξεργασία Ερωτημάτων στο σύστημα PrefDB

Στην παρούσα ενότητα αναλύουμε τον τρόπο επεξεργασίας ερωτημάτων με προτιμήσεις στο σύστημα PrefDB. Οι ενότητες 3.4.1 και 3.4.2 περιγράφουν το υποσύστημα ανάλυσης και το υποσύστημα βελτιστοποίησης ερωτημάτων αντιστοίχως. Στην ενότητα 3.4.3 παρουσιάζουμε αλγορίθμους εκτέλεσης ερωτημάτων. Θα περιγράψουμε τη λειτουργικότητα των επιμέρους υποσυστημάτων με τη βοήθεια του παρακάτω παραδείγματος:

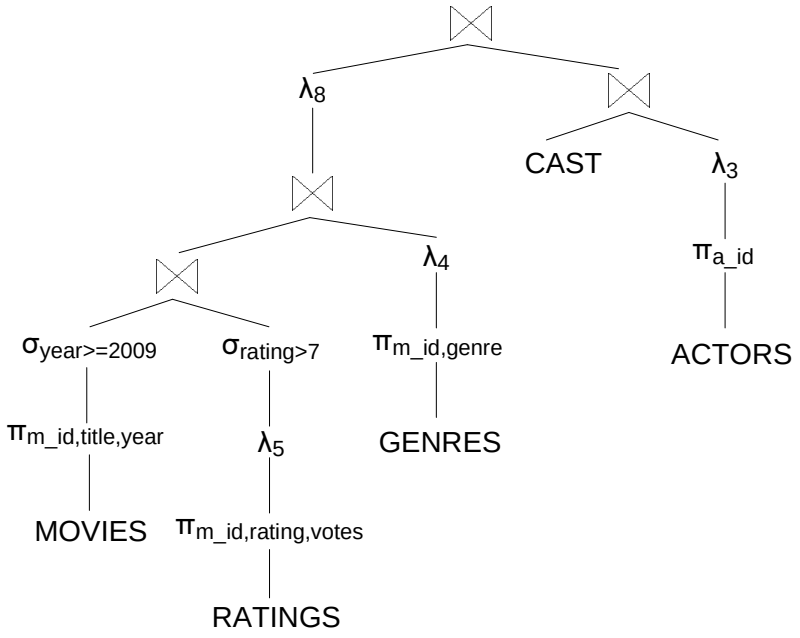
```
SELECT m.m_id, m.title, r.rating, g.genre
FROM MOVIES m, RATINGS r, GENRES g
WHERE m.m_id = r.m_id AND m.m_id = g.m_id AND
m.year ≥ 2009 AND r.rating > 7
```

Για το συγκεκριμένο ερώτημα θα χρησιμοποιήσουμε τις προτιμήσεις p_3, p_4, p_5 και p_8 του Πίνακα 3.2. Χάριν ευκολίας ξαναδίνουμε τον ορισμό τους εδώ:

$p_3[ACTORS] = (\sigma_{a\_id=a_1}, 1, 1)$
$p_4[GENRES] = (\sigma_{genre=\text{`}Comedy\text{'}}, 1, 0.8)$
$p_5[RATINGS] = (\sigma_{votes \geq 100}, S_r(rating), 0.9)$
$p_8[MOVIES \bowtie GENRES] = (\sigma_{genre=\text{`}Romance\text{'}}, S_m(y, 2012), 0.8)$

3.4.1 Ανάλυση Ερωτημάτων

Δοθέντος ενός ερωτήματος και ενός συνόλου προτιμήσεων, το υποσύστημα ανάλυσης ερωτημάτων αρχικά κατασκευάζει ένα επεκτεταμένο πλάνο εκτέλεσης το οποίο περιέχει όλους τους επεκτεταμένους τελεστές και τον τελεστή προτίμησης. Στο παραγόμενο πλάνο εκτέλεσης, οι σειρά των τελεστών ακολουθεί τη σειρά εμφάνισής τους στο αρχικό ερώτημα. Το υποσύστημα ανάλυσης ερωτημάτων προσθέτει έναν τελεστή προτίμησης για κάθε προτίμηση. Επίσης, για κάθε προτίμηση ελέγχεται αν εμπλέκει κάποιο γνώρισμα (είτε στη συνθήκη προτίμησης είτε στη συνάρτηση βαθμολόγησης) το οποίο δεν περιέχεται στο ερώτημα. Σε αυτή την περίπτωση οι τελεστές προβολής τροποποιούνται κατάλληλα ώστε να περιλαμβάνουν επίσης και τα συγκεκριμένα γνωρίσματα. Επιπλέον, το υποσύστημα ανάλυσης ερωτημάτων, προσθέτει επιπλέον συνθήκες σύζευξης με σχέσεις οι οποίες μπορεί να μην εμφανίζονται στο αρχικό SQL ερώτημα, αλλά εμπλέκονται στις προτιμήσεις που έχουν δηλωθεί μαζί με το ερώτημα. Για παράδειγμα, η προτίμηση p_3 είναι ορισμένη στη σχέση $ACTORS$, και επομένως το υποσύστημα ανάλυσης ερωτημάτων θα προσθέσει την παρακάτω συνθήκη σύζευξης $MOVIES \bowtie CAST \bowtie ACTORS$. Το Σχήμα 3.7 δείχνει το πλάνο εκτέλεσης που παράγεται για το παραπάνω παράδειγμα ερώτησης.



Σχήμα 3.7: *Ένα επεκτεταμένο πλάνο εκτέλεσης*

3.4.2 Βελτιστοποίηση Ερωτημάτων

Όπως περιγράψαμε παραπάνω, ένα επεκτεταμένο πλάνο εκτέλεσης περιλαμβάνει επεκτεταμένους τελεστές καθώς και τελεστές προτίμησης. Είναι προφανές λοιπόν πως ένας συμβατικός βελτιστοποιητής ερωτημάτων μιας βάσης δεδομένων δεν μπορεί να χρησιμοποιηθεί για τέτοιου είδους πλάνα εκτέλεσης. Παρόλαυτα, η υλοποίηση των επεκεταταμένων σχέσεων που ακολουθούμε στο σύστημα PrefDB διαχωρίζει κατά κάποιο τρόπο τις εγγραφές που σχετίζονται με την αποτίμηση προτιμήσεων (αυτές που περιέχονται στους πίνακες σκορ) από αυτές που σχετίζονται με το υπόλοιπο κομμάτι της ερώτησης (αυτές που περιέχονται στους πίνακες βάσης). Επομένως, μπορούμε να υποθέσουμε με ασφάλεια ότι το συνολικό κόστος εκτέλεσης αποτελείται από δύο κομμάτια, ένα κομμάτι που σχετίζεται με τον υπολογισμό και την συνάθροιση σκορ και βαθμών εμπιστοσύνης στους πίνακες σκορ, και ένα κομμάτι που σχετίζεται με τους υπόλοιπους τελεστές (επιλογές, συζεύξεις, κλπ.) στους πίνακες βάσης.

Οι επεκτεταμένοι σχεσιακοί τελεστές δεν αλλάζουν τον τρόπο που οι εγγραφές φιλτράρονται με βάση τον συμβατικό τους ορισμό. Επίσης, η εκτέλεση ενός τελεστή προτίμησης δεν προκαλεί την απόρριψη καμιάς εγγραφής. Συνεπώς, οι νέοι τελεστές δεν επηρεάζουν το κομμάτι του κόστους επεξεργασίας που σχετίζεται με τους πίνακες βάσης. Αναμένεται επομένως ότι η σειρά εκτέλεσης joins που θα επέλεγε ο βελτιστοποιητής ερωτημάτων της βάσης για ένα ερώτημα αν αφαιρεθούν οι τελεστές προτίμησης, θα έχει καλή απόδοση και για το ίδιο ερώτημα μαζί με τις προτιμήσεις. Με βάση αυτή την παρατήρηση, ο βελτιστοποιητής ερωτημάτων του συστήματος PrefDB διατηρεί την σειρά εκτέλεσης joins που επιλέγεται από τη βάση δεδομένων, και θεωρεί το συγκεκριμένο κομμάτι του κόστους εκτέλεσης ως σταθερό. Συνεπώς, στόχος του βελτιστοποιητή είναι η ελαχιστοποίηση του υπόλοιπου κόστους, δηλαδή του κόστους που σχετίζεται με την αποτίμηση προτιμήσεων.

Συνήθως η πιο καθοριστική παράμετρος που επηρεάζει το κόστος επεξεργασίας ενός ερωτήματος είναι ο αριθμός λειτουργιών εισόδου/εξόδου από τον δίσκο, ο οποίος

είναι ανάλογος με τον αριθμό εγγραφών που 'ρέουν' από έναν τελεστή του πλάνου εκτέλεσης στον επόμενο τελεστή. Κάνοντας την παραδοχή ότι οι υπόλοιποι τελεστές έχουν σταθερή θέση στο πλάνο εκτέλεσης, ουσιαστικά ο στόχος του βελτιστοποιητή είναι να τοποθετήσει τους τελεστές προτίμησης στο πλάνο εκτέλεσης, έτσι ώστε ο αριθμός εγγραφών που μεταφέρονται μέσω των πινάκων σκορ να είναι ο ελάχιστος δυνατός.

Με βάση τα παραπάνω, οι στρατηγική του βελτιστοποιητή περιλαμβάνει τα παρακάτω βήματα. Αρχικά, εξάγεται η σειρά εκτέλεσης joins που θα επέλεγε ο βελτιστοποιητής ερωτημάτων της βάσης για το αρχικό ερώτημα χωρίς τις προτιμήσεις. Στη συνέχεια, κάνοντας χρήση των αλγεβρικών ιδιοτήτων που περιγράψαμε στην ενότητα 3.2.3, ο βελτιστοποιητής εφαρμόζει ένα σύνολο ευριστικών κανόνων που έχουν σκοπό να βελτιώσουν την θέση των τελεστών προτίμησης στο πλάνο εκτέλεσης. Ο σκοπός του βήματος αυτού είναι διπλός: (α) επιχειρούμε να ελαχιστοποιήσουμε τον αριθμό εγγραφών που δίνονται ως είσοδος στους τελεστές προτίμησης, και (β) απορρίπτουμε λιγότερο αποδοτικά (suboptimal) πλάνα και με αυτό τον τρόπο περιορίζουμε τον χώρο αναζήτησης εναλλακτικών πλάνων που θα εξετάσουμε στο επόμενο βήμα. Τελικά, προτείνονται δύο αλγόριθμοι εκτίμησης κόστους οι οποίοι (α) εξετάζουν εναλλακτικές θέσεις των τελεστών προτίμησης, και (β) επιλέγουν το πλάνο με το ελάχιστο εκτιμώμενο κόστος. Στη συνέχεια αναλύουμε την προσέγγισή μας. Στην ενότητα 3.4.2.1 συζητάμε κάποιους ευριστικούς κανόνες βελτιστοποίησης, ενώ στην ενότητα 3.4.2.2 προτείνουμε ένα μοντέλο κόστους και μεθόδους εκτίμησης κόστους.

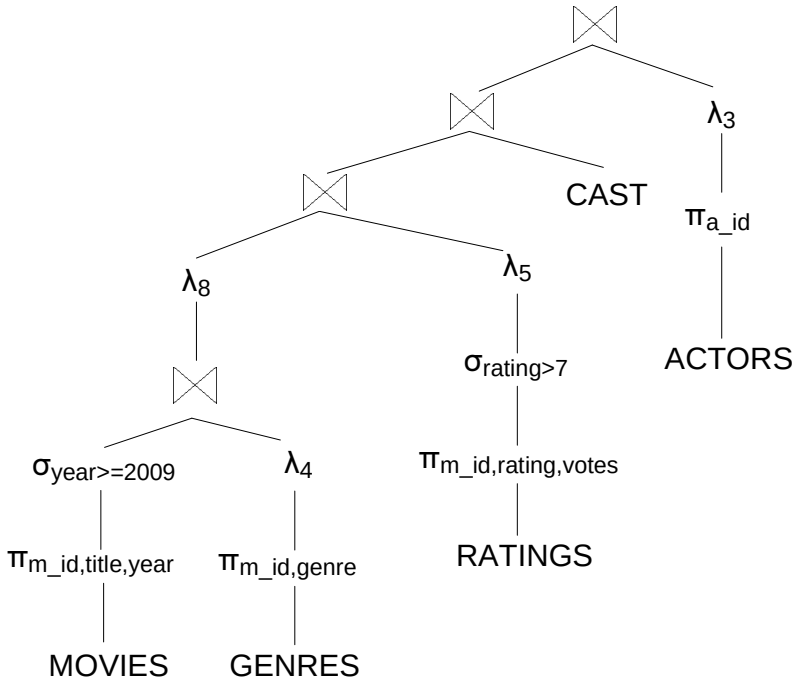
3.4.2.1 Βελτιστοποίηση ερωτημάτων βασισμένη σε κανόνες

Αρχικά, ο βελτιστοποιητής ανακτά το πλάνο εκτέλεσης που θα επέλεγε ο βελτιστοποιητής της βάσης δεδομένων.[1] Στη συνέχεια, ο βελτιστοποιητής αναδιατάσσει το πλάνο που έλαβε ως είσοδο από τον αναλυτή ερωτημάτων, έτσι ώστε να ακολουθεί την προτεινόμενη σειρά εκτέλεσης joins. Τέλος, το παραγόμενο πλάνο τροποποιείται εφαρμόζοντας ένα σύνολο κανόνων ισοδυναμίας οι οποίοι εκμεταλλεύονται τις αλγεβρικές ιδιότητες του τελεστή προτίμησης. Στόχος είναι να ελαχιστοποιηθεί κατά το δυνατόν ο αριθμός εγγραφών που 'ρέουν' διαμέσου των τελεστών του πλάνου εκτέλεσης. Πιο συγκεκριμένα, ο βελτιστοποιητής ερωτημάτων εφαρμόζει τους παρακάτω ευριστικούς κανόνες:

1. Οι τελεστές επιλογής που εμπλέκουν γνωρίσματα εκτός των σκορ και βαθμών εμπιστοσύνης, σπρώχνονται όσο χαμηλότερα γίνεται στο πλάνο εκτέλεσης. Αν μια συνθήκη επιλογής εμπλέκει περισσότερες των μία σχέσεων, η συνθήκη σπάει και κάθε κομμάτι μεταφέρεται χαμηλότερα ξεχωριστά. Με βάση την Ιδιότητα 1, είναι δυνατόν ένας τελεστής επιλογής να μεταφερθεί χαμηλότερα από έναν τελεστή προτίμησης.

2. Οι τελεστές προβολής σπρώχνονται όσο χαμηλότερα γίνεται στο πλάνο εκτέλεσης, διασφαλίζοντας όμως ότι όλα τα απαιτούμενα γνωρίσματα θα είναι διαθέσιμα

[1] Συνήθως το προτεινόμενο πλάνο εκτέλεσης είναι δυνατόν να ανακτηθεί χωρίς να εκτελεστεί το ερώτημα, εκτελώντας μια εντολή SHOW PLAN ή EXPLAIN με αμελητέο κόστος.



Σχήμα 3.8: *Παραγόμενο πλάνο μετά την εφαρμογή των ευριστικών κανόνων* για τους επόμενους τελεστές.

3. Αν ένας τελεστής προτίμησης είναι τοποθετημένος ψηλότερα από έναν δυαδικό τελεστή (π.χ. σύζευξη, ένωση, τομή) και ταυτόχρονα εμπλέκει γνωρίσματα μόνο από μια από τις δύο σχέσεις, τότε μεταφέρεται χαμηλότερα από τον δυαδικό τελεστή στην αντίστοιχη σχέση.

4. Αν ένας τελεστής προτίμησης και ένας τελεστής επιλογής είναι ορισμένοι πάνω στην ίδια σχέση, τότε όλες οι συνθήκες επιλογές που εμπλέκουν γνωρίσματα εκτός του σκορ και του βαθμού εμπιστοσύνης προστίθενται στη συνθήκη προτίμησης του τελεστή προτίμησης.

Οι ευριστικοί κανόνες 1 και 2 χρησιμοποιούνται ευρέως και στη βελτιστοποίηση συμβατικών πλάνων εκτέλεσης, με στόχο τον περιορισμό του αριθμού και του μεγέθους, αντιστοίχως, των εγγραφών που ρέουν μέσω του πλάνου εκτέλεσης. Ο ευριστικός κανόνας 3 βασίζεται στην Ιδιότητα 3. Ο συγκεκριμένος κανόνας εφαρμόζεται έτσι ώστε κάθε τελεστής προτίμησης να τοποθετηθεί στη χαμηλότερη δυνατή θέση του πλάνου εκτέλεσης. Ο ευριστικός κανόνας 4 βασίζεται στην Ιδιότητα 4 και η σκοπιμότητά του θα εξηγηθεί στην ενότητα 3.4.3 όπου προτείνουμε στρατηγικές εκτέλεσης ερωτημάτων.

Παράδειγμα 3.12. *Το Σχήμα 3.8 απεικονίζει το παραγόμενο πλάνο εκτέλεσης μετά την εφαρμογή των ευριστικών κανόνων στο πλάνο του Σχήματος 3.7. Μπορούμε να παρατηρήσουμε ότι ο τελεστής* λ_5 *έχει αλλάξει θέση με τον τελεστή* $\sigma_{rating>7}$*, ενώ ο τελεστής* λ_8 *έχει μεταφερθεί χαμηλότερα από τον τελεστή σύζευξης, αφού εμπλέκει γνωρίσματα μόνο από το αριστερό κομμάτι του τελεστή (*$MOVIES$ *και* $GENRES$*).* □

3.4.2.2 Βελτιστοποίηση ερωτημάτων βασισμένη στο κόστος

Στην ενότητα αυτή, αρχικά εισάγουμε ένα μοντέλο κόστους για επιμέρους τελεστές και στη συνέχεια περιγράφουμε πώς μπορεί να εκτιμηθεί το συνολικό κόστος ενός πλάνου εκτέλεσης με βάση τις θέσεις των τελεστών προτίμησης. Τέλος, προτείνουμε δύο αλγορίθμους εκτίμησης κόστους οι οποίοι επιλέγουν το πιο αποδοτικό πλάνο εκτέλεσης.

Μοντέλο κόστους. Θα θεωρήσουμε το κόστος ενός τελεστή προτίμησης ανάλογο με τον αριθμό εγγραφών που επηρεάζονται από τον τελεστή, ακολουθώντας μια προσέγγιση παρόμοια με αυτή που ακολουθείται για την εκτίμηση του κόστους των συμβατικών τελεστών της σχεσιακής άλγεβρας.[2] Οι εγγραφές που επηρεάζονται προέρχονται τόσο από τους πίνακες βάσης όσο και από τους πίνακες σκορ. Έστω $cost(R_{B_i}, \lambda_j)$ το κόστος αποτίμησης ενός τελεστή προτίμησης λ_j στον πίνακα βάσης R_{B_i}. Έχουμε:

$$cost(R_{B_i}, \lambda_j) \simeq sel(R_{B_i}, \lambda_j) \cdot |R_{B_i}| + \alpha \cdot sel(R_{S_i}, \lambda_j) \cdot |R_{S_i}|$$

όπου R_{B_i} και R_{S_i} συμβολίζουν τους πίνακες βάσεις και σκορ και $sel(R_{B_i}, \lambda_j)$ και $sel(R_{S_i}, \lambda_j)$ συμβολίζουν την επιλεκτικότητα του τελεστή λ_j όταν εφαρμοστεί στους πίνακες R_{B_i} και R_{S_i} αντιστοίχως. Σημειώνουμε ότι το κόστους υπολογισμού νέων σκορ και βαθμών εμπιστοσύνης (το οποίο σχετίζεται με τις εγγραφές που περιέχονται στον πίνακα βάσης) είναι διαφορετικό από το κόστος συνάθροισης σκορ και βαθμών εμπιστοσύνης (το οποίο σχετίζεται με τις εγγραφές που περιέχονται στον πίνακα σκορ). Γι'αυτό το λόγο, χρησιμοποιούμε έναν συντελεστή κανονικοποίησης όταν προσθέτουμε τα δύο παραπάνω κόστη.

Για την εκτίμηση του κόστους των τελεστών ένωσης, τομής και σύζευξης, θα χρησιμοποιήσουμε το μέγεθος του πίνακα βάσης που αναμένεται να παραχθεί, σύμφωνα με τα στατιστικά της βάσης δεδομένων. Εδώ θα εστιάσουμε στο κόστος της συνάθροισης σκορ και βαθμών εμπιστοσύνης. Επειδή η κατανομή των εγγραφών στους πίνακες αυτούς εξαρτάται από τις προτιμήσεις που έχουν αποτιμηθεί σε προηγούμενα βήματα της εκτέλεσης του ερωτήματος, θα χρησιμοποιήσουμε τις παρακάτω εκτιμήσεις που αντιστοιχούν στη χειρότερη περίπτωση:

$$cost(R_i \cup R_j) \simeq |R_{S_i}| + |R_{S_j}|$$

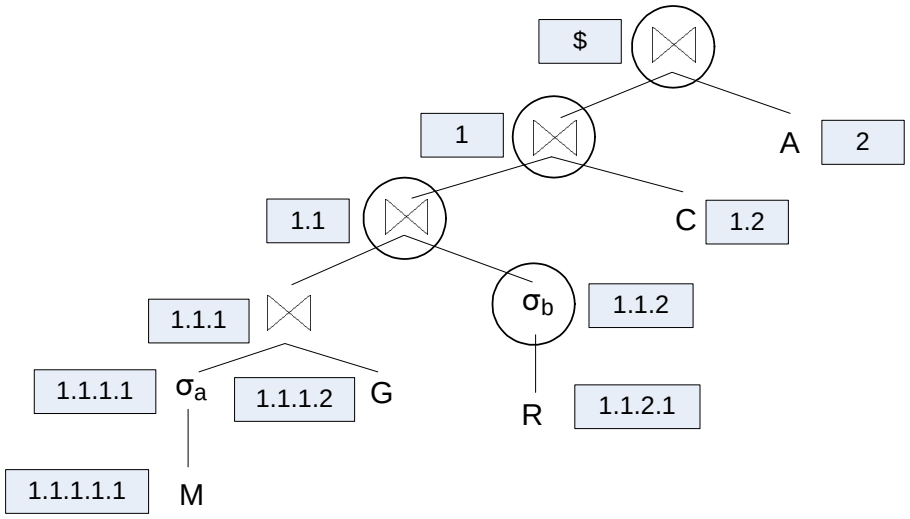
$$cost(R_i \cap R_j) \simeq min\{|R_{S_i}|, |R_{S_j}|\}$$

$$cost(R_i \bowtie R_j) \simeq sel(R_{B_i}, R_{B_j}) \cdot |R_{S_i}| \cdot |R_{S_j}|$$

όπου $sel(R_{B_i}, R_{B_j})$ είναι η επιλεκτικότητα σύζευξης των πινάκων R_{B_i}, R_{B_j}.

Στη συνέχεια περιγράφουμε πώς μπορεί να εκτιμηθεί το συνολικό κόστος του πλάνου εκτέλεσης. Όπως εξηγήσαμε παραπάνω, μια ακριβής εκτίμηση κόστους θα πρέπει

[2] Οι διαθέσιμες μέθοδοι προσπέλασης των δεδομένων σε φυσικό επίπεδο, καθώς και το κόστος υπολογισμού των σκορ και βαθμών εμπιστοσύνης μπορεί επίσης να επηρεάζουν σημαντικά το κόστος εκτέλεσης. Χάριν απλότητας, στο προτεινόμενο μοντέλο κόστους θα θεωρήσουμε ότι για όλες τις σχέσεις υπάρχει μια προκαθορισμένη μέθοδος πρόσβασης και ότι οι συναρτήσεις βαθμολόγησης που χρησιμοποιούμε δεν απαιτούν πολύπλοκους υπολογισμούς.



Σχήμα 3.9: *Εναλλακτικές θέσεις για τον τελεστή* λ_5

να λαμβάνει υπόψη της: (α) το κόστος που σχετίζεται με τους πίνακες σκορ, και (β) το κόστος που σχετίζεται με το συμβατικό κομμάτι της εκτέλεσης ενός ερωτήματος (π.χ. τελεστές επιλογής, προβολής, σύζευξης στους πίνακες βάσης). Όμως, αφού η σειρά εκτέλεσης των joins είναι προκαθορισμένη και ίδια για όλα τα πλάνα εκτέλεσης που θα εξεταστούν, το κόστος που σχετίζεται με τους πίνακες βάσης είναι σταθερό. Συνεπώς, το κόστος ενός πλάνου εκτέλεσης $plan_q$ καθορίζεται από το κομμάτι που σχετίζεται με την αποτίμηση προτιμήσεων, δηλαδή:

$$cost(plan_q) \simeq \sum_{p\_op \in plan_q} cost(p\_op)$$

όπου με p_op συμβολίζουμε έναν τελεστή του πλάνου εκτέλεσης.

Εκτίμηση κόστους. Στη συνέχεια περιγράφουμε πώς ο βελτιστοποιητής ερωτημάτων εφαρμόζει το προτεινόμενο πλάνο εκτέλεσης με σκοπό να επιλέξει ένα όσο το δυνατόν πιο αποδοτικό πλάνο εκτέλεσης. Το Σχήμα 3.9 απεικονίζει το πλάνο που παρήχθη από το πρώτο βήμα βελτιστοποίησης βάσει των ευριστικών κανόνων (Σχήμα 3.8) όπου χάριν απλότητας στην παρουσίαση έχουν αφαιρεθεί οι τελεστές προβολής και προτίμησης. Έστω λ_j ένας τελεστής προτίμησης που αποτιμά μια προτίμηση $p_j[R]$. Σύμφωνα με τον ευριστικό κανόνα 1, ο τελεστής λ_j θα πρέπει να έχει ήδη τοποθετηθεί ψηλότερα από τελεστές επιλογής ή προβολής, εφόσον υπάρχουν. Βάσει των αλγεβρικών ιδιοτήτων του τελεστή προτίμησης (Ιδιότητες 1, 2 και 3) και κάνοντας την παραδοχή ότι εκτελούνται οι κατάλληλες προβολές στα γνωρίσματα που απαιτούνται, μπορούμε να επιλέξουμε να εφαρμόσουμε τον τελεστή λ_j σε οποιαδήποτε θέση ανάμεσα στην τρέχουσα και τη ρίζα του πλάνου εκτέλεσης. Οι εναλλακτικές θέσεις για τον τελεστή λ_5 απεικονίζονται με κύκλο στο Σχήμα 3.9.

Θεωρώντας μια προκαθορισμένη σειρά εκτέλεσης joins, ο χώρος αναζήτησης αποτελείται από όλα τα πλάνα που ακολουθούν τη συγκεκριμένη σειρά εκτέλεσης joins, αλλά διαφέρουν στην ακριβή θέση των τελεστών προτίμησης. Επομένως, ο στόχος βελτιστοποίησης είναι η εύρεση ενός πλάνου εκτέλεσης όπου οι τελεστές προτίμησης είναι τοποθετημένοι με τέτοιον τρόπο ώστε να ελαχιστοποιείται το παραπάνω κόστος $cost(plan_q)$.

Έστω h το ύψος του πλάνου εκτέλεσης και έστω P ο αριθμός των τελεστών προτίμησης που περιλαμβάνονται στο πλάνο. Μια εξαντλητική εξέταση όλων των εναλλακτικών πλάνων θα απαιτούσε $O(h^P)$ εκτιμήσεις κόστους, κάτι που είναι απαγορευτικά ακριβό από πλευράς χρόνου ακόμα και για μικρές τιμές των h ή P. Αξίζει να σημειωθεί βεβαίως, ότι χάρη στο πρώτο βήμα βελτιστοποίησης βάσει ευριστικών κανόνων, έχει ήδη μειωθεί το μέγεθος του χώρου αναζήτησης, καθώς έχουν απορριφθεί κάποια πλάνα τα οποία είναι λιγότερο αποδοτικά. Για παράδειγμα, επανερχόμενοι στο Σχήμα 3.9, κατά το δεύτερο βήμα βελτιστοποίησης βασισμένη στο κόστος, δεν θα εξεταστούν πλάνα στα οποία ο τελεστής λ_5 εφαρμόζεται απευθείας πάνω στη σχέση R, καθώς ο τελεστής επιλογής έχει μεταφερθεί χαμηλότερα και προηγείται του τελεστή προτίμησης λ_5.

Στη συνέχεια προτείνουμε δύο προσεγγιστικούς αλγορίθμους εκτίμησης κόστους οι οποίοι καταλήγουν σε ένα αποδοτικό πλάνο εκτέλεσης εξετάζοντας ένα υποσύνολο των εναλλακτικών πλάνων. Δανειζόμενοι κάποιες ιδέες από συμβατικές μεθόδους απαρίθμησης πλάνων εκτέλεσης, αρχικά περιγράφουμε μια τεχνική βασισμένη σε δυναμικό προγραμματισμό. Για πιο σύνθετα ερωτήματα (μεγαλύτερο αριθμό συζεύξεων ή προτιμήσεων) ακόμα και ο περιορισμένος αριθμός πλάνων εκτέλεσης που εξετάζονται από έναν αλγόριθμο δυναμικού προγραμματισμού, απαιτεί εκθετικό αριθμό υπολογισμών. Η συμπεριφορά αυτή επιβεβαιώνεται και από τα πειράματά μας (ενότητα 3.5).

Για το λόγο αυτό, προτείνουμε επίσης έναν άπληστο αλγόριθμο που προσδιορίζει διαδοχικά τη θέση κάθε τελεστή προτίμησης στο πλάνο εκτέλεσης. Τα πειράματά μας δείχνουν ότι και οι δύο αλγόριθμοι παράγουν πλάνα εκτέλεσης τα οποία απαιτούν χρόνο εκτέλεσης που είναι συγκρίσιμος με αυτόν που απαιτείται από το βέλτιστο πλάνο, όπως αυτό θα προέκυπτε από μια εξαντλητική αναζήτηση στο χώρο όλων των πιθανών πλάνων. Παρακάτω παρουσιάζουμε τις λεπτομέρειες των προτεινόμενων αλγορίθμων.

Αλγόριθμος Δυναμικού Προγραμματισμού. Η συγκεκριμένη τεχνική δανείζεται ιδέες από τις μεθόδους απαρίθμησης πλάνων εκτέλεσης που χρησιμοποιούνται στη βελτιστοποίηση συμβατικών ερωτημάτων. Σε εκείνη την περίπτωση, ο στόχος βελτιστοποίησης είναι να προκύψει μια αποδοτική σειρά εκτέλεσης joins [63]. Το πρόβλημα της βελτιστοποίησης επεκτεταμένων πλάνων εκτέλεσης είναι κάπως πιο πολύπλοκο. Συγκεκριμένα, το κόστος δεν εξαρτάται μόνο από τη σειρά των τελεστών αλλά και από την ακριβή θέση τους στο πλάνο εκτέλεσης. Ο πρώτος λόγος έχει να κάνει με το γεγονός ότι για κάθε θέση στο πλάνο προκύπτει διαφορετικός αναμενόμενος αριθμός εγγραφών, κάτι που επηρεάζει το μέγεθος του πίνακα βάσης στον οποίο ένας τελεστής προτίμησης εφαρμόζεται. Ο δεύτερος λόγος είναι ότι όταν υπολογίζουμε το τελικό κόστος, πρέπει επίσης να συμπεριλάβουμε το κόστος συνάθροισης των τιμών για τα skor και βαθμούς εμπιστοσύνης. Τέλος, πρέπει να σημειωθεί ότι η διαδικασία εκτίμησης κόστους για κάθε τελεστή απαιτεί την εξέταση όλων των εναλλακτικών θέσεων για τον τελεστή, κάτι που χρειάζεται $O(h)$ υπολογισμούς.

Στη συνέχεια δίνουμε τις λεπτομέρειες εκτέλεσης του αλγορίθμου. Ο αλγόριθμος ξεκινάει εξετάζοντας όλες τις πιθανές θέσεις για ένα ζεύγος τελεστών προτίμησης. Υπολογίζεται το κόστος κάθε πλάνου και για κάθε τελεστή προτίμησης σημειώνεται η θέση η οποία ελαχιστοποιεί το εκτιμώμενο κόστος. Στο i-οστό βήμα εκτέλεσης, ο αλγόριθμος εξετάζει όλα τα υποσύνολα τελεστών προτίμησης μεγέθους i και για κάθε συνδυασμό σημειώνει το σύνολο των βέλτιστων θέσεων. Στο επόμενο βήμα,

παράγονται όλοι οι συνδυασμοί τελεστών μεγέθους $i+1$ επεκτείνοντας τα υποσύνολα μεγέθους i με έναν επιπλέον τελεστή προτίμησης. Για κάθε υποσύνολο τελεστών που έχουν ήδη εξεταστεί χρησιμοποιούμε τις βέλτιστες θέσεις τους όπως υπολογίστηκαν σε κάποιο από τα προηγούμενα βήματα. Ο αλγόριθμος τερματίζει όταν ολοκληρωθεί η εκτίμηση κόστους και των P τελεστών προτίμησης, οπότε επιστρέφεται το σύνολο των βέλτιστων θέσεων.

Πρέπει να σημειωθεί ότι κάθε εκτίμηση κόστους απαιτεί τον προσδιορισμό όλων των θέσεων που βρίσκονται ψηλότερα από την τρέχουσα θέση ενός τελεστή, έτσι ώστε να εξακριβωθούν οι έγκυρες θέσεις και να ενημερωθεί το αναμενόμενο κόστος κατάλληλα. Για να γίνει αυτό πιο αποδοτικά, χρησιμοποιούμε ένα σχήμα κωδικοποίησης για δέντρα (tree labeling scheme). Συγκεκριμένα, χρησιμοποιήσαμε την κωδικοποίηση Dewey[3], η οποία επιτρέπει την απάντηση ερωτημάτων εύρεσης κόμβων-προγόνων (ancestor queries) σε σταθερό χρόνο. Στο Σχήμα 3.9, κάθε κόμβος είναι κωδικοποιημένος κατά Dewey. Για παράδειγμα οι κόμβοι 1.1 και 1 είναι εναλλακτικές θέσεις για τον τελεστή λ_5, αφού οι ετικέτες τους περιέχονται στην ετικέτα 1.1.2.

Άπληστος Αλγόριθμος. Ο άπληστος αλγόριθμος τοποθετεί διαδοχικά τους τελεστές προτίμησης στο πλάνο εκτέλεσης, εξετάζοντάς τους με αύξουσα σειρά ελάχιστου κόστους. Σε κάθε βήμα, ο άπληστος αλγόριθμος επιλέγει μεταξύ των τελεστών προτίμησης των οποίων δεν έχει προσδιοριστεί ακόμα η ακριβής θέση, αυτόν που έχει το χαμηλότερο κόστος εκτέλεσης. Ο αλγόριθμος σημειώνει τη συγκεκριμένη θέση ως τη βέλτιστη για τον τρέχοντα εξεταζόμενο τελεστή προτίμησης, και ενημερώνει κατάλληλα τα κόστη εκτέλεσης όλων των τελεστών που είναι τοποθετημένοι ψηλότερα στο πλάνο εκτέλεσης. Ο αλγόριθμος εκτελείται σε επαναλήψεις έως ότου η θέση όλων των τελεστών προτίμησης έχει προσδιοριστεί, οπότε τροποποιεί κατάλληλα το πλάνο εκτέλεσης με σκοπό να ανταναλά τις νέες επιλεγμένες θέσεις των τελεστών προτίμησης. Η εκτέλεση του αλγορίθμου θα γίνει πιο κατανοητή με τη βοήθεια του επόμενου παραδείγματος.

Παράδειγμα 3.13. *Έστω το πλάνο εκτέλεσης του Σχήματος 3.9. Τα αρχικά κόστη εκτέλεσης κάθε τελεστή προτίμησης φαίνονται στο Σχήμα 3.10(α΄). Κατά την πρώτη επανάληψη, ο αλγόριθμος επιλέγει τον τελεστή λ_8 ο οποίος έχει το ελάχιστο τρέχον εκτιμώμενο κόστος ανάμεσα σε όλους τους τελεστές και σημειώνει τη θέση του στον κόμβο 1.1.1. Επιπλέον, ενημερώνει τα εκτιμώμενα κόστη εκτέλεσης όλων των κόμβων που βρίσκονται ψηλότερα από τη θέση 1.1.1. Το Σχήμα 3.10(β΄) απεικονίζει τις ενημερωμένες τιμές για τα κόστη. Ομοίως, στις παρακάτω επαναλήψεις ο αλγόριθμος προσδιορίζει τις βέλτιστες θέσεις των τελεστών λ_5, λ_4 και λ_3 ως 1.1.2, 1.1.1 και τη ρίζα του πλάνου εκτέλεσης αντιστοίχως. Οι τιμές με έντονη γραμματοσειρά αντιπροσωπεύουν τα τρέχοντα εκτιμώμενα ελάχιστα κόστη ανά τελεστή. Το Σχήμα 3.11 δείχνει το τελικό πλάνο εκτέλεσης που παράγεται από την εκτέλεση του άπληστου αλγορίθμου.*

3.4.3 Εκτέλεση Ερωτημάτων

Το υποσύστημα εκτέλεσης ερωτημάτων του συστήματος PrefDB είναι υπεύθυνο για την επεξεργασία ερωτημάτων με προτιμήσεις και υποστηρίζει ένα σύνολο αλγορίθμων

[3]http://www.oclc.org/dewey/

Κόμβος	λ_3	λ_4	λ_5	λ_8
1.1.1.2	-	100	-	-
1.1.1	-	**40**	-	**30**
1.1.2	-	-	50	-
1.1	-	150	**40**	100
1	-	100	150	100
2	150	-	-	-
$	**50**	150	200	130

(α′) Αρχικά επιμέρους κόστη τελεστών

Κόμβος	λ_3	λ_4	λ_5
1.1.1.2	-	100	-
1.1.1	-	**55**	-
1.1.2	-	-	**50**
1.1	-	165	55
1	-	115	165
2	150	-	-
$	**65**	165	215

(β′) Κόστη εκτέλεσης μετά τον προσδιορισμό της θέσης του τελεστή λ_8

Κόμβος	λ_3	λ_4
1.1.1.2	-	100
1.1.1	-	**55**
1.1.2	-	-
1.1	-	80
1	-	190
2	150	-
$	**90**	240

(γ′) Κόστη εκτέλεσης μετά τον προσδιορισμό της θέσης του τελεστή λ_5

Κόμβος	λ_3
1.1.1.2	-
1.1.1	-
1.1.2	-
1.1	-
1	-
2	150
$	**92.5**

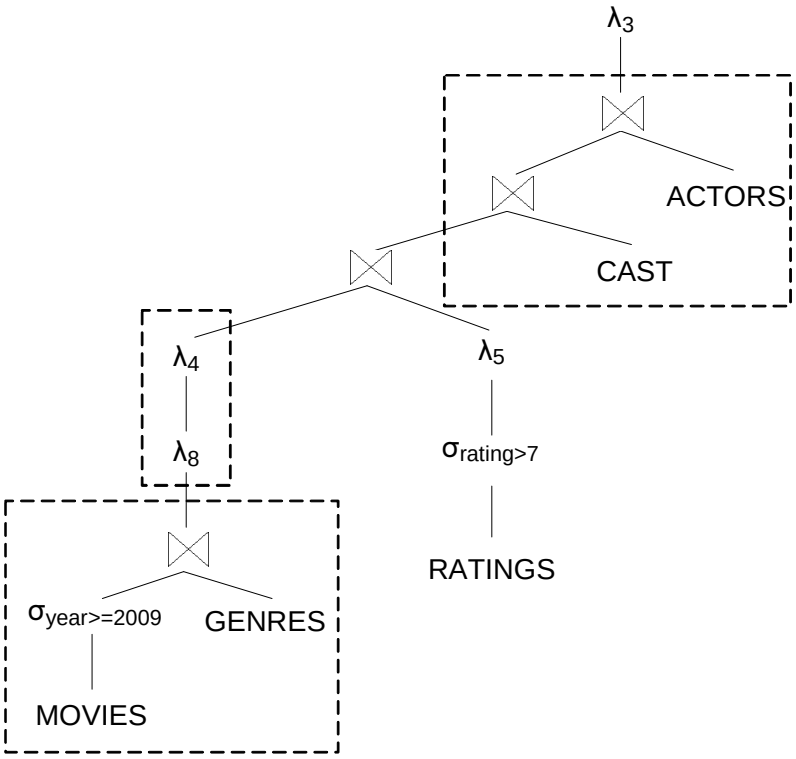
(δ′) Κόστη εκτέλεσης μετά τον προσδιορισμό της θέσης του τελεστή λ_4

Σχήμα 3.10: *Παράδειγμα εκτέλεσης του άπληστου αλγορίθμου απαρίθμησης πλάνων εκτέλεσης*

που έχουμε αναπτύξει. Σε αυτή την ενότητα εστιάζουμε σε αλγορίθμους που εφαρμόζονται σε επεκτεταμένα πλάνα εκτέλεσης. Η εργασία [8] περιγράφει δύο plug-in αλγορίθμους που έχουμε επίσης υλοποιήσει. Επίσης η ενότητα ;; περιλαμβάνει διεξοδική πειραματική σύγκριση όλων των μεθόδων.

Κατά την εκτέλεση ενός ερωτήματος θα παραμείνουμε συνεπείς με τον στόχο βελτιστοποίησης, δηλαδή να ελαχιστοποιήσουμε όσο το δυνατόν τον αριθμό εγγραφών που ρέουν μέσω των τελεστών. Συγκεκριμένα, εδώ ο στόχος μας είναι να ελαχιστοποιήσουμε τον αριθμό των ενδιάμεσων εγγραφών στο δίσκο κατά τη διάρκεια της εκτέλεσης ενός ερωτήματος. Γι'αυτό το λόγο καταφεύγουμε: (α) σε ομαδοποίηση της εκτέλεσης κάποιων τελεστών, και (β) σε χρήση του υποσυστήματος εκτέλεσης ερωτημάτων της βάσης δεδομένων, όπου αυτό είναι εφικτό. Οι συγκεκριμένες μέθοδοι βελτιστοποίησης θα φανούν καλύτερα με τη βοήθεια ενός παραδείγματος. Έστω το πλάνο εκτέλεσης του Σχήματος 3.11. Μετά από μια πιο προσεκτική εξέταση του πλάνου εκτέλεσης μπορούμε να κάνουμε τις παρακάτω παρατηρήσεις:

- Πριν εκτελεστεί ο τελεστής λ_8 οι αντίστοιχες επεκτεταμένες σχέσεις είναι στην ουσία 'συμβατικές ' σχέσεις, αφού καμιά εγγραφή με σκορ δεν έχει παραχθεί ακόμα. Επομένως, μπορούμε να εκτελέσουμε τα $\sigma_{year \geq 2009}$ και $MOVIES \bowtie$



Σχήμα 3.11: *Οι τελεστές μέσα σε διακεκομμένες γραμμές μπορούν να εκτελεστούν σε ένα βήμα*

$GENRES$ με τον συμβατικό τρόπο. Επιπλέον, είναι δυνατόν να συνδυάσουμε την εκτέλεση των $\sigma_{year \geq 2009}$ και $MOVIES \bowtie GENRES$ σε ένα βήμα, το οποίο μπορεί να εκτελεστεί σαν υποερώτημα από τη βάση δεδομένων.

- Εξαιτίας του ευριστικού κανόνα 4 που περιγράψαμε στην ενότητα 3.4.2, ο βελτιστοποιητής ερωτημάτων έχει ήδη προσθέσει τη συνθήκη $\sigma_{rating>7}$ στη συνθήκη προτίμησης του τελεστή λ_5. Επομένως, θα μπορούσαμε να αποφύγουμε να εκτελέσουμε τον τελεστή $\sigma_{rating>7}$ ξεχωριστά στον πίνακα σκορ.

- Έστω ότι έχουν ήδη εκτελεστεί οι δύο πρώτες συνθήκες σύζευξης και έστω T_{S_i} ο ενδιάμεσος πίνακας σκορ που περιέχει τις εγγραφές που έχουν παραχθεί μαζί με τα αντίστοιχα σκορ και βαθμούς εμπιστοσύνης. Από αυτό το συγκεκριμένο σημείο και έπειτα μέχρι το σημείο εκτέλεσης του τελεστή λ_3, δεν είναι απαραίτητο να εκτελέσουμε τους ενδιάμεσους τελεστές στον πίνακα σκορ. Αντιθέτως, μπορούμε απλά να μεταφέρουμε τον πίνακα T_{S_i} παραπάνω στο πλάνο εκτέλεσης και να εκτελέσουμε σε ένα βήμα όλους τους τελεστές την εκτέλεση των οποίων αναβάλαμε.

- Οι τελεστές προτίμησης λ_8 και λ_4 μπορούν να εκτελεστούν σε ένα βήμα, αποφεύγοντας έτσι την εγγραφή ενός ενδιάμεσου πίνακα σκορ στο δίσκο.

Βασικός Αλγόριθμος. Μια πρώτη προσέγγιση επεξεργασίας ενός επεκτεταμένου πλάνου εκτέλεσης θα ήταν να εκτελέσουμε μία postorder διάσχιση του πλάνου εκτέλεσης και να εφαρμόσουμε τον κάθε τελεστή ακολουθώντας τη σειρά εκτέλεσης που ορίζει το πλάνο. Θα αναφερόμαστε σε αυτό τον αλγόριθμο ως *Bottom-Up (BU)*. Ο

αλγόριθμος BU διασχίζει το πλάνο που επιλέγει ο βελτιστοποιητής ερωτημάτων και εκτελεί κάθε τελεστή ξεχωριστά. Για κάθε ενδιάμεσο κόμβο του πλάνου εκτέλεσης n_i, ο αλγόριθμος BU συγκρατεί τα αποτελέσματα της εκτέλεσης του συγκεκριμένου τελεστή σε ένα ζεύγος προσωρινών πινάκων T_{B_i} και T_{S_i}. Ο αλγόριθμος BU τερματίζει όταν εκτελεστεί και ο τελεστής που βρίσκεται τοποθετημένος στη ρίζα του πλάνου εκτέλεσης. Τα τελικά αποτελέσματα του ερωτήματος προκύπτουν εκτελώντας ένα join μεταξύ των πινάκων T_{B_r} και T_{S_r}.

Αλγόριθμος GBU. Είναι προφανές ότι ο αλγόριθμος BU δεν είναι αποδοτικός καθώς χρειάζεται να εγγράψει στο δίσκο όλους τους ενδιάμεσους πίνακες που παράγονται κατά την εκτέλεση του ερωτήματος. Συνεπώς, με βάση τις παρατηρήσεις που κάναμε παραπάνω, προτείνουμε ένα πιο αποδοτικό αλγόριθμο, ονόματι *Group Bottom-Up (GBU)* ο οποίος: (α) επιχειρεί να αποφύγει κάποιες εγγραφές στο δίσκο συνδυάζοντας τελεστές, και (β) χρησιμοποιεί το υποσύστημα εκτέλεσης ερωτημάτων της βάσης δεδομένων για επιμέρους τμήματα της εκτέλεσης ενός ερωτήματος. Συνδυάζοντας κάποιους τελεστές, επιτυγχάνουμε να αποφύγουμε την εγγραφή στο δίσκο όλων των ενδιάμεσων πινάκων. Παράλληλα, συνδυάζοντας για παράδειγμα δύο ή περισσότερους τελεστές προτίμησης, η αντίστοιχη αποθηκευμένη διαδικασία (stored procedure) η οποία υλοποιεί τον τελεστή προτίμησης καλείται μόνο μία φορά. Επίσης, μεταβιβάζοντας κάποια τμήματα του ερωτήματος στη βάση δεδομένων, εκμεταλλευόμαστε κάποιες βελτιστοποιήσεις που παρέχονται από τη βάση, όπως αποδοτικές μεθόδους πρόσβασης στα δεδομένα, αποδοτική υλοποίηση τελεστών και διασωλήνωση (pipelining) μεταξύ τελεστών. Μένοντας πιστοί στον αρχικό μας στόχο, δηλαδή την ελαχιστοποίηση του αριθμού εγγραφών στο δίσκο, παρακάτω παραθέτουμε ένα σύνολο κανόνων που ο αλγόριθμος GBU εξετάζει με σκοπό να αποφασίσει αν ένας τελεστής μπορεί να συνδυαστεί με επόμενους τελεστές:

- Η εκτέλεση ενός τελεστή επιλογής ή ενός τελεστή προβολής μπορεί να αναβληθεί.
- Διαδοχικοί τελεστές προτίμησης μπορούν να εκτελεστούν σε ένα βήμα.
- Η εκτέλεση ενός τελεστή σύζευξης μπορεί να αναβληθεί εφόσον τουλάχιστον ένας από τους δύο αρχικούς πίνακες είναι κενός.

Για να γίνει περισσότερο κατανοητός ο τελευταίος κανόνας, επανερχόμενοι στο Σχήμα 3.11, η εκτέλεση του $GENRES \bowtie MOVIES \bowtie RATINGS$ δεν μπορεί να συνδυαστεί με επόμενους τελεστές, αφού σε αυτό το σημείο της εκτέλεσης και οι δύο πίνακες σκορ που δίνονται ως είσοδος περιέχουν εγγραφές. Αντιθέτως, οι επόμενοι τελεστές σύζευξης μπορούν να συνδυαστούν αφού για κάθε έναν από αυτούς μόνο το ένα υπόδεντρο αποτελείται από μη κενό πίνακα σκορ.

Ο αλγόριθμος GBU ακολουθεί μια postorder διάσχιση του επεκτεταμένου πλάνου εκτέλεσης. Κατά τη διάσχιση, αντί να εκτελέσει κατευθείαν κάθε τελεστή n_i, ο αλγόριθμος GBU εξετάζει αν η εκτέλεση του n_i μπορεί να αναβληθεί σύμφωνα με τους κανόνες που παραθέσαμε παραπάνω. Για να είναι δυνατή η εκτέλεση τελεστών σε ομάδες, ο αλγόριθμος GBU διατηρεί κατά την εκτέλεση έναν κατευθυνόμενο ακυκλικό γράφο (directed acyclic graph (DAG) G ο οποίος περιέχει τους ενδιάμεσους πίνακες

Algorithm 1: Group Bottom-Up

Input: Q_P a query plan, n_r the root of Q_P
Output: $R_Q = \{(t, S_t, C_t) | t \in R_{NP}\}$ where R_{NP} is the result of executing the non-preference part of Q

```
1 begin
2     G := ∅;
3     GBU(n_r, G);
4     extract(G, n_r) → q'; //Remove remaining operators, combine extracted operators into q'
5     execute(q') → T_{B_r}, T_{S_r};
6     R_Q := T_{B_r} ⋈ T_{S_r};
7     return R_Q;
```

που παράγονται κατά την εκτέλεση του πλάνου οι οποίοι δεν έχουν ακόμα χρησιμοποιηθεί, καθώς και όσους τελεστές η εκτέλεση των οποίων έχει αναβληθεί. Όποτε ο αλγόριθμος GBU προσπελαύνει έναν κόμβο n_i η εκτέλεση του οποίου μπορεί να αναβληθεί, τότε ο n_i αντιγράφεται στο G (γραμμές 6, 8, 12 και 22 στη συνάρτηση GBU). Αν ο αλγόριθμος επισκεφτεί έναν τελεστή n_i ο οποίος πρέπει να εκτελεστεί άμεσα, τότε όλοι οι κόμβοι που ανήκουν στο υπόδεντρο του κόμβου n_i διαγράφονται από τον γράφο G και συνδυάζονται σε ένα υποερώτημα (γραμμές 15 και 24-25), το οποίο διαβιβάζεται στο υποσύστημα εκτέλεσης ερωτημάτων της βάσης δεδομένων (γραμμή 16). Τα αποτελέσματα του ερωτήματος εγγράφονται σε ένα ζεύγος προσωρινών πινάκων και στη συνέχεια δίνονται ως είσοδος του τελεστή n_i (γραμμές 17 και 26). Οι ενδιάμεσοι πίνακες που παράγονται εκτελώντας τον τελεστή n_i εισάγονται ξανά στον γράφο G (γραμμές 18 και 27). Όταν ο αλγόριθμος GBU φτάσει στη ρίζα του πλάνου εκτέλεσης, τότε όλοι οι τελεστές που έχουν απομείνει στον γράφο G συνδυάζονται και εκτελούνται σε ένα βήμα (γραμμές 4-5 στον Αλγόριθμο 1). Τέλος, εκτελείται ένα join μεταξύ των πινάκων T_{B_r}, T_{S_r} που παράγονται ως αποτέλεσμα της εκτέλεσης του τελεστή που είναι τοποθετημένος στη ρίζα του πλάνου εκτέλεσης (γραμμή 6) και επιστρέφονται τα τελικά αποτελέσματα (γραμμή 7).

3.5 Πειραματική αξιολόγηση

Σε αυτή την ενότητα, αξιολογούμε πειραματικά τις προτεινόμενες μεθόδους. Στόχος της πειραματικής αξιολόγησης είναι (α) να αξιολογηθεί η επίδραση της αποτίμησης προτιμήσεων στην απόδοση της εκτέλεσης ενός ερωτήματος χρησιμοποιώντας τόσο τις προτεινόμενες μεθόδους εκτέλεσης, όσο και plug-in μεθόδους, και (β) να αξιολογηθούν οι προτεινόμενοι αλγόριθμοι βασισμένης στο κόστος βελτιστοποίησης.

3.5.1 Πειραματική μεθοδολογία

Πειραματικό Περιβάλλον. Όλα τα πειράματα εκτελέστηκαν σε ένα σύστημα με επεξεργαστή 2.0 GHz Intel Xeon CPU, 16 GB RAM με λειτουργικό Debian Linux 4.4.5 και εγκατεστημένη τη βάση δεδομένων PostgreSQL 9.1.1. Οι τιμές των παραμέτρων `shared_buffers` (shared memory buffer size) και `work_mem` (internal memory for sorting and hashing) της PostgreSQL ρυθμίστηκαν σε 4 GB και 1 GB αντιστοίχως. Το σύστημα PrefDB λειτουργεί ως επέκταση της PostgreSQL. Ο τελεστής προτίμησης και οι επεκτεταμένοι τελεστές έχουν υλοποιηθεί ως user defined functions - UDFs κάνοντας χρήση της γλώσσας pgSQL. Τα υπόλοιπα υποσυστήματα (διαχειριστής προτι-

Function GBU

Input: n_i node in Q_P, G a DAG containing all operators that can be combined and intermediate tables produced during query execution
Output: G

```
1  begin
2      if n_i is null then
3          exit;
4      GBU(n_i.left, G); GBU(n_i.right, G);
5      if n_i is a relation (leaf node) R_i then
6          insert R_i into G;
7      else if n_i is a project or select operator then
8          insert n_i into G;
9      else if n_i is a prefer operator then
10         let n_p be the parent node of n_i;
11         if n_p is a prefer operator then
12             insert n_i into G;
13         else
14             let n_j be the child node of n_i;
15             extract(G, n_j) → q'; //Remove all not executed operators in the subtree of n_j,
               combine extracted operators into q'
16             execute(q') → T_Bj, T_Sj;
17             evaluate(n_i, T_Bj, T_Sj) → T_Si;
18             insert T_Bj into G; insert T_Si into G;
19     else if n_i is a join or set operator then
20         let n_j, n_k be the children nodes of n_i;
21         if ((n_i is a join) and (|T_Sj| = 0 or |T_Sk| = 0)) then
22             insert n_i into G;
23         else
24             extract(G, n_j) → q_j; //Remove all not executed operators from both subtrees
25             extract(G, n_k) → q_k;
26             evaluate(n_i, q_j, q_k) → T_Bi, T_Si; //combine all extracted operators with n_i, execute
               query
27             insert T_Bi into G; insert T_Si into G;
```

μήσεων, αναλυτής, βελτιστοποιητής και μηχανή εκτέλεσης ερωτημάτων) υλοποιήθηκαν σε Java.

Σύνολα Δεδομένων. Στα πειράματα χρησιμοποιήθηκαν δύο πραγματικά σύνολα δεδομένων. Η βάση δεδομένων IMDB[4] (Πίνακας 3.1) αποτελεί στιγμιότυπο της αντίστοιχης βάσης δεδομένων ταινιών το οποίο ανακτήθηκε τον Μάρτιο του 2010. Η βάση δεδομένων DBLP[5] (Πίνακας 3.4) αποτελεί στιγμιότυπο της βάσης δεδομένων επιστημονικών δημοσιεύσεων το οποίο ανακτήθηκε τον Ιούνιο 2011.

Ερωτήματα. Χρησιμοποιήθηκαν 8 βασικά ερωτήματα των οποίων οι λεπτομέρειες είναι διαθέσιμες στο σύνδεσμο [1]. Προσπαθήσαμε να επιλέξουμε ερωτήματα με διαφοροποιημένα χαρακτηριστικά σε σχέση με το πλήθος αποτελεσμάτων, τον αριθμό joins, το πλήθος προτιμήσεων, κλπ. με σκοπό να εξετάσουμε την συμπεριφορά των μεθόδων σε διαφορετικά σενάρια χρήσης. Ο Πίνακας 3.5(α′) περιγράφει τις πιο σημαντικές ιδιότητες των ερωτημάτων που εξετάστηκαν. Με $|Q|$ συμβολίζουμε το πλήθος αποτελεσμάτων του ερωτήματος χωρίς προτιμήσεις, $|R|$ είναι το πλήθος σχέσεων που συμμετέχουν στο ερώτημα και $|P|$ είναι το πλήθος προτιμήσεων. Οι προτιμήσεις είναι ομοιόμορφα κατανεμημένες στις σχέσεις που συμμετέχουν στο ερώτημα. Για το βασικό σενάριο πειραμάτων χρησιμοποιήσαμε μια προτίμηση ανά σχέση. Για παράδειγμα, το ερώτημα IMDB-3 εμπλέκει τρεις προτίμησεις, κάθε μια από τις οποίες σχετίζεται με

[4]http://www.imdb.com
[5]http://dblp.uni-trier.de/

PUBLICATIONS(p_id, title, pub_type), AUTHORS(a_id, name),
PUB_AUTHORS(p_id, a_id), CITATIONS(p1_id, p2_id),
CONFERENCES(p_id, name, year, location),
JOURNALS(p_id, name, year, volume)

Πίνακας 3.4: *Το σχήμα δεδομένων της βάσης DBLP*

Ερώτημα	$\|Q\|$	$\|R\|$	$\|P\|$
IMDB-1	119840	4	3
IMDB-2	159211	3	3
IMDB-3	194448	4	3
IMDB-4	263829	3	3
IMDB-5	3584662	5	4
DBLP-1	11290	4	2
DBLP-2	36269	3	2
DBLP-3	1401127	2	1

(α′) Βασικές παράμετροι ερωτημάτων

Αλγόριθμος	**Ακρωνύμιο**
Non-Preference Query	SQL
Pure Plug-In	PL
Hybrid Plug-In	FP
GBU-Greedy	GBU-G
GBU-Dynamic	GBU-D
GBU-Exhaustive	GBU-E

(β′) Αλγόριθμοι υπό εξέταση

Πίνακας 3.5: *Παράμετροι πειραμάτων*

μία σχέση, ενώ μια σχέση δεν συνδέεται με καμιά προτίμηση.

Παράμετροι. Έχουμε διεξαγάγει ανάλυση ευαισθησίας σε σχέση με τις παρακάτω παραμέτρους: το πλήθος αποτελεσμάτων $|Q|$, τον αριθμό σχέσεων που συμμετέχουν στο ερώτημα $|R|$, το μέγεθος των σχέσεων N, το πλήθος των προτιμήσεων ανά σχέση $|P|$, την επιλεκτικότητα των προτιμήσεων PS καθώς και την κατανομή των προτιμήσεων στους πίνακες του ερωτήματος PD. Σε κάθε πείραμα, για κάθε βασικό ερώτημα μεταβάλλουμε μια μόνο παράμετρο και κρατάμε όλες τις υπόλοιπες σταθερές στις προκαθορισμένες τους τιμές. Για παράδειγμα, στο πείραμα όπου μεταβάλλουμε το πλήθος προτιμήσεων ανά σχέση, κρατάμε τις υπόλοιπες συνθήκες του ερωτήματος, την κατανομή προτιμήσεων στις σχέσεις κλπ. αμετάβλητες. Θα περιγράψουμε αναλυτικά πώς τροποποιούμε κατάλληλα τα ερωτήματα σε κάθε πείραμα. Το σύνολο ερωτημάτων που χρησιμοποιήθηκε στα πειράματα περιλαμβάνει συνολικά 121 ερωτήματα. Παρακάτω παρουσιάζουμε τα πειραματικά αποτελέσματα για ένα υποσύνολο ερωτημάτων. Επιπλέον διαγράμματα είναι διαθέσιμα στο σύνδεσμο [8].

Αλγόριθμοι. Έχουμε συνδυάσει τις δύο μεθόδους βελτιστοποίησης που παρουσιάστηκαν στην ενότητα 3.4.2.2 με τον αλγόριθμο εκτέλεσης ερωτημάτων GBU. Θα χρησιμοποιήσουμε τα ακρωνύμια GBU-D και GBU-G για να συμβολίσουμε τον αλγόριθμο δυναμικού προγραμματισμού και τον άπληστο αλγόριθμο αντίστοιχα. Συγκρίναμε την απόδοση των παρακάτω μεθόδων (α) με δύο παραλλαγές των μεθόδων plug-in που υλοποιήσαμε, και (β) με τον αλγόριθμο GBU αν εκτελεστεί στο βέλτιστο πλάνο όπως αυτό προκύπτει με μια εξαντλητική αναζήτηση στον χώρο των πιθανών πλάνων εκτέλεσης. Θα αναφερόμαστε σε αυτό τον αλγόριθμο ως GBU-E. Επίσης σε ορισμένες περιπτώσεις χρησιμοποιούμε το ακρωνύμιο GBU-* για να αναφερόμαστε συνολικά σε όλες τις παραλλαγές που εξετάσαμε και είναι βασισμένες στον αλγόριθμο εκτέλεσης GBU.

Ο πρώτος αλγόριθμος plug-in (θα αναφερόμαστε σε αυτόν ως PL) μεταφράζει τις

Αλγόριθμος	Μέσος χρόνος βελτιστοποίησης	(%) Βελτίωση έναντι PL
GBU-G	123 msecs	41.93%
GBU-D	234 msecs	44.44%
GBU-E	403 msecs	51.40%

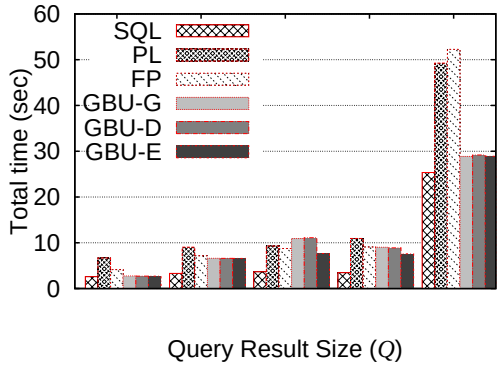
Πίνακας 3.6: *Συνολικά πειραματικά αποτελέσματα*

προτιμήσεις σε υποερωτήματα και στη συνέχεια τα διαβιβάζει στο υποσύστημα εκτέλεσης ερωτημάτων της βάσης δεδομένων. Ο δεύτερος αλγόριθμος (θα αναφερόμαστε σε αυτόν ως FP) ακολουθεί επίσης μια plug-in στρατηγική αλλά βασίζεται στο επεκτεταμένο μοντέλο δεδομένων και σχεσιακή άλγεβρα. Ακολουθώντας αυτή την προσέγγιση, αρχικά το ερώτημα χωρίς τις προτιμήσεις εκτελείται και στη συνέχεια κάθε προτίμηση εφαρμόζεται χρησιμοποιώντας την υλοποίηση του τελεστή προτίμησης που ακολουθεί το σύστημα PrefDB. Περισσότερες λεπτομέρειες για τους αλγορίθμους plug-in που εξετάστηκαν είναι διαθέσιμες στο σύνδεσμο [8].

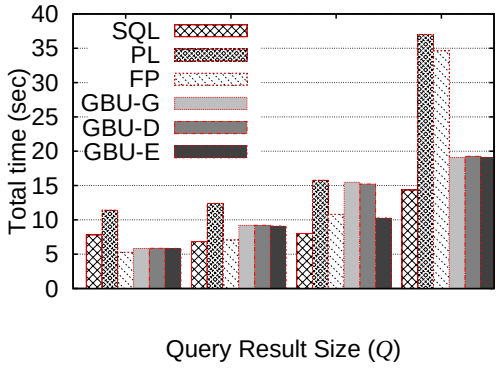
Η υλοποίηση του αλγορίθμου GBU-E έγινε ως εξής. Αρχικά εκτελέσαμε τον αλγόριθμο GBU για κάθε πιθανό πλάνο εκτέλεσης. Στη συνέχεια υπολογίσαμε το συνολικό κόστος επεξεργασίας αθροίζοντας το κόστος απαρίθμησης όλων των πλάνων μαζί με το βέλτιστο κόστος εκτέλεσης που βρέθηκε. Σημειώνουμε επίσης ότι δεν συμπεριλάβαμε τον βασικό αλγόριθμο BU στην πειραματική αξιολόγηση, καθώς ο αλγόριθμος GBU αποτελεί βελτιωμένη εκδοχή του BU. Τέλος, για να φανεί καλύτερα η επιπλέον επίπτωση στο κόστος που προκαλείται ως συνέπεια της αποτίμησης προτιμήσεων, σε όλα τα διαγράμματα που θα ακολουθήσουν έχουν συμπεριληφθεί ο χρόνος που απαιτείται για την εκτέλεση του ερωτήματος χωρίς προτιμήσεις, για τον οποίο θα χρησιμοποιήσουμε το ακρωνύμιο SQL. Ο Πίνακας 3.5(β′) συνοψίζει τα ακρωνύμια κάθε αλγορίθμου.

Μετρήσεις. Κάθε πείραμα μετράει τον συνολικό χρόνο επεξεργασίας που απαιτεί κάθε αλγόριθμος που εξετάστηκε. Ο συνολικός χρόνος αποτελείται από (α) τον χρόνο που απαιτήθηκε για τη βελτιστοποίηση του ερωτήματος, και (β) τον χρόνο εκτέλεσης του ερωτήματος. Οι χρόνοι εκτέλεσης μετρήθηκαν με τη βοήθεια της εντολής `EXPLAIN ANALYZE` της PostgreSQL. Για να πετύχουμε όσο το δυνατόν μεγαλύτερη ακρίβεια στις μετρήσεις, πριν εκτελέσουμε κάθε ερώτημα αδειάσαμε την μνήμη cache του συστήματος. Σημειώνουμε ότι οι αλγόριθμοι SQL, PL και FP δεν έχουν φάση βελτιστοποίησης.

Με την εξαίρεση των τροποποιημένων ερωτημάτων που θα εξετάσουμε στο πείραμα όπου μεταβάλλουμε το πλήθος προτιμήσεων $|P|$, στα υπόλοιπα πειράματα, οι μετρούμενοι χρόνοι βελτιστοποίησης είναι αμελητέοι σε σχέση με τους αντίστοιχους χρόνους εκτέλεσης των ερωτημάτων. Ο Πίνακας 3.6 δείχνει τους μέσους χρόνους βελτιστοποίησης που αντιστοιχούν στον άπληστο αλγόριθμο, τον αλγόριθμο δυναμικού προγραμματισμού και την εξαντλητική αναζήτηση όπως μετρήθηκαν σε όλο το σύνολο των ερωτημάτων που εξετάστηκαν στα πειράματά μας (χωρίς να συμπεριλάβουμε τα πειράματα όπου μεταβάλλουμε το πλήθος προτιμήσεων $|P|$). Όπως δείχνει ο Πίνακας, οι μέσοι χρόνοι βελτιστοποίησης είναι 123, 234 και 403 msecs αντιστοίχως. Ως εκ τούτου, για όλα τα πειράματα εκτός αυτών όπου μεταβάλλουμε το πλήθος προτιμήσεων, τα αντίστοιχα διαγράμματα απεικονίζουν τους συνολικούς χρόνους εκτέλεσης. Για το πείραμα όπου μεταβάλλουμε το πλήθος προτιμήσεων $|P|$ έχουμε συμπεριλάβει ξεχωριστά διαγράμματα που δείχνουν τους χρόνους βελτιστοποίησης.

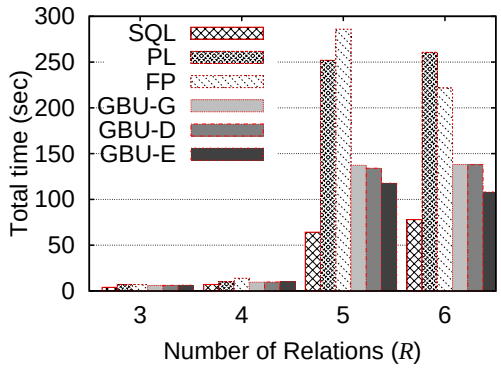


(α′) IMDB-Q

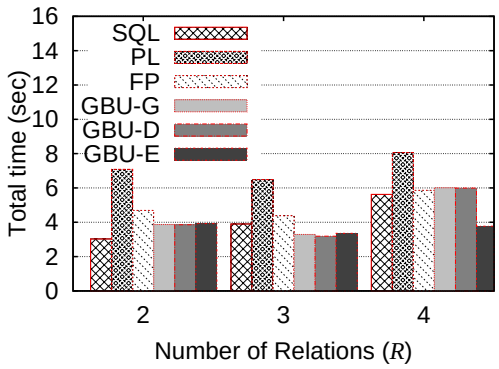


(β′) DBLP-Q

Σχήμα 3.12: *Συνολικός χρόνος εκτέλεσης σε σχέση με το πλήθος αποτελεσμάτων*



(α′) IMDB-R1



(β′) IMDB-R2

Σχήμα 3.13: *Συνολικός χρόνος εκτέλεσης σε σχέση με τον αριθμό πινάκων του ερωτήματος*

3.5.2 Πειραματικά αποτελέσματα

Επίδοση σε σχέση με το πλήθος αποτελεσμάτων. Στο πρώτο πείραμα, μετρήσαμε την επίδοση όλων των μεθόδων σε σχέση με το πλήθος αποτελεσμάτων $|Q|$. Για να απομονώσουμε το πλήθος αποτελεσμάτων ως παράμετρο, κάθε ερώτημα που εκτελέσαμε περιέχει ένα join. Κρατώντας σταθερό το ένα από τα δύο μέρη του join και χρησιμοποιώντας διαφορετικές σχέσεις για το άλλο, προκύπτουν ερωτήματα με διαφορετικό πλήθος αποτελεσμάτων. Όπως δείχνουν τα διαγράμματα 3.12(α′) και 3.12(β′), και οι δύο παραλλαγές του αλγορίθμου GBU που εξετάσαμε εμφανίζουν καλύτερες δυνατότητες κλιμάκωσης συγκρινόμενες με τις μεθόδους plug-in PL και FP σε σχέση με το πλήθος αποτελεσμάτων ενός ερωτήματος. Το συγκεκριμένο αποτέλεσμα αποτελεί ισχυρό κίνητρο για την εφαρμογή μεθόδων εκτέλεσης βασισμένων στο κόστος οι οποίες θα επέλεγαν να εφαρμόσουν τις προτιμήσεις στους επιμέρους αντί για τον παραγόμενο πίνακα.

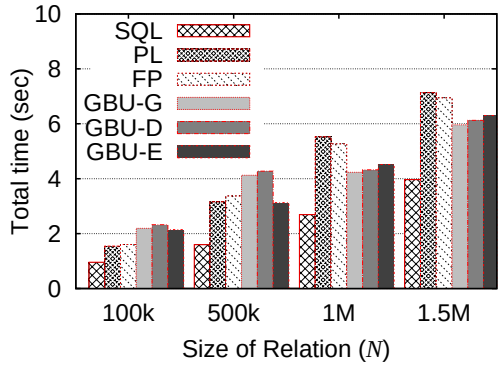
Επίδοση σε σχέση με τον αριθμό πινάκων ενός ερωτήματος. Στη συνέχεια συγκρίνουμε πειραματικά όλες τις μεθόδους καθώς μεταβάλλουμε τον αριθμό πινάκων $|R|$ που συμμετέχουν σε ένα ερώτημα. Για το συγκεκριμένο πείραμα χρησι-

μοποιήσαμε τα εξής δύο ερωτήματα IMDB-R1 με $|R| = 3$ και IMDB-R2 με $|R| = 2$ προσθέτοντας σταδιακά επιπλέον πίνακες έως $|R| = 6$ ($|R| = 4$ αντίστοιχα). Τα διαγράμματα 3.13(α′) και 3.13(β′) απεικονίζουν τους μετρούμενους συνολικούς χρόνους εκτέλεσης σε σχέση με τον αριθμό πινάκων ενός ερωτήματος.

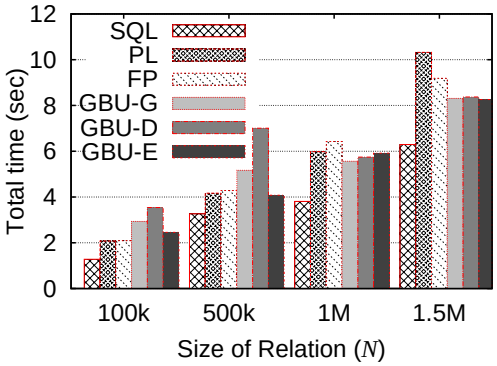
Όπως φαίνεται στα διαγράμματα, τόσο ο GBU-G όσο και ο GBU-D είναι σημαντικά πιο γρήγοροι και εμφανίζουν καλύτερη συμπεριφορά κλιμάκωσης για ερωτήματα με πολλαπλά joins συγκρινόμενοι με τις μεθόδους plug-in που εξετάσαμε. Για να γίνει αυτό πιο κατανοητό ας επιστρέψουμε στο παράδειγμα μας. Μια μέθοδος plug-in θα ανακτήσει αρχικά όλα τα αποτελέσματα που ικανοποιούν τις συνθήκες του ερωτήματος, το οποίο θα έχει ως αποτέλεσμα να παραχθούν εγγραφές όπου η κάθε ταινία εμφανίζεται με πολλά είδη και ηθοποιούς (σχέση 1:πολλά). Στη συνέχεια, όλες οι προτιμήσεις πρέπει να αποτιμηθούν σε αυτό το τεράστιο σύνολο αποτελεσμάτων, κάτι που είναι ιδιαιτέρως ακριβό από πλευράς κόστους. Αντιθέτως, οι αλγόριθμοι GBU-G και GBU-D κάνουν χρήση του μοντέλου κόστους που παρουσιάστηκε στην ενότητα 3.4.2.2, διασφαλίζοντας ότι οι προτιμήσεις θα αποτιμηθούν σε τέτοια θέση του πλάνου εκτέλεσης ώστε να ελαχιστοποιείται το κόστος αποτίμησης των προτιμήσεων. Για παράδειγμα, οι προτιμήσεις p_4 και p_8, οι οποίες εμπλέκουν τη σχέση $GENRES$ θα αποτιμηθούν μετά την εκτέλεση του join $MOVIES \bowtie GENRES$.

Επίδοση σε σχέση με το μέγεθος των πινάκων ενός ερωτήματος. Έπειτα εξετάζουμε το κόστος επεξεργασίας όλων των αλγορίθμων καθώς μεταβάλλεται το μέγεθος των πινάκων που συμμετέχουν σε ένα ερώτημα. Για το σκοπό αυτό κατασκευάσαμε υποσύνολα των αρχικών πινάκων ως εξής. Επιλέξαμε ως βασικούς πίνακες για τις βάσεις δεδομένων IMDB και DBLP τους πίνακες $MOVIES$ και $PUBLICATIONS$ αντίστοιχα. Στη συνέχεια διαλέξαμε με τυχαίο τρόπο 1M εγγραφές από τον πίνακα $MOVIES$, έπειτα 500K από αυτές κ.ο.κ. Επίσης, από τους υπόλοιπους πίνακες κρατήσαμε μόνο τις εγγραφές που έχουν join με τις επιλεγμένες. Τα διαγράμματα 3.14(α′) - 3.14(δ′) απεικονίζουν το κόστος επεξεργασίας όλων των αλγορίθμων για τα ερωτήματα IMDB-2, IMDB-3, DBLP-1 και DBLP-2. Ο άξονας x αναπαριστά το μέγεθος του βασικού πίνακα.

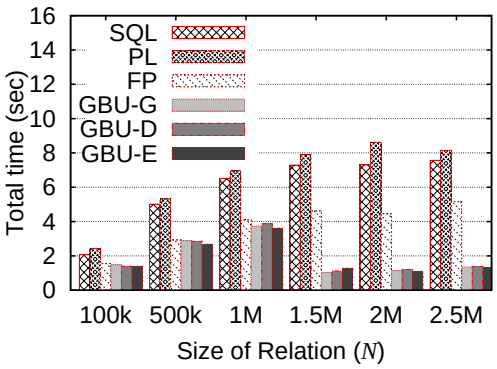
Όπως μπορούμε να παρατηρήσουμε, για ερωτήματα όπου το μέγεθος των πινάκων είναι σχετικά μικρό, οι μέθοδοι plug-in έχουν παρόμοια επίδοση με αυτή των παραλλαγών του αλγορίθμου GBU-*. Αντιθέτως, καθώς το μέγεθος των πινάκων αυξάνεται, το κόστος αποτίμησης προτιμήσεων γίνεται υπολογίσιμος παράγοντας στη διαμόρφωση του συνολικού κόστους. Σε αυτή την περίπτωση οι αλγόριθμοι εκτέλεσης που λαμβάνουν υπόψη το κόστος εκτέλεσης γίνονται αρκετά πιο αποδοτικοί από τις μεθόδους plug-in. Αξίζει να σημειωθεί επίσης ότι σε ορισμένα διαγράμματα ο συνολικός χρόνος που απαιτείται για την εκτέλεση του αλγόριθμου εξαντλητικής αναζήτησης GBU-E είναι αρκετά μεγαλύτερος από αυτόν που απαιτούν οι προσεγγιστικοί αλγόριθμοι GBU-G και GBU-D εξαιτίας του πολύ υψηλού κόστους που συνεπάγεται η απαρίθμηση και εκτίμηση κόστους όλων των πιθανών πλάνων εκτέλεσης. Το γεγονός αυτό καταδεικνύει ότι ακόμα και για σχετικά απλά ερωτήματα όπου μια εξαντλητική αναζήτηση στο χώρο των πιθανών πλάνων εκτέλεσης είναι εφικτή, είναι συχνά πιο αποδοτικό να επιλεγεί ένα λιγότερο βέλτιστο πλάνο ελαχιστοποιώντας ταυτόχρονα τον χρόνο που απαιτείται για την βελτιστοποίηση του ερωτήματος. Τα πειράματά μας δείχνουν ότι και οι δύο προτει-



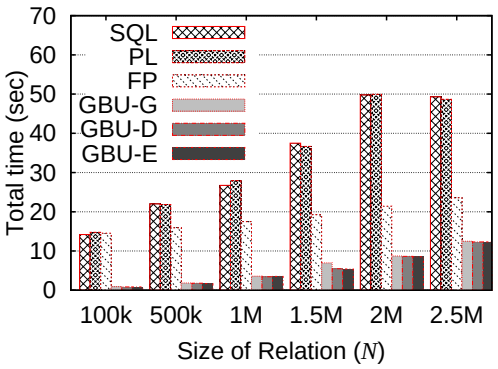
(α′) IMDB-2



(β′) IMDB-3



(γ′) DBLP-1



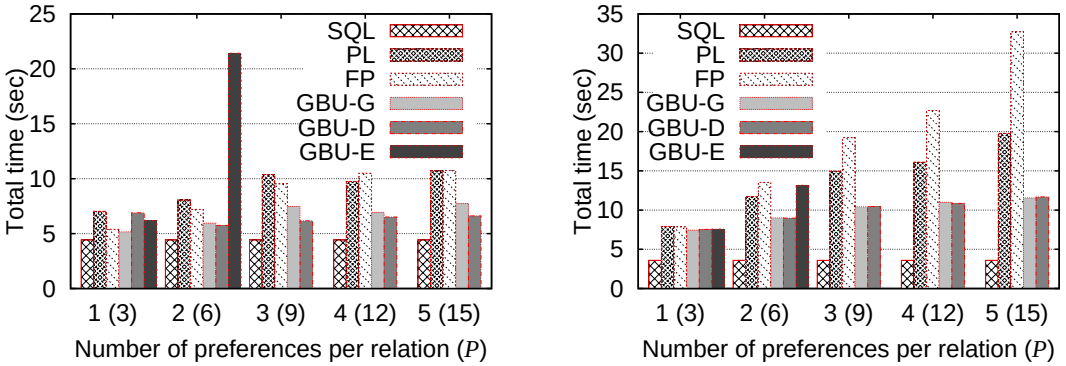
(δ′) DBLP-2

Σχήμα 3.14: *Συνολικός χρόνος εκτέλεσης σε σχέση με το μέγεθος των πινάκων*

νόμενοι αλγόριθμοι βελτιστοποίησης επιτυγχάνουν στις περισσότερες περιπτώσεις να επιλέξουν αρκετά αποδοτικά πλάνα εκτέλεσης.

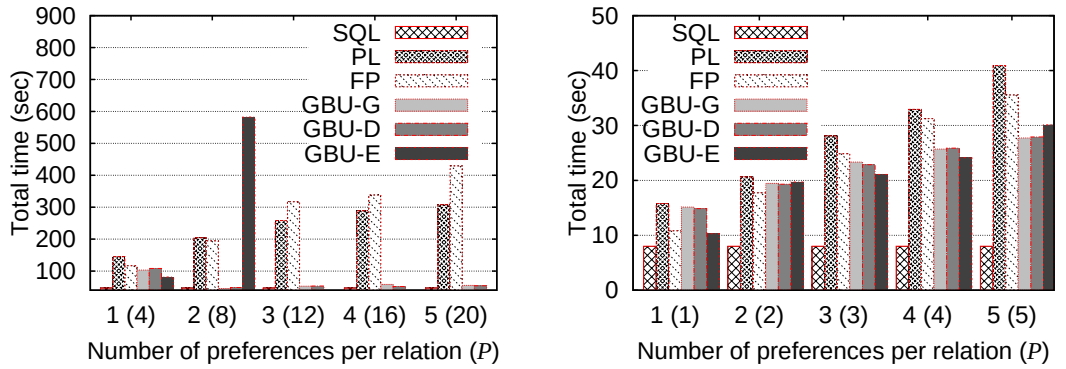
Επίδοση σε σχέση με το πλήθος προτιμήσεων. Σε αυτό το πείραμα, εξετάζουμε την επίδοση όλων των μεθόδων καθώς μεταβάλλεται το πλήθος προτιμήσεων $|P|$ που συνδέονται με το ερώτημα. Πιο συγκεκριμένα μεταβάλλουμε το πλήθος προτιμήσεων ως εξής. Για κάθε σχέση που έχει σχετιζόμενες προτίμησεις, μεταβάλλουμε τον αριθμό των σχετιζόμενων προτιμήσεων από 1 έως 5. Τα διαγράμματα 3.15(α′) - 3.15(δ′) απεικονίζουν τους συνολικούς χρόνους εκτέλεσης για τα ερωτήματα IMDB-1, IMDB-4, IMDB-5 και DBLP-3. Τα διαγράμματα 3.16(α′) - 3.16(δ′) δείχνουν τους αντίστοιχους χρόνους βελτιστοποίησης. Οι αριθμοί εντός παρενθέσεων αντιπροσωπεύουν τον συνολικό αριθμό προτιμήσεων στο ερώτημα.

Όπως φαίνεται στα διαγράμματα 3.15(α′) - 3.15(δ′), οι αλγόριθμοι GBU-G και GBU-D είναι οι πιο αποδοτικοί και παρουσιάζουν καλύτερη συμπεριφορά κλιμάκωσης καθώς αυξάνεται ο αριθμός των προτιμήσεων σε ένα ερώτημα. Το κέρδος στην απόδοση εξαρτάται από τον συνολικό αριθμό προτιμήσεων του ερωτήματος. Όσο περισσότεροι οι τελεστές προτίμησης που χρειάζεται να αποτιμηθούν τόσο μεγαλύτερο το κέρδος στον χρόνο εκτέλεσης των μεθόδων που είναι βασισμένες στον αλγόριθμο GBU-* σε σύγκριση με τις μεθόδους plug-in. Αυτό φαίνεται πιο ξεκάθαρα στο διάγραμμα 3.15(γ′), το οποίο το έχουμε συμπεριλάβει με σκοπό να δείξουμε τη συμπεριφορά τον αλγορίθμων



(α′) IMDB-1

(β′) IMDB-4



(γ′) IMDB-5

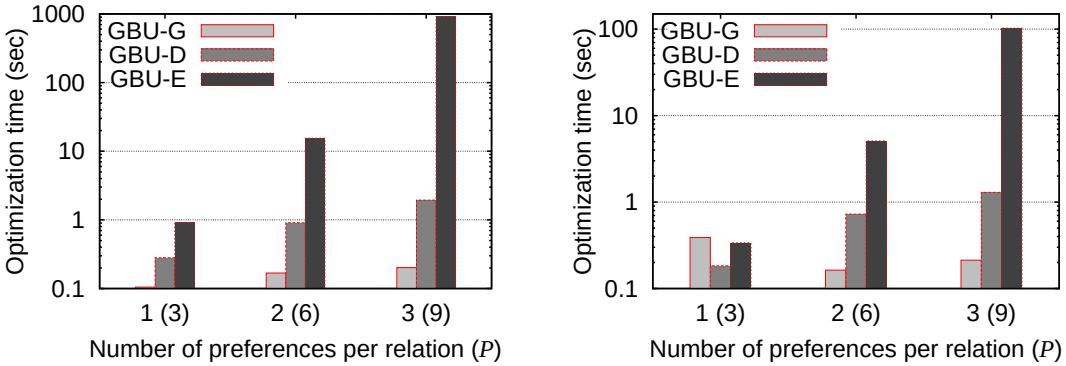
(δ′) DBLP-3

Σχήμα 3.15: *Συνολικός χρόνος εκτέλεσης σε σχέση με το πλήθος προτιμήσεων*

για ερωτήματα που περιέχουν μεγάλο πλήθος προτιμήσεων. Αξίζει να σημειωθεί ότι σε κάποια διαγράμματα έχουμε παραλείψει τους χρόνους εκτέλεσης του αλγόριθμου GBU-E, καθώς η εκτέλεσή του δεν είχε ολοκληρωθεί σε εύλογο χρονικό διάστημα.

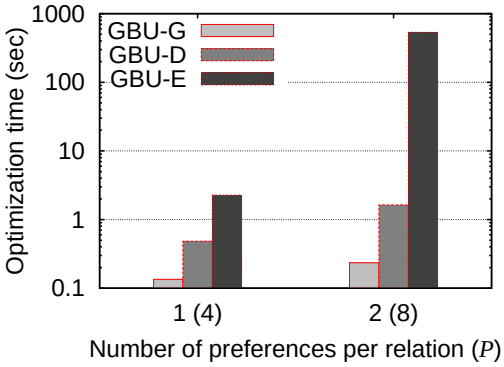
Σε σχέση με τους χρόνους βελτιστοποίησης, τόσο ο αλγόριθμος δυναμικού προγραμματισμού όσο και ο αλγόριθμος εξαντλητικής αναζήτησης είναι στην πράξη μη εφαρμόσιμοι για ερωτήματα με μεγάλο πλήθος προτιμήσεων. Σύμφωνα με τα πειράματά μας, η εξαντλητική αναζήτηση δεν μπορεί να εφαρμοστεί για περισσότερες από 5 προτιμήσεις, ενώ για πάνω από 7 προτιμήσεις ο αλγόριθμος δυναμικού προγραμματισμού συνεπάγεται υψηλότερους συνολικούς χρόνους σε σχέση με τον άπληστο αλγόριθμο. Απομονώνοντας τους χρόνους εκτέλεσης, οι δύο αλγόριθμοι GBU-G και GBU-D έχουν παραπλήσια επίδοση με αυτή ενός αλγορίθμου που επιλέγει το βέλτιστο πλάνο εκτέλεσης.

Επίδοση σε σχέση με την επιλεκτικότητα των προτιμήσεων. Στη συνέχεια εξετάζουμε την επίδραση της επιλεκτικότητας των προτιμήσεων στην επίδοση όλων των μεθόδων. Για το πείραμα αυτό χρησιμοποιήσαμε τα ερωτήματα IMDB-3, IMDB-4, IMDB-5 και DBLP-3, όπου μεταβάλαμε τη συνθήκη προτίμησης από περισσότερο σε λιγότερο επιλεκτική. Τα διαγράμματα 3.17(α′)-3.17(δ′) δείχνουν τα αποτελέσματα κάθε πειράματος. Όπως αναμενόταν, η επίδοση όλων των μεθόδων χειροτερεύει όταν οι προτιμήσεις επηρεάζουν περισσότερες εγγραφές. Παρολαυτά, οι παραλλαγές του

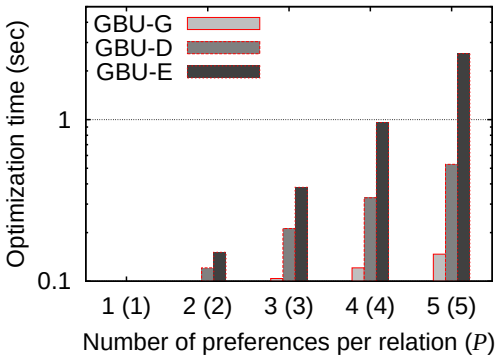


(α′) IMDB-1

(β′) IMDB-4



(γ′) IMDB-5



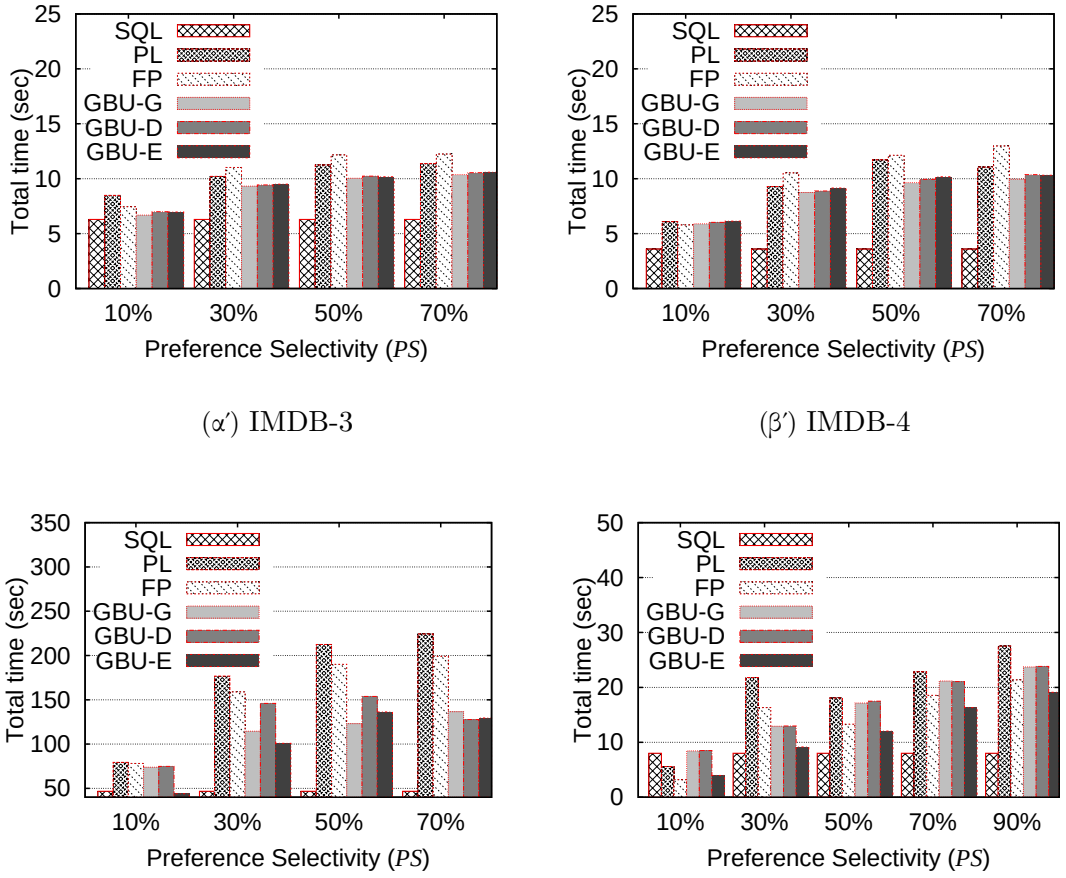
(δ′) DBLP-3

Σχήμα 3.16: *Χρόνος βελτιστοποίησης ερωτήματος σε σχέση με το πλήθος προτιμήσεων*

αλγορίθμου GBU είναι στις περισσότερες περιπτώσεις οι πιο αποδοτικές, ανεξαρτήτως επιλεκτικότητας.

Επίδοση σε σχέση με την κατανομή των προτιμήσεων. Στο τελευταίο πείραμα εξετάσαμε το πώς επηρεάζεται το κόστος επεξεργασίας των ερωτημάτων από την κατανομή των προτιμήσεων στους πίνακες του ερωτήματος. Ουσιαστικά, τροποποιούμε κατάλληλα τα ερωτήματα έτσι ώστε οι προτιμήσεις να εφαρμόζονται σε έναν, περισσότερους ή όλους τους πίνακες ενός ερωτήματος. Συγκεκριμένα, για τα ερωτήματα IMDB-1 - IMDB-4 κρατήσαμε το κομμάτι που δεν σχετίζεται με προτιμήσεις σταθερό και μεταβάλαμε τον αριθμό των πινάκων που έχουν προτιμήσεις προσθέτοντας σταδιακά μία προτίμηση σε έναν νέο πίνακα. Τα διαγράμματα 3.18(α′)-3.18(δ′) δείχνουν τους μετρούμενους χρόνους επεξεργασίας καθώς μεταβάλλεται ο αριθμός πινάκων που σχετίζονται με προτιμήσεις. Όπως προκύπτει, οι χρόνοι επεξεργασίας αυξάνονται καθώς αυξάνεται το πλήθος πινάκων που επηρεάζονται από προτιμήσεις εξαιτίας του κόστους αποτίμησης των προτιμήσεων. Γενικά, οι παραλλαγές του αλγόριθμου GBU είναι οι πιο αποδοτικές μεταξύ των εξεταζόμενων μεθόδων.

Σύνοψη συμπερασμάτων. Ο Πίνακας 3.6 συνοψίζει ορισμένα συνολικά στατιστικά που ελήφθησαν από το σύνολο των πειραμάτων που εκτελέστηκαν. Μεταξύ των δύο μεθόδων plug-in που εξετάσαμε, τα πειράματά μας δείχνουν ότι η μέθοδος FP είναι γενικά πιο αποδοτική από τη μέθοδο PL (μετρήσαμε μια μέση μείωση χρόνου εκτέλε-



(α′) IMDB-3

(β′) IMDB-4

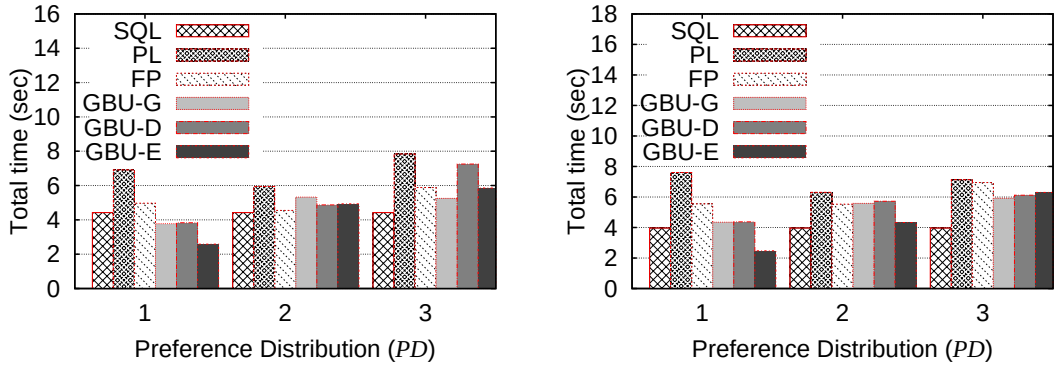
(γ′) IMDB-5

(δ′) DBLP-3

Σχήμα 3.17: *Συνολικός χρόνος εκτέλεσης σε σχέση με την επιλεκτικότητα των προτιμήσεων*

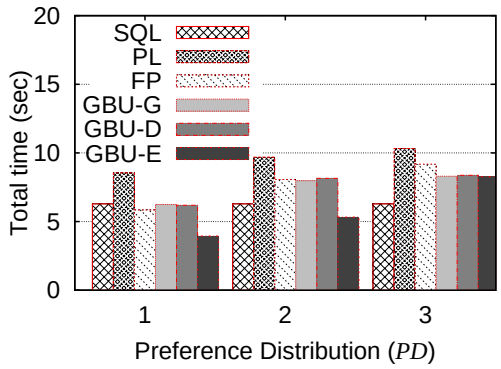
σης της τάξης του 14.63%), γεγονός που αποτελεί μια ένδειξη ότι ακολουθώντας την υλοποίηση του επεκτεταμένου σχεσιακού μοντέλου προκύπτει ελαφρά βελτίωση στην επίδοση. Επίσης, για σχετικά απλά ερωτήματα (διάγραμμα 3.12(α′)), μικρό πλήθος αποτελεσμάτων (διάγραμμα 3.13(α′)) ή μικρό μέγεθος πινάκων (διαγράμματα 3.14(α′)-3.14(δ′)), οι αλγόριθμοι plug-in έχουν παρόμοια συμπεριφορά με τους αλγορίθμους που είναι βασισμένοι στον GBU. Όμως, δεν παρουσιάζουν καλή συμπεριφορά κλιμάκωσης για ερωτήματα με πολλαπλά joins, μεγαλύτερο πλήθος αποτελεσμάτων ή προτιμήσεων.

Αντιθέτως, ο προτεινόμενος αλγόριθμος GBU είναι αρκετά πιο αποδοτικός και παρουσιάζει καλύτερη κλιμάκωση για τέτοιου τύπου ερωτήματα. Όπως δείχνει ο Πίνακας 3.6, όλες οι μέθοδοι που είναι βασισμένες στον αλγόριθμο GBU επιτυγχάνουν βελτίωση μεγαλύτερη του 41% σε σχέση με μια μέθοδο plug-in. Επιπλέον, τα πειράματά μας δείχνουν ότι ο αλγόριθμος GBU δεν επηρεάζεται από την επιλεκτικότητα και την κατανομή των προτιμήσεων στους πίνακες ενός ερωτήματος. Εδώ πρέπει να τονίσουμε ότι τα βασικά ερωτήματα και οι προκαθορισμένες τιμές παραμέτρων που χρησιμοποιήσαμε στην πειραματική μας αξιολόγηση είναι μάλλον ευνοϊκές για τις μεθόδους plug-in αφού θεωρήσαμε μόνο μια προτίμηση ορισμένη σε κάθε σχέση. Στην πράξη, ένα σύστημα συνήθως έχει συλλέξει πλήθος προτιμήσεων για κάθε χρήστη, για παράδειγμα παρακολουθώντας την συμπεριφορά του ή με βάση άλλες τεχνικές εξόρυξης

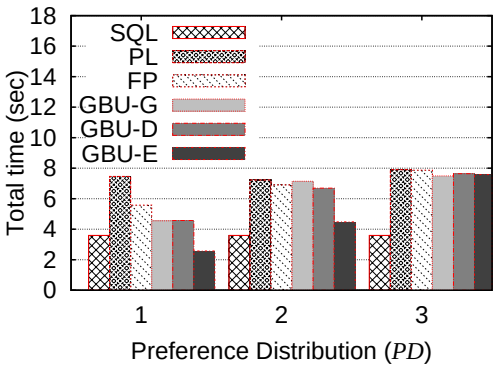


(α′) IMDB-1

(β′) IMDB-2



(γ′) IMDB-3



(δ′) IMDB-4

Σχήμα 3.18: *Συνολικός χρόνος εκτέλεσης σε σχέση με την κατανομή προτιμήσεων*

γνώσης. Στην περίπτωση όπου υπάρχει μεγαλύτερος αριθμός προτιμήσεων που πρέπει να αποτιμηθεί, το κέρδος του αλγορίθμου GBU από πλευράς απόδοσης είναι πολύ μεγαλύτερο (π.χ. παρέβαλε τα διαγράμματα 3.15(α′)-3.15(δ′)).

Αξιολογώντας το προτεινόμενο μοντέλο κόστους και τεχνικές βελτιστοποίησης, τα πειράματά μας δείχνουν ότι τόσο ο άπληστος αλγόριθμος όσο και ο αλγόριθμος δυναμικού προγραμματισμού παράγουν πλάνα εκτέλεσης που αποδίδουν αρκετά καλά συγκρινόμενα με το βέλτιστο πλάνο. Όπως αναμενόταν, σε ερωτήματα με σχετικά μικρό αριθμό προτιμήσεων ή joins, ο αλγόριθμος δυναμικού προγραμματισμού είναι ελαφρά καλύτερος από τον άπληστο αλγόριθμο (το μέσο ποσοστό βελτίωσης που μετρήσαμε ήταν 4.32%). Αντιθέτως, για ερωτήματα με πολλαπλά joins ή μεγάλο αριθμό προτιμήσεων, (π.χ. $|P| \geq 6$) ο άπληστος αλγόριθμος συνήθως είναι προτιμότερος εξαιτίας του μικρότερου κόστους βελτιστοποίησης που απαιτεί σε σχέση με τον αλγόριθμο δυναμικού προγραμματισμού.

Κεφάλαιο 4

Αλγόριθμοι Εξατομίκευσης από την πλευρά των χρηστών

Στην ενότητα 4.1 ορίζουμε την έννοια του ερωτήματος κορυφογραμμής που εξαρτάται από το περιβάλλον χρήσης. Στις ενότητες 4.2 και 4.3 προτείνουμε αλγορίθμους για την αποτίμηση τέτοιων ερωτημάτων. Τέλος, η ενότητα 4.4 περιέχει την πειραματική αξιολόγηση των προτεινόμενων μεθόδων.

4.1 Βασικοί ορισμοί

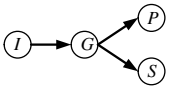
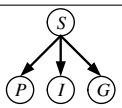
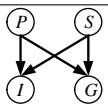
Η ενότητα 4.1.1 εισάγει τις βασικές έννοιες του context, των προτιμήσεων και ενός ερωτήματος κορυφογραμμής. Στη συνέχεια, στην ενότητα 4.1.2 επεκτείνουμε τους προηγούμενους ορισμούς στην περίπτωση αβέβαιων προτιμήσεων και δίνουμε τον επίσημο ορισμό του προβλήματος. Στην ενότητα 4.1.3 προτείνουμε μια μέθοδο εξαγωγής αβέβαιων προτιμήσεων. Ο Πίνακας 4.1 συνοψίζει τα σύμβολα που θα χρησιμοποιήσουμε στη συνέχεια.

4.1.1 Ερωτήματα κορυφογραμμής εξαρτώμενα από το περιβάλλον χρήσης

Έστω μια σχέση $\mathcal{R}$ που αποτελείται από ένα σύνολο γνωρισμάτων $\mathbf{A} = \{A_1, \ldots, A_d\}$. Ορίζουμε ως context C μία συγκεκριμένη κατάσταση που σχετίζεται με έναν χρήστη ή ερώτημα. Κάθε context μπορεί να αναπαρασταθεί με τη βοήθεια ενός συνόλου ζευγών παραμέτρων-τιμών [3, 66]. Για παράδειγμα το context C_1 του Πίνακα 4.2 μπορεί να αναπαρασταθεί ως {Purpose=Business, Period=June}. Θα χρησιμοποιήσουμε τον όρο *υποκειμενικά γνωρίσματα* (relatively preferred - RP) για να αναφερόμαστε σε γνωρίσματα για τα οποία οι προτιμήσεις εξαρτώνται από το τρέχον context. Αντίστοιχα, για γνωρίσματα των οποίων οι προτιμήσεις είναι προκαθορισμένες και ισχύουν καθολικά ανεξαρτήτως context θα χρησιμοποιήσουμε τον όρο *αντικειμενικά γνωρίσματα* (statically preferred - SP). Για παράδειγμα στο Σχήμα 4.1(α′), η τιμή και η απόσταση ενός ξενοδοχείου από τη θάλασσα είναι αντικειμενικά γνωρίσματα, ενώ οι προσφερόμενες υπηρεσίες (amenities) είναι υποκειμενικό γνώρισμα.

Σύμβολο	Ορισμός
$\mathcal{R}$	πίνακας
$\text{dom}(A_j)$	πεδίο ορισμού γνωρίσματος A_j
C_i	context
$u \succ_{A_j} v \,\|\, C_i$	η τιμή u προτιμάται έναντι της v για το context C_i, όπου $u, v \in \text{dom}(A_j)$
$\text{CSQ}(\mathcal{R}\|C_q)$	contextual skyline query
$Pr[u \succ_{A_j} v \,\|\, C_i]$	πιθανότητα ότι ισχύει $u \succ_{A_j} v \,\|\, C_i$
$t \succ t'$	η εγγραφή t κυριαρχεί επί της t'
$Pr[t \succ t' \,\|\, C_q]$	πιθανότητα κυριαρχίας της εγγραφής t επί της t' για το context C_q
$P_{sky}^{C_q}(t)$	πιθανότητα η εγγραφή t να ανήκει στην κορυφογραμμή
p-$\text{CSQ}(\mathcal{R}\|C_q)$	probabilistic contextual skyline query
$\mathcal{G}_i$	σύνολο εγγραφών
e_j^i, e_j^{i-}, e_j^{i+}	κόμβος ενός aR-tree T_i, και οι αντίστοιχες πάνω δεξιά και κάτω αριστερά κορυφές του
$p(e_j^i, t)$	εκτιμώμενη πιθανότητα κυριαρχίας του κόμβου e_j^i επί της t

Πίνακας 4.1: *Πίνακας συμβόλων*

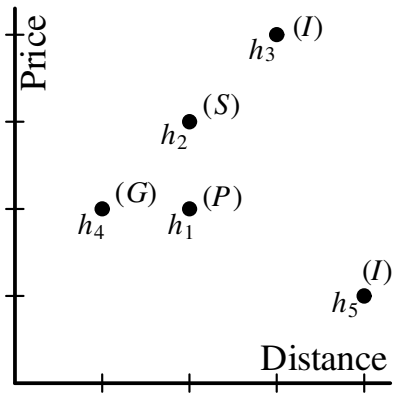
Context	Προτιμήσεις	Κορυφογραμμή
C_1: Business, June		h_3, h_4, h_5
C_2: Vacation		h_2, h_4, h_5
C_3: Summer		h_1, h_2, h_4, h_5
C_q: Business, Summer	—	;

Πίνακας 4.2: *Contexts, προτιμήσεις και ερωτήματα κορυφογραμμής εξαρτώμενα από το τρέχον context*

Για να αναπαραστήσουμε τις προτιμήσεις ενός χρήστη θα ακολουθήσουμε την 'ποιοτική' προσέγγιση. Συγκεκριμένα θα θεωρήσουμε ότι κάθε προτίμηση εφαρμόζεται σε ένα γνώρισμα A_j και έχει τη μορφή $u \succ_{A_j} v$, όπου $u, v \in \text{dom}(A_j)$. Η προτίμηση αυτή υποδηλώνει ότι αν για δύο εγγραφές $t, t' \in \mathcal{R}$ ισχύει ότι $t.A_j = u$, $t'.A_j = v$ και $t.A_k = t'.A_k$ για όλα τα υπόλοιπα $k \neq j$, τότε η εγγραφή t προτιμάται έναντι της t'. Για υποκειμενικά γνωρίσματα θα θεωρήσουμε προτιμήσεις οι οποίες εξαρτώνται από το context. Για παράδειγμα αν η παραπάνω προτίμηση ισχύει για το context C_i, τότε μπορεί να οριστεί ως $u \succ_{A_j} v \,|\, C_i$. Εφόσον οι προτιμήσεις για αντικειμενικά γνωρίσματα ισχύουν για όλα τα δυνατά contexts, θα ακολουθήσουμε την ακόλουθη ενοποιημένη ορολογία $u \succ_{A_j} v \,|\, C_i$ για όλα τα A_j. Επανερχόμενοι στο παράδειγμά μας, ένας χρήστης έχει δηλώσει τις ακόλουθες προτιμήσεις: $I \succ_A G \,|\, C_1$, $I \succ_A P \,|\, C_1$, $I \succ_A S \,|\, C_1$, $G \succ_A P \,|\, C_1$, $G \succ_A S \,|\, C_1$ που ισχύουν για το context C_1 σε σχέση με το γνώρισμα Amenities (εν συντομία A). Θα κάνουμε την παραδοχή ότι για ένα συγκεκριμένο context C_i και γνώρισμα A_j οι προτιμήσεις ενός χρήστη είναι μη αντικρουόμενες, δηλαδή δεν μπορεί να ισχύει ταυτόχρονα $u \succ_{A_j} v \,|\, C_i$ και $v \succ_{A_j} u \,|\, C_i$ για $u \neq v \in \text{dom}(A_j)$.

Hotel	Price	Distance	Amenity
h_1	200	10	Pool (P)
h_2	300	10	Spa (S)
h_3	400	15	Internet (I)
h_4	200	5	Gym (G)
h_5	100	20	Internet (I)

(α′) Dataset



(β′) Απεικόνιση σε 2 διαστάσεις

Σχήμα 4.1: *Βάση δεδομένων με πληροφορίες για ξενοδοχεία*

Με άλλα λόγια το σύνολο προτιμήσεων που ισχύουν για ένα γνώρισμα A_j και context C_i ορίζει μία αυστηρά μερική διάταξη (strict partial order). Μια μερική διατάξη μπορεί να αναπαρασταθεί με τη βοήθεια ενός κατευθυνόμενου γράφου. Ο Πίνακας 4.2 απεικονίζει τους κατευθυνόμενους γράφους που που αντιστοιχούν στις προτιμήσεις που έχουν οριστεί.

Θα λέμε ότι μια εγγραφή t *κυριαρχεί* (dominates) επί μιας άλλης εγγραφής t' για το context C_i, και θα συμβολίζουμε με $t \succ t' \,|\, C_i$, αν η εγγραφή t προτιμάται ή είναι ισοδύναμη της t' για όλα τα γνωρίσματα της σχέσης $\mathcal{R}$, και υπάρχει τουλάχιστον ένα γνώρισμα στο οποίο η εγγραφή t προτιμάται αυστηρά έναντι της t', δηλαδή $\forall j\ t.A_j \succeq_{A_j} t'.A_j \,|\, C_i$[1] και $\exists k\ t.A_k \succ_{A_k} t'.A_k \,|\, C_i$. Ένα *ερώτημα κορυφογραμμής εξαρτώμενο από το περιβάλλον χρήσης* (contextual skyline query - CSQ) για ένα context C_i, το οποίο θα συμβολίζουμε ως CSQ($\mathcal{R}|C_i$), επιστρέφει όλες τις εγγραφές οι οποίες δεν κυριαρχούνται από καμιά άλλη εγγραφή για το context C_i.

Σημειώνουμε ότι ο παραπάνω ορισμός βρίσκεται σε συμφωνία με τον ορισμό ενός συμβατικού ερωτήματος κορυφογραμμής. Ένα συμβατικό ερώτημα κορυφογραμμής δεν χρειάζεται να συνδέεται με ένα συγκεκριμένο context καθώς όλα τα γνωρίσματα θεωρούνται ως αντικειμενικά. Αντιθέτως, στα δυναμικά ερωτήματα κορυφογραμμής όπου υπάρχουν υποκειμενικά γνωρίσματα, οι προηγούμενες εργασίες θεωρούν ένα τρέχον context για το οποίο ισχύουν οι προτιμήσεις, χωρίς όμως να το περιγράφουν ευθέως. Σε αυτή την εργασία, θεωρούμε όλα τα ερωτήματα κορυφογραμμής ως εξαρτώμενα από το context, δηλαδή θεωρούμε ότι συνδέονται με μια συγκεκριμένη κατάσταση η οποία τυπικά αναφέρεται ως context.

4.1.2 Πιθανοτικά ερωτήματα κορυφογραμμής εξαρτώμενα από το τρέχον περιβάλλον χρήσης

Προηγούμενες εργασίες ασχολούνται με την αποτίμηση CSQs, όπου οι προτιμήσεις ενός χρήστη για τα υποκειμενικά γνωρίσματα είναι γνωστές για το τρέχον context. Το πρόβλημα που αντιμετωπίζουμε σε αυτή την εργασία μπορεί να περιγραφεί ως ε-

[1] Η συντόμευση $u \succeq_A v$ ισοδυναμεί με $u \succ_A v \vee u = v$.

ξής: δοθέντος ενός συνόλου προτιμήσεων (προφίλ) που έχουν εξαχθεί για διαφορετικά contexts από το τρέχον, θέλουμε να προσδιορίσουμε τις εγγραφές που ανήκουν στην κορυφογραμμή για το τρέχον context.

Αν το τρέχον context ταιριάζει απόλυτα με ένα από τα contexts για τα οποία οι προτιμήσεις ενός χρήστη είναι γνωστές, τότε το πρόβλημα μπορεί να μετασχηματιστεί σε ένα κλασικό ερώτημα κορυφογραμμής. Πώς θα πρέπει όμως να αντιμετωπίσουμε την περίπτωση όπου το τρέχον context δεν περιλαμβάνεται στο προφίλ του χρήστη· Στην παρούσα ενότητα υποστηρίζουμε ότι είναι εφικτό να παραχθούν αποτελέσματα για ένα ερώτημα κορυφογραμμής που εξαρτάται από το context ακόμα και αν οι ακριβείς προτιμήσεις ενός χρήστη για το τρέχον context δεν είναι γνωστές. Η μόνη προϋπόθεση για να συμβεί αυτό είναι να υπάρχει διαθέσιμη πληροφορία για τις προτιμήσεις ενός χρήστη για άλλα contexts. Εναλλακτικά μπορούν επίσης να χρησιμοποιηθούν οι προτιμήσεις χρηστών με παρόμοια χαρακτηριστικά ακολουθώντας μια προσέγγιση παρόμοια με αυτή που ακολουθείται στα συστήματα συστάσεων που χρησιμοποιούν την τεχνική collaborative filtering.

Σε αυτή την περίπτωση το σύστημα μπορεί να υπολογίσει μια εκτίμηση των προτιμήσεων που αναμένεται ότι θα ισχύουν για το ζητούμενο context, όμως η συγκεκριμένη διαδικασία εισάγει μια αβεβαιότητα στην ισχύ των προτιμήσεων. Αξίζει εδώ να σημειωθεί ότι η αβεβαιότητα είναι μια εγγενής ιδιότητα των προτιμήσεων, την οποία εισαγάγαμε για πρώτη φορά στο μοντέλο αναπαράστασης των προτιμήσεων που περιγράφεται στο Κεφάλαιο 3. Με παρόμοιο τρόπο, εδώ η αβεβαιότητα αναπαρίσταται με τη βοήθεια των πιθανοτήτων. Για παράδειγμα, μια αβέβαιη προτίμηση που εξαρτάται από το context, έστω $u \succ_{A_j} v \,|\, C_i$ ισχύει με πιθανότητα $Pr[u \succ_{A_j} v \,|\, C_i]$. Όπως και στην περίπτωση των βέβαιων προτιμήσεων, θα κάνουμε την παραδοχή ότι οι προτιμήσεις είναι μη αντικρουόμενες. Συνεπώς ισχύει ότι για κάθε $u \neq v \in \mathrm{dom}(A_j)$, $Pr[u \succ_{A_j} v \,|\, C_i] \leq 1 - Pr[v \succ_{A_j} u \,|\, C_i]$. Η ανισότητα ισχύει έτσι ώστε να συμπεριλάβουμε και την περίπτωση όπου οι τιμές u, v είναι μη συγκρίσιμες.

Θα επιχειρήσουμε να επιλύσουμε ένα ερώτημα κορυφογραμμής για ένα νέο context, μετασχηματίζοντάς το σε δύο υποπροβλήματα. Για το πρώτο υποπρόβλημα θέλουμε να εξαγάγουμε ένα σύνολο αβέβαιων προτιμήσεων για το τρέχον context. Για το δεύτερο υποπρόβλημα, χρησιμοποιώντας το σύνολο αβέβαιων προτιμήσεων ως είσοδο, επιλύουμε ένα πιθανοτικό ερώτημα κορυφογραμμής το οποίο επιστρέφει εκείνες τις εγγραφές οι οποίες ανήκουν στην κορυφογραμμή με μεγάλη πιθανότητα. Παρακάτω αναλύουμε τα δύο υποπροβλήματα.

Πρόβλημα 1 [Εξαγωγή αβέβαιων προτιμήσεων] Με βάση το προφίλ ενός χρήστη, προσδιόρισε ένα σύνολο αβέβαιων προτιμήσεων για το τρέχον context C_q.

Η αβεβαιότητα που υπάρχει για τις προτιμήσεις που ισχύουν οδηγεί επίσης σε αβέβαιες σχέσεις κυριαρχίας (dominance relationships) μεταξύ των εγγραφών. Θεωρώντας τις προτιμήσεις ανεξάρτητες μεταξύ τους, η πιθανότητα μια εγγραφή t να κυριαρχεί επί της t' για το context C_i είναι:

$$Pr[t \succ t' \,|\, C_i] = \begin{cases} \prod_j Pr[t.A_j \succeq_{A_j} t'.A_j \,|\, C_i], & \text{if } t \neq t' \\ 0, & \text{if } t = t', \end{cases} \tag{4.1}$$

όπου η πρώτη περίπτωση έχει εφαρμογή μόνο αν οι δύο εγγραφές δεν έχουν ακριβώς τις ίδιες τιμές για όλα τα γνωρίσματα. Αξίζει να αναφερθεί ότι ο συγκεκριμένος ορισμός μετασχηματίζεται στον συμβατικό ορισμό της κυριαρχίας όταν όλες οι προτιμήσεις είναι βέβαιες, δηλαδή $Pr[t \succ t' \,|\, C_i] = 1$ αν και μόνο αν $t \succ t' \,|\, C_i$, και 0 σε άλλη περίπτωση.

Με βάση τα παραπάνω, προσαρμόζουμε τον ορισμό ενός ερωτήματος κορυφογραμμής. Εφόσον το ενδεχόμενο μια εγγραφή να κυριαρχείται από κάποια άλλη είναι αβέβαιο, τότε και το ενδεχόμενο η εγγραφή αυτή να ανήκει στην κορυφογραμμή είναι επίσης αβέβαιο. Η πιθανότητα μια εγγραφή t να ανήκει στην κορυφογραμμή για ένα context C_i προκύπτει ως εξής:

$$P_{sky}^{C_i}(t) = \prod_{t' \neq t} (1 - Pr[t' \succ t \,|\, C_i])\,. \tag{4.2}$$

Σε συμφωνία με την βέβαιη εκδοχή του ερωτήματος, αν όλες οι προτιμήσεις είναι βέβαιες, τότε $P_{sky}^{C_i}(t) = 1$ αν η εγγραφή t κυριαρχείται από κάποια άλλη εγγραφή, και 0 σε άλλη περίπτωση. Στη συνέχεια ορίζουμε το δεύτερο υποπρόβλημα:

Πρόβλημα 2 [Probabilistic Contextual Skyline Query (p-CSQ)] Δοθείσης μιας σχέσης $\mathcal{R}$ και ενός συνόλου αβέβαιων προτιμήσεων, επέστρεψε τις εγγραφές $t \in \mathcal{R}$ των οποίων η πιθανότητα να ανήκουν στην κορυφογραμμή είναι μεγαλύτερη από μια τιμή κατωφλίου $t \in p\text{-CSQ}(\mathcal{R}|C_i) \Leftrightarrow P_{sky}^{C_i}(t) \geq p$.

Η συγκεκριμένη εργασία εστιάζει στην αποδοτική αντιμετώπιση του δεύτερου προβλήματος. Συνεπώς, στην ενότητα 4.1.3 παραθέτουμε απλά μια μεθοδολογία για την επίλυση του Προβλήματος 1 και δεν θα αναφερθούμε αναλυτικά σε άλλους εναλλακτικούς τρόπους επίλυσης που μπορούν επίσης να εφαρμοστούν. Στη συνέχεια, στις ενότητες 4.2 και 4.3 εισάγουμε αλγορίθμους για την αποδοτική επεξεργασία του δεύτερου υποπροβλήματος.

4.1.3 Εξαγωγή αβέβαιων προτιμήσεων

Αρχικά ορίζουμε ένα μέτρο ομοιότητας μεταξύ contexts. Θα θεωρήσουμε ότι για κάθε παράμετρο του context X_i, υπάρχει μια συνάρτηση sim_{X_i} η οποία υπολογίζει την ομοιότητα ανάμεσα σε δύο τιμές στο πεδίο τιμών $\text{dom}(X_i)$ και επιστρέφει μια τιμή ομοιότητας στο διάστημα $[0, 1]$, όπου υψηλότερες τιμές υποδηλώνουν μεγαλύτερη ομοιότητα. Ανάλογα με τον τύπο της κάθε παραμέτρου διαφορετικές συναρτήσεις ομοιότητας μπορούν να εφαρμοστούν. Για παράδειγμα, αν μια παράμετρος έχει αριθμητικό πεδίο ορισμού, τότε μπορούμε να χρησιμοποιήσουμε τη συνάρτηση $\text{sim}_{X_i}(a, b) = 1 - \frac{|a-b|}{M-m}$, όπου M και m είναι η μέγιστη και η ελάχιστη τιμή στο $\text{dom}(X_i)$. Αντίστοιχα, για κατηγορικά ή ιεραρχικά πεδία ορισμού, μια πιθανή επιλογή για συνάρτηση ομοιότητας είναι ο συντελεστής Jaccard: $\text{sim}_{X_i}(a, b) = \frac{|\text{lvs}(a) \cap \text{lvs}(b)|}{|\text{lvs}(a) \cup \text{lvs}(b)|}$, όπου με $\text{lvs}(a)$ συμβολίζουμε το σύνολο κόμβων στο υπόδεντρο ενός κόμβου a. Για δυαδικά (boolean) γνωρίσματα, $\text{sim}_{X_i}(a, b) = 1$ αν $a = b$, και 0 διαφορετικά. Χρησιμοποιώντας τέτοιου είδους συναρτήσεις για τον υπολογισμό της ομοιότητας μεταξύ των τιμών παραμέτρων του context, ορίζουμε την ομοιότητα μεταξύ δύο contexts C, C' ως $\text{sim}(C, C') = \prod_i \text{sim}_{X_i}(C.X_i, C'.X_i)$. Επίσης θα θεωρήσουμε ότι αν για κάποιο context η τιμή μιας παραμέτρου δεν έχει προσδιο-

ριστεί, τότε ταιριάζει με οποιαδήποτε άλλη τιμή του πεδίου τιμών. Για παράδειγμα στον Πίνακα 4.2 για το context C_2 δεν έχει προσδιοριστεί τιμή για την παράμετρο του χρόνου, συνεπώς μπορούμε να θεωρήσουμε ότι οι αντίστοιχες προτιμήσεις ισχύουν ανεξαρτήτως χρόνου.

Θα εφαρμόσουμε τον παραπάνω ορισμό για να υπολογίσουμε την ομοιότητα μεταξύ του τρέχοντος context C_q με όλα τα contexts τα οποία εμφανίζονται στο προφίλ ενός χρήστη. Στη συνέχεια θα χρησιμοποιήσουμε τις τιμές ομοιότητας που υπολογίσαμε για να εξαγάγουμε ένα σύνολο αβέβαιων προτιμήσεων. Διαισθητικά, αναμένεται η πιθανότητα μια τιμή u να προτιμάται σε σχέση με μια τιμή v για το τρέχον context C_q να εξαρτάται από το αν η προτίμηση $u \succ v$ εμφανίζεται σε άλλα contexts και από το πόσο μοιάζουν τα contexts αυτά με το C_q. Για ένα ζεύγος τιμών $u, v \in \mathrm{dom}(A_i)$, ορίζουμε:

$$Pr[u \succ_{A_i} v \,|\, C_q] = \frac{\sum_j (\mathrm{sim}(C_q, C_j) \cdot |u \succ_{A_i} v \,|\, C_j|)}{\sum_j \mathrm{sim}(C_q, C_j)}, \tag{4.3}$$

όπου $|u \succ_{A_i} v \,|\, C_j| = 1$ αν υπάρχει η συγκεκριμένη προτίμηση και 0 σε άλλη περίπτωση.

Σύμφωνα με την εξίσωση 4.3, αν το context C_q ταιριάζει απόλυτα με κάποιο από τα contexts C_i, τότε $Pr[u \succ_{A_i} v \,|\, C_q] = 1$ αν και μόνο αν η προτίμηση $u \succ_{A_i} v \,|\, C_i$ ισχύει, και 0 διαφορετικά. Αν $u \succ v$ για όλα τα contexts, τότε $Pr[u \succ_{A_i} v \,|\, C_q] = 1$. Τέλος, εφόσον οι προτιμήσεις είναι μη αντικρουόμενες, οι παραγόμενες αβέβαιες προτιμήσεις είναι επίσης μη αντικρουόμενες και συνεπώς ισχύει ότι $Pr[u \succ_{A_i} v \,|\, C_q] \leq 1 - Pr[v \succ_{A_i} u \,|\, C_q]$.

Παράδειγμα 4.14. *Η χρήση των παραπάνω ορισμών θα γίνει πιο κατανοητή με τη βοήθεια ενός παραδείγματος. Για το συγκεκριμένο παράδειγμα θα χρησιμοποιήσουμε τη βάση δεδομένων του Σχήματος 4.1(α΄) και τις προτιμήσεις του Πίνακα 4.2. Έστω ένα ερώτημα p-CSQ με ελάχιστη τιμή κατωφλίου 0.5.*

Πρόβλημα 1: *Αρχικά θα υπολογίσουμε τις τιμές ομοιότητας των contexts C_1, C_2 και C_3 με το τρέχον context C_q. Θα θεωρήσουμε τις παρακάτω τιμές ομοιότητας μεταξύ των παραμέτρων του context:* $\mathrm{sim}_{purpose}(Business, Vacation) = 0$ *και* $\mathrm{sim}_{period}(June, Summer) = 1/3$. *Με βάση τις τιμές αυτές προκύπτουν οι εξής τιμές ομοιότητας μεταξύ contexts:* $\mathrm{sim}(C_q, C_1) = 1/3$, $\mathrm{sim}(C_q, C_2) = 0$ *και* $\mathrm{sim}(C_q, C_3) = 1$. *Στη συνέχεια, βάσει της εξίσωσης 4.3 υπολογίζουμε τις αβέβαιες προτιμήσεις που φαίνονται στον Πίνακα 4.3, όπου η τιμή κάθε κελιού αντιστοιχεί στην πιθανότητα η τιμή της γραμμής να είναι προτιμότερη από την αντίστοιχη τιμή της στήλης. Για παράδειγμα για τις τιμές I, P έχουμε* $Pr[I \succ_A P \,|\, C_q] = \frac{1/3+0+0}{1+0+1/3} = 1/4$ *και* $Pr[P \succ_A I \,|\, C_q] = \frac{0+0+1}{1+0+1/3} = 3/4$.

Πρόβλημα 2: *Στη συνέχεια, για να προσδιορίσουμε τις εγγραφές που ανήκουν στην κορυφογραμμή με πιθανότητα μεγαλύτερη ή ίση του 0.5, χρειάζεται πρώτα να υπολογίσουμε τις πιθανότητες των αβέβαιων σχέσεων κυριαρχίας μεταξύ όλων των εγγραφών της βάσης δεδομένων. Οι πιθανότητες αυτές φαίνονται στον Πίνακα 4.4, όπου η τιμή κάθε κελιού αντιστοιχεί στην πιθανότητα η εγγραφή της γραμμής να κυριαρχεί επί της εγγραφής της στήλης. Για παράδειγμα, έστω η εγγραφή h_4. Σύμφωνα με το Σχήμα 4.1(α΄) η εγγραφή h_4 δεν κυριαρχεί επί της εγγραφής h_5 όσον*

u \ v	I	G	P	S
I	—	1/4	1/4	1/4
G	0	—	1/4	1/4
P	3/4	3/4	—	1/4
S	3/4	3/4	0	—

Πίνακας 4.3: *Πιθανότητες προτιμήσεων $Pr[u \succ_A v \,|\, C_q]$ βάσει του Πίνακα 4.2*

t' \ t	h_1	h_2	h_3	h_4	h_5
h_1	0	0	3/4	0	0
h_2	0	0	3/4	0	0
h_3	0	0	0	0	0
h_4	1/4	1/4	0	0	0
h_5	0	0	0	0	0
$\boldsymbol{P_{sky}^{C_q}(t)}$	**3/4**	**3/4**	**1/16**	**1**	**1**

Πίνακας 4.4: *Πιθανότητες κυριαρχίας $Pr[t' \succ t \,|\, C_q]$ για τη βάση δεδομένων του Σχήματος 4.1(α')*

αφορά τα αντικειμενικά γνωρίσματα (τιμή, απόσταση από τη θάλασσα). Συνεπώς, δεν μπορεί να κυριαρχεί επί της h_5 ούτε και στο σύνολο των γνωρισμάτων, άρα η πιθανότητα κυριαρχίας είναι 0. Επίσης, η εγγραφή h_4 κυριαρχεί επί των h_1, h_2, h_3 όσον αφορά τα αντικειμενικά γνωρίσματα. Επειδή η τιμή της εγγραφής h_4 για το γνώρισμα amenity, G, προτιμάται έναντι της αντίστοιχης τιμής P της εγγραφής h_1 με πιθανότητα 1/4, άρα ισχύει $Pr[h_4 \succ h_1 \,|\, C_q] = 1 \cdot 1 \cdot 1/4 = 1/4$. Παρομοίως προκύπτει ότι $Pr[h_4 \succ h_2 \,|\, C_q] = 1 \cdot 1 \cdot 1/4 = 1/4$ αφού $Pr[G \succ_A S \,|\, C_q] = 1/4$. Αντιθέτως, η τιμή G δεν προτιμάται έναντι της I σε καμία περίπτωση και επομένως $Pr[h_4 \succ h_3 \,|\, C_q] = 1 \cdot 1 \cdot 0 = 0$. Επίσης σημειώνουμε ότι η εγγραφή h_4 δεν δύναται να κυριαρχεί επί του εαυτού της σύμφωνα και με την εξίσωση 4.1.

Το τελικό βήμα συνεπάγεται τον υπολογισμό της πιθανότητας μιας εγγραφής να ανήκει στην κορυφογραμμή. Έστω η εγγραφή h_3. Σύμφωνα με τον Πίνακα 4.4, η εγγραφή h_3 μπορεί να κυριαρχηθεί μόνο από τις εγγραφές h_1, h_2. Έτσι, σύμφωνα με την εξίσωση 4.2 έχουμε ότι $P_{sky}^{C_q}(h_3) = (1 - 3/4) \cdot (1 - 3/4) \cdot 1 \cdot 1 = 1/16$. Παρομοίως υπολογίζουμε τις πιθανότητες των υπόλοιπων εγγραφών να ανήκουν στην κορυφογραμμή, οι οποίες φαίνονται στην τελευταία γραμμή του Πίνακα 4.4. Συνεπώς στο αποτέλεσμα του ερωτήματος 0.5-$CSQ(\mathcal{R}|C_q)$ ανήκουν οι εγγραφές h_1, h_2, h_4, h_5.

4.2 Αλγόριθμοι για μη δεικτοδοτούμενα δεδομένα

Σε αυτή την ενότητα περιγράφουμε αλγορίθμους για την αποτίμηση ερωτημάτων p-CSQ για μη δεικτοδοτούμενα δεδομένα. Οι αλγόριθμοι αυτοί επιχειρούν να αποφύγουν να

υπολογίσουν όλες τις πιθανότητες κυριαρχίας μεταξύ των εγγραφών ανά δύο. Στην ενότητα 4.2.1 παρουσιάζουμε μια επέκταση του αλγορίθμου block nested loops - BNL [14] για ερωτήματα p-CSQ, ενώ στην ενότητα 4.2.2 κάνουμε κάποιες παρατηρήσεις και περιγράφουμε ένα σύνολο ιδιοτήτων οι οποίες μας βοηθούν να βελτιώσουμε την επίδοση του προτεινόμενου αλγορίθμου.

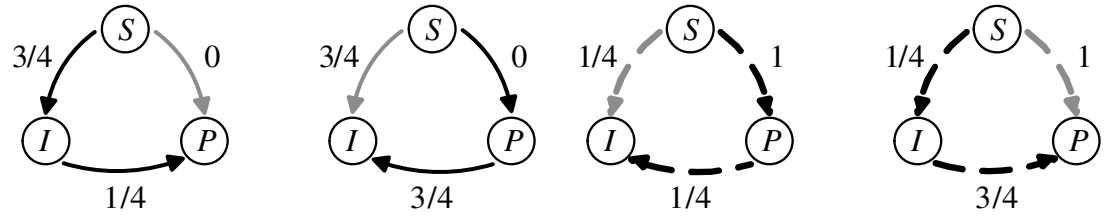
4.2.1 Βασικός Επαναληπτικός Αλγόριθμος

Η πλειονότητα των αλγορίθμων που έχουν προταθεί για την αποτίμηση ερωτημάτων κορυφογραμμής ακολουθούν μια μονοτονική σειρά στον τρόπο με τον οποίο εξετάζουν τις εγγραφές της βάσης. Η σειρά αυτή άλλοτε υποδηλώνεται ρητά, για παράδειγμα ταξινομώντας τις εγγραφές με κάποιο κριτήριο πριν την έναρξη του αλγορίθμου [67, 21, 29, 56, 11], είτε έμμεσα, για παράδειγμα χρησιμοποιώντας μια ουρά προτεραιότητας (priority queue) [39, 55, 48, 62]. Αξίζει να σημειωθεί ότι ακόμα και στην περίπτωση όπου οι τιμές των εγγραφών εμφανίζουν κάποιο βαθμό αβεβαιότητας, μπορεί να προκύψει μια μονοτονική σειρά εξέτασης αν κάποιος λάβει υπόψη του το ελάχιστο περιέχοντα κύβο (minimum bounding box - MBB) κάθε (αβέβαιης) εγγραφής [56, 48]. Εξαιτίας της μονοτονικής ιδιότητας που ισχύει για τις σχέσεις κυριαρχίας ($t_1 \succ t_2, t_2 \succ t_3 \Rightarrow t_1 \succ t_3$), μια μονοτονική σειρά εξέτασης των πλειάδων μειώνει τον μέσο αριθμό ελέγχων κυριαρχίας που απαιτούνται και επιπλέον επιτρέπει την προοδευτική παραγωγή αποτελεσμάτων κατά τη διάρκεια της εκτέλεσης του αλγορίθμου.

Παρακάτω θα δείξουμε ότι η μεταβατική ιδιότητα και η μονοτονικότητα δεν ισχύουν στην περίπτωση αβέβαιων προτιμήσεων και επομένως ούτε και στην περίπτωση αβέβαιων σχέσεων κυριαρχίας. Ας θεωρήσουμε τις αβέβαιες προτιμήσεις του Πίνακα 4.3 που αναφέρονται στις τιμές S, I, P για το τρέχον context C_q. Η τιμή S προτιμάται έναντι της τιμής I με πιθανότητα 3/4 και η τιμή I έναντι της τιμής P με πιθανότητα 1/4, όπως δείχνει το Σχήμα 4.2(α′). Αν ίσχυε η μεταβατική ιδιότητα, τότε θα περιμέναμε η τιμή S να προτιμάται έναντι της P (γκρι βέλος στο Σχήμα 4.2(α′)) με κάποια πιθανότητα π.χ., 3/16. Όμως όπως προκύπτει από τον Πίνακα 4.3, έχουμε $Pr[S \succ_A P \,|\, C_q] = 0$, δηλαδή η πιθανότητα είναι μικρότερη από την αναμενόμενη. Παρομοίως ενώ έχουμε $Pr[S \succ_A P \,|\, C_q] = 0$ και $Pr[P \succ_A I \,|\, C_q] = 3/4$, ισχύει επίσης $Pr[S \succ_A I \,|\, C_q] = 3/4$, δηλαδή περισσότερο από ό,τι αναμενόταν, όπως δείχνει το Σχήμα 4.2(β′). Παρόμοια αποτελέσματα ισχύουν και για την συμπληρωματική πιθανότητα ‘μη προτίμησης ’ $1 - Pr[\cdot \succ_A \cdot \,|\, C_q]$, όπως δείχνουν τα διακεκομμένα βέλη στα Σχήματα 4.2(γ′) και 4.2(δ′). Επομένως, όπως δείχνει το συγκεκριμένο παράδειγμα, αν οι προτιμήσεις είναι αβέβαιες δεν μπορεί να προκύψει στη γενική περίπτωση μερική διάταξη από τη λιγότερο προς την περισσότερο προτιμητέα τιμή.

Το γεγονός ότι η μεταβατική ιδιότητα και η μονοτονικότητα των πιθανοτήτων δεν ισχύουν υποδηλώνει ότι αλγόριθμοι που βασίζονται σε ταξινόμηση των εγγραφών πριν την εξέτασή τους (π.χ. SFS [21]) δεν μπορούν να εφαρμοστούν στην περίπτωση ερωτημάτων p-CSQ. Παρακάτω, παρουσιάζουμε μια βασική τεχνική αποτίμησης p-CSQ για μη δεικτοδοτούμενα δεδομένα, με την ονομασία Βασικός Επαναληπτικός Αλγόριθμος (Basic Iterative Algorithm - BIA).

Ο αλγόριθμος BIA υπολογίζει την πιθανότητα μια εγγραφή να ανήκει στην κορυ-



Σχήμα 4.2: *Η μεταβατική ιδιότητα και η μονοτονικότητα των πιθανοτήτων δεν ισχύουν για αβέβαιες προτιμήσεις. Η πιθανότητα προτίμησης μειώνεται (α), αυξάνεται (β). Η πιθανότητα μη προτίμησης μειώνεται (γ), αυξάνεται (δ)*

φογραμμή σαρώνοντας όλα τα δεδομένα, ακολουθώντας μια μέθοδο παρόμοια με αυτήν που ακολουθεί ο αλγόριθμος εμφωλευμένων βρόχων BNL. Συγκεκριμένα, έστω M και N τα μεγέθη της κύριας μνήμης και των δεδομένων, μετρούμενα σε blocks του δίσκου. Ο αλγόριθμος BIA διαχωρίζει το σύνολο δεδομένων σε $\frac{N}{M-1}$ ομάδες (batches) των $M-1$ blocks και τα εξετάζει διαδοχικά. Ο αλγόριθμος BIA φορτώνει την κάθε ομάδα και εκτελεί την παρακάτω διαδικασία.

Αρχικά η πιθανότητα κάθε εγγραφής t της ομάδας να ανήκει στην κορυφογραμμή τίθεται σε $P_{sky}^{C_q}(t) = 1$. Στη συνέχεια, ο αλγόριθμος BIA σαρώνει τα δεδομένα φορτώνοντας στη μνήμη ένα block κάθε φορά. Για κάθε εγγραφή t της ομάδας και κάθε εγγραφή t' του block που ανακτήθηκε, ο αλγόριθμος BIA υπολογίζει την πιθανότητα κυριαρχίας της εγγραφής t' επί της t και ενημερώνει κατάλληλα την πιθανότητα να ανήκει στην κορυφογραμμή ως εξής: $P_{sky}^{C_q}(t) := P_{sky}^{C_q}(t) \cdot (1 - Pr[t' \succ t \,|\, C_q])$. Αν η πιθανότητα $P_{sky}^{C_q}(t)$ πέσει κάτω από την προκαθορισμένη τιμή κατωφλίου, τότε η εγγραφή t απορρίπτεται και δεν συμπεριλαμβάνεται στους επόμενους ελέγχους κυριαρχίας. Αν όλες οι εγγραφές μιας ομάδας απορριφθούν, ο BIA αντικαθιστά την ομάδα αυτή φορτώνοντας το επόμενο block από το δίσκο. Όταν ολόκληρη η βάση δεδομένων έχει σαρωθεί τότε οι τρέχουσες πιθανότητες όλων των εγγραφών που δεν έχουν απορριφθεί είναι οι τελικές και επομένως οι εγγραφές αυτές επιστρέφονται ως αποτέλεσμα. Εύκολα μπορούμε να υπολογίσουμε ότι η πολυπλοκότητα του αλγορίθμου BIA είναι $O\left(\frac{N^2}{M}\right)$.

4.2.2 Αλγόριθμος Επιλογής Υποψηφίων

Η ενότητα αυτή παρουσιάζει δύο βήματα προεπεξεργασίας τα οποία μειώνουν σε μεγάλο βαθμό τον αριθμό των απαιτούμενων ελέγχων κυριαρχίας. Έστω $\{\mathcal{G}_i\}$ το σύνολο σχέσεων που προκύπτουν εφαρμόζοντας έναν τελεστή ομαδοποίησης (group-by) στα υποκειμενικά γνωρίσματα (SP) της σχέσης $\mathcal{R}$ και αφού εκτελέσουμε μια προβολή στα αντικειμενικά γνωρίσματα (SP). Κάθε $\mathcal{G}_i$ περιέχει εγγραφές για έναν συγκεκριμένο συνδυασμό τιμών των υποκειμενικών γνωρισμάτων. Για παράδειγμα, για τη βάση δεδομένων του Σχήματος 4.1 έχουμε ένα υποκειμενικό γνώρισμα με 4 πιθανές τιμές. Επομένως προκύπτουν οι ακόλουθες ομάδες: $\mathcal{G}_I = \{h_3, h_5\}$, $\mathcal{G}_S = \{h_2\}$, $\mathcal{G}_P = \{h_1\}$, $\mathcal{G}_G = \{h_4\}$. Έστω $\mathrm{CSQ}(\mathcal{G}_i)$ το σύνολο εγγραφών που ανήκουν στην κορυφογραμμή μιας ομάδας $\mathcal{G}_i$. Τότε ισχύουν τα ακόλουθα:

Ιδιότητα 5. *Για κάθε τιμή κατωφλίου $p > 0$ και context C_q, p-CSQ$(\mathcal{R}|C_q) \subseteq \bigcup_i$ CSQ$(\mathcal{G}_i)$.*

Απόδειξη. Θα αποδείξουμε την παραπάνω ιδιότητα με τη μέθοδο της εις άτοπον απαγωγής. Έστω ότι υπάρχει μια εγγραφή t τέτοια ώστε $t \in p$-CSQ$(\mathcal{R}|C_q)$ αλλά $t \notin \bigcup_i$ CSQ$(\mathcal{G}_i)$, για κάποια p και C_q. Επιπλέον, έστω $\mathcal{G}_k$ η ομάδα στην οποία ανήκει η εγγραφή t. Αφού $t \notin$ CSQ$(\mathcal{G}_k)$, τότε θα πρέπει να υπάρχει μία άλλη εγγραφή $t' \in \mathcal{G}_k$, τέτοια ώστε η t' να κυριαρχεί επί της t σε σχέση με τα αντικειμενικά γνωρίσματα, δηλαδή $t' \succ_{SP} t$. Αφού οι εγγραφές t, t' ανήκουν στην ίδια ομάδα, άρα έχουν ίσες τιμές για όλα τα υποκειμενικά γνωρίσματα και επομένως η εγγραφή t' κυριαρχεί επί της t για κάθε context, δηλαδή $Pr[t' \succ t \,|\, C_q] = 1$. Άρα η πιθανότητα να ανήκει η εγγραφή t στην κορυφογραμμή είναι μηδενική $P_{sky}^{C_q}(t) = 0$ και επομένως $t \notin p$-CSQ$(\mathcal{R}|C_q)$, το οποίο είναι άτοπο. □

Στο προηγούμενο παράδειγμα, καμιά εγγραφή δεν μπορεί να απορριφθεί βάσει της παραπάνω ιδιότητας, καθώς κάθε μια ανήκει στην κορυφογραμμή της ομάδας της.

Ιδιότητα 6. *Για κάθε τιμή κατωφλίου $p > 0$ και context C_q, αν για μια εγγραφή $t \in CSQ(\bigcup_i \mathcal{G}_i)$ δεν υπάρχει άλλη εγγραφή $t' \in \mathcal{R}$ τέτοια ώστε οι εγγραφές t, t' να έχουν ίσες τιμές για τα αντικειμενικά γνωρίσματα, τότε $t \in p$-CSQ$(\mathcal{R}|C_q)$.*

Απόδειξη. Έστω μια εγγραφή t τέτοια ώστε $t \in$ CSQ$(\bigcup_i \mathcal{G}_i)$ για την οποία δεν υπάρχει άλλη εγγραφή $t' \in \mathcal{R}$ τέτοια ώστε οι εγγραφές t, t' να έχουν ίσες τιμές για τα αντικειμενικά γνωρίσματα. Η πρώτη συνθήκη υποδηλώνει ότι η εγγραφή t ανήκει στην κορυφογραμμή όλης της βάσης δεδομένων αν λάβουμε υπόψη μας μόνο τα αντικειμενικά κριτήρια. Θα αποδείξουμε ότι για κάθε $t^* \neq t \in \mathcal{R}$ ισχύει $Pr[t^* \succ t \,|\, C_q] = 0$. Έστω ότι δεν ισχύει. Τότε μια εγγραφή t^* θα πρέπει να είναι προτιμότερη με μη μηδενική πιθανότητα ή ισοδύναμη με την εγγραφή t για όλα τα γνωρίσματα. Όμως αυτό δεν μπορεί να ισχύει για τα αντικειμενικά γνωρίσματα καθώς: (α) δεν υπάρχει εγγραφή με ίσες τιμές για όλα τα γνωρίσματα (αφού η εγγραφή t είναι μοναδική), και (β) καμιά εγγραφή δεν προτιμάται έναντι της t σε σχέση με τα αντικειμενικά γνωρίσματα (αφού η εγγραφή t ανήκει στην κορυφογραμμή σε σχέση με τα αντικειμενικά γνωρίσματα). Συνεπώς η παραδοχή μας ήταν λάθος και ισχύει $Pr[t^* \succ t \,|\, C_q] = 0$ για όλες τις εγγραφές $t^* \neq t \in \mathcal{R}$. Συνεπώς, η εξίσωση 4.2 μας δίνει $Pr_{sky}^{C_q}(t) = 1$, το οποίο σημαίνει ότι η εγγραφή t ανήκει στο p-CSQ$(\mathcal{R}|C_q)$ για κάθε p, C_q. □

Στο παράδειγμα του Σχήματος 4.1(β′), οι εγγραφές h_4, h_5 ανήκουν στην κορυφογραμμή σε σχέση με τα αντικειμενικά γνωρίσματα. Επομένως, ανήκουν και στο αποτέλεσμα για κάθε ερώτημα p-CSQ, όπως προκύπτει επίσης και από τους Πίνακες 4.2 και 4.4.

Αξίζει να σημειωθεί ότι η κορυφογραμμή της ένωσης των ομάδων είναι ισοδύναμη με την ένωση της κορυφογραμμής των επιμέρους ομάδων, δηλαδή, CSQ$(\bigcup_i \mathcal{G}_i) =$ CSQ$\left(\bigcup_i \text{CSQ}(\mathcal{G}_i)\right)$. Συνεπώς, CSQ$(\bigcup_i \mathcal{G}_i) \subseteq \bigcup_i$ CSQ$(\mathcal{G}_i)$. Με άλλα λόγια, χρησιμοποιώντας τις ιδιότητες 5, 6 μπορούμε να προσδιορίσουμε άμεσα τις εγγραφές εκείνες οι οποίες ανήκουν οπωσδήποτε στο αποτέλεσμα για κάθε p-CSQ, τις οποίες μπορούμε και να επιστρέψουμε ως μέρος του αποτελέσματος. Επίσης αξίζει να σημειωθεί ότι το ποιες

εγγραφές ανήκουν στην κορυφογραμμή κάθε ομάδας $\mathrm{CSQ}(\mathcal{G}_i)$ εξαρτάται αποκλειστικά από τις αντικειμενικά τους γνωρίσματα για τα οποία οι ισχύουσες προτιμήσεις είναι σταθερές. Συνεπώς, οι εγγραφές που ανήκουν στα επιμέρους σύνολα κορυφογραμμής κάθε ομάδας μπορούν να έχουν προεπεξεργαστεί πριν την εκκίνηση του αλγορίθμου.

Ο Αλγόριθμος Επιλογής Υποψηφίων (Candidate Selection Algorithm - CSA χρησιμοποιεί τις παραπάνω ιδιότητες για να επιταχύνει την εκτέλεση του ερωτήματος. Αρχικά υπολογίζει το σύνολο $\mathcal{C}$ των εγγραφών οι οποίες είναι υποψήφιες να ανήκουν στο αποτέλεσμα του ερωτήματος. Το σύνολο $\mathcal{C}$ θα περιέχει όλες τις εγγραφές t που ανήκουν στο $\bigcup_i \mathrm{CSQ}(\mathcal{G}_i)$ μη συμπεριλαμβανομένων αυτών που με βάση την ιδιότητα 6 έχουν $Pr_{sky}^{C_q}(t) = 1$ και επομένως είναι ήδη γνωστό ότι ανήκουν στο αποτέλεσμα. Η εκτέλεση του αλγορίθμου CSA είναι πανομοιότυπη με του BIA με μόνη διαφορά τη διαδικασία ομαδοποίησης. Συγκεκριμένα αντί να διαχωρίσει όλο το σύνολο των δεδομένων σε ομάδες (batches), ο αλγόριθμος CSA ταξινομεί τις εγγραφές που ανήκουν στο $\mathcal{C}$ χρησιμοποιώντας μια συνάρτηση γεμίσματος χώρου (π.χ. καμπύλη Hilbert) και στη συνέχεια διαχωρίζει τις εγγραφές σε ομάδες ώστε να χωρούν στην κύρια μνήμη. Στη συνέχεια, ο αλγόριθμος CSA υπολογίζει την πιθανότητα κάθε εγγραφής του $\mathcal{C}$ να ανήκει στην κορυφογραμμή κατά τα γνωστά. Σημειώνουμε ότι όπως και στην περίπτωση του αλγορίθμου BIA, ο αλγόριθμος CSA χρειάζεται να σαρώσει ολόκληρη τη βάση δεδομένων και όχι απλά τις εγγραφές που ανήκουν στο $\mathcal{C}$, καθώς εγγραφές που δεν ανήκουν στο $\mathcal{C}$ μπορούν να κυριαρχούν επί εγγραφών του $\mathcal{C}$ με μη μηδενική πιθανότητα. Στην περίπτωση που όλες οι υποψήφιες εγγραφές (δηλαδή αυτές που ανήκουν στο $\mathcal{C}$) χωρούν στην κύρια μνήμη, τότε η πολυπλοκότητα του αλγορίθμου CSA γίνεται $O(N)$.

4.3 Αλγόριθμοι για δεικτοδοτούμενα δεδομένα

Στην ενότητα αυτή συζητάμε μεθόδους για την αποτίμηση ερωτημάτων p-CSQ σε δεικτοδοτούμενα δεδομένα. Στα συμβατικά ερωτήματα κορυφογραμμής κάθε έλεγχος κυριαρχίας έχει ως στόχο να απαντήσει αν μια εγγραφή κυριαρχείται από κάποια άλλη. Αντιθέτως, στην περίπτωση ερωτημάτων p-CSQ κάθε έλεγχος κυριαρχίας επιχειρεί να προσδιορίσει τον αριθμό των εγγραφών από τις οποίες είναι πιθανό να κυριαρχείται μια εγγραφή καθώς και τις αντίστοιχες πιθανότητες. Για το λόγο αυτό, για τους αλγορίθμους που θα παρουσιάσουμε θα βασιστούμε σε δείκτες οι οποίοι περιέχουν συναθροιστική πληροφορία (aggregate information). Σημειώνουμε ότι οι δείκτες που περιγράφονται στη συνέχεια μπορούν να κατασκευαστούν ανεξαρτήτως του τρέχοντος context και των αντίστοιχων προτιμήσεων και επομένως μπορούν να χρησιμοποιηθούν για την αποτίμηση ερωτημάτων p-CSQ για κάθε πιθανό context.

4.3.1 Αλγόριθμος Απαρίθμησης με Ομάδες

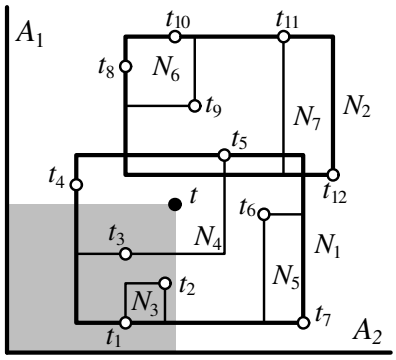
Ο Αλγόριθμος Απαρίθμησης με Ομάδες (Basic Group Counting - BGC) βασίζεται σε δύο παρατηρήσεις. Η πρώτη παρατήρηση είναι ότι μπορούμε να διαχωρίσουμε το κομμάτι ενός ελέγχου κυριαρχίας που σχετίζεται με τα αντικειμενικά γνωρίσματα από το κομμάτι που σχετίζεται με τα υποκειμενικά γνωρίσματα. Για να έχει μια εγγραφή t μη μηδενική πιθανότητα να κυριαρχεί επί της t' θα πρέπει η t να κυριαρχεί επί της t'

όσον αφορά τα αντικειμενικά γνωρίσματα και επίσης οι t, t' να μην έχουν ίσες τιμές. Η δεύτερη παρατήρηση που μπορούμε να κάνουμε είναι ότι όλες οι εγγραφές που ανήκουν σε μια ομάδα $\mathcal{G}_i$ έχουν την ίδια πιθανότητα κυριαρχίας επί μιας εγγραφής t' σε σχέση με τα υποκειμενικά γνωρίσματα. Συνδυάζοντας τις παραπάνω παρατηρήσεις, ο στόχος του αλγορίθμου BGC είναι να μετρήσει το πλήθος εγγραφών που κυριαρχούν επί της t' σε σχέση με τα αντικειμενικά γνωρίσματα για κάθε ομάδα $\mathcal{G}_i$.

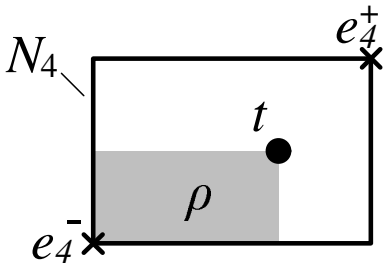
Με σκοπό τον αποδοτικό υπολογισμό του πλήθους εγγραφών για κάθε ομάδα, ο αλγόριθμος BGC δεικτοδοτεί τις εγγραφές που ανήκουν σε κάθε ομάδα με τη βοήθεια ενός COUNT aggregate R-tree (aR-tree). Με παρόμοιο τρόπο προς αυτόν ενός R-tree, σε ένα aR-tree οι εγγραφές ομαδοποιούνται και ανατίθενται σε κόμβους. Στη συνέχεια, τα minimum bounding boxes (MBBs) κάθε κόμβου ομαδοποιούνται ιεραρχικά προς τα πάνω σχηματίζοντας κόμβους υψηλότερου επιπέδου. Κάθε κόμβος ενός aR-tree περιέχει καταχωρήσεις της μορφής: $\langle e_i, \text{MBB}_i, c_i \rangle$ για κάθε έναν από τους κόμβους - άμεσους απογόνους N_i, όπου e_i είναι ένας δείκτης προς τον κόμβο N_i, MBB_i είναι το MBB του κόμβου N_i και c_i είναι η συναθροιστική πληροφορία που περιέχει ο κόμβος, δηλαδή ο αριθμός εγγραφών που περικλείονται στο υπόδεντρο με ρίζα τον κόμβο N_i. Κάθε MBB αναπαρίσταται από την κάτω αριστερή και την πάνω δεξιά κορυφή του e_i^- και e_i^+ αντίστοιχα. Το Σχήμα 4.3 δείχνει ένα aR-tree για μια ομάδα 12 εγγραφών όπου κάθε κόμβος έχει χωρητικότητα 3. Συγκεκριμένα, το Σχήμα 4.3(α′) απεικονίζει τα MBBs, το Σχήμα 4.3(β′) εστιάζει στον κόμβο N_4 δείχνοντας την κάτω αριστερά και την πάνω δεξιά κορυφή του, ενώ το Σχήμα 4.3(γ′) δείχνει τη δομή κάθε κόμβου.

Ο αλγόριθμος BGC υπολογίζει την πιθανότητα να ανήκουν στην κορυφογραμμή μόνο για τις υποψήφιες εγγραφές που ανήκουν στο σύνολο $\mathcal{C}$, ακολουθώντας τη μεθοδολογία της ενότητας 4.2.2. Η κορυφογραμμή κάθε ομάδας μπορεί να υπολογιστεί με τη βοήθεια της μεθόδου BBS [55] (αγνοώντας τη συναθροιστική πληροφορία των κόμβων). Παρακάτω, γράφουμε $\succ_{SP}$ και $\succ_{RP}$ για να συμβολίσουμε την κυριαρχία όσον αφορά τα αντικειμενικά και τα υποκειμενικά γνωρίσματα αντιστοίχως. Οι αντίστοιχες πιθανότητες μπορούν να υπολογιστούν με τη βοήθεια της εξίσωσης 4.1 αν χρησιμοποιήσουμε μόνο τα αντικειμενικά ή υποκειμενικά γνωρίσματα αντιστοίχως. Χωρίς βλάβη της γενικότητας, για τα αντικειμενικά γνωρίσματα θα θεωρήσουμε αριθμητικές τιμές και ότι οι μικρότερες τιμές είναι προτιμότερες. Επιπλέον, θα συμβολίζουμε με $Pr[\mathcal{G}_{t'} \succ_{RP} \mathcal{G}_t \,|\, C_q]$ την πιθανότητα μια οποιαδήποτε εγγραφή $t' \in \mathcal{G}_{t'}$ να κυριαρχεί όσον αφορά τα υποκειμενικά γνωρίσματα επί μιας οποιασδήποτε εγγραφής $t \in \mathcal{G}_t$, ήτοι $Pr[t' \succ_{RP} t \,|\, C_q]$.

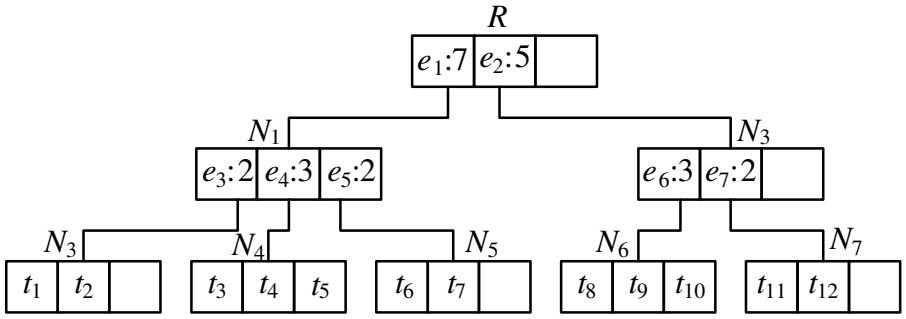
Έστω μια εγγραφή $t \in \mathcal{C}$ και έστω $\mathcal{G}_t$ η ομάδα υποκειμενικών τιμών γνωρισμάτων στην οποία ανήκει η t. Ένας πρώτος τρόπος για να υπολογιστεί η πιθανότητα η εγγραφή t να ανήκει στην κορυφογραμμή, είναι να εξετάσουμε όλους τους κόμβους του aR-tree διαδοχικά τον έναν μετά τον άλλο. Αρχικά θέτουμε $Pr_{sky}^{C_q}(t) = 1$. Για κάθε κόμβο T_i ο οποίος περιέχει εγγραφές από μια ομάδα $\mathcal{G}_i$, εκτελώντας ένα ερώτημα περιοχής (range query) στο aR-tree μπορούμε να βρούμε το πλήθος εγγραφών που κυριαρχούν επί της t όσον αφορά τα αντικειμενικά γνωρίσματα. Η σκιασμένη περιοχή του Σχήματος 4.3(α′) αντιστοιχεί σε ένα τέτοιο ερώτημα περιοχής. Αφού ο αλγόριθμος επισκεφτεί διαδοχικά τους κόμβους R, N_1 και N_4 προκύπτει ότι η απάντηση είναι 3



(α′) aR-tree MBBs



(β′) Κόμβος N_4



(γ′) aR-tree structure

Σχήμα 4.3: *Παράδειγμα aggregate R-tree*

εγγραφές (η απάντηση περιέχει την εγγραφή t_3 και δύο εγγραφές που αντιστοιχούν στον κόμβο N_3). Επομένως, η πιθανότητα η εγγραφή t να ανήκει στην κορυφογραμμή πρέπει να ενημερωθεί ως εξής: $Pr^{c_q}_{sky}(t) := Pr^{c_q}_{sky}(t) \cdot (1 - Pr[\mathcal{G}_i \succ_{RP} \mathcal{G}_t \,|\, C_q])^{n_i}$, όπου n_i είναι ο αριθμός εγγραφών που κυριαρχούν επί της t όσον αφορά τα αντικειμενικά γνωρίσματα.

Ενώ η παραπάνω προσέγγιση είναι σωστή, είναι πιθανό να οδηγήσει σε περιττές λειτουργίες εισόδου/εξόδου από τον δίσκο για μια εγγραφή που τελικά θα απορριφθεί από το αποτέλεσμα του ερωτήματος. Ο βασικός μας στόχος είναι να εντοπίσουμε και να απορρίψουμε όσο το δυνατόν ταχύτερα τέτοιες εγγραφές. Για το λόγιο αυτό, ο αλγόριθμος BGC διασχίζει τους κόμβους του aR-tree με σειρά αυξανόμενης πιθανότητας απόρριψης, δηλαδή ξεκινώντας από τις εγγραφές που έχουν μεγαλύτερη πιθανότητα να επιστραφούν και καταλήγοντας σε αυτές με τη μικρότερη πιθανότητα. Σημειώνουμε ότι η αλλαγή στη σειρά εξέτασης των κόμβων δεν επηρεάζει το πλήθος των κόμβων που εξετάζει ο αλγόριθμος.

Ο ψευδοκώδικας του αλγορίθμου BGC φαίνεται στη λίστα 2. Ο αλγόριθμος BGC συντηρεί έναν σωρό $\mathcal{H}$ που περιέχει εγγραφές της μορφής $\langle e^i_j, p(e^i_j, t)\rangle$, όπου με e^i_j συμβολίζουμε έναν κόμβο στο aR-tree T_i και $p(e^i_j, t)$ είναι το κλειδί του $\mathcal{H}$, το οποίο αντιπροσωπεύει τη συμβολή του κόμβου e^i_j στην πιθανότητα της εγγραφής t να ανήκει στην κορυφογραμμή. Συγκεκριμένα, $p(e^i_j, t)$ είναι η εκτιμώμενη πιθανότητα κάποια από τις εγγραφές του κόμβου e^i_j να κυριαρχεί επί της t. Ο τρόπος υπολογισμού αυτής της

Algorithm 2: Basic/Super Group Counting

Input: $\mathcal{C}$, p, C_q, aR-trees $\{T_i\}$
Output: S the answer to p-CSQ($\mathcal{D}|C_q$)
Variables: $\mathcal{H}$ a minheap with entries $\langle e, p(e,t)\rangle$ and key $p(e,t)$

```
1  begin
2      S := ∅ ;
3      foreach t ∈ C do
4          Pr^{Cq}_{SKY}(t) := 1 ;
5          foreach T_i do
6              p(e^i_R) := ComputeProb (e^i_R, t, C_q)        // for the root e^i_R of T_i ;
7              enheap ⟨e^i_R, p(e^i_R, t)⟩ ;
8          while H not empty and Pr^{Cq}_{SKY}(t) ≥ p do
9              deheap ⟨e^i_j, p(e^i_j, t)⟩ ;
10             if e^{i+}_j ≻_SP t then
11                 Pr^{Cq}_{SKY}(t) := Pr^{Cq}_{SKY}(t) · p(e^i_j, t) ;
12             else
13                 foreach child e^i_k of e^i_j do
14                     if e^{i−}_k ≻_SP t then
15                         p(e^i_k, t) := ComputeProb (e^i_k, t, C_q) ;
16                         enheap ⟨e^i_k, p(e^i_k, t)⟩ ;
17         if Pr^{Cq}_{SKY}(t) ≥ p then
18             insert t in S ;
19     return S;
```

πιθανότητας (συνάρτηση BCGComputeProb) θα περιγραφεί στη συνέχεια.

Ο αλγόριθμος BGC επαναλαμβάνει την ακόλουθη διαδικασία για κάθε υποψήφια εγγραφή t. Αρχικά, η πιθανότητα η εγγραφή t να ανήκει στην κορυφογραμμή τίθεται ίση με 1 (γραμμή 4). Επίσης, για τη ρίζα e^i_R κάθε δέντρου T_i, δημιουργείται μια καταχώριση $\langle e^i_R, p(e^i_R, t)\rangle$ η οποία προστίθεται στο σωρό (γραμμές 5–7)[2]. Ο αλγόριθμος συνεχίζει εξετάζοντας τα περιεχόμενα του σωρού (γραμμές 8–16), έως ότου είτε ο σωρός αδειάσει οπότε η εγγραφή t επιστρέφεται ως αποτέλεσμα (γραμμές 17-18), είτε η πιθανότητα της πέσει κάτω από την τιμή κατωφλίου. Έστω $\langle e^i_j, p(e^i_j, t)\rangle$ η εξεταζόμενη καταχώριση, δηλαδή αυτή με το ελάχιστο κλειδί (γραμμή 9). Αν η πάνω δεξιά κορυφή του e^i_j κυριαρχεί επί της t σε σχέση με τα αντικειμενικά γνωρίσματα (γραμμή 10), τότε αυτό συνεπάγεται και ότι όλες οι εγγραφές μέσα στην e^i_j επίσης κυριαρχούν επί της εγγραφής t. σε σχέση με τα αντικειμενικά γνωρίσματα. Επομένως, η αναμενόμενη τιμή της πιθανότητας $p(e^i_j, t)$ ενημερώνεται χρησιμοποιώντας το πλήθος εγγραφών μέσα στην e^i_j (γραμμή 11). Αν η πάνω δεξιά κορυφή του e^i_j δεν κυριαρχεί επί της t, τότε ο αλγόριθμος BGC χρειάζεται να διαβάσει τα περιεχόμενα του κόμβου e^i_j και να εξετάσει τους κόμβους - απογόνους (γραμμές 13–16). Σημειώνουμε ότι ένας κόμβος απόγονος e^i_k εισάγεται ως νέο στοιχείο στο σωρό μόνο αν περιέχει τουλάχιστον μια εγγραφή η οποία μπορεί κυριαρχεί επί της t, δηλαδή αν και μόνο αν η κάτω αριστερή κορυφή του κόμβου e^i_j κυριαρχεί επί της t σε σχέση με τα αντικειμενικά γνωρίσματα (γραμμή 14). Σε αυτή την περίπτωση, η εκτιμώμενη πιθανότητα κυριαρχίας του κόμβου e^i_k υπολογίζεται (γραμμή 15) και η αντίστοιχη καταχώριση προστίθεται στο σωρό (γραμμή 16).

[2]Για να αποφύγουμε την υπερχείλιση της μνήμης που έχει ανατεθεί για το σωρό, αν ο αριθμός aR-trees είναι μεγάλος, αντί να εξετάσουμε όλα τα aR-trees μαζί, τα χωρίζουμε και τα εξετάζουμε σε ομάδες.

Function BCGComputeProb

Input: e, t, C_q
Output: $p(e,t)$ the expected $\prod_{t' \in e}(1 - Pr[t' \succ t])$
Variables: c the count associated with entry e,;
$\mathcal{G}_e$ the group of tuples contained in e,;
$\mathcal{G}_t$ the group of t,;
ρ the fraction of e volume that dominates t w.r.t. SP

```
1 begin
2     ρ := 1 ;
3     foreach SP attribute A_k do
4         ρ := ρ · max{ (min{t.A_k, e+.A_k} − e−.A_k) / (e+.A_k − e−.A_k), 0 } ;
5     p(e,t) := (1 − Pr[G_e ≻_RP G_t | C_q])^(ρ·c) ;
6     return p(e,t);
```

Στη συνέχεια αναλύουμε τον τρόπο υπολογισμού της τιμής $p(e,t)$ η οποία αντιπροσωπεύει την εκτιμώμενη πιθανότητα με την οποία ένας κόμβος e κυριαρχεί επί της εγγραφής t. Κάθε εγγραφή t' που περιέχεται στον κόμβο e η οποία κυριαρχεί επί της t σε σχέση με τα αντικειμενικά γνωρίσματα συνεισφέρει με $1 - Pr[\mathcal{G}_e \succ_{RP} \mathcal{G}_t \,|\, C_q]$ στην πιθανότητα της εγγραφής t να ανήκει στην κορυφογραμμή t. Το ερώτημα είναι πόσες τέτοιες εγγραφές υπάρχουν. Για να καταλήξουμε σε μια εκτίμηση του πλήθους αυτού, θα κάνουμε την παραδοχή ότι οι εγγραφές είναι ομοιόμορφα ταξινομημένες μέσα σε κάθε κόμβο. Με βάση αυτή την παραδοχή μπορούμε να υποθέσουμε ότι ο αριθμός των εγγραφών που κυριαρχούν επί της t είναι ανάλογος με το μέρος του όγκου του MBB στο οποίο περιέχονται. Το Σχήμα 4.3(β′) δείχνει ένα παράδειγμα για την εγγραφή t και τον κόμβο N_4. Κάθε εγγραφή στην σκιασμένη περιοχή μπορεί να κυριαρχεί επί της t. Έστω $\rho = \prod_k \max\left\{\frac{\min\{t.A_k, e^+.A_k\} - e^-.A_k}{e^+.A_k - e^-.A_k}, 0\right\}$ το αντίστοιχο κλάσμα όγκου. Στον όγκο αυτό, αναμένεται να περιέχονται $\rho \cdot c$ εγγραφές, όπου c είναι το συνολικό πλήθος εγγραφών του κόμβου. Επομένως η εκτιμωμένη πιθανότητα θα είναι ίση με $p(e,t) = (1 - Pr[\mathcal{G}_e \succ_{RP} \mathcal{G}_t \,|\, C_q])^{\rho \cdot c}$, και υπολογίζεται με τη βοήθεια της συνάρτησης BCGComputeProb.

4.3.2 Αλγόριθμος Απαρίθμησης με Υπερ-ομάδες

Η απόδοση του αλγορίθμου BGC χειροτερεύει καθώς ο αριθμός ομάδων αυξάνεται. Αυτό συμβαίνει γιατί όταν το πλήθος εγγραφών σε κάθε aR-tree μειώνεται, ο χώρος που καταλαμβάνουν γίνεται πιο αραιός και πλέον κάθε κόμβος (για να χωρέσει τον ίδιο αριθμό εγγραφών) χρειάζεται να καταλαμβάνει μεγαλύτερο όγκο. Επομένως, η πιθανότητα ένας ολόκληρος κόμβος (ουσιαστικά η πάνω δεξιά κορυφή του) να κυριαρχεί επί μιας εγγραφής t μειώνεται δραστικά, πράγμα το οποίο συνεπάγεται πολύ λιγότερα κλαδέματα κόμβων και συνεπώς περισσότερες λειτουργίες εισόδου/εξόδου από το δίσκο. Για να αντιμετωπίσουμε το ζήτημα αυτό, προτείνουμε τον Αλγόριθμο Απαρίθμησης με Υπερ-ομάδες (Super Group Counting - SGC), ο οποίος αναθέτει ομάδες σε υπερ-ομάδες και κατασκευάζει ένα κατάλληλα τροποποιημένο aR-tree για κάθε υπερ-ομάδα.

Αφού κάθε aR-tree περιέχει εγγραφές από διαφορετικές ομάδες, η συναθροιστική πληροφορία που περιέχει κάθε κόμβος θα πρέπει να τροποποιηθεί κατάλληλα. Στο τροποποιημένο aR-tree κάθε κόμβος N_i είναι της μορφής $\langle e_i, \text{MBB}_i, c_i[]\rangle$, όπου $c_i[]$ είναι ένας πίνακας που περιέχει τα πλήθη εγγραφών κάθε ομάδας N_i που περικλείονται

Function SGCComputeProb

Input: e, t, C_q
Output: $p(e,t)$ the expected $\prod_{t' \in e}(1 - Pr[t' \succ t])$
Variables: $c[]$ the count array associated with entry e,;
$\{\mathcal{G}_j\}$ the groups of tuples contained in e,;
$\mathcal{G}_t$ the group of t,;
ρ the fraction of e volume that dominates t w.r.t. SP

```
1 begin
2     ρ := 1 ;
3     foreach SP attribute A_k do
4         ρ := ρ · max{ (min{t.A_k, e⁺.A_k} − e⁻.A_k) / (e⁺.A_k − e⁻.A_k), 0 } ;
5     p(e,t) := 1 ;
6     foreach group G_j ≠ G_t do
7         p(e,t) := p(e,t) · (1 − Pr[G_j ≻_RP G_t | C_q])^{ρ·c[j]} ;
8     return p(e,t);
```

στο υπόδεντρό του. Για παράδειγμα η τιμή $c_i[j]$ αντιστοιχεί στο πλήθος εγγραφών της ομάδας G_j. Ο αλγόριθμος SGC εκτελείται ακριβώς όπως και ο BGC (λίστα 2). Η μόνη διαφορά είναι ότι η συνάρτηση SGCComputeProb πρέπει αντίστοιχα να τροποποιηθεί κατάλληλα ώστε να λαμβάνει υπόψη της πολλαπλές ομάδες.

4.3.3 Αλγόριθμος Ομαδοποίησης

Οι προηγούμενες μέθοδοι έχουν το μειονέκτημα ότι χρειάζεται να εξετάσουν κάθε κόμβο του aR-tree πολλές φορές, μια φορά για κάθε υποψήφια εγγραφή του $\mathcal{C}$. Με τον Αλγόριθμο Ομαδοποίησης (Batch Counting Algorithm - BCA) προτείνουμε μια πιο αποδοτική προσέγγιση η οποία επιτρέπει την ταυτόχρονη επεξεργασία πολλαπλών εγγραφών. Θα θεωρήσουμε ότι όλες οι υποψήφιες εγγραφές χωρούν στην κύρια μνήμη, αλλιώς ο αλγόριθμος BCA χωρίζει το σύνολο υποψηφίων εγγραφών σε ομάδες τις οποίες επεξεργάζεται ξεχωριστά, με παρόμοιο τρόπο όπως ο αλγόριθμος CSA που προτείναμε για μη δεικτοδοτούμενα δεδομένα. Σημειώνουμε ότι ο αλγόριθμος BCA μπορεί να εφαρμοστεί σε aR-trees που έχουν κατασκευαστεί ανά ομάδα ή υπερ-ομάδα. Εφεξής δεν κάνουμε κανένα διαχωρισμό μεταξύ των δύο αυτών επιλογών.

Η εξέταση των εγγραφών κατά ομάδες εισάγει διάφορα προβλήματα. Για παράδειγμα έστω οι εγγραφές t και t' και έστω e ο κόμβος aR-tree υπό εξέταση. Έστω ότι η πάνω δεξιά κορυφή e^+ του κόμβου κυριαρχεί σε σχέση με τα αντικειμενικά γνωρίσματα επί της t αλλά όχι επί της t', δηλαδή μόνο η κάτω δεξιά κορυφή e^- κυριαρχεί σε σχέση με τα αντικειμενικά γνωρίσματα επί της t'. Ο αλγόριθμος BCA θα ενημερώσει την πιθανότητα η εγγραφή t να ανήκει στην κορυφογραμμή εφαρμόζοντας τη συνάρτηση BCGComputeProb. Όμως σε αυτή την περίπτωση το υπόδεντρο με ρίζα τον κόμβο e δεν μπορεί να απορριφθεί καθώς περιλαμβάνει κάποιες εγγραφές οι οποίες κυριαρχούν σε σχέση με τα αντικειμενικά γνωρίσματα επί της εγγραφής t'. Επομένως οι κόμβοι - απόγονοι του e θα πρέπει να εισαχθούν στο σωρό. Όμως, ο αλγόριθμος BCA οφείλει να διασφαλίσει ότι για τους κόμβους - απογόνους του e οι οποίοι επίσης κυριαρχούν σε σχέση με τα αντικειμενικά γνωρίσματα επί της εγγραφής t, η συνεισφορά τους αυτή δεν θα υπολογιστεί και δεύτερη φορά στον υπολογισμό της πιθανότητας. Μια πιθανή αντιμετώπιση αυτού του προβλήματος θα ήταν να συσχετίσουμε κάθε κόμβο e με το σύνολο των εγγραφών για τις οποίες θα πρέπει να εξεταστεί. Όμως ακόμα και

Algorithm 3: Batch Counting Algorithm

Input: $\mathcal{C}$, p, C_q, aR-trees $\{T_i\}$
Output: S the answer to p-CSQ($\mathcal{D}|C_q$)
Variables: $\mathcal{H}$ a minheap with entries $\langle e, b^+\rangle$ and key MINDIST(e)

```
1  begin
2      S := C ;
3      foreach t ∈ S do
4          Pr_SKY^{C_q}(t) := 1 ;
5      foreach T_i do
6          enheap ⟨e_R^i, e_R^{i+}⟩                    // for the root e_R^i of T_i ;
7      while H and S not empty do
8          deheap ⟨e_j^i, b^+⟩ ;
9          foreach t ∈ S do
10             if e_j^{i+} ≻_SP t and b^+ ⊁_SP t then
11                 p(e_j^i, t) := ComputeProb (e_j^i, t, C_q) ;
12                 Pr_SKY^{C_q}(t) := Pr_SKY^{C_q}(t) · p(e_j^i, t) ;
13                 if Pr_SKY^{C_q}(t) < p then
14                     remove t from S ;
15             else if e_j^{i-} ≻_SP t then
16                 foreach child e_k^i of e_j^i do
17                     enheap ⟨e_k^i, e_j^{i+}⟩ ;
18     return S;
```

αν χρησιμοποιηθούν συμπιεσμένες δομές όπως Bloom filters, ο απαιτούμενος χώρος που χρειάζεται γι'αυτό είναι πολύ μεγάλος. Για το λόγο αυτό ο αλγόριθμος BCA ακολουθεί διαφορετική προσέγγιση. Συσχετίζει με κάθε κόμβο e, την πάνω δεξιά κορυφή του κόμβου - πατέρα b^+. Αν το b^+ κυριαρχεί σε σχέση με τα αντικειμενικά γνωρίσματα επί της εγγραφής t, αυτό σημαίνει ότι ο κόμβος - πατέρας του e επίσης κυριαρχεί σε σχέση με τα αντικειμενικά γνωρίσματα επί της εγγραφής t. Συνεπώς, η συνεισφορά του e έχει ήδη προσμετρηθεί και επομένως δεν προσμετρείται ξανά για τον υπολογισμό της πιθανότητας κυριαρχίας επί της εγγραφής t.

Ο ψευδοκώδικας του αλγορίθμου BCA δίνεται στη λίστα 3. Όπως και στην περίπτωση των αλγορίθμων BGC και SGC, ο αλγόριθμος BCA συντηρεί έναν σωρό $\mathcal{H}$. Στην περίπτωση όμως του αλγορίθμου BCA, τα περιεχόμενα του σωρού περιλαμβάνουν κόμβους που ανήκουν σε διαφορετικά aR-trees. Σημειώνουμε επίσης ότι τα στοιχεία του σωρού του αλγορίθμου BCA είναι της μορφής $\langle e, b^+\rangle$, όπου e είναι ένας κόμβος aR-tree και b^+ είναι η πάνω δεξιά κορυφή του κόμβου - πατέρα του e. Επιπλέον, σε αντίθεση με τους αλγορίθμους BGC και SGC, τα στοιχεία του σωρού του αλγορίθμου BCA είναι ταξινομημένα με κριτήριο την Ευκλείδεια απόσταση τους (MINDIST) από την αρχή των αξόνων.

Αρχικά η πιθανότητα κάθε υποψήφιας εγγραφής τίθεται στην τιμή 1 (γραμμές 3–4) και για τη ρίζα κάθε aR-tree εισάγεται στο σωρό μια καταχώριση της μορφής $\langle e_R^i, e_R^{i+}\rangle$ (γραμμές 5–6). Επειδή η ρίζα δεν έχει κόμβο πατέρα, εισάγεται η δική της πάνω δεξιά κορυφή ως b^+. Ο αλγόριθμος BCA εκτελείται σε επαναλήψεις έως ότου είτε ο σωρός αδειάσει είτε δεν έχει απομείνει πλέον καμιά υποψήφια εγγραφή προς εξέταση (γραμμές 7–18). Έστω $\langle e_j^i, b^+\rangle$ το στοιχείο του σωρού με την τρέχουσα ελάχιστη Ευκλειδεια απόσταση από την αρχή των αξόνων (γραμμή 8). Οι υποψήφιες εγγραφές εξετάζονται σειριακά, έστω t η τρέχουσα εξεταζόμενη εγγραφή. Αν η πάνω δεξιά

Algorithm	Acronym	Section
Basic Iterative Algorithm	BIA	4.2.1
Candidate Selection Algorithm	CSA	4.2.2
Super Group Counting	SGC	4.3.1, 4.3.2
Batch Counting Algorithm	BCA	4.3.3

Πίνακας 4.5: *Αλγόριθμοι υπό εξέταση*

κορυφή e_j^{i+} κυριαρχεί σε σχέση με τα αντικειμενικά γνωρίσματα επί της t αλλά το σημείο b^+ όχι (γραμμή 10), τότε όλες οι εγγραφές στο υπόδεντρο του κόμβου e_j^i πρέπει να συμπεριληφθούν στον υπολογισμό της πιθανότητας η εγγραφή t να ανήκει στην κορυφογραμμή. Επομένως, η τιμή της πιθανότητας $Pr_{SKY}^{C_q}(t)$ ενημερώνεται κατάλληλα (γραμμές 11–12). Αν η τιμή της πιθανότητας πέσει κάτω από την τιμή κατωφλίου τότε η εγγραφή t απορρίπτεται από το σύνολο πιθανών αποτελεσμάτων (γραμμές 13–14). Αν η πάνω δεξιά κορυφή e_j^{i+} δεν κυριαρχεί σε σχέση με τα αντικειμενικά γνωρίσματα επί της t αλλά η κάτω αριστερή κορυφή κυριαρχεί (γραμμές 15–18), τότε οι κόμβοι - απόγονοι e_k^i πρέπει επίσης να εξεταστούν (γραμμές 16–18). Τελικά ο αλγόριθμος BCA επιστρέφει ως αποτέλεσμα όλες τις εγγραφές που δεν έχουν απορριφθεί κατά τη λήξη της εκτέλεσής του.

4.4 Πειραματική αξιολόγηση

4.4.1 Πειραματική μεθοδολογία

Σε αυτή την ενότητα αξιολογούμε πειραματικά τους προτεινόμενους αλγορίθμους. Για την πειραματική αξιολόγηση θα θεωρήσουμε ότι οι πιθανότητες ισχύος κάθε προτίμησης έχουν εξαχθεί για το τρέχον context βάσει μιας μεθοδολογίας όπως αυτή που παρουσιάσαμε στην ενότητα 4.1.3. Χρησιμοποιούμε αυτές τις πιθανότητες ως είσοδο για τους αλγόριθμους που θα εξετάσουμε στη συνέχεια.

Στα πειράματά μας χρησιμοποιήσαμε μια γεννήτρια συνθετικών δεδομένων[3]. Με τη βοήθεια της γεννήτριας κατασκευάσαμε δύο σύνολα δεδομένων που ακολουθούν διαφορετικές κατανομές όσον αφορά τις τιμές των γνωρισμάτων τους. Συγκεκριμένα, για το σύνολο uniform οι τιμές επιλέγονται από μία ομοιόμορφη κατανομή. Για το σύνολο anticorrelated οι εγγραφές με προτιμότερες (χαμηλές) τιμές για κάποιο γνώρισμα είναι περισσότερο πιθανό να έχουν υψηλότερες (λιγότερο καλές) τιμές για τα υπόλοιπα γνωρίσματα, για παράδειγμα ένα ξενοδοχείο που είναι πιο κοντά στη θάλασσα είναι πιο πιθανό να είναι ακριβότερο. Το μέγεθος του πεδίου τιμών για τα αντικειμενικά γνωρίσματα είναι 10000, ενώ για κάθε υποκειμενικό γνώρισμα χρησιμοποιήσαμε στα πειράματα από 8 έως 128 διαφορετικές τιμές.

Οι αλγόριθμοι τους οποίους αξιολογήσαμε στα πειράματά μας μαζί με τα ακρωνύμιά τους συνοψίζονται στον Πίνακα 4.5. Όλοι οι αλγόριθμοι υλοποιήθηκαν σε C++, μεταγλωττίστηκαν με gcc και εκτελέστηκαν σε ένα σύστημα με επεξεργαστή 2 GHz Intel Core 2 Duo. Το μέγεθος κάθε σελίδας (μπλοκ) στον δίσκο είναι 4096 bytes ενώ θεωρήσαμε ότι κάθε λειτουργία εισόδου/εξόδου από τον δίσκο έχει κόστος 10 msecs.

[3]http://randdataset.projects.postgresql.org

Παράμετρος	Εύρος Τιμών
Πλήθος εγγραφών (N)	100K, 500K, **1M**, 5M, 10M
Αριθμός αντικειμενικών γνωρισμάτων (d_{SP})	2, **3**, 4
Αριθμός υποκειμενικών γνωρισμάτων (d_{RP})	**1**, 2
Μέγεθος πεδίου τιμών υποκειμενικών γνωρισμάτων ($\|RP\|$)	8, 16, **32**, 64, 128
Τιμή κατωφλίου πιθανότητας (p)	0.1, 0.3, **0.5**, 0.7, 0.9
Αριθμός ομάδων ανά υπερ-ομάδα ($\|sg\|$)	1, 4, **8**, 16, 32

Πίνακας 4.6: *Παράμετροι πειραμάτων*

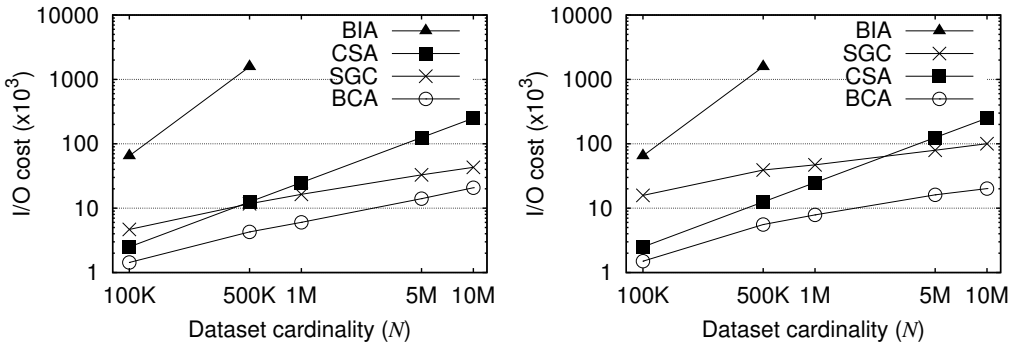
Επίσης χρησιμοποιήσαμε έναν LRU buffer μεγέθους 100 σελίδων (410 KB). Για κάθε εγγραφή θεωρήσαμε ένα σταθερό μέγεθος 100 bytes. Οι αλγόριθμοι CSA και BCA χρησιμοποιούν τον buffer για να επεξεργαστούν τις εγγραφές σε ομάδες (batches). Αντιθέτως, ο αλγόριθμος SGC επεξεργάζεται μια υποψήφια εγγραφή σε κάθε επανάληψη και συνεπώς ο buffer χρησιμοποιείται για να κρατήσει στην μνήμη cache τους κόμβους aR-tree που προσπελάστηκαν πιο πρόσφατα. Σε κάθε πείραμα, μεταβάλαμε μια μόνο παράμετρο κρατώντας τις υπόλοιπες παραμέτρους στις προκαθορισμένες τους τιμές. Ο Πίνακας 4.6 δείχνει τις παραμέτρους υπό εξέταση μαζί με το εύρος τιμών που δοκιμάσαμε για κάθε παράμετρο. Οι προκαθορισμένες τιμές διακρίνονται με έντονους χαρακτήρες.

4.4.2 Πειραματικά αποτελέσματα

Επίδοση σε σχέση με το πλήθος εγγραφών. Στο πρώτο πείραμα εξετάζουμε την επίδοση κάθε αλγορίθμου σε σχέση με το πλήθος εγγραφών. Συγκεκριμένα, μεταβάλαμε το πλήθος εγγραφών N από 100K έως 10M εγγραφές και μετρήσαμε τον αριθμό λειτουργιών εισόδου/εξόδου από το δίσκο (I/Os), τον χρόνο επεξεργασίας και τον συνολικό χρόνο (χρόνος επεξεργασίας συν κόστος εισόδου/εξόδου από το δίσκο). Τα αποτελέσματα του πειράματος φαίνονται στα Σχήματα 4.4, 4.5 και 4.6 αντιστοίχως.

Μελετώντας το Σχήμα 4.4 βλέπουμε ότι, όπως αναμέναμε, καθώς το μέγεθος της βάσης δεδομένων μεγαλώνει, όλοι οι αλγόριθμοι απαιτούν περισσότερες λειτουργίες εισόδου/εξόδου. Ο αλγόριθμος BIA έχει κόστος εκτέλεσης που είναι ανάλογο με το τετράγωνο του πλήθους εγγραφών N και επομένως γίνεται πρακτικά μη εφαρμόσιμος καθώς το πλήθος εγγραφών αυξάνεται (σύμφωνα με το Σχήμα 4.4 αυτό συμβαίνει για μεγέθη μεγαλύτερα των 500K εγγραφών). Αντιθέτως, όλοι οι υπόλοιποι αλγόριθμοι βασίζονται στις παρατηρήσεις που κάναμε στην ενότητα 4.2.2 εξετάζοντας μόνο τις υποψήφιες εγγραφές και επομένως η κλιμάκωσή τους είναι σχεδόν γραμμική με το πλήθος εγγραφών. Συγκεκριμένα, αν όλες οι υποψήφιες εγγραφές χωρούν στην κύρια μνήμη, τότε ο αλγόριθμος CSA έχει γραμμική εξάρτηση από το πλήθος εγγραφών αφού χρειάζεται μόνο μια σάρωση της βάσης δεδομένων.

Το Σχήμα 4.4(α′) δείχνει ότι οι αλγόριθμοι CSA, SGC και BCA παρουσιάζουν δύο τάξεις μεγέθους βελτίωση σε σχέση με τον αλγόριθμο BIA για $N = 500$K εγγραφές στο σύνολο δεδομένων που ακολουθεί την ομοιόμορφη κατανομή. Μεταξύ των τριών αλγορίθμων, οι αλγόριθμοι SGC και BCA που βασίζονται στη χρήση aR-tree είναι σημαντικά πιο αποδοτικοί απαιτώντας περίπου μια τάξη μεγέθους λιγότερα I/Os για το μεγαλύτερο σύνολο δεδομένων: 43123 και 20741 αντιστοίχως έναντι 250000 I/Os για



(α′) uniform dataset

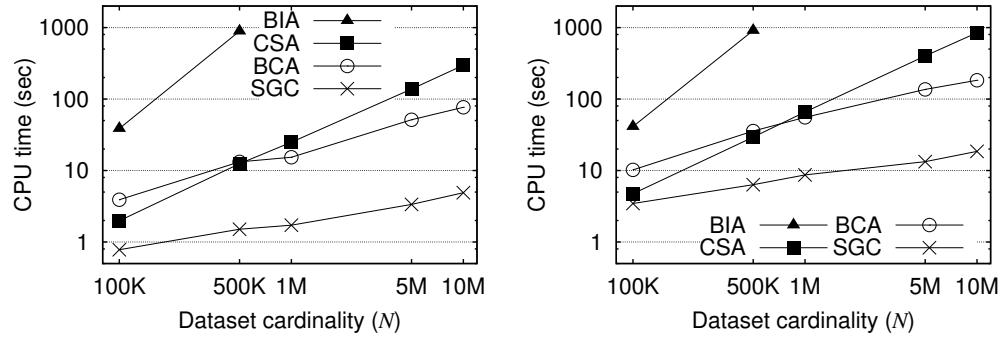
(β′) anticorrelated dataset

Σχήμα 4.4: *Αριθμός λειτουργιών εισόδου/εξόδου (I/O) σε σχέση με το πλήθος εγγραφών*

τον αλγόριθμο CSA. Μεταξύ των αλγορίθμων SGC και BCA, ο δεύτερος απαιτεί κατά μέσο όρο δύο φορές λιγότερα I/Os. Το Σχήμα 4.4(β′) παρουσιάζει παρόμοια τάση στη συμπεριφορά όλων των αλγορίθμων και στην περίπτωση δεδομένων που ακολουθούν την anticorrelated κατανομή. Αξίζει να σημειωθεί όμως ότι η επίδοση του αλγορίθμου SGC χειροτερεύει πολύ πιο γρήγορα σε σχέση με την uniform κατανομή καθώς στο anticorrelated σύνολο δεδομένων ο αριθμός υποψηφίων εγγραφών είναι πολύ μεγαλύτερος. Η πολυπλοκότητα I/O του αλγορίθμου SGC εξαρτάται σε μεγάλο βαθμό από το μέγεθος του συνόλου $\mathcal{C}$ αφού χρειάζεται να εξετάσει όλα τα aR-trees μια φορά για κάθε υποψήφια εγγραφή.

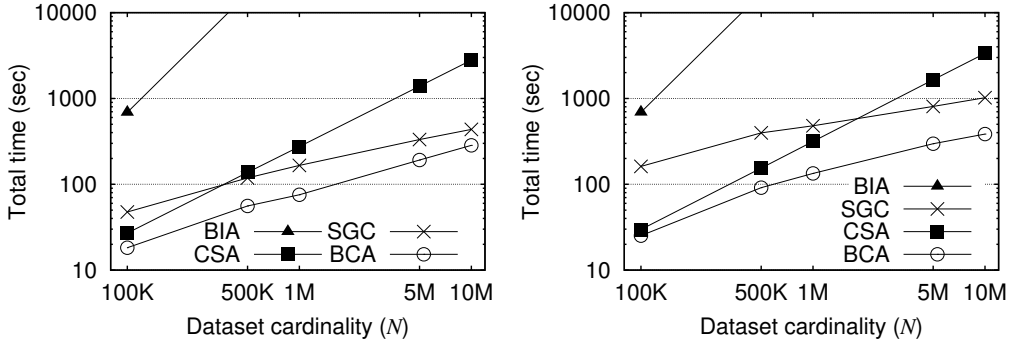
Το Σχήμα 4.5 απεικονίζει τον χρόνο επεξεργασίας κάθε αλγορίθμου καθώς αυξάνεται το πλήθος δεδομένων. Η συμπεριφορά των αλγορίθμων που δεν βασίζονται σε δεικτοδοτούμενα δεδομένα παρουσιάζει παρόμοια συμπεριφορά με αυτή του Σχήματος 4.4. Αντιθέτως, όπως μπορούμε να παρατηρήσουμε ο χρόνος επεξεργασίας που απαιτεί ο αλγόριθμος BCA είναι αρκετά μεγαλύτερος σε σχέση με τον χρόνο που απαιτεί ο αλγόριθμος SGC. Αυτό οφείλεται κυρίως στο μεγαλύτερο μέγεθος του σωρού που συντηρεί ο αλγόριθμος BCA. Στο βασικό σενάριο πειραμάτων το μέγιστο μέγεθος του σωρού που μετρήθηκε ήταν 6036 καταχωρήσεις (60 KB) για τον αλγόριθμο BCA, έναντι μόλις 57 καταχωρίσεων (0.4 KB) για τον αλγόριθμο SGC. Αυτό συμβαίνει επειδή σύμφωνα με τον αλγόριθμο BCA, ένας κόμβος aR-tree πρέπει να προστεθεί στο σωρό αν κυριαρχεί επί τουλάχιστον μιας υποψήφιας εγγραφής, κάτι που συμβαίνει πολύ συχνά. Επιπλέον, για κάθε καταχώρηση $\langle e, b^+ \rangle$ που αφαιρείται από το σωρό, απαιτείται η εκτέλεση ελέγχων κυριαρχίας με όλες τις υποψήφιες εγγραφές, τόσο για τον κόμβο e όσο και για το σημείο b^+. Επομένως, κάθε καταχώρηση του σωρού απαιτεί κατά μέσο όρο περίπου $3|S|$ ελέγχους κυριαρχίας, όπου S είναι το τρέχον σύνολο υποψηφίων αποτελεσμάτων. Για το anti-correlated σύνολο δεδομένων τα αποτελέσματα φαίνονται στο Σχήμα 4.5(β′). Γενικότερα, γι'αυτό το σύνολο δεδομένων οι εγγραφές είναι πιο απίθανο να κυριαρχούν η μια έναντι της άλλης και επομένως ο αριθμός ελέγχων κυριαρχίας και ο χρόνος επεξεργασίας που απαιτείται αυξάνεται σε αυτή την περίπτωση.

Το Σχήμα 4.6 δείχνει τον συνολικό χρόνο εκτέλεσης (κόστος CPU και I/O) καθώς το πλήθος εγγραφών της βάσης αυξάνεται. Είναι σημαντικό να παρατηρήσουμε ότι ο καθοριστικός παράγοντας για το συνολικό κόστος είναι ο απαιτούμενος αριθ-



(α′) uniform dataset (β′) anticorrelated dataset

Σχήμα 4.5: *Χρόνος επεξεργασίας σε σχέση με το πλήθος εγγραφών*



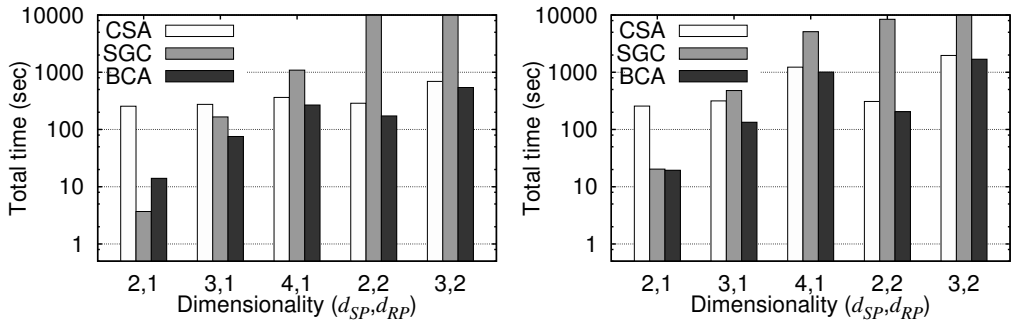
(α′) uniform dataset (β′) anticorrelated dataset

Σχήμα 4.6: *Συνολικό κόστος επεξεργασίας σε σχέση με το πλήθος εγγραφών*

μός λειτουργιών I/O. Όπως φαίνεται στο Σχήμα 4.6(α′) οι αλγόριθμοι που βασίζονται σε δεικτοδοτούμενα δεδομένα είναι σημαντικά ταχύτεροι από τον αλγόριθμο CSA ακολουθώντας την τάση του Σχήματος 4.4(α′). Το Σχήμα 4.6(β′) δείχνει παρόμοια συμπεριφορά για όλους τους αλγορίθμους και για τα anti-correlated δεδομένα. Σε αυτή την περίπτωση, ο αλγόριθμος BCA είναι από 6.4 έως και 8.7 φορές ταχύτερος από τους αλγορίθμους CSA και SGC αντιστοίχως.

Στη συνέχεια της πειραματικής μας αξιολόγησης δεν συμπεριλαμβάνουμε τον αλγόριθμο BIA, καθώς είναι πολύ χειρότερος από πλευράς κόστους σε σχέση με τους υπόλοιπους αλγόριθμους σε όλα τα πειράματα.

Επίδοση σε σχέση με τον αριθμό διαστάσεων. Στο Σχήμα 4.7 εξετάζουμε την επίπτωση του αριθμού διαστάσεων (dimensionality) στο συνολικό κόστος επεξεργασίας. Όταν ο αριθμός αντικειμενικών γνωρισμάτων d_{SP} αυξάνεται, το πλήθος πιθανών αποτελεσμάτων αυξάνεται δραματικά. Όταν ο αριθμός υποκειμενικών γνωρισμάτων d_{RP} αυξάνεται, ο αριθμός υποψηφίων αποτελεσμάτων αυξάνεται με ακόμα μεγαλύτερο βαθμό επειδή τότε ο αριθμός ομάδων υποκειμενικών γνωρισμάτων αυξάνεται εκθετικά. Ο αλγόριθμος SGC επηρεάζεται κυρίως εξαιτίας της ευαισθησίας του στον αριθμό υποψηφίων αποτελεσμάτων. Αυτό φαίνεται πιο ξεκάθαρα στο Σχήμα 4.7, όπου παρότι ο αλγόριθμος SGC είναι ο πιο αποδοτικός σε χαμηλότερο αριθμό δια-



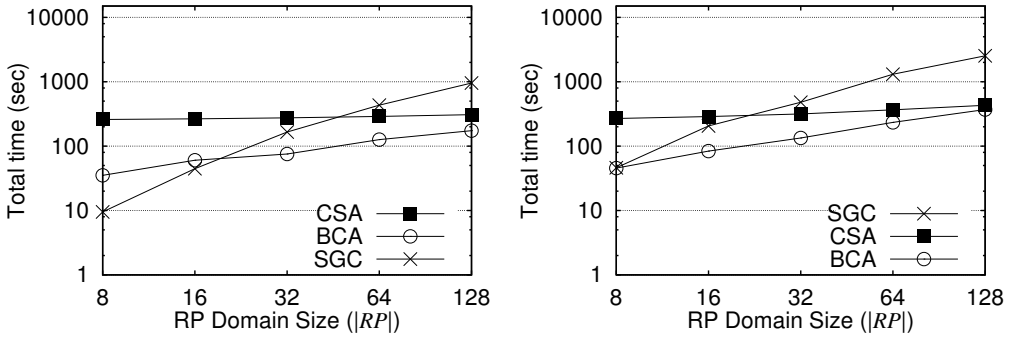
(α′) uniform dataset (β′) anticorrelated dataset

Σχήμα 4.7: *Συνολικό κόστος σε σχέση με τον αριθμό αντικειμενικών (d_{SP}) και υποκειμενικών (d_{RP}) γνωρισμάτων*

στάσεων, γίνεται γρήγορα μη αποδοτικός σε πάνω από 4 διαστάσεις. Αντιθέτως, οι αλγόριθμοι CSA και BCA συνεχίζουν να έχουν αποδεκτούς χρόνους εκτέλεσης ακόμα και για μεγαλύτερο αριθμό διαστάσεων.

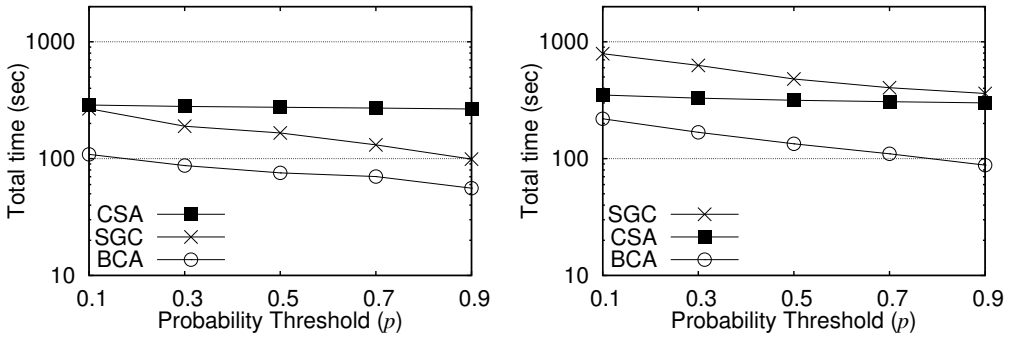
Επίδοση σε σχέση με το μέγεθος του πεδίου τιμών των υποκειμενικών γνωρισμάτων. Το Σχήμα 4.8 δείχνει τον συνολικό χρόνο επεξεργασίας ως συνάρτηση του μεγέθους του πεδίου τιμών των υποκειμενικών γνωρισμάτων. Το κόστος του προτεινόμενου αλγορίθμου για μη δεικτοδοτούμενα δεδομένα CSA παραμένει σε μεγάλο βαθμό ανεπηρέαστο από το μέγεθος του πεδίου τιμών $|RP|$. Αυτό συμβαίνει επειδή, αν και το μέγεθος των εγγραφών που ανήκουν στην κορυφογραμμή αυξάνεται με το μέγεθος του $|RP|$, οι εγγραφές αυτές εξακολουθούν να χωρούν στην κύρια μνήμη. Συνεπώς ο αλγόριθμος CSA χρειάζεται μόνο μια σάρωση της βάσης δεδομένων. Αντιθέτως, ο αλγόριθμος BCA χρειάζεται να εκτελέσει περισσότερες λειτουργίες εισόδου/εξόδου και απαιτεί μεγαλύτερο χρόνο επεξεργασίας όταν το μέγεθος του $|RP|$ μεγαλώνει, επειδή ο μεγαλύτερος αριθμός υποψήφιων αποτελεσμάτων οδηγεί σε αύξηση του μεγέθους του σωρού. Όπως περιγράψαμε και προηγουμένως στο Σχήμα 4.4(β′), η επίδοση του αλγορίθμου SGC είναι ιδιαίτερα ευαίσθητη στο μέγεθος του συνόλου $\mathcal{C}$, και επομένως η αύξηση των υποψήφιων αποτελεσμάτων οδηγεί σε κατακόρυφη αύξηση του συνολικού χρόνου και για τις δύο κατανομές δεδομένων.

Επίδοση σε σχέση με την τιμή κατωφλίου πιθανότητας. Το Σχήμα 4.9 δείχνει τον συνολικό χρόνο επεξεργασίας ως συνάρτηση της τιμής κατωφλίου p. Σημειώνουμε ότι η τιμή του κατωφλίου δεν έχει καμιά επίδραση στον αριθμό υποψήφιων αποτελεσμάτων. Όμως μεγαλύτερες τιμές κατωφλίου έχουν αποτέλεσμα οι υποψήφιες εγγραφές να μπορούν να απορριφθούν πιο γρήγορα. Η συγκεκριμένη τάση παρατηρείται στους αλγορίθμους SGC και BCA και για τις δύο κατανομές δεδομένων. Αντιθέτως αν υπάρχει έστω και μια υποψήφια εγγραφή που μπορεί να ανήκει στο αποτέλεσμα, ο αλγόριθμος CSA υποχρεούται να σαρώσει ολόκληρη τη βάση δεδομένων. Αυτό συμβαίνει σε όλα τα σενάρια που εξετάζονται στο Σχήμα 4.9. Επομένως, ο αριθμός λειτουργιών εισόδου/εξόδου που απαιτούνται από τον αλγόριθμο CSA δεν αλλάζει, μειώνεται όμως ο χρόνος επεξεργασίας που απαιτείται. Σε σχέση με το συνολικό κόστος εκτέλεσης, παρατηρούμε μόνο οριακή μείωση καθώς αυξάνεται η τιμή p.



(α′) uniform dataset (β′) anticorrelated dataset

Σχήμα 4.8: *Συνολικό κόστος σε σχέση με το μέγεθος του πεδίου τιμών των υποκειμενικών γνωρισμάτων*

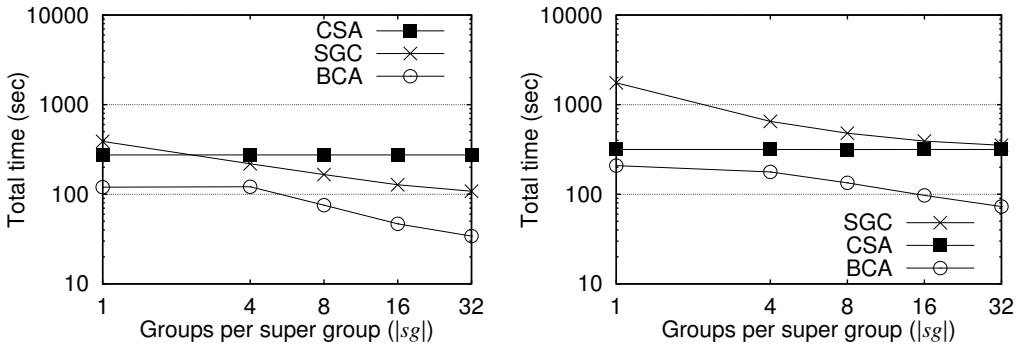


(α′) uniform dataset (β′) anticorrelated dataset

Σχήμα 4.9: *Συνολικό κόστος σε σχέση με την τιμή κατωφλίου πιθανότητας*

Επίδοση σε σχέση με τον αριθμό ομάδων ανά υπερ-ομάδα. Στο τελευταίο πείραμα (Σχήμα 4.10) εξετάζουμε τον συνολικό χρόνο επεξεργασίας των αλγορίθμων για δεικτοδοτούμενα δεδομένα, καθώς μεταβάλλουμε τον αριθμό ομάδων $|sg|$ που ανατίθενται σε κάθε υπερ-ομάδα. Για λόγους σύγκρισης, στο Σχήμα 4.10 έχουμε συμπεριλάβει επίσης τον χρόνο που απαιτείται για την εκτέλεση του αλγορίθμου CSA, ο οποίος βεβαίως είναι ανεξάρτητος από τον αριθμό ομάδων ανά υπερ-ομάδα. Η τιμή $|sg| = 1$ αντιστοιχεί σε μια ομάδα ανά υπερ-ομάδα, που σημαίνει ουσιαστικά ότι ο αλγόριθμος SGC εκφυλίζεται στον αλγόριθμο BGC. Επίσης είναι προφανές ότι σε όλες τις περιπτώσεις όπου $|sg| > 1$, ο αλγόριθμος BGC είναι υποδεέστερος του SGC. Γενικότερα και οι δύο εξεταζόμενοι αλγόριθμοι SGC και BCA επωφελούνται από την ύπαρξη λιγότερων aR-trees όπως εξηγήσαμε στην ενότητα 4.3.2. Στο βασικό σενάριο των πειραμάτων έχουμε ένα μόνο υποκειμενικό γνώρισμα με μέγεθος πεδίου ορισμού των αντικειμενικών γνωρισμάτων ίσο με 32. Αυτό σημαίνει ότι για την περίπτωση όπου $|sg| = 32$ ουσιαστικά υπάρχει μόνο ένα aR-tree το οποίο δεικτοδοτεί το σύνολο των δεδομένων. Σε αυτό το ακραίο σενάριο, οι αλγόριθμοι SGC και BCA επιτυγχάνουν τη βέλτιστη επίδοση.

Εν κατακλείδι μπορούμε να κάνουμε τις ακόλουθες παρατηρήσεις σε σχέση με τους προτεινόμενους αλγορίθμους για την επεξεργασία ερωτημάτων p-CSQ. Ο βασικός αλ-



(α′) uniform dataset (β′) anticorrelated dataset

Σχήμα 4.10: *Συνολικό κόστος σε σχέση με τον αριθμό ομάδων ανά υπερ-ομάδα*

γόριθμος BIA δεν είναι πρακτικά εφαρμόσιμος στις περισσότερες των περιπτώσεων. Ο προτεινόμενος αλγόριθμος για μη δεικτοδοτούμενα δεδομένα CSA αποδίδει αρκετά καλά για σύνολα δεδομένων μέσου μεγέθους και παραμένει αποδοτικός ακόμα και όταν το μέγεθος του πεδίου τιμών για τα υποκειμενικά γνωρίσματα είναι αρκετά μεγάλο. Ο αλγόριθμος SGC εμφανίζει χαμηλούς χρόνους επεξεργασίας, όμως απαιτεί σχετικά μεγάλο αριθμό λειτουργιών εισόδου/εξόδου και επομένως είναι ελκυστική επιλογή μόνο στην περίπτωση που ο αριθμός υποψήφιων αποτελεσμάτων είναι μικρός, όπως για παράδειγμα για σύνολα δεδομένων που ακολουθούν την κανονική κατανομή, ή έχουν μικρό μέγεθος ή μικρό αριθμό διαστάσεων. Ο αλγόριθμος BCA είναι σε γενικές γραμμές ο πιο αποδοτικός καθώς εμφανίζει σταθερά καλή επίδοση στα περισσότερα από τα σενάρια πειραμάτων που εξετάστηκαν.

Κεφάλαιο 5

Αλγόριθμοι Εξατομίκευσης από την πλευρά των παρόχων

Στην ενότητα 5.1 παραθέτουμε κάποιους βασικούς ορισμούς και ιδιότητες και περιγράφουμε τον πιο αποδοτικό αλγόριθμο που έχει προταθεί στην έως τώρα βιβλιογραφία των αντίστροφων ερωτημάτων κορυφογραμμής. Στην ενότητα 5.2 προτείνουμε έναν νέο αλγόριθμο, ονόματι RSA. Στη συνέχεια στην ενότητα 5.3 ορίζουμε τα ερωτήματα k-MAC και περιγράφουμε έναν άπληστο αλγόριθμο που βρίσκει τα καλύτερα k υποψήφια προϊόντα. Έπειτα στην ενότητα 5.4 παρουσιάζουμε έναν νέο αλγόριθμο για την αποτίμηση πολλαπλών αντίστροφων ερωτημάτων κορυφογραμμής. Τέλος η ενότητα 5.5 περιέχει την πειραματική αξιολόγηση των προτεινόμενων μεθόδων.

5.1 Βασικοί Ορισμοί

Στην ενότητα 5.1.1 δίνουμε κάποιους βασικούς ορισμούς για τα αντίστροφα ερωτήματα κορυφογραμμής. Στην ενότητα 5.1.2 παρουσιάζουμε τις έννοια του συνόλου και της ζώνης επιρροής και παραθέτουμε κάποιες σημαντικές ιδιότητες που ισχύουν. Τέλος, στην ενότητα 5.1.3 περιγράφουμε τον πιο αποδοτικό αλγόριθμο που έχει προταθεί στην έως σήμερα βιβλιογραφία των αντίστροφων ερωτημάτων κορυφογραμμής. Ο Πίνακας 5.1 συνοψίζει κάποια συχνά χρησιμοποιούμενα σύμβολα και όρους.

5.1.1 Αντίστροφα Ερωτήματα Κορυφογραμμής

Έστω P και C δύο σύνολα εγγραφών ενός πίνακα μιας βάσης δεδομένων που αποτελείται από ένα σύνολο γνωρισμάτων $\mathbf{A} = \{A_1, \ldots, A_D\}$. Μπορούμε εναλλακτικά να θεωρήσουμε κάθε εγγραφή ως σημείο σε έναν πολυδιάστατο χώρο D διαστάσεων[1]. Με μικρά γράμματα θα συμβολίζουμε μια εγγραφή που ανήκει στο αντίστοιχο σύνολο με κεφαλαία γράμματα, π.χ. $p \in P$. Θα αναφερόμαστε σε κάθε εγγραφή $p \in P$ ως *προϊόν*, ενώ με p_i θα συμβολίζουμε την τιμή του γνωρίσματος A_i για το προϊόν p. Επιπλέον κάθε εγγραφή $c \in C$ αναπαριστά τα ιδανικά χαρακτηριστικά ενός προϊόντος

[1] Στα επόμενα θα χρησιμοποιούμε τους όρους γνώρισμα και διάσταση χωρίς διάκριση.

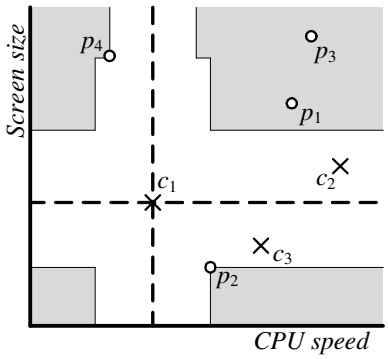
Σύμβολο	Ορισμός
P, C, Q	σύνολα προϊόντων, προτιμήσεων και υποψηφίων
$p_i \prec_c p_j$	το προϊόν p_i κυριαρχεί έναντι του p_j ως προς τις προτιμήσεις του καταναλωτή c
$m_i(q)$	ενδιάμεσο σημείο (midpoint) του p_i ως προς q
T_P, T_C	R-tree στο σύνολο προϊόντων και καταναλωτών αντίστοιχα
e_p, e_c	κόμβος που ανήκει στο T_P, T_C αντίστοιχα
$e_x^-(q)$, $e_x^+(q)$	min-corner του e_x (ως προς q)
$SKY(c)$	κορυφογραμμή ως προς c
$RSKY(q)$	σύνολο επιρροής του q
$IS(q)$	σκορ επιρροής του q
$IR(q)$	ζώνη επιρροής του q
$E_P(q)$, $E_C(q)$	ουρές προτεραιότητας ως προς q
L, U	σύνολα min-corners, minmax-corners για ένα σύνολο $E_P(q)$
$IR^-(q)$, $IR^+(q)$	κάτω και άνω όριο της ζώνης επιρροής του IR (q)
$RSKY(Q)$	από κοινού σύνολο επιρροής του συνόλου Q
$IS(Q)$	από κοινού σκορ επιρροής του συνόλου Q

Πίνακας 5.1: *Πίνακας συμβόλων*

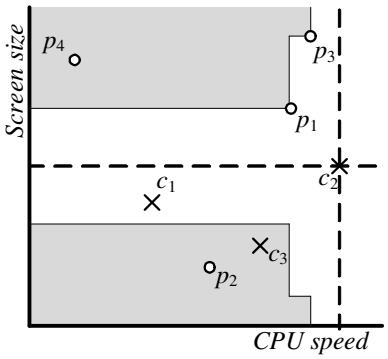
σύμφωνα με κάποιον καταναλωτή. Θα αναφερόμαστε εν συντομία σε κάθε εγγραφή c ως *καταναλωτή*.

Για παράδειγμα, ας υποθέσουμε ότι οι εγγραφές της βάσης δεδομένων περιέχουν πληροφορίες για τα χαρακτηριστικά διαθέσιμων μοντέλων φορητών υπολογιστών. Στην περίπτωση αυτή, πιθανά γνωρίσματα (διαστάσεις) μπορεί να είναι η τιμή, το βάρος, το μέγεθος της οθόνης, το μέγεθος της μνήμης, κλπ. Όπως είδαμε και στο Κεφάλαιο 4 κάποια από τα γνωρίσματα έχουν αντικειμενικά βέλτιστες τιμές. Για παράδειγμα μεταξύ δύο μοντέλων με ακριβώς ίδια χαρακτηριστικά, ένας αγοραστής θα προτιμήσει πάντα το φτηνότερο ή το ελαφρύτερο μοντέλο. Είναι σαφές όμως ότι για κάποια γνωρίσματα η βέλτιστη τιμή είναι υποκειμενική. Για παράδειγμα, μια μεγάλη οθόνη είναι πιο πρακτική αλλά ταυτόχρονα δυσχεραίνει τη φορητότητα ενός υπολογιστή. Παρομοίως, ένας πολύ γρήγορος επεξεργαστής συνήθως προκαλεί περισσότερη θερμότητα και θόρυβο ενώ παράλληλα μειώνει την αυτονομία. Κάποιοι αγοραστές προτιμούν έναν ισχυρό και μεγαλύτερου μεγέθους φορητό υπολογιστή (desktop replacement laptop), ενώ κάποιοι άλλοι ένα μοντέλο με λιγότερη ισχύ και μικρότερο μέγεθος (π.χ. netbook ή tablet). Προφανώς, ένας αγοραστής είναι πιο πιθανό να ενδιαφερθεί περισσότερο για προϊόντα τα οποία ταιριάζουν αρκετά με τις προτιμήσεις του. Λαμβάνοντας υπόψη την ύπαρξη προτιμήσεων για τα υποκειμενικά γνωρίσματα μιας σχέσης, ακολούθως παραθέτουμε τον ορισμό της *δυναμικής κυριαρχίας (dynamic dominance)*, όπως δίνεται στην εργασία [24].

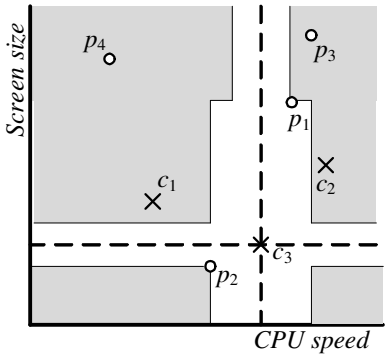
Ορισμός 5.4. *(Δυναμική Κυριαρχία): Έστω δύο σημεία $c \in C$, $p, p' \in P$. Θα λέμε ότι ένα προϊόν p κυριαρχεί δυναμικά επί ενός προϊόντος p' ως προς τις προτιμήσεις ενός καταναλωτή c, και θα συμβολίζουμε με $p \prec_c p'$, αν για κάθε διάσταση ισχύει $|p_i - c_i| \leq |p'_i - c_i|$ και υπάρχει τουλάχιστον μια διάσταση τέτοια ώστε $|p_i - c_i| < |p'_i - c_i|$.*



(α′) Δυναμικό Ερώτημα Κορυφογραμμής ως προς το c_1



(β′) Δυναμικό Ερώτημα Κορυφογραμμής ως προς το c_2



(γ′) Δυναμικό Ερώτημα Κορυφογραμμής ως προς το c_3

Products p_i
Customers c_i
SKY(c_1): {p_2, p_4}
SKY(c_2): {p_1, p_3}
SKY(c_3): {p_1, p_2}
RSKY(p_1): {c_2, c_3}
RSKY(p_2): {c_1,c_3}
RSKY(p_3): {c_2}
RSKY(p_4): {c_1}

(δ′) Ερωτήματα Κορυφογραμμής και Σύνολα Επιρροής

Σχήμα 5.1: *Παράδειγμα Δυναμικού Ερωτήματος Κορυφογραμμής*

Σημειώνουμε ότι ο συγκεκριμένος ορισμός μπορεί να καλύψει και τις αντικειμενικές διαστάσεις, κάνοντας την παραδοχή ότι οι μικρότερες τιμές είναι προτιμότερες και θέτοντας την προτίμηση c_i στην ελάχιστη τιμή για το γνώρισμα A_i. Για παράδειγμα, θεωρώντας ότι ένας ελαφρύτερος υπολογιστής είναι πάντα προτιμότερος μπορούμε απλά να θεωρήσουμε ότι για όλους τους καταναλωτές ισχύει η προτίμηση $c_{weight} = 0$. Στη συνέχεια παραθέτουμε τον ορισμό ενός *δυναμικού ερωτήματος κορυφογραμμής (dynamic skyline query)* (από την εργασία [24]).

Ορισμός 5.5. *(Δυναμικό Ερώτημα Κορυφογραμμής): Ένα δυναμικό ερώτημα κορυφογραμμής ως προς τις προτιμήσεις ενός καταναλωτή $c \in C$, το οποίο θα συμβολίζουμε ως $SKY(c)$, επιστρέφει όλα τα προϊόντα $p \in P$ τα οποία δεν κυριαρχούνται δυναμικά ως προς c από κάποιο άλλο προϊόν $p' \in P$.*

Έστω ένα σύνολο προϊόντων $P = \{p_1, p_2, p_3, p_4\}$ και ένα σύνολο καταναλωτών $C = \{c_1, c_2, c_3\}$. Το Σχήμα 5.1(α′) δείχνει ένα δυναμικό ερώτημα κορυφογραμμής ως προς

το σημείο c_1 θεωρώντας δύο διαστάσεις: επεξεργαστική ισχύς και μέγεθος οθόνης. Το αποτέλεσμα του ερωτήματος αποτελείται από τα προϊόντα p_2 και p_4. Τα σημεία του σχήματος που βρίσκονται μέσα σε γκρίζες περιοχές κυριαρχούνται δυναμικά από σημεία που ανήκουν στη δυναμική κορυφογραμμή ως προς c_1. Επειδή μας ενδιαφέρει μόνο η απόλυτη διαφορά των τιμών ενός γνωρίσματος, ένα προϊόν είναι δυνατόν να κυριαρχεί έναντι κάποιου προϊόντος που βρίσκεται σε διαφορετικό τεταρτημόριο ως προς το c_1. Για παράδειγμα τα σημεία p_1 και p_3 στο πάνω δεξιά τεταρτημόριο κυριαρχούνται δυναμικά από το σημείο p_2 που ανήκει στο κάτω δεξιά τεταρτημόριο, καθώς το p_2 έχει επεξεργαστική ισχύ και μέγεθος οθόνης που είναι και τα δύο πιο κοντά στις προτιμήσεις του c_1 από τα αντίστοιχα χαρακτηριστικά των σημείων p_1 και p_3. Τα Σχήματα 5.1(β′) και 5.1(γ′) δείχνουν δυναμικά ερωτήματα κορυφογραμμής ως προς τα σημεία c_2 και c_3 αντιστοίχως.

Στη συνέχεια παρουσιάζουμε το πρόβλημα από την αντίστροφη πλευρά, δηλαδή ενός προϊόντος, παραθέτοντας τον ορισμό ενός *διχρωματικού αντίστροφου ερωτήματος κορυφογραμμής (bichromatic reverse skyline query)*[2] όπως δίνεται στην εργασία [48].

Ορισμός 5.6. *(Διχρωματικό Αντίστροφο Ερώτημα Κορυφογραμμής): Έστω P και C δύο σύνολα προϊόντων και καταναλωτών αντίστοιχα. Ένα διχρωματικό αντίστροφο ερώτημα κορυφογραμμής ως προς ένα σημείο $p \in P$, το οποίο θα συμβολίζουμε ως $RSKY(p)$, επιστρέφει όλα τα σημεία $c \in C$ τέτοια ώστε $p \in SKY(c)$, δηλαδή το σημείο p ανήκει στο δυναμικό ερώτημα κορυφογραμμής ως προς το σημείο c.*

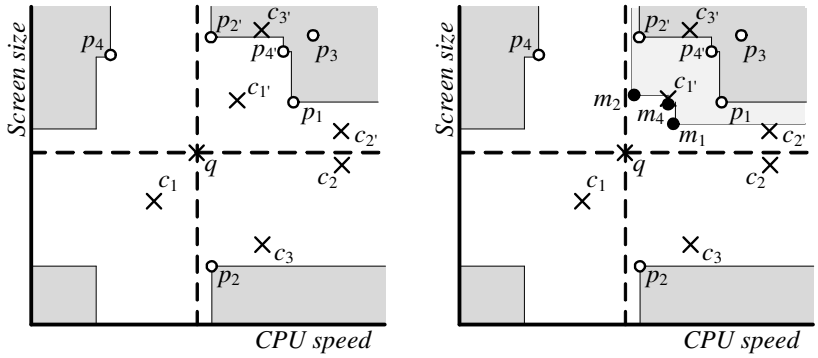
Με άλλα λόγια, ένα διχρωματικό αντίστροφο ερώτημα κορυφογραμμής ως προς ένα προϊόν p επιστρέφει όλους τους καταναλωτές $c \in C$ οι οποίοι βρίσκουν το σημείο p ως 'ελκυστικό'. Στη συνέχεια θα αναφερόμαστε στο αποτέλεσμα ενός διχρωματικού αντίστροφου ερωτήματος κορυφογραμμής ως προς p ως το *σύνολο επιρροής (influence set)* ως προς p. Στο Σχήμα 5.1(δ′) φαίνονται τα σύνολα επιρροής των σημείων p_1, p_2, p_3 και p_4.

Το μέγεθος του συνόλου επιρροής ενός προϊόντος p, $RSKY(p)$, μπορεί να θεωρηθεί ως ένας τρόπος μέτρησης της απήχησης του προϊόντος στην αγορά. Θα αναφερόμαστε στο μέγεθος του συνόλου επιρροής ενός προϊόντος p, $|RSKY(p)|$, ως *σκορ επιρροής (influence score)* και θα το συμβολίζουμε ως IS (p). Στο παράδειγμα του Σχήματος 5.1, IS (p_1) = IS (p_2) = 2 και IS (p_3) = IS (p_4) = 1.

5.1.2 Περιοχή Επιρροής

Έστω ένα σημείο (προϊόν) q. Το σημείο αυτό χωρίζει έναν χώρο D σε 2^D τεταρτημόρια Ω_i, όπου το καθένα μπορεί να προσδιοριστεί με τη βοήθεια ενός αριθμού στην περιοχή $[0, 2^D - 1]$. Στο παράδειγμα του Σχήματος 5.2 όπου $D = 2$, το σημείο q χωρίζει τον χώρο σε 4 τεταρτημόρια. Επειδή στην περίπτωση των δυναμικών ερωτημάτων κορυφογραμμής μας ενδιαφέρει μόνο η απόλυτη διαφορά τιμών των γνωρισμάτων, μπορούμε

[2]Ο όρος 'διχρωματικό' χρησιμοποιείται για να διασαφηνίσει ότι το σημείο αναφοράς και τα σημεία από τα οποία προέρχονται τα αποτελέσματα προέρχονται από διαφορετικά σύνολα δεδομένων, εδώ προϊόντα και καταναλωτικές προτιμήσεις. Σε αντιδιαστολή, σε ένα μονοχρωματικό ερώτημα τόσο το σημείο αναφοράς όσο και τα αποτελέσματα προέρχονται από το ίδιο σύνολο δεδομένων.



(α′) Μετασχηματισμένος χώρος ως προς το 1ο τεταρτημόριο Ω_0 του σημείου q

(β′) Ενδιάμεσα σημεία (midpoints) ως προς το q

Σχήμα 5.2: *Περιοχή επιρροής ως προς q*

να μετασχηματίσουμε τα σημεία από όλα τα τεταρτημόρια στο 1ο τεταρτημόριο Ω_0 όπως δείχνει το Σχήμα 5.2(α′). Για λόγους ευκολότερης παρουσίασης, στη συνέχεια επικεντρωνόμαστε στο πρώτο τεταρτημόριο Ω_0 ως προς ένα σημείο αναφοράς q.

Για κάθε σημείο p_i που ανήκει στην κορυφογραμμή ως προς το q, έστω $m_i(q)$ το σημείο που βρίσκεται στο μέσο του ευθυγράμμου τμήματος που συνδέει το q με το p_i. Στο Σχήμα 5.2(β′), τα σημεία που αναπαρίστανται με μαύρο στίγμα m_1, m_2 και m_4 αντιπροσωπεύουν τα ενδιάμεσα σημεία των p_1, p_2 και p_4 ως προς q. Στη συνέχεια, όποτε αναφερόμαστε σε ένα προϊόν p_i θα εννοούμε το αντίστοιχο ενδιάμεσο σημείο $m_i(q)$ ως προς q. Επίσης θα θεωρήσουμε ότι κάθε σημείο p_i μπορεί να μετασχηματιστεί με έναν απλό υπολογισμό στο αντίστοιχο $m_i(q)$ κατά την επεξεργασία ενός ερωτήματος κορυφογραμμής.

Ενώνοντας όλες τις περιοχές του χώρου οι οποίες δεν κυριαρχούνται δυναμικά ως προς q από τα ενδιάμεσα σημεία κορυφογραμμής του q, προκύπτει η λεγόμενη *ζώνη επιρροής (influence region)* του σημείου q, την οποία θα συμβολίζουμε με IR (q). Στο Σχήμα 5.2(β′) η μη σκιασμένη περιοχή στο πρώτο τεταρτημόριο Ω_0 δείχνει την ζώνη επιρροής του q. Να σημειωθεί ότι τα σημεία που ανήκουν στην κορυφογραμμή δεν αποτελούν τα ίδια κομμάτι της ζώνης επιρροής, καθώς σύμφωνα με τον ορισμό της κυριαρχίας, δύο σημεία με ίσες τιμές σε κάθε διάσταση δεν κυριαρχούν το ένα έναντι του άλλου. Στη συνέχεια παραθέτουμε μια χρήσιμη ιδιότητα που ισχύει σε σχέση με τη ζώνη επιρροής, όπως περιγράφεται στην εργασία [48].

Ιδιότητα 7. *Ένα σημείο (καταναλωτής) c ανήκει στο σύνολο επιρροής $RSKY(q)$ ενός προϊόντος q αν και μόνο αν το σημείο c ανήκει στην περιοχή επιρροής ως προς q δηλαδή, $c \in IR\ (q) \Leftrightarrow c \in RSKY(q)$.*

Επιστρέφοντας στο παράδειγμα του Σχήματος 5.2(β′), παρατηρούμε ότι μόνο το σημείο c_2 ανήκει στο IR (q). Επομένως $RSKY(q) = \{c_2\}$.

Στη συνέχεια θα θεωρήσουμε ότι όλα τα σημεία (είτε ανήκουν στο σύνολο προϊόντων είτε στο σύνολο καταναλωτών) δεικτοδοτούνται με τη βοήθεια ενός R-tree. Σε ένα R-tree, τα σημεία με παρόμοιες τιμές γνωρισμάτων (διαστάσεων) ομαδοποιούνται και

ανατίθενται σε κόμβους. Κάθε κόμβος περιέχει ένα minimum bounding box (MBB) το οποίο περικλείει ένα πλήθος σημείων των οποίων οι ακριβείς τιμές δεν είναι γνωστές στον συγκεκριμένο κόμβο. Το Σχήμα 5.3(α′) δείχνει ένα MBB e. Θα αναφερόμαστε ως min-corner και θα συμβολίζουμε με $e^-(q)$ τη κορυφή του MBB η οποία έχει την ελάχιστη Ευκλείδεια απόσταση από το σημείο q. Μεταξύ των σημείων του MBB, το σημείο $e^-(q)$ είναι αυτό που κυριαρχεί επί του μεγαλύτερου κομματιού του πολυδιάστατου χώρου. Επιπλέον, εξαιτίας του τρόπου κατασκευής ενός MBB το οποίο αποτελεί το μικρότερο δυνατό παραλληλεπίπεδο που περικλείει ένα σύνολο σημείων, κάθε πλευρά του MBB πρέπει να περιέχει τουλάχιστον ένα σημείο. Στη χειρότερη περίπτωση το σημείο αυτό θα βρίσκεται σε κάποια από τις κορυφές. Θα αναφερόμαστε στις κορυφές που ανήκουν στις d πλευρές του MBB όπου ανήκει και το q και που είναι ακριβώς αντιδιαμετρικά του q ως minmax-corners. Κάθε MBB περιέχει ακριβώς d τέτοιες κορυφές. Ανεξαρτήτως της κατανομής των σημείων μέσα σε ένα MBB e, κάθε σημείο στο e κυριαρχεί επί της περιοχής που κυριαρχεί οποιοδήποτε από τα minmax-corners, ενώ στην καλύτερη περίπτωση κυριαρχεί της περιοχής όπου κυριαρχεί το min-corner.

Δοθέντος ενός συνόλου MBB μπορούμε να εξάγουμε δύο σύνολα: ένα σύνολο L που περιέχει όλα τα min-corners και ένα σύνολο U που περιέχει όλα τα minmax-corners ως προς q. Το Σχήμα 5.3(β′) δείχνει ένα τέτοιο παράδειγμα για το σύνολο κόμβων $E_P = \{e_{p1}, e_{p2}, e_{p3}, e_{p4}\}$ όπου το e_{p_i} συμβολίζει ένα κόμβο που περιέχει σημεία-προϊόντα. Στα Σχήματα 5.3(β′) και 5.3(γ′) τα παραλληλεπίπεδα αντιπροσωπεύουν midpoints και τα σημεία με μαύρα και κούφια στίγματα δείχνουν τα αντίστοιχα min-corners και minmax-corners.

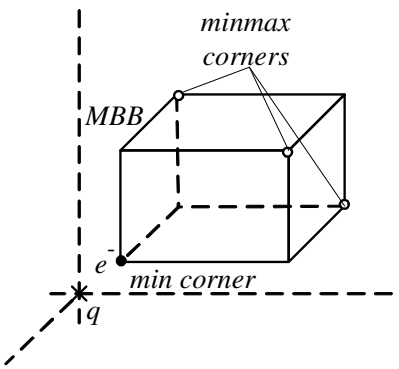
Συνεχίζοντας το παράδειγμα του Σχήματος 5.3(β′), η γκρίζα περιοχή αντιστοιχεί σε ένα κάτω όριο της ζώνης επιρροής του q, $\mathrm{IR}^-(q)$, και ορίζεται ως η περιοχή που δεν κυριαρχείται από κανένα από τα min-corners που ανήκουν στο σύνολο L. Παρομοίως, η γκρίζα περιοχή στο Σχήμα 5.3(γ′) αντιστοιχεί σε ένα άνω όριο της ζώνης επιρροής του q, $\mathrm{IR}^+(q)$, και ορίζεται ως η περιοχή που δεν κυριαρχείται από κανένα από τα minmax-corners που ανήκουν στο σύνολο U. Σύμφωνα με την εργασία [81] ισχύει η παρακάτω ιδιότητα:

Ιδιότητα 8. *Αν ένας κόμβος e_c κυριαρχείται από κάποιο σημείο $u \in U$, δηλαδή ο κόμβος e_c βρίσκεται εντελώς εκτός του άνω ορίου της ζώνης επιρροής $IR^+(q)$, τότε ο κόμβος e_c δεν είναι δυνατόν να περιέχει κανένα σημείο μέσα στην (πραγματική) ζώνη επιρροής IR (q). Επομένως, σύμφωνα με την Ιδιότητα 7, ο κόμβος e_c μπορεί να κλαδευτεί.*

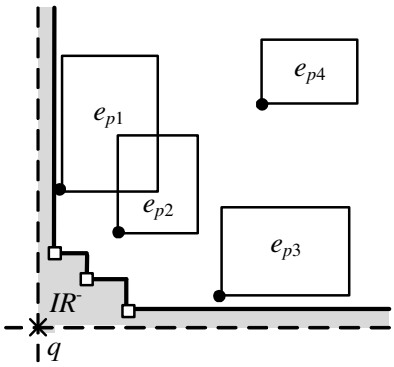
Για παράδειγμα ο κόμβος e_{c1} στο Σχήμα 5.3(δ′) μπορεί να κλαδευτεί αφού βρίσκεται εντελώς εκτός του $\mathrm{IR}^+(q)$.

5.1.3 Ο αλγόριθμος BRS

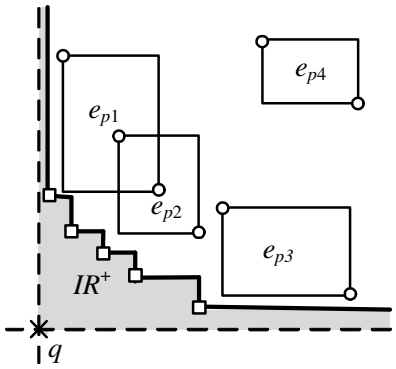
Στη συνέχεια παρουσιάζουμε εν συντομία τον πιο αποδοτικό αλγόριθμο που έχει προταθεί στην έως τώρα βιβλιογραφία των αντίστροφων ερωτημάτων κορυφογραμμής, με την ονομασία Bichromatic Reverse Skyline - BRS [81]. Ο αλγόριθμος BRS στοχεύει στην ελαχιστοποίηση του αριθμού λειτουργιών εισόδου/εξόδου (I/Os) (α) εκλεπτύνοντας προοδευτικά τα όρια της ζώνης επιρροής ενός σημείου q έως ώτου εξαχθεί η



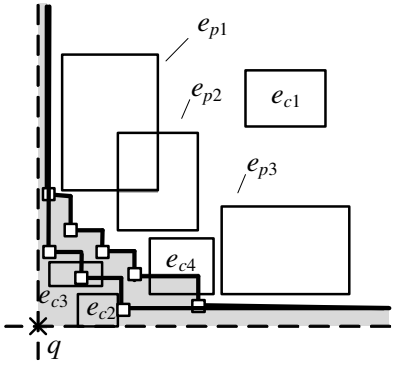
(α′) Παράδειγμα MBB



(β′) Κάτω όριο της ζώνης επιρροής IR (q)



(γ′) Άνω όριο της ζώνης επιρροής IR (q)



(δ′) Παράδειγμα κλαδέματος κόμβου

Σχήμα 5.3: *Ζώνες επιρροής*

τελική ζώνη επιρροής, και (β) εφαρμόζοντας την Ιδιότητα 8 για να κλαδέψει κάποιους κόμβους που δεν συνεισφέρουν στο σύνολο επιρροής $RSKY(q)$.

Ο αλγόριθμος BRS χρησιμοποιεί δύο R-trees T_P και T_C τα οποία δεικτοδοτούν αντίστοιχα τα σύνολα δεδομένων P και C. Επιπλέον για κάθε σύνολο δεδομένων διατηρεί μια ουρά προτεραιότητας (σωρό) E_P και E_C αντίστοιχα η οποία είναι ταξινομημένη με κριτήριο την Ευκλείδεια απόσταση κάθε κόμβου από το σημείο q. Ο αλγόριθμος BRS εκτελείται σε επαναλήψεις. Αρχικά, ο αλγόριθμος προσθέτει τη ρίζα του δέντρου T_P (αντίστοιχα T_C) στην ουρά. Σε κάθε επανάληψη ο αλγόριθμος BRS κατασκευάζει τα σύνολα L και U που αποτελούνται από όλα τα min-corners και όλα τα minmax-corners κάθε $e_p \in E_P$. Στη συνέχεια υπολογίζει τις κορυφογραμμές των συνόλων L και U, τις οποίες συμβολίζουμε ως $SKY(L)$ και $SKY(U)$ αντίστοιχα.

Σε κάθε επανάληψη ο αλγόριθμος αφαιρεί από την ουρά προτεραιότητας E_P τον κόμβο με την μικρότερη Ευκλείδεια απόσταση από το σημείο q και ενημερώνει κατάλληλα τα τρέχοντα σύνολα L και U και τα αντίστοιχα σύνολα κορυφογραμμής $SKY(L)$ και $SKY(U)$. Έπειτα, για κάθε κόμβο $e_c \in E_C$ ελέγχει αν κυριαρχείται από τα $SKY(L)$ και $SKY(U)$. Αν ο κόμβος e_c δεν κυριαρχείται από το $SKY(L)$, δηλαδή έχει τομή με το κάτω όριο της ζώνης επιρροής του q, $\text{IR}^-(q)$, τότε ο αλγόριθμος προσπελαύνει τον

κόμβο e_c καθώς είναι πιθανό να περιέχει σημεία που να βρίσκονται εντός της ζώνης επιρροής IR (q). Στο παράδειγμα του Σχήματος 5.3(δ′), ο κόμβος e_{c3} τέμνει το $\text{IR}^-(q)$, επομένως θα πρέπει να προσπελαστεί και οι κόμβοι-παιδιά του να προστεθούν στην ουρά προτεραιότητας. Αντιθέτως, αν ένας κόμβος e_c κυριαρχείται από το $SKY(U)$ (όπως π.χ. ο κόμβος e_{c1} στο Σχήμα 5.3(δ′)), τότε ο κόμβος e_c μπορεί να κλαδευτεί βάση της Ιδιότητας 8. Ο αλγόριθμος BRS τερματίζει όταν αδειάσει η ουρά E_C, δηλαδή η ακριβής θέση κάθε κόμβου e_c έχει προσδιοριστεί, είτε εντός της ζώνης επιρροής IR (q) οπότε όλα τα φύλλα του υπόδεντρου ανήκουν στην αντίστροφη κορυφογραμμή ως προς q, είτε εκτός της ζώνης επιρροής IR (q) οπότε απορρίπτονται από το αποτέλεσμα.

5.2 Αποτίμηση Αντίστροφων Ερωτημάτων Κορυφογραμμής

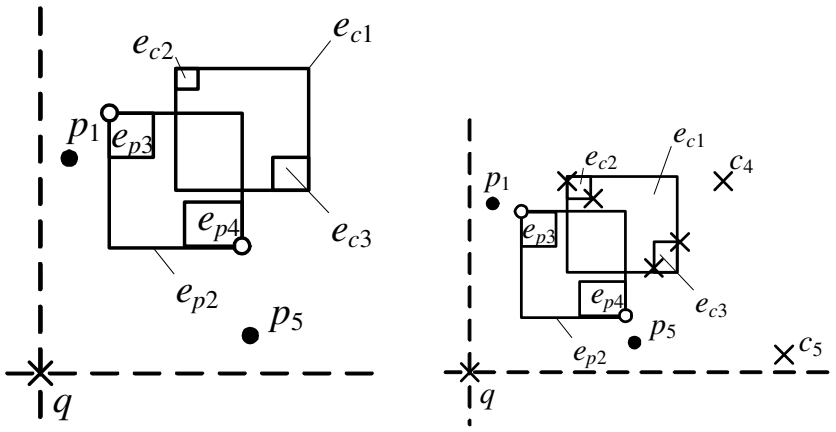
Σε αυτή την ενότητα αναλύουμε κάποιους περιορισμούς της προσέγγισης που ακολουθεί ο αλγόριθμος BRS και στη συνέχεια περιγράφουμε έναν πιο αποδοτικό αλγόριθμο για την αποτίμηση αντίστροφων ερωτημάτων κορυφογραμμής, με την ονομασία RSA.

5.2.1 Περιορισμοί του αλγορίθμου BRS

Ανάλυση πολυπλοκότητας. Έστω p_k και c_k τα τρέχοντα μεγέθη που αντιστοιχούν στις ουρές προτεραιότητας E_P και E_C κατά την k-οστή επανάληψη του αλγορίθμου BRS. Στην χειρότερη περίπτωση τα μεγέθη των p_k και c_k είναι ίσα με τα μεγέθη των συνόλων δεδομένων $|P|$ και $|C|$. Όπως περιγράψαμε παραπάνω, σε κάθε επανάληψη ο αλγόριθμος BRS διατηρεί τα σύνολα κορυφογραμμής $SKY(L)$ και $SKY(U)$ τα οποία έχουν μέγεθος $O(|P|)$ και $O(D|P|)$ αντιστοίχως, όπου D είναι ο αριθμός διαστάσεων. Ο αλγόριθμος BRS εκτελεί ελέγχους κυριαρχίας μεταξύ κάθε κόμβου που ανήκει στα E_P και E_C με τα σύνολα κορυφογραμμής $SKY(L)$ και $SKY(U)$. Επομένως, κάθε επανάληψη απαιτεί $O(D|P| \times (|P| + |C|))$ ελέγχους κυριαρχίας, ή αλλιώς $O(D^2|P| \times (|P| + |C|))$ συγκρίσεις, αφού κάθε έλεγχος κυριαρχίας ισοδυναμεί με $O(D)$ συγκρίσεις.

Όπως προκύπτει από την παραπάνω ανάλυση, το κόστος επεξεργασίας του αλγορίθμου BRS εξαρτάται ουσιαστικά από το μέγεθος των συνόλων κορυφογραμμής $SKY(L)$ και $SKY(U)$. Η εργασία [12] δείχνει ότι για ομοιόμορφα κατανεμημένα δεδομένα, το μέγεθος της κορυφογραμμής είναι $\Theta(\frac{(ln|P|)^{D-1}}{D!})$. Συνεπώς, το κόστος επεξεργασίας του αλγορίθμου BRS γίνεται ουσιαστικά απαγορευτικό για μεγαλύτερα σύνολα δεδομένων ή υψηλότερο αριθμό διαστάσεων. Η πειραματική μας αξιολόγηση (ενότητα 5.5) επιβεβαιώνει την παραπάνω ανάλυση. Πιο συγκεκριμένα τα πειράματά μας δείχνουν ότι το κόστος επεξεργασίας του αλγορίθμου BRS αυξάνεται δραστικά για $|P| \geq 10^6$ ή $D \geq 4$.

Με σκοπό να αντιμετωπίσουμε το πρόβλημα της κλιμάκωσης στην αποτίμηση αντίστροφων ερωτημάτων κορυφογραμμής, στη συνέχεια προτείνουμε έναν πιο αποδοτικό αλγόριθμο που ονομάζουμε RSA ο οποίος αποφεύγει σε μεγάλο βαθμό τον υπολογισμό των συνόλων κορυφογραμμής $SKY(L)$ και $SKY(U)$ και γι'αυτό το λόγο εμφανίζει



(α′) Ο αλγόριθμος BRS θα προσπελάσει τον κόμβο e_{p_2} εκτελώντας ένα περιττό I/O

(β′) Ο αλγόριθμος RSA αποφεύγει την προσπέλαση του κόμβου e_{p_2} κλαδεύοντας το c_4 με τη βοήθεια του σημείου p_1

Σχήμα 5.4: *Σειρά επεξεργασίας και προσπελάσεις στον δίσκο*

καλύτερη συμπεριφορά κλιμάκωσης για δεδομένα περισσότερων διαστάσεων και γενικότερα για δεδομένα με μεγάλο μέγεθος κορυφογραμμής.

Σειρά επεξεργασίας. Ο αλγόριθμος BRS διασχίζει με ασύγχρονο τρόπο τα R-trees T_P και T_C, ακολουθώντας μια διάταξη που βασίζεται στην Ευκλείδεια απόσταση κάθε κόμβου από το σημείο q. Η συγκεκριμένη σειρά επεξεργασίας διασφαλίζει την ελαχιστοποίηση των λειτουργιών εισόδου/εξόδου I/Os που απαιτούνται πάνω στον δείκτη T_P. Όμως, εστιάζοντας στον συνολικό αριθμό λειτουργιών εισόδου/εξόδου που απαιτούνται, ο αλγόριθμος BRS πολλές φορές εκτελεί κάποια περιττά I/Os. Το Σχήμα 5.4(α′) απεικονίζει μια τέτοια περίπτωση, όπου οι κόμβοι e_{p_2} και e_{c_1} δεν έχουν προσπελαστεί ακόμα.[3] Στο επόμενο βήμα εκτέλεσης ο αλγόριθμος BRS θα προσπελάσει τον κόμβο e_{p_2} και θα προσθέσει τους κόμβους - απογόνους e_{p_3} και e_{p_4} στον σωρό. Όμως, ο κόμβος e_{c_1} δεν επηρεάζεται από τη συγκεκριμένη προσπέλαση και συνεπώς θα πρέπει επίσης να προσπελαστεί. Αντιθέτως, αν προσπελάσουμε πρώτα τον κόμβο e_{c_1}, η συγκεκριμένη προσπέλαση θα βρει τους κόμβους - απογόνους e_{c_2} και e_{c_3} οι οποίοι κυριαρχούνται από τα σημεία p_1 και p_5 αντιστοίχως και συνεπώς μπορούν να κλαδευτούν. Με αυτό τον τρόπο αποφεύγεται η επιπλέον προσπέλαση του κόμβου e_{p_2}. Ο προτεινόμενος αλγόριθμος RSA ακολουθεί μια διαφορετική σειρά επεξεργασίας η οποία βασίζεται κυρίως στο επίπεδο κάθε κόμβου στο T_C. Όπως επιβεβαιώνεται και πειραματικά, η συγκεκριμένη σειρά απαιτεί συνολικά λιγότερες λειτουργίες εισόδου/εξόδου.

Προοδευτική παραγωγή αποτελεσμάτων. Ο αλγόριθμος BRS εκλεπτύνει προοδευτικά τα όρια της ζώνης επιρροής IR (q) και επιστρέφει όλα τα σημεία που ανήκουν στο σύνολο C και βρίσκονται εντός του κάτω ορίου $\text{IR}^-(q)$. Λόγω της σειράς επεξεργασίας που ακολουθεί ο BRS, συνήθως απαιτούνται αρκετές επαναλήψεις ώσπου να βρεθούν τα πρώτα αποτελέσματα, γεγονός το οποίο είναι μη επιθυμητό ιδιαιτέρως

[3] Θυμίζουμε ότι με e_p στην πραγματικότητα αναφερόμαστε στο αντίστοιχο midpoint ως προς q.

για εφαρμογές οι οποίες χρειάζονται μόνο ένα μέρος του συνόλου αποτελεσμάτων ή απαιτούν γρήγορη απόκριση. Ο αλγόριθμος RSA επιχειρεί να αντιμετωπίσει και το συγκεκριμένο πρόβλημα, παράγοντας τα πρώτα αποτελέσματα αρκετά ταχύτερα από τον αλγόριθμο BRS.

5.2.2 Ο αλγόριθμος RSA

Στη συνέχεια παρουσιάζουμε τον αλγόριθμο RSA (Reverse Skyline Algorithm), ο οποίος στοχεύει να αντιμετωπίσει τα προβλήματα που περιγράψαμε παραπάνω.
Γενική ιδέα. Ο αλγόριθμος RSA:

- Αποφεύγει τον υπολογισμό των $SKY(L)$ και $SKY(U)$ σε κάθε επανάληψη και συνεπώς έχει χαμηλότερο κόστος επεξεργασίας.
- Σε κάθε επανάληψη εξετάζει έναν κόμβο από την ουρά προτεραιότητας E_C ακολουθώντας μια σειρά επεξεργασίας που βασίζεται σε δύο κριτήρια: (α) στο επίπεδο του κόμβου στο δέντρο, και (β) στην Ευκλείδεια απόσταση του κόμβου από το σημείο q.
- Προσπελαύνει έναν κόμβο από την ουρά προτεραιότητας E_P μόνο αν η συγκεκριμένη προσπέλαση είναι απαραίτητη για να προσδιοριστεί αν ένα σημείο του συνόλου C ανήκει στο σύνολο επιρροής $RSKY(q)$.

Ο αλγόριθμος RSA συντηρεί δύο ουρές προτεραιότητας E_P και E_C καθώς και ένα σύνολο $SKY(q)$ που περιέχει όσα midpoint skylines έχουν βρεθεί έως την τρέχουσα επανάληψη. Οι δύο ουρές προτεραιότητας είναι ταξινομημένες με βάση δύο κριτήρια: αρχικά με βάση το επίπεδο του κόμβου στο αντίστοιχο R-tree, και δευτερευόντως με βάση την Ευκλείδεια απόσταση του κόμβου από το σημείο q. Έτσι, οι κόμβοι φύλλα έχουν πάντοτε υψηλότερη προτεραιότητα και εξετάζονται πρώτοι, ενώ η προσπέλαση των ενδιάμεσων κόμβων παρατείνεται όσο το δυνατόν για αργότερα. Ακολουθώντας τη συγκεκριμένη σειρά επεξεργασίας είναι δυνατόν ένας κόμβος φύλλο να αποκαλύψει κάποιο midpoint skyline το οποίο στη συνέχεια μπορεί να χρησιμοποιηθεί για να κλαδέψει κάποιον ενδιάμεσο κόμβο e_c βάσει της Ιδιότητας 8. Η ίδια λογική ισχύει και για κόμβους e_p καθώς ένα midpoint skyline μπορεί επίσης να χρησιμοποιηθεί για να κλαδέψει κάποιον ενδιάμεσο κόμβο e_p αν ο κόμβος e_p δεν συνεισφέρει στην κορυφογραμμή. Αρχικά, ο τρέχων εξεταζόμενος κόμβος e_c ελέγχεται για κυριαρχία με το $SKY(q)$ και στη συνέχεια με όλους τους κόμβους φύλλα που ανήκουν στην ουρά προτεραιότητας E_P. Αν δεν υπάρχει φύλλο που να κυριαρχεί επί του κόμβου e_c, τότε ο αλγόριθμος RSA αναγκαστικά θα προσπελάσει με τη σειρά τον επόμενο ενδιάμεσο κόμβο e_p. Η συγκεκριμένη αλλαγή στη σειρά επεξεργασίας μειώνει επίσης και τον αριθμό I/Os στο T_P. Για παράδειγμα, στο Σχήμα 5.4(β′) ο αλγόριθμος RSA θα εξετάσει πρώτα τον κόμβο-φύλλο p_1 και θα ανακαλύψει ότι ο συγκεκριμένος κόμβος κυριαρχεί επί του c_4, συνεπώς το σημείο c_4 δεν ανήκει στο σύνολο επιρροής $RSKY(q)$. Με αυτό τον τρόπο αποφεύγεται η προσπέλαση του κόμβου e_{p_2}.
Περιγραφή αλγορίθμου. Ο ψευδοκώδικας του αλγορίθμου RSA φαίνεται στη λίστα 4. Αρχικά ο αλγόριθμος RSA προσθέτει στην ουρά E_P (αντίστοιχα E_C) τους

Algorithm 4: RSA

```
Input: q a query point, T_P R-tree on products, T_C R-tree on customers, E_P(q) priority queue on
       products, E_C(q) priority queue on customers
Output: RSKY(q) reverse skylines of q
Variables: SKY(q) currently found midpoint skylines of products w.r.t. q
1  begin
2      SKY(q) := ∅; RSKY(q) := ∅;
3      while E_C ≠ ∅ do
4          dominated := false;
5          E_C(q).pop() → e_c;
6          if dominated(e_c, SKY(q)) then
7              dominated := true;
8              continue;
9          if e_c is a non-leaf entry then
10             Expand e_c, insert children entries in E_C(q);
11         else
12             foreach e_p ∈ E_P(q) do
13                 midpoint(e_p, q) → m;
14                 if e_c is dominated by m then
15                     if e_p is a leaf entry then
16                         if (dominated(m, SKY(q)) == false) then
17                             SKY(q).push(m);
18                         dominated := true;
19                         break;
20                     else
21                         Expand e_p, insert children entries in E_P(q);
22                     E_P(q).remove(e_p);
23             if (dominated == false) then
24                 RSKY(q).push(e_c);
25     return RSKY(q);
```

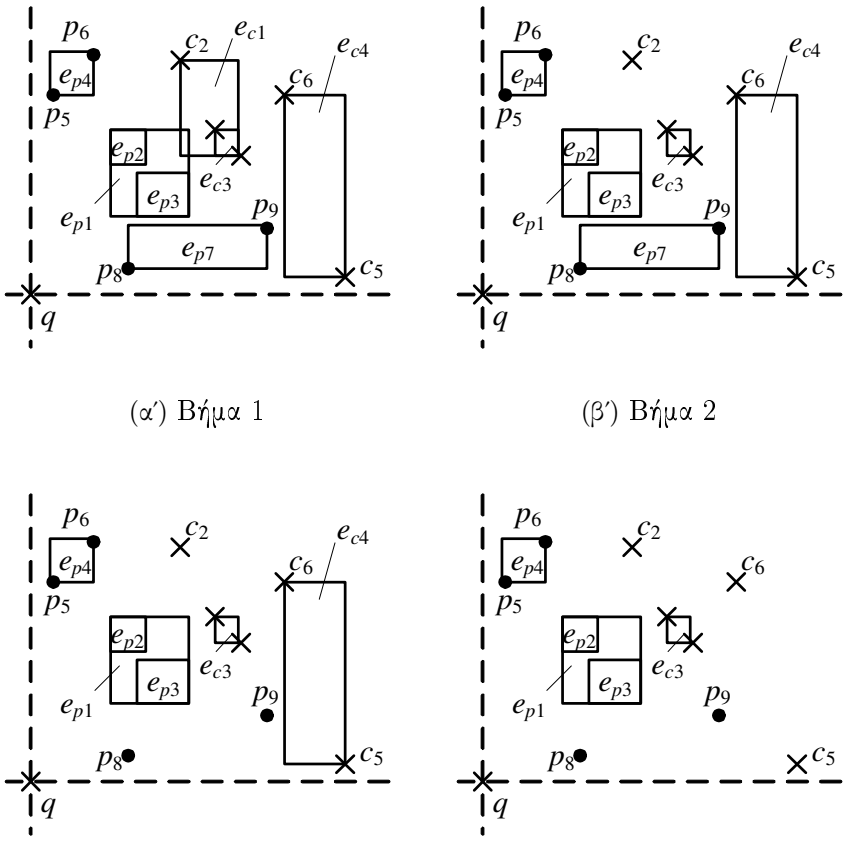
κόμβους - απογόνους της ρίζας του T_P (αντίστοιχα T_C). Στη συνέχεια ο αλγόριθμος εκτελείται σε επαναλήψεις. Σε κάθε επανάληψη ο RSA αφαιρεί έναν κόμβο από την ουρά E_C (γραμμή 5) και ελέγχει τις παρακάτω συνθήκες κλαδέματος:

1. Αν ο κόμβος e_c κυριαρχείται από κάποιο σημείο που ανήκει στο τρέχον σύνολο των midpoint skylines, $SKY(q)$, τότε ο κόμβος e_c απορρίπτεται από τα αποτελέσματα βάσει της Ιδιότητας 7 (γραμμές 6–8).

2. Αλλιώς αν ο κόμβος e_c είναι ενδιάμεσος κόμβος (γραμμή 9), τότε ο κόμβος e_c προσπελαύνεται και οι κόμβοι-απόγονοί του προστίθενται στην ουρά E_C (γραμμή 10).

3. Διαφορετικά, για κάθε κόμβο e_p που ανήκει στην ουρά E_P (γραμμές 12-22):

   - Αν ο κόμβος e_c κυριαρχείται από το midpoint κάποιου κόμβου φύλλου $e_p \in E_P$ (γραμμή 15), τότε ο κόμβος e_c απορρίπτεται από τα αποτελέσματα βάσει της Ιδιότητας 7, ενώ το σημείο midpoint του e_p προστίθεται στο σύνολο $SKY(q)$ (γραμμή 17).
   - Διαφορετικά αν ο κόμβος e_c κυριαρχείται από το midpoint του min-corner e_p^- ενός ενδιάμεσου κόμβου $e_p \in E_P$ (γραμμή 20), τότε ο κόμβος e_p προσπελαύνεται και οι κόμβοι-απόγονοί του προστίθενται στην ουρά E_P (γραμμή 21).



(α′) Βήμα 1 (β′) Βήμα 2

(γ′) Βήμα 3 (δ′) Βήμα 4

Σχήμα 5.5: *Παράδειγμα εκτέλεσης του αλγορίθμου RSA*

Τέλος, αν ο κόμβος e_c δεν έχει κλαδευτεί από κάποια από τις παραπάνω συνθήκες (γραμμή 23), τότε ο e_c ανήκει στην αντίστροφη κορυφογραμμή του q και μπορεί να επιστραφεί άμεσα ως αποτέλεσμα (γραμμή 24). Ο αλγόριθμος RSA τερματίζει όταν η ουρά προτεραιότητας E_C αδειάσει οπότε επιστρέφεται το σύνολο επιρροής $RSKY(q)$ (γραμμή 25).

Παράδειγμα 5.15. *Η εκτέλεση του αλγορίθμου RSA θα φανεί καλύτερα με τη βοήθεια του παραδείγματος του Σχήματος 5.5. Αρχικά έχουμε* $E_P(q) = \{e_{p_7}, e_{p_1}, e_{p_4}\}$ *και* $E_C(q) = \{e_{c_1}, e_{c_4}\}$ *(οι κόμβοι βρίσκονται στο ίδιο επίπεδο οπότε ταξινομούνται ως προς την Ευκλείδεια απόστασή τους από το σημείο q). Κατά την πρώτη επανάληψη ο αλγόριθμος RSA θα εξετάσει τον κόμβο* e_{c_1} *ο οποίος έχει την ελάχιστη απόσταση από το q. Επειδή πρόκειται για ενδιάμεσο κόμβο ο RSA προσπελαύνει τον κόμβο* e_{c_1} *(γραμμή 10) και προσθέτει τους κόμβους - απογόνους* c_2 *και* e_{c_3} *στην ουρά προτεραιότητας* $E_C(q)$ *(βλέπε Σχήμα 5.5(β′)). Σε αυτό το σημείο έχουμε* $E_C(q) = \{c_2, e_{c_4}, e_{c_3}\}$ *οπότε ο αλγόριθμος θα επιλέξει να εξετάσει τον κόμβο* c_2*. Αφού το τρέχον σύνολο κορυφογραμμής είναι κενό, ο κόμβος* c_2 *δεν κυριαρχείται από κανένα σημείο-προϊόν οπότε ο αλγόριθμος συνεχίζει για να ελέγξει αν ο κόμβος* c_2 *κυριαρχείται από κάποιον κόμβο που περιέχεται στην ουρά* $E_P(q)$*. Ο κόμβος* c_2 *κυριαρχείται από το min-corner του πρώτου κόμβου στην ουρά* $E_P(q)$*,* e_{p_7} *(γραμμή 14). Συνεπώς ο αλγόριθμος πρέπει να εξετάσει αν υπάρχει κάποιο σημείο (κόμβος - φύλλο) εντός του* e_{p_7} *το οποίο να κυριαρ-*

χεί έναντι του c_2 ως προς q. Γι'αυτό το λόγο ο αλγόριθμος προσπελαύνει τον κόμβο e_{p_7} (γραμμή 21), προσθέτοντας τους κόμβους - απογόνους p_8 και p_9 στην ουρά $E_P(q)$. Πλέον έχουμε $E_P(q) = \{p_8, p_9, e_{p_1}, e_{p_4}\}$ (βλέπε Σχήμα 5.5(γ')). Σε αυτό το σημείο ο αλγόριθμος βρίσκει ότι ο κόμβος c_2 κυριαρχείται από το σημείο p_8. Επομένως, το σημείο p_8 προστίθεται στην κορυφογραμμή (γραμμή 17) και ο κόμβος c_2 απορρίπτεται από τα αποτελέσματα. Στην επόμενη επανάληψη, ο αλγόριθμος RSA θα εξετάσει τον κόμβο e_{c_4}. Ο κόμβος e_{c_4} δεν κυριαρχείται από το τρέχον σύνολο κορυφογραμμής και αφού είναι ενδιάμεσος κόμβος προσπελαύνεται και οι κόμβοι-απόγονοί του c_5 και c_6 προστίθενται στην ουρά $E_C(q)$ (γραμμή 10) (βλέπε Σχήμα 5.5(δ')). Πλέον έχουμε $E_C(q) = \{c_5, c_6, e_{c_3}\}$. Έπειτα ο αλγόριθμος θα εξετάσει τον κόμβο c_5. Ο κόμβος c_5 δεν κυριαρχείται επομένως επιστρέφεται ως αποτέλεσμα (γραμμή 24). Στη συνέχεια ο αλγόριθμος RSA εξετάζει τον κόμβο c_6 ο οποίος κυριαρχείται από το σημείο p_8 που ανήκει στην τρέχουσα κορυφογραμμή (γραμμή 6) και επομένως απορρίπτεται. Στην τελευταία επανάληψη ο αλγόριθμος θα εξετάσει τον κόμβο e_{c_3}. Και ο κόμβος e_{c_3} κυριαρχείται από το σημείο p_8 και επομένως απορρίπτεται. Η ουρά προτεραιότητας $E_C(q)$ έχει πλέον αδειάσει επομένως ο αλγόριθμος RSA τερματίζει και επιστρέφει το σημείο c_5 ως το τελικό αποτέλεσμα του ερωτήματος. □

Ανάλυση πολυπλοκότητας. Ο αλγόριθμος RSA απαιτεί στην χειρότερη περίπτωση $|C|$ επαναλήψεις, μία για κάθε σημείο του συνόλου C. Βέβαια στην πράξη αρκετοί κόμβοι θα απορριφθούν με τη βοήθεια του τρέχοντος συνόλου κορυφογραμμής $SKY(q)$ (γραμμή 6). Κάθε επανάληψη περιλαμβάνει έναν έλεγχο κυριαρχίας (α) με το τρέχον σύνολο κορυφογραμμής $SKY(q)$, και (β) με το σύνολο των κόμβων που ανήκουν στην ουρά προτεραιότητας E_P. Και τα δύο σύνολα έχουν στη χειρότερη περίπτωση μέγεθος $O(|P|)$ οπότε συνολικά ο αλγόριθμος RSA απαιτεί $O(|P||C|)$ ελέγχους κυριαρχίας, ή αλλιώς $O(D|P||C|)$ συγκρίσεις.

Προοδευτική παραγωγή αποτελεσμάτων. Επανερχόμενοι στη συζήτηση για την προοδευτική παραγωγή αποτελεσμάτων, θυμίζουμε ότι ο αλγόριθμος RSA εξετάζει πάντα πρώτα τους κόμβους φύλλα οι οποίοι ταυτόχρονα έχουν την ελάχιστη Ευκλείδεια απόσταση από το σημείο q. Με άλλα λόγια αυτό έχει ως συνέπεια ότι οι πρώτες επαναλήψεις αφορούν σημεία τα οποία βρίσκονται πολύ κοντά στο q. Διαισθητικά, όσο εγγύτερα στο σημείο q βρίσκεται ένα σημείο του συνόλου C, τόσο πιο πιθανό είναι να μην κυριαρχείται από κάποιο άλλο σημείο του συνόλου P ως προς q. Επομένως, τα σημεία που εξετάζονται στις πρώτες επαναλήψεις έχουν μεγάλη πιθανότητα να ανήκουν στο σύνολο επιρροής $RSKY(q)$. Επιπλέον, λόγω του κριτηρίου ταξινόμησης στην ουρά E_C, οι πρώτοι κόμβοι που θα εξεταστούν θα είναι κόμβοι-φύλλα και επομένως δεν θα χρειαστούν να προσπελαστούν οι αντίστοιχοι κόμβοι στο δίσκο, κάτι που συνεπάγεται ότι οι πρώτες επαναλήψεις θα είναι εν γένει ταχύτερες από τις επόμενες. Λόγω των παραπάνω (και όπως επιβεβαιώνουν και τα πειράματά μας στην ενότητα 5.5) ο αλγόριθμος RSA χρειάζεται συγκριτικά ελάχιστο χρόνο για να βρει τα πρώτα αποτελέσματα ενός αντίστροφου ερωτήματος κορυφογραμμής. Σε αντιδιαστολή, θυμίζουμε ότι ο αλγόριθμος BRS απαιτεί αρκετές επαναλήψεις ώστε να προσδιορίσει τη ζώνη επιρροής με τέτοιο βαθμό ακρίβειας που να μπορεί να επιστρέψει τα πρώτα αποτελέσματα.

5.3 Ερωτήματα Εύρεσης των k πιο Ελκυστικών Υποψηφίων

Στη συνέχεια προτείνουμε έναν νέο τύπο ερωτήματος στο οποίο θα αναφερόμαστε ως *Ερώτημα Εύρεσης των k πιο Ελκυστικών Υποψηφίων (k-Most Attractive Candidates - k-MAC)*. Τα ερωτήματα k-MAC αποτελούν γενίκευση προβλημάτων που έχουν μελετηθεί σε πρόσφατες εργασίες [57, 49] καλύπτοντας και την περίπτωση όπου οι προτιμήσεις των καταναλωτών περιλαμβάνουν υποκειμενικά γνωρίσματα. Αρχικά περιγράφουμε ένα παράδειγμα εφαρμογής που δείχνει τη χρησιμότητα τέτοιων ερωτημάτων. Στη συνέχεια θα παρουσιάσουμε τον αυστηρό ορισμό των ερωτημάτων k-MAC.

Παράδειγμα εφαρμογής ερωτημάτων k-MAC. Έστω μια εταιρία κατασκευαστής φορητών υπολογιστών η οποία σκοπεύει να θέσει σε παραγωγή μια νέα σειρά μοντέλων. Για να αποφασίσει ποια μοντέλα θα επιλέξει να θέσει σε παραγωγή, η εταιρία πρέπει να λάβει υπόψη της: (α) το σύνολο των ανταγωνιστικών μοντέλων P που υπάρχουν στην αγορά, (β) το σύνολο των καταναλωτικών προτιμήσεων C που έχουν εκφραστεί ως προς τις προδιαγραφές ενός μοντέλου, και (γ) το σύνολο υποψηφίων νέων μοντέλων Q όπως αυτά έχουν προταθεί από το σχεδιαστικό τμήμα. Στόχος είναι να προσδιοριστούν τα k μοντέλα από το σύνολο Q τα οποία αναμένεται συνδυαστικά να έχουν τη μεγαλύτερη απήχηση στην αγορά, δηλαδή αυτά από κοινού εκτιμάται ότι θα προσελκύσουν το μέγιστο δυνατό αριθμό αγοραστών. Διευκρινίζουμε ότι το ζητούμενο σύνολο k μοντέλων είναι διαφορετικό από αυτό που θα προέκυπτε αν επιλεγούν τα k επιμέρους πιο ελκυστικά μοντέλα, καθώς δεν έχει νόημα να επιλεγούν στο αποτέλεσμα δύο μοντέλα τα οποία αναμένεται να προσελκύσουν το ίδιο σύνολο αγοραστών.

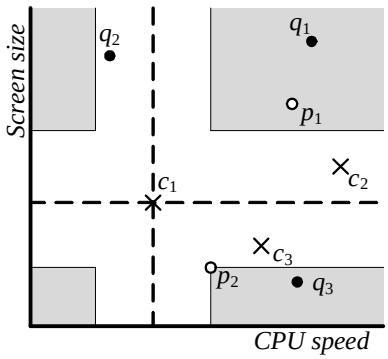
Ορισμός Προβλήματος. Αρχικά ορίζουμε το *από κοινού σύνολο επιρροής* για ένα σύνολο υποψηφίων Q. Στη συνέχεια παραθέτουμε τον ορισμό του *από κοινού σκορ επιρροής* και εισάγουμε την έννοια των ερωτημάτων k-MAC.

Ορισμός 5.7. *(Από Κοινού Σύνολο Επιρροής): Δοθέντος ενός συνόλου προϊόντων P, ενός συνόλου καταναλωτικών προτιμήσεων C και ενός συνόλου νέων (υποψήφιων) προϊόντων Q, τα από κοινού σύνολο επιρροής του Q, το οποίο θα συμβολίζουμε ως $RSKY(Q)$, ορίζεται ως ένωση των επιμέρους συνόλων επιρροής όλων των $q_i \in Q$:*

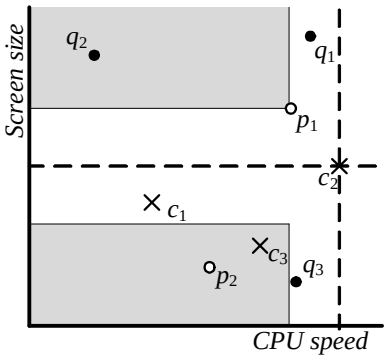
$$RSKY(Q) = \bigcup_{q_i \in Q} RSKY(q_i)$$

Με βάση των παραπάνω ορισμό, το *από κοινού σκορ επιρροής* IS (Q) ενός συνόλου υποψήφιων προϊόντων Q είναι ίσο με το μέγεθος του από κοινού συνόλου επιρροής του Q, $|RSKY(Q)|$. Στη δίνουμε τον ορισμό ενός ερωτήματος k-MAC ως εξής:

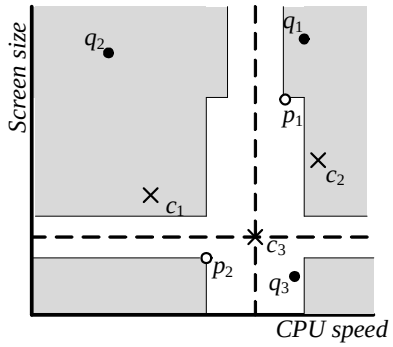
Ορισμός 5.8. *(Ερώτημα Εύρεσης των k πιο Ελκυστικών Υποψηφίων k-Most Attractive Candidates Query - k-MAC): Δοθέντος ενός συνόλου προϊόντων P, ενός συνόλου καταναλωτικών προτιμήσεων C, ενός συνόλου νέων (υποψήφιων) προϊόντων Q και ενός θετικού ακεραίου $k > 1$, ένα ερώτημα k-MAC επιστρέφει το υποσύνολο $Q' \subseteq Q$, με μέγεθος $|Q'| = k$ το οποίο μεγιστοποιεί το από κοινού σκορ επιρροής IS(Q').*



(α′) Δυναμική κορυφογραμμή ως προς c_1



(β′) Δυναμική κορυφογραμμή ως προς c_2



(γ′) Δυναμική κορυφογραμμή ως προς c_3

Products p_i
Customers c_i
Candidates q_i
RSKY(q_1): {c_2}
RSKY(q_2): {c_1}
RSKY(q_3): {c_2, c_3}
RSKY{q_1,q_2}: {c_1, c_2}
RSKY{q_1,q_3}: {c_2, c_3}
RSKY{q_2,q_3}: {c_1, c_2, c_3}
2-MAC{q_1, q_2, q_3}: {q_2,q_3}

(δ′) Από κοινού σύνολα επιρροής

Σχήμα 5.6: *Παράδειγμα ερωτήματος k-MAC*

Το Σχήμα 5.6 δείχνει ένα παράδειγμα ερωτήματος k-MAC όπου έχουμε θεωρήσει 2 υπάρχοντα (ανταγωνιστικά) προϊόντα p_1 και p_2 και 3 υποψήφια νέα μοντέλα. Όπως προκύπτει, το αποτέλεσμα ενός ερωτήματος 1-MAC θα επιστρέψει το μοντέλο q_3 για το οποίο $\text{IS}(q_3) = 2$. Αντίστοιχα, ένα ερώτημα 2-MAC θα επιστρέψει τα μοντέλα $\{q_2, q_3\}$ τα οποία έχουν από κοινού σκορ επιρροής 3.

Αξίζει να σημειωθεί ότι είναι δυνατόν περισσότερα του ενός υποψήφια προϊόντα να είναι ελκυστικά με βάση τις προτιμήσεις ενός καταναλωτή. Για παράδειγμα, στο Σχήμα 5.6(β′), τόσο το q_1 όσο και το q_3 ανήκουν στο $SKY(c_2)$. Επιπλέον διευκρινίζουμε ότι κατά την αποτίμηση ερωτημάτων k-MAC, κάθε υποψήφιο προϊόν $q \in Q$ λαμβάνεται υπόψη ανεξάρτητα από τα υπόλοιπα υποψήφια και μόνο σε σύγκριση με τα ήδη υπάρχοντα προϊόντα του συνόλου P. Με άλλα λόγια, οι σχέσεις κυριαρχίας μεταξύ υποψήφιων προϊόντων δεν λαμβάνονται υπόψη. Η παραδοχή αυτή είναι συνεπής με μια πραγματική εφαρμογή όπου μια εταιρία ενδιαφέρεται να συγκρίνει το σύνολο προϊόντων της μόνο σε σχέση με τον ανταγωνισμό. Επίσης, στο τέλος της ενότητας περιγράφουμε πώς χειριζόμαστε περιπτώσεις όπου δύο υποψήφια προϊόντα έχουν ισοδύναμα σκορ επιρροής.

Προηγούμενες εργασίες [57, 49] επιχειρούν να αντιμετωπίσουν ένα παρόμοιο ερώτημα με αυτό που προτείνουμε εισάγοντας τα ερωτήματα k-MAC. Όμως, οι εργασίες αυτές θεωρούν μόνο την περίπτωση όπου όλα τα γνωρίσματα της βάσης δεδομένων είναι αντικειμενικά, δηλαδή έχουν μια γενικά βέλτιστη τιμή (για παράδειγμα στην περίπτωση ενός φορητού υπολογιστή: μηδενική τιμή αγοράς, άπειρη αυτονομία κλπ.). Τα ερωτήματα k-MAC επεκτείνουν αυτές τις εργασίες καλύπτοντας επίσης και υποκειμενικά γνωρίσματα, όπου η βέλτιστη τιμή εξαρτάται από τις προτιμήσεις κάθε καταναλωτή. Επιπλέον, στις εργασίες [57, 49] γίνεται η παραδοχή ότι οι σχέσεις κυριαρχίας μεταξύ των ανταγωνιστικών και υποψήφιων προϊόντων είναι εκ των προτέρων γνωστές, κάτι που είναι δυστυχώς εφικτό μόνο στην περίπτωση αντικειμενικών γνωρισμάτων. Επομένως, η συνεισφορά των εργασιών [57, 49] περιορίζεται στο να προτείνουν αποδοτικούς αλγόριθμους για την επιλογή των υποψηφίων προϊόντων που ανήκουν στο αποτέλεσμα του ερωτήματος. Αντιθέτως, στη δική μας περίπτωση η έμφαση δίνεται στον αποδοτικό προσδιορισμό του συνόλου επιρροής.

Παραλλαγές των ερωτημάτων k-MAC. Με βάση τον ορισμό ενός ερωτήματος k-MAC, αν πολλαπλά προϊόντα έχουν πανομοιότυπα χαρακτηριστικά, τότε αρκεί να εξετάσουμε κάθε τέτοια ομάδα προϊόντων μόνο μια φορά. Αυτό ισχύει γιατί δύο προϊόντα με ίσες τιμές γνωρισμάτων έχουν πανομοιότυπες ζώνες επιρροής. Με παρόμοιο τρόπο μπορούμε να χειριστούμε ένα σύνολο SC καταναλωτών με πανομοιότυπες προτιμήσεις, όπου $|SC| > 1$. Συγκεκριμένα αρκεί: (α) να θεωρήσουμε μόνο έναν καταναλωτή για κάθε τέτοια ομάδα με βάρος ίσο με $|SC|$, και (β) να λάβουμε υπόψη το συγκεκριμένο βάρος όταν υπολογίζουμε τα από κοινού σκορ επιρροής.

Μια άλλη παραλλαγή του ερωτήματος k-MAC θα ήταν να συσχετίσουμε κάθε πιθανό αγοραστή c_i που ανήκει στο σύνολο επιρροής $RSKY(q)$ με ένα βάρος w_i. Η συγκεκριμένη τιμή για το βάρος w_i αντιπροσωπεύει την πιθανότητα ο καταναλωτής c_i να αγοράσει τελικά το προϊόν q. Για παράδειγμα, μια παράμετρος που μπορεί να χρησιμοποιηθεί για να υπολογιστεί η συγκεκριμένη πιθανότητα είναι η απόσταση ανάμεσα στα σημεία c_i και q στον πολυδιάστατο χώρο.

Άπληστος Αλγόριθμος. Η επεξεργασία ενός ερωτήματος k-MAC είναι κάθε άλλο παρά απλή. Συγκεκριμένα το πρόβλημα μπορεί να διαχωριστεί σε δύο υποπροβλήματα: (α) τον υπολογισμό των επιμέρους συνόλων επιρροής για ένα σύνολο υποψήφιων προϊόντων, και (β) την εύρεση ενός υποσυνόλου μεγέθους k το οποίο μεγιστοποιεί το κέρδος μετρούμενο ως το σύνολο των πιθανών αγοραστών (από κοινού σκορ επιρροής). Στην επόμενη ενότητα προτείνουμε τεχνικές για την αποδοτική επεξεργασία του πρώτου μέρους. Θεωρώντας τα επιμέρους σύνολα επιρροής γνωστά, το δεύτερο υποπρόβλημα μπορεί να μετασχηματιστεί σε ένα πιο γενικό πρόβλημα γνωστό ως maximum k-coverage. Το συγκεκριμένο πρόβλημα είναι NP-hard και συνεπώς μια εξαντλητική εξέταση όλων των πιθανών υποσυνόλων μεγέθους k δεν είναι εφικτή επιλογή. Επομένως, στη συνέχεια θα προτείνουμε έναν αποδοτικό άπληστο αλγόριθμο για την επίλυση του προβλήματος. Η λύση που προτείνουμε αποτελεί μια παραλλαγή της πιο γενικής μεθόδου κάλυψης k-βημάτων (k-stage covering algorithm) η οποία περιγράφεται στην εργασία [31]. Όπως μας εξασφαλίζει η παρακάτω ιδιότητα, το κέρδος που προκύπτει από τη λύση που δίνει η προτεινόμενη μέθοδος θα απέχει απόσταση το πολύ ίση με $1 - 1/e$ από το κέρδος της βέλτιστης λύσης.

Algorithm 5: k-stage Greedy Selection Algorithm

```
Input: Q a set of candidates, RSKY(q_i) reverse skylines of q_i, k
Output: Q' the most attractive set of candidates where |Q'| = k
1  begin
2      Q' := ∅; TempRSKY := ∅; MaxRSKY := ∅;
3      while |Q'| < k do
4          TempRSKY := MaxRSKY;
5          foreach q_i ∈ Q do
6              if |RSKY(q_i) ⋃ MaxRSKY| > |TempRSKY| then
7                  TempRSKY := RSKY(q_i) ⋃ MaxRSKY;
8                  BestCand := {q_i};
9          MaxRSKY := TempRSKY;
10         Q := Q − BestCand;
11         Q' := Q' ⋃ BestCand;
12     return Q';
```

Ιδιότητα 9. *Ο αλγόριθμος κάλυψης k-βημάτων επιστρέφει μια προσεγγιστική λύση η οποία έχει κέρδος που απέχει το πολύ $1 - 1/e$ από το κέρδος της βέλτιστης λύσης.*

Στη συνέχεια περιγράφουμε πώς προσαρμόζουμε τον αλγόριθμο κάλυψης k-βημάτων στην περίπτωση ερωτημάτων k-MAC. Ο αλγόριθμος μας (k-stage Greedy Selection Algorithm - kGSA) λαμβάνει ως είσοδο ένα σύνολο Q υποψήφιων προϊόντων και επιστρέφει ένα υποσύνολο $Q' \subseteq Q$, όπου $|Q'| = k$, το οποίο αποτελεί μια $(1 - 1/e)$-προσεγγιστική λύση του ερωτήματος k-MAC. Ο αλγόριθμος kGSA εκτελείται σε επαναλήψεις. Σε κάθε επανάληψη ο αλγόριθμος εξετάζει όλα τα υποψήφια προϊόντα και επιλέγει αυτό το οποίο αν προστεθεί στο τρέχον αποτέλεσμα θα οδηγήσει στη μέγιστη δυνατή αύξηση του από κοινού σκορ επιρροής. Αν περισσότερα του ενός υποψήφια προϊόντα έχουν ως αποτέλεσμα ισόποση αύξηση του IS (Q'), τότε ο αλγόριθμος kGSA επιλέγει εκείνο το προϊόν που έχει το ελάχιστο άθροισμα αποστάσεων από τα σημεία που ανήκουν στο σύνολο επιρροής του. Ο λόγος που επιλέγουμε το συγκεκριμένο κριτήριο είναι ότι διαισθητικά, όσο πιο κοντά είναι οι προδιαγραφές ενός προϊόντος στις προτιμήσεις ενός καταναλωτή, τόσο περισσότερο πιθανό είναι ο καταναλωτής να ενδιαφερθεί για την αγορά του συγκεκριμένου προϊόντος. Ο αλγόριθμος kGSA τερματίζει μετά από k επαναλήψεις και επιστρέφει ως αποτέλεσμα το σύνολο Q'.

5.4 Επεξεργασία Πολλαπλών Αντίστροφων Ερωτημάτων Κορυφογραμμής

Τα ερωτήματα k-MAC αποτελούν ένα παράδειγμα ερωτήματος που καταδεικνύει την ανάγκη ταυτόχρονης αποτίμησης πολλαπλών αντίστροφων ερωτημάτων κορυφογραμμής. Στην παρούσα ενότητα επεκτείνουμε τον αλγόριθμο RSA που προτείναμε για απλά αντίστροφα ερωτήματα κορυφογραμμής για την περίπτωση πολλαπλών ερωτημάτων.

Βασικός Αλγόριθμος. Η πιο απλή μέθοδος επεξεργασίας πολλαπλών αντίστροφων ερωτημάτων κορυφογραμμής είναι να εφαρμόσουμε έναν αλγόριθμο για απλά ερωτήματα, όπως ο BRS ή ο RSA ξεχωριστά για κάθε σημείο. Όμως η συγκεκριμένη προσέγγιση δεν είναι καθόλου αποδοτική ως προς τον αριθμό εισόδων/εξόδων από το δίσκο που απαιτούνται. Συγκεκριμένα, αρκετοί κόμβοι e_p (e_c) θα χρειαστεί να προσπελαστούν πολλές φορές εφόσον εμφανίζονται στην ουρά προτεραιότητας περισσότερων

του ενός υποψήφιου.

Αλγόριθμος bRSA. Στην συνέχεια περιγράφουμε τον αλγόριθμο bRSA, ο οποίος αποτελεί επέκταση του αλγορίθμου RSA που προτείναμε για την περίπτωση απλών ερωτημάτων κορυφογραμμής (ενότητα 5.2). Βασικός στόχος του αλγορίθμου bRSA είναι να μειώσει τον συνολικό αριθμό λειτουργιών εισόδου/εξόδου που απαιτούνται, εκμεταλλευόμενος κατά το δυνατόν τη γειτνίαση μεταξύ υποψηφίων και επιτρέποντας τον διαμοιρασμό κάποιου τμήματος της επεξεργασίας. Αξίζει να σημειωθεί ότι ο αλγόριθμος bRSA που προτείνουμε μπορεί να εφαρμοστεί εκτός των ερωτημάτων k-MAC και σε άλλου τύπου ερωτήματα τα οποία απαιτούν την επεξεργασία πολλαπλών αντίστροφων ερωτημάτων κορυφογραμμής.

Ο αλγόριθμος bRSA επεξεργάζεται πολλαπλά ερωτήματα παράλληλα, ομαδοποιώντας τα με τέτοιο τρόπο ώστε τα σημεία που βρίσκονται σε μια ομάδα να επωφελούνται κατά το δυνατό από την επεξεργασία των άλλων μελών της ομάδας. Ο αλγόριθμος επιχειρεί να αποφύγει περιττές λειτουργίες εισόδου/εξόδου χρησιμοποιώντας κόμβους που προσπελάστηκαν κατά την εκτέλεση ενός τμήματος του αλγορίθμου RSA που αφορά ένα ερώτημα, για να κλαδέψει κόμβους που ανήκουν στις ουρές προτεραιότητας των υπολοίπων μελών της ομάδας. Συγκεκριμένα, όποτε εκτελείται προσπέλαση σε έναν κόμβο και οι κόμβοι-απόγονοι του εισάγονται στην αντίστοιχη ουρά προτεραιότητας, ενημερώνονται ταυτόχρονα οι ουρές προτεραιότητας όλων των μελών της ομάδας που περιείχαν τον αρχικό κόμβο. Επομένως, για κάθε κόμβο εκτελείται αποκλειστικά μία προσπέλαση ανά ομάδα. Επιπλέον, με σκοπό να βελτιώσει περισσότερο το κόστος επεξεργασίας, ο αλγόριθμος bRSA διατηρεί ένα σύνολο προϊόντων (φύλλων στο αντίστοιχο R-tree) τα οποία εκτιμά ότι έχουν δυνατότητα να κλαδέψουν μεγάλο μέρος του χώρου. Στη συνέχεια θα χρησιμοποιούμε τον όρο σημεία υπεροχής (vantage points) για να αναφερόμαστε σε αυτά τα σημεία. Η χρήση τους θα εξηγηθεί παρακάτω καθώς περιγράφουμε αναλυτικά την εκτέλεση του αλγορίθμου bRSA.

Σε αυτό το σημείο είναι σημαντικό να αναφέρουμε ότι οι δομές δεδομένων που χρειάζονται για την εκτέλεση του αλγόριθμου bRSA (π.χ. ουρές προτεραιότητας, σύνολα κορυφογραμμής κλπ.) καταλαμβάνουν σημαντικό μέρος της κύριας μνήμης. Στη γενική περίπτωση, ιδιαιτέρως για μεγαλύτερα $|Q|$, δεν μπορούμε με ασφάλεια να υποθέσουμε ότι όλες αυτές οι δομές δεδομένων μπορούν να χωρέσουν στην κύρια μνήμη. Με βάση τις δυνατότητες του διαθέσιμου συστήματος, στη συνέχεια θα θεωρήσουμε ότι μόνο G ερωτήματα μπορούν να επεξεργαστούν παράλληλα, όπου $G \ll |Q|$. Όπως θα δούμε και κατά την πειραματική αξιολόγηση του αλγορίθμου (ενότητα 5.5), ο αλγόριθμος bRSA εμφανίζει τη βέλτιστη συμπεριφορά κρατώντας την τιμή G σε σχετικά μικρό μέγεθος (π.χ. έως 10 ερωτήματα ανά ομάδα). Ο λόγος είναι ότι μεγαλύτερα μεγέθη ομάδων οδηγούν σε εκρηκτική αύξηση του κόστους επεξεργασίας που συνδέεται με την διαχείριση των ουρών προτεραιότητας και τους ελέγχους κυριαρχίας που απαιτούνται, κάτι που γρήγορα αντισταθμίζει το όφελος από τον μειωμένο αριθμό λειτουργιών εισόδου/εξόδου.

Σημεία τα οποία γειτνιάζουν στον πολυδιάστατο χώρο είναι περισσότερο πιθανό να επωφεληθούν από την παράλληλη επεξεργασία. Για το λόγο αυτό ο αλγόριθμος bRSA αρχικά κατατμίζει το σύνολο Q σε $\lceil |Q|/G \rceil$ ομάδες με τη βοήθεια μιας καμπύλης γεμίσματος χώρου (π.χ. Hilbert curve). Στη συνέχεια, ο αλγόριθμος επεξεργάζεται τις

Algorithm 6: bRSA

Input: Q a set of candidates, T_P R-tree on products, T_C R-tree on customers
Variables: $E_P(q_i)$ priority queue on products for q_i, $E_C(q_i)$ priority queue on customers for q_i, $RSKY(q_i)$ reverse skylines for q_i, $SKY(q_i)$ midpoint skylines of q_i, G_j batches with $|G_j| = G$

```
1  begin
2      partition Q into ⌈|Q|/G⌉ batches → G_j;
3      foreach G_j do
4          while (RSKY(q_i) for all q_i ∈ G_j have not been found) do
5              selectCandidate → q_i;
               /* Process q_i until IS(q_i) has been completely determined */
6              if E_C(q_i) ≠ ∅ then
7                  Batch-RSA (q_i, G_j, T_P, T_C, E_P(q_i), E_C(q_i), RSKY(q_i), SKY(q_i), H_P);
```

Function Batch-RSA

Input: G a group of candidates, T_P R-tree on products, T_C R-tree on customers, $E_P(q_i)$ priority queue on products for q_i, $E_C(q_i)$ priority queue on customers for q_i, $RSKY(q_i)$ reverse skylines of q_i, $SKY(q_i)$ midpoint skylines of q_i, H_P priority queue on product leaf entries (vantage points)
Output: $RSKY(q_i)$ reverse skylines of q_i

```
1  begin
2      while E_C(q_i) ≠ ∅ do
3          dominated := false;
4          E_C(q_i).pop() → e_c;
5          if dominated(e_c, SKY(q_i)) OR dominated(e_c, H_P) then
6              dominated := true; continue;
7          if e_c is a non-leaf entry then
8              Expand e_c for all relevant q_i, insert children into E_C(q_i);
9          else
10             foreach e_p ∈ E_P(q_i) do
11                 midpoint(e_p, q_i) → m;
12                 if e_c is dominated by m then
13                     if e_p is a leaf entry then
14                         if (dominated(m, SKY(q_i)) == false) then
15                             SKY(q_i).push(m);
16                         H_P.push(e_p);
17                         dominated := true; break;
18                     else
19                         Expand e_p for all relevant q_i, insert children into E_P(q_i);
20                     E_P(q_i).remove(e_p);
21             if (dominated == false) then
22                 RSKY(q_i).push(e_c);
23     return RSKY(q_i);
```

ομάδες την μία μετά την άλλη. Για κάθε ομάδα επιλέγει σε κάθε επανάληψη με κυκλικό τρόπο ένα ερώτημα (υποψήφιο προϊόν) (γραμμή 5 του αλγορίθμου 6) και εκτελεί μια τροποποιημένη εκδοχή του αλγορίθμου RSA. Η εκδοχή αυτή (με την ονομασία Batch-RSA) επεκτείνει τον αλγόριθμο RSA στην περίπτωση παράλληλης επεξεργασίας μιας ομάδας ερωτημάτων. Στη συνέχεια περιγράφουμε τις διαφορές του αλγορίθμου Batch-RSA σε σχέση με τον RSA.

Καταρχήν, όποτε προσπελαύνεται ένας κόμβος e_x, οι ουρές προτεραιότητας όλων των μελών της ομάδας στις οποίες εμφανίζεται ο e_x ενημερώνονται κατάλληλα. Επίσης, αν βρεθεί ένα σημείο-φύλλο έστω p_i (γραμμή 12 της συνάρτησης 6), ο αλγόριθμος αποφασίζει αν το p_i θα πρέπει να εισαχθεί σε έναν buffer H_P ο οποίος περιέχει σημεία υπεροχής (vantage points), δηλαδή αυτά που μπορούν να χρησιμοποιηθούν στο κλάδεμα κόμβων άλλων υποψηφίων. Διαισθητικά, όσο πλησιέστερα σε ένα υποψήφιο βρίσκεται ένα σημείο, τόσο μεγιστοποιούνται οι δυνατότητες κλαδέματός του ενώ κυ-

ριαρχεί σε μεγαλύτερο κομμάτι του πολυδιάστατου χώρου. Ακολουθώντας αυτή τη λογική υλοποιήσαμε τον buffer H_P ως μία ουρά προτεραιότητας με κλειδί την ελάχιστη Ευκλείδεια απόσταση ενός κόμβου από οποιοδήποτε υποψήφιο της ομάδας. Όταν το H_P γεμίσει το πιο απομακρυσμένο σημείο στο H_P αντικαθίσταται με ένα νέο σημείο p_i. Τα vantage points (ουσιαστικά τα αντίστοιχα midpoints) χρησιμοποιούνται με σκοπό να εφαρμοστούν έλεγχοι κυριαρχίας επιπλέον του συνόλου κορυφογραμμής (δεύτερη συνθήκη στη διάζευξη - γραμμή 5 της συνάρτησης 6), με στόχο την αποφυγή κάποιων περιττών λειτουργιών εισόδου/εξόδου.

5.5 Πειραματική Αξιολόγηση

Στην ενότητα αυτή αξιολογούμε πειραματικά τους προτεινόμενους αλγορίθμους. Όλοι οι αλγόριθμοι που εξετάσαμε υλοποιήθηκαν σε C++, μεταγλωττίστηκαν με gcc και εκτελέστηκαν σε ένα σύστημα με επεξεργαστή 2 GHz Intel Xeon, μνήμη RAM 4 GB και λειτουργικό σύστημα Debian Linux. Ο πηγαίος κώδικας που υλοποιεί τον αλγόριθμο BRS μας χορηγήθηκε από τους συγγραφείς της εργασίας [81].

5.5.1 Πειραματική Μεθοδολογία

Στα πειράματά μας χρησιμοποιήσαμε μια γεννήτρια συνθετικών δεδομένων[4] με σκοπό να κατασκευάσουμε σύνολα δεδομένων που ακολουθούν διαφορετικές κατανομές όσον αφορά τις τιμές των γνωρισμάτων τους. Συγκεκριμένα, για το σύνολο uniform οι τιμές επιλέγονται από μία ομοιόμορφη κατανομή. Για το σύνολο anticorrelated οι εγγραφές με προτιμότερες (χαμηλές) τιμές για κάποιο γνώρισμα είναι περισσότερο πιθανό να έχουν υψηλότερες (λιγότερο καλές) τιμές για τα υπόλοιπα γνωρίσματα.

Εκτός των συνθετικών δεδομένων, αξιολογήσαμε πειραματικά όλους τους αλγορίθμους και σε δύο σύνολα πραγματικών δεδομένων. Το σύνολο δεδομένων NBA αποτελείται από 17265 εγγραφές 5 διαστάσεων, όπου οι διαστάσεις αντιπροσωπεύουν τη μέση τιμή της απόδοσης ενός παίκτη σε στατιστικές κατηγορίες όπως πόντοι, rebounds, assists, κλεψίματα και κοψίματα. Το σύνολο δεδομένων HOUSEHOLD, αποτελείται από 127930 εγγραφές 6 διαστάσεων που αντιπροσωπεύουν δημογραφικά δεδομένα Αμερικανικών νοικοκυριών. Τα γνωρίσματα που περιλαμβάνει κάθε εγγραφή αφορούν τις ετήσιες δαπάνες κάθε νοικοκυριού για φυσικό αέριο, ηλεκτρική ενέργεια, ύδρευση, θέρμανση, ασφάλεια και φόρο ακίνητης περιουσίας. Για τα πραγματικά σύνολα δεδομένων, κατασκευάσαμε επιπλέον σύνολα που αντιστοιχούν σε καταναλωτικές προτιμήσεις και υποψήφια προϊόντα προσθέτοντας θόρυβο που ακολουθεί την κανονική κατανομή με μέση τιμή την πραγματική τιμή του κάθε γνωρίσματος. Επιπλέον, τόσο για τα συνθετικά όσο και για τα πραγματικά δεδομένα, οι τιμές των γνωρισμάτων κανονικοποιήθηκαν στο πεδίο τιμών [0,10000]. Τέλος δεικτοδοτήσαμε τα δύο σύνολα δεδομένων (προϊόντα και προτιμήσεις καταναλωτών) με τη βοήθεια ενός R-tree θεωρώντας το μέγεθος της κάθε σελίδας ίσο με 4096 bytes.

Στα πειράματά μας συγκρίναμε την επίδοση των προτεινόμενων αλγορίθμων RSA

[4]http://randdataset.projects.postgresql.org

Παράμετρος	**Εύρος Τιμών**
Αριθμός διαστάσεων (D)	2, **3**, 4, 5
Μέγεθος συνόλου δεδομένων P ($\|P\|$)	10K, **100K**, 500K, 1M
Μέγεθος συνόλου δεδομένων C ($\|C\|$)	10K, **100K**, 500K, 1M
Μέγεθος κρυφής μνήμης/Συνολικό μέγεθος δεδομένων (M)	6.25%, **12.5%**, 25%
Πλήθος ερωτημάτων ανά ομάδα (G)	5, **10**, 20, 50, 100

Πίνακας 5.2: *Παράμετροι πειραμάτων*

και bRSA σε σχέση με τον αλγόριθμο BRS. Για κάθε αλγόριθμο μετρήσαμε το χρόνο εκτέλεσης και τον αριθμό λειτουργιών εισόδου/εξόδου που απαιτούνται για την επεξεργασία (α) ενός συνόλου $|Q|$ αντίστροφων ερωτημάτων κορυφογραμμής, και (β) ενός ερωτήματος k-MAC δοθέντος ενός συνόλου $|Q|$ υποψήφιων προϊόντων. Ο αλγόριθμος bRSA εξετάστηκε μόνο στην περίπτωση ερωτημάτων k-MAC. Πιο αναλυτικά σε κάθε πείραμα μετρήθηκαν:

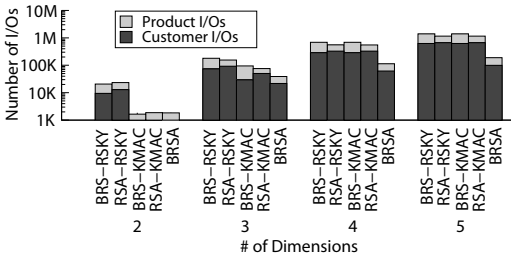
- Ο αριθμός λειτουργιών εισόδου/εξόδου από το δίσκο ξεχωριστά για προϊόντα και καταναλωτικές προτιμήσεις. Για κάθε σύνολο δεδομένων χρησιμοποιήσαμε έναν buffer με μέγεθος ίσο με 100 σελίδες (410 KB), το οποίο για το βασικό μας πείραμα αντιπροσωπεύει ποσοστό 12.5% του συνολικού μεγέθους των δεδομένων. Για τον buffer ακολουθήσαμε την στρατηγική αντικατάστασης των λιγότερο πρόσφατα χρησιμοποιούμενων σελίδων (Least Recently Used cache replacement policy - LRU).

- Ο χρόνος που δαπανήθηκε στη CPU.

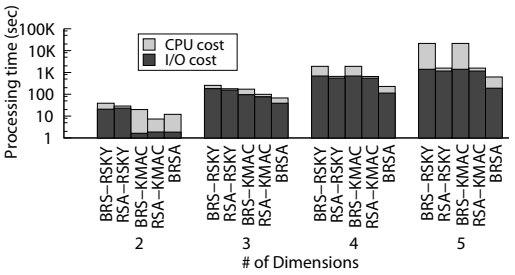
- Ο συνολικός χρόνος επεξεργασίας αποτελούμενος από το χρόνο που δαπανήθηκε στην CPU συν το κόστος εισόδου/εξόδου από το δίσκο, κάνοντας την παραδοχή ότι κάθε είσοδος/έξοδος απαιτεί 1 millisecond.

Θυμίζουμε ότι στην περίπτωση απλών αντίστροφων ερωτημάτων κορυφογραμμής, οι αλγόριθμοι BRS και RSA θα εκτελέσουν ένα σύνολο ερωτημάτων σειριακά το ένα μετά το άλλο. Για να αξιολογήσουμε τους συγκεκριμένους αλγορίθμους στην περίπτωση ερωτημάτων k-MAC προσθέσαμε δύο επιπλέον βήματα εκτέλεσης: (α) ένα βήμα προεπεξεργασίας το οποίο ταξινομεί τα σημεία με βάση την τιμή κατακερματισμού τους σύμφωνα με την συνάρτηση Hilbert, και (β) ένα τελικό στάδιο το οποίο εκτελεί τον άπληστο αλγόριθμο kGSA με σκοπό να επιλέξει k υποψήφια προϊόντα. Στα πειράματά μας, οι χρόνοι εκτέλεσης των δύο παραπάνω βημάτων ήταν αμελητέοι σε σχέση με τον χρόνο που απαιτεί η αποτίμηση ενός αντίστροφου ερωτήματος κορυφογραμμής. Επιπλέον αξίζει να σημειωθεί ότι κανείς από τους αλγορίθμους δεν επηρεάζεται πρακτικά από το πλήθος αποτελεσμάτων k, καθώς θα πρέπει ούτως ή άλλως να προσδιορίσουν πρώτα τα σύνολα επιρροής για όλα τα Q υποψήφια προϊόντα.

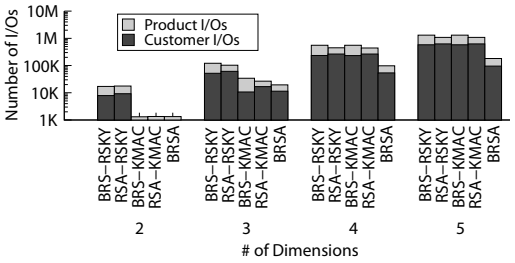
Σε κάθε πείραμα, μεταβάλαμε μια μόνο παράμετρο κρατώντας τις υπόλοιπες παραμέτρους στις προκαθορισμένες τους τιμές. Ο Πίνακας 5.2 δείχνει τις παραμέτρους υπό εξέταση μαζί με το εύρος τιμών που δοκιμάσαμε για κάθε παράμετρο. Οι προκαθορισμένες τιμές διακρίνονται με έντονους χαρακτήρες.



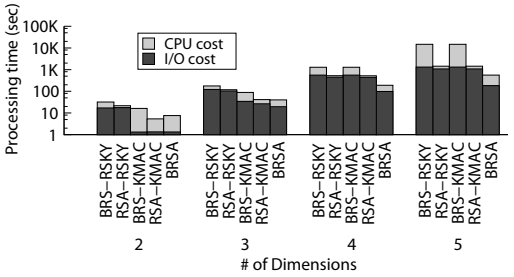
(α′) Πλήθος I/Os σε σχέση με τον αριθμό διαστάσεων (uniform dataset)



(β′) Συνολικό κόστος επεξεργασίας σε σχέση με τον αριθμό διαστάσεων (uniform dataset)



(γ′) Πλήθος I/Os σε σχέση με τον αριθμό διαστάσεων (anticorrelated dataset)



(δ′) Συνολικό κόστος επεξεργασίας σε σχέση με τον αριθμό διαστάσεων (anticorrelated dataset)

	2-D data		3-D data		4-D data		5-D data	
	BRS	RSA	BRS	RSA	BRS	RSA	BRS	RSA
I/O cost (sec)	20.7	23.3	179.9	156.0	687.6	555.0	1404	1161
CPU cost (sec)	18.4	5.5	76.9	23.2	1222.6	106.2	20042	428
Total cost (sec)	39.1	28.8	256.8	179.2	1910.2	661.2	21446	1589

	2-D data	3-D data	4-D data	5-D data
	BRS / RSA	BRS / RSA	BRS / RSA	BRS / RSA
I/O cost ratio	0.89	1.15	1.24	1.21
CPU cost ratio	3.35	3.32	11.51	46.81
Total cost ratio	1.36	1.43	2.89	13.50

(ε′) Επιδόσεις αλγορίθμων RSA & BRS σε σχέση με τον αριθμό διαστάσεων (uniform dataset)

Σχήμα 5.7: *Επίδοση αλγορίθμων σε σχέση με τον αριθμό διαστάσεων*

5.5.2 Πειραματικά Αποτελέσματα

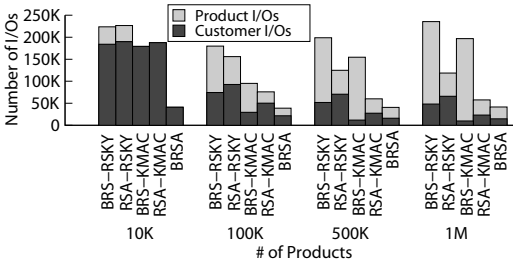
Επίδοση σε σχέση με τον αριθμό διαστάσεων. Στο πρώτο πείραμα εξετάζουμε την επίδοση όλων των αλγορίθμων καθώς μεταβάλλουμε τον αριθμό διαστάσεων από 2 έως 5. Τα Σχήματα 5.7(α′)-5.7(β′) απεικονίζουν σε λογαριθμική κλίμακα τον μετρούμενο αριθμό λειτουργιών εισόδου/εξόδου και τον συνολικό χρόνο επεξεργασίας

αντιστοίχως για τα σύνολα δεδομένων με ομοιόμορφη (uniform) κατανομή. Οι ακριβείς μετρήσεις παρουσιάζονται επίσης στο Σχήμα 5.7(ε′). Όπως αναμενόταν με βάση την ανάλυση πολυπλοκότητας που παρουσιάσαμε στην ενότητα 5.2, ο αλγόριθμος BRS έχει απαγορευτικά υψηλό κόστος εκτέλεσης για δεδομένα με τουλάχιστον 3 διαστάσεις. Συγκεκριμένα δαπανά 3.35 φορές περισσότερο χρόνο στην CPU σε σχέση με τον αλγόριθμο RSA ακόμα και για 2 διαστάσεις, ενώ είναι περίπου 46 φορές πιο αργός σε σχέση με τον χρόνο που δαπανάται στην CPU και 13.5 φορές πιο αργός σε σχέση με τον συνολικό χρόνο για δεδομένα 5 διαστάσεων. Σύμφωνα με τα Σχήματα 5.7(γ′)-5.7(δ′) οι αλγόριθμοι εμφανίζουν αντίστοιχη συμπεριφορά και στα σύνολα δεδομένων με anticorrelated κατανομή. Αξίζει να αναφέρουμε ότι για τα πειράματά μας δοκιμάσαμε επίσης ακόμα υψηλότερες τιμές για τον αριθμό διαστάσεων, π.χ. για $D = 6$, τα οποία δεν έχουμε συμπεριλάβει στα διαγράμματα. Σε αυτή την περίπτωση ο αλγόριθμος BRS χρειάστηκε περίπου 15 ώρες για να τερματίσει στο σύστημά μας, ενώ ο αλγόριθμος RSA χρειάστηκε 20.2 λεπτά. Αντίστοιχη συμπεριφορά παρατηρήσαμε και στα πειράματα που εκτελέσαμε με πραγματικά δεδομένα, γεγονός που επιβεβαιώνει τη συζήτησή μας στην ενότητα 5.2.1 για τους περιορισμούς του αλγορίθμου BRS. Επίσης σε όλα τα πειράματα παρατηρούμε ότι ο αλγόριθμος bRSA έχει πολύ καλύτερη επίδοση σε σχέση με τους BRS και RSA για την περίπτωση ερωτημάτων k-MAC.

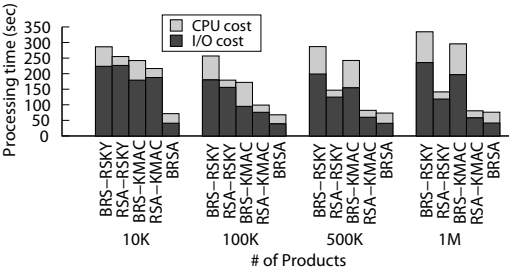
Μια άλλη σημαντική παρατήρηση που μπορούμε να κάνουμε είναι ότι όταν έχουμε μεγαλύτερο αριθμό διαστάσεων, το συνολικό κόστος επεξεργασίας καθορίζεται σε μεγάλο βαθμό από το χρόνο επεξεργασίας στην CPU και όχι από τις εισόδους/εξόδους από το δίσκο.[5] Αυτό συμβαίνει γιατί τα μεγέθη των συνόλων κορυφογραμμής $SKY(L)$ και $SKY(U)$ αυξάνονται δραματικά με τον αριθμό διαστάσεων (σύμφωνα και με την εργασία [12]). Επομένως, ο αριθμός ελέγχων κυριαρχίας που απαιτούνται για την εκτέλεση ενός αντίστροφου ερωτήματος κορυφογραμμής αυξάνεται κατακόρυφα. Σημειώνουμε ότι στο υπόλοιπο μέρος της πειραματικής μας αξιολόγησης χρησιμοποιήσαμε ως προκαθορισμένη τιμή για τον αριθμό διαστάσεων την τιμή $D = 3$, η οποία είναι σχετικά μικρή για πραγματικά δεδομένα. Συνεπώς, τα πειραματικά μας σενάρια είναι μάλλον ευνοϊκά προς τον ανταγωνιστικό αλγόριθμο BRS. Τα κέρδη στην απόδοση του αλγορίθμου RSA σε σχέση με τον BRS είναι σημαντικά μεγαλύτερα αν θεωρήσουμε μεγαλύτερο αριθμό διαστάσεων.

Επίδοση σε σχέση με το μέγεθος του συνόλου δεδομένων. Στη συνέχεια αναλύουμε πειραματικά την εξάρτηση της επίδοσης των αλγορίθμων από το μέγεθος του συνόλου δεδομένων. Αρχικά μεταβάλλουμε το μέγεθος του συνόλου δεδομένων για τα προϊόντα $|P|$. Τα Σχήματα 5.8(α′)-5.8(β′) και 5.8(γ′)-5.8(δ′) δείχνουν τα αποτελέσματα για τα σύνολα δεδομένων με uniform και anticorrelated κατανομές. Αξίζει να παρατήρησει κανείς τη διαφορετική συμπεριφορά μεταξύ των αλγορίθμων BRS και RSA σε σχέση με τον τύπο προσπελάσεων στο δίσκο (Σχήμα 5.8(α′)), συνέπεια της διαφορετικής σειράς επεξεργασίας που ακολουθείται. Ο αλγόριθμος BRS απαιτεί περισσότερες προσπελάσεις στο δίσκο για δεδομένα του συνόλου P, ενώ ο αλγόριθμος RSA εκτελεί περισσότερες προσπελάσεις για κόμβους του συνόλου C. Στην περίπτωση που τα μεγέθη των συνόλων P και C είναι παραπλήσια (100K εγγραφές το καθένα)

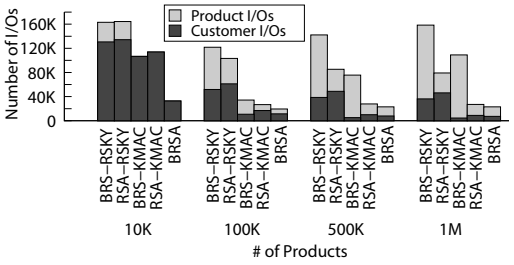
[5] Υπενθυμίζουμε ότι τα Σχήματα 5.7(α′)-5.7(δ′) είναι σε λογαριθμική κλίμακα.



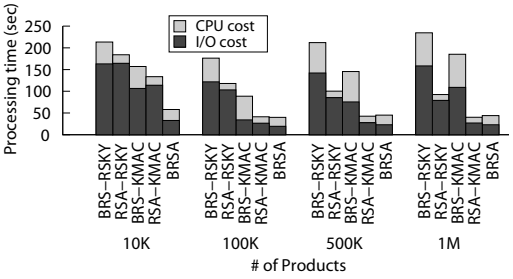
(α′) Πλήθος I/Os σε σχέση με το μέγεθος του συνόλου δεδομένων P (uniform dataset)



(β′) Συνολικό κόστος επεξεργασίας σε σχέση με το μέγεθος του συνόλου δεδομένων P (uniform dataset)



(γ′) Πλήθος I/Os σε σχέση με το μέγεθος του συνόλου δεδομένων P (anticorrelated dataset)



(δ′) Συνολικό κόστος επεξεργασίας σε σχέση με το μέγεθος του συνόλου δεδομένων P (anticorrelated dataset)

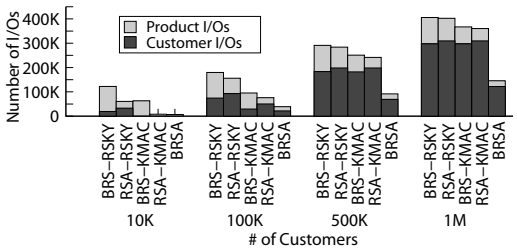
	10K		100K		500K		1M	
	BRS	RSA	BRS	RSA	BRS	RSA	BRS	RSA
I/O cost (sec)	223.6	226.5	179.9	156.0	198.9	125.0	235.5	118.5
CPU cost (sec)	62.7	28.6	76.9	23.2	87.9	22.0	98.8	22.9
Total cost (sec)	286.3	255.1	256.8	179.2	286.8	147.0	334.3	141.4

	10K	100K	500K	1M
	BRS / RSA	BRS / RSA	BRS / RSA	BRS / RSA
I/O cost ratio	0.99	1.15	1.59	1.99
CPU cost ratio	2.19	3.32	3.99	4.32
Total cost ratio	1.12	1.43	1.95	2.36

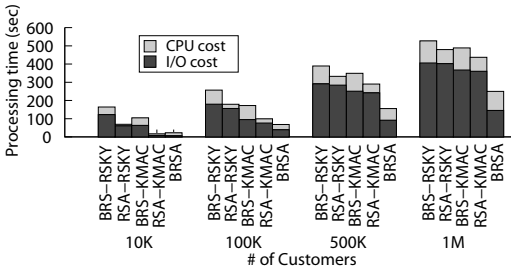
(ε′) Επιδόσεις αλγορίθμων RSA & BRS σε σχέση με το μέγεθος του συνόλου δεδομένων P (uniform dataset)

Σχήμα 5.8: *Επίδοση αλγορίθμων σε σχέση με το μέγεθος του συνόλου P*

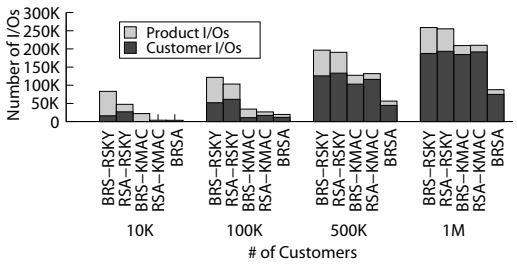
οι δύο αλγόριθμοι απαιτούν περίπου τον ίδιο αριθμό λειτουργιών εισόδου/εξόδου. Όμως, καθώς ο αριθμός προϊόντων αυξάνεται, όπως δείχνει το διάγραμμα 5.8(α′), η στρατηγική που ακολουθείται από τον αλγόριθμο RSA αποδεικνύεται πιο αποδοτική σε σχέση με τον συνολικό αριθμό I/Os που απαιτούνται. Επιπλέον ο χρόνος που δαπανάται στην CPU από τον αλγόριθμο RSA είναι σημαντικά μικρότερος και εμφανίζει



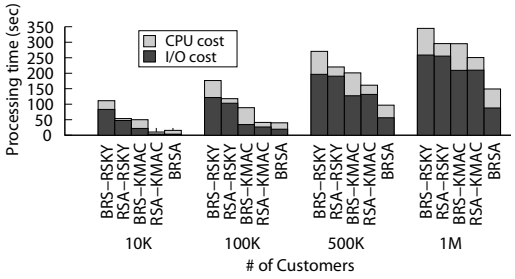
(α′) Πλήθος I/Os σε σχέση με το μέγεθος του συνόλου δεδομένων C (uniform dataset)



(β′) Συνολικό κόστος επεξεργασίας σε σχέση με το μέγεθος του συνόλου δεδομένων C (uniform dataset)



(γ′) Πλήθος I/Os σε σχέση με το μέγεθος του συνόλου δεδομένων C (anticorrelated dataset)



(δ′) Συνολικό κόστος επεξεργασίας σε σχέση με το μέγεθος του συνόλου δεδομένων C (anticorrelated dataset)

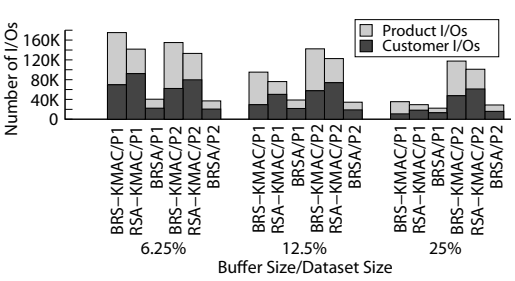
	10K		100K		500K		1M	
	BRS	RSA	BRS	RSA	BRS	RSA	BRS	RSA
I/O cost (sec)	122.4	60.6	179.9	156.0	291.0	284.1	405.8	402.4
CPU cost (sec)	41.6	8.5	76.9	23.2	98.1	48.1	121.5	77.0
Total cost (sec)	164.0	69.1	256.8	179.2	389.1	332.2	527.3	479.4

	10K	100K	500K	1M
	BRS / RSA	BRS / RSA	BRS / RSA	BRS / RSA
I/O cost ratio	2.02	1.15	1.02	1.01
CPU cost ratio	4.91	3.32	2.04	1.58
Total cost ratio	2.37	1.43	1.17	1.10

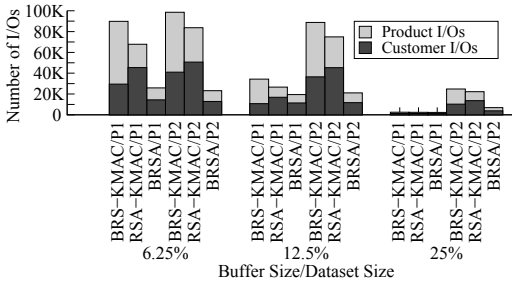
(ε′) Επιδόσεις αλγορίθμων RSA & BRS σε σχέση με το μέγεθος του συνόλου δεδομένων C (uniform dataset)

Σχήμα 5.9: *Επίδοση αλγορίθμων σε σχέση με το μέγεθος του συνόλου C*

καλύτερη συμπεριφορά κλιμάκωσης καθώς αυξάνεται το μέγεθος του συνόλου P. Για περισσότερη λεπτομέρεια, οι ακριβείς μετρήσεις φαίνονται στο Σχήμα 5.8(ε′). Τα Σχήματα 5.8(γ′)-5.8(δ′) δείχνουν αντίστοιχη συμπεριφορά για τους αλγορίθμους και στην περίπτωση δεδομένων που ακολουθούν την anticorrelated κατανομή, με μόνη διαφορά ότι οι χρόνοι εκτέλεσης είναι ελαφρά υψηλότεροι. Επίσης, όπως δείχνουν τα σχήμα-



(α′) Πλήθος I/Os σε σχέση με το μέγεθος της κρυφής μνήμης (uniform dataset)



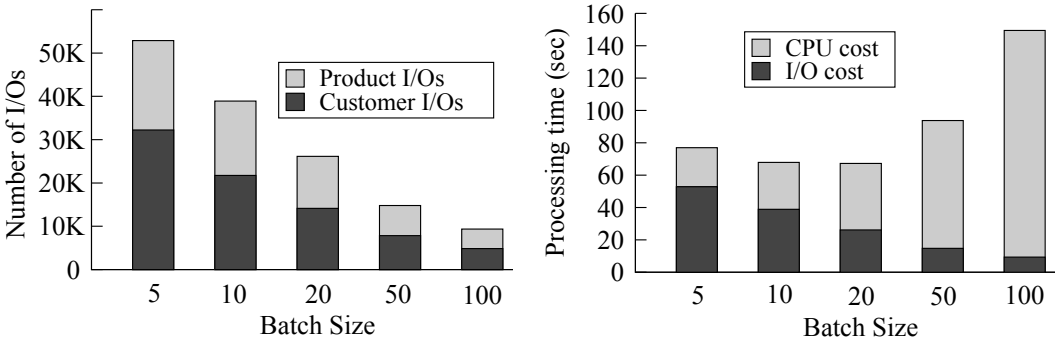
(β′) Πλήθος I/Os σε σχέση με το μέγεθος της κρυφής μνήμης (anticorrelated dataset)

Σχήμα 5.10: *Επίδοση αλγορίθμων σε σχέση με το μέγεθος της κρυφής μνήμης*

τα, ο αλγόριθμος bRSA είναι με διαφορά η πιο αποδοτική επιλογή για την περίπτωση ερωτημάτων k-MAC, ενώ δεν φαίνεται να επηρεάζεται ιδιαίτερα από τη μεταβολή του μεγέθους του συνόλου P.

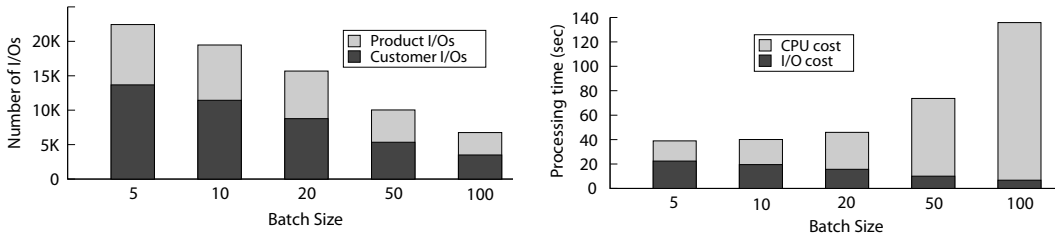
Στη συνέχεια συγκρίναμε την επίδοση όλων των αλγορίθμων καθώς μεταβάλλεται το μέγεθος του συνόλου C. Στα Σχήματα 5.9(α′)-5.9(β′) και 5.9(γ′)-5.9(δ′) απεικονίζεται ο αριθμός I/Os και οι χρόνοι επεξεργασίας για τα σύνολα δεδομένων που ακολουθούν uniform και anticorrelated κατανομές αντίστοιχα. Όπως φαίνεται στα διαγράμματα οι αλγόριθμοι BRS και RSA απαιτούν περίπου τον ίδιο αριθμό λειτουργιών εισόδου/εξόδου για τα μεγαλύτερα σύνολα δεδομένων C που εξετάσαμε. Στην περίπτωση που το μέγεθος του συνόλου C είναι αρκετά μεγαλύτερο από το μέγεθος του συνόλου P, η στρατηγική που ακολουθεί ο αλγόριθμος BRS φαίνεται περισσότερο ελπιδοφόρα, καθώς ο αριθμός επαναλήψεων του αλγορίθμου RSA είναι $O(|C|)$. Όμως, όπως φαίνεται από τα πειράματα (Σχήματα 5.9(β′)-5.9(ε′)), ακόμα και στο συγκεκριμένο σενάριο που είναι αρκετά δυσμενές για τον αλγόριθμο RSA, το συνολικό κόστος επεξεργασίας του αλγορίθμου RSA είναι μικρότερο από αυτό του αλγορίθμου BRS κυρίως εξαιτίας του σημαντικά χαμηλότερου χρόνου που δαπανάται στην CPU. Ομοίως και σε αυτό το πείραμα, ο αλγόριθμος bRSA είναι σημαντικά πιο αποδοτικός από τους αλγορίθμους BRS και RSA στην περίπτωση εκτέλεσης πολλαπλών αντίστροφων ερωτημάτων κορυφογραμμής.

Επίδοση σε σχέση με το μέγεθος της κρυφής μνήμης. Σε αυτό το πείραμα συγκρίνουμε τον αριθμό προσπελάσεων στο δίσκο που απαιτεί κάθε αλγόριθμος σε σχέση με το μέγεθος της μνήμης cache. Συγκεκριμένα μεταβάλαμε το μέγεθος του buffer από 50 σελίδες (το οποίο αντιστοιχεί σε 6.25% του μεγέθους των δεδομένων) έως 200 σελίδες (το οποίο αντιστοιχεί σε 25% του μεγέθους των δεδομένων). Πειραματιστήκαμε με δύο διαφορετικές στρατηγικές αντικατάστασης σελίδων. Η πρώτη στρατηγική την οποία συμβολίζουμε με $P1$ ακολουθεί τη λογική LRU. Για τη δεύτερη στρατηγική, θεωρήσαμε ότι οι κόμβοι ενός R-tree που βρίσκονται σε υψηλότερα επίπεδα έχουν μεγαλύτερη συχνότητα προσπέλασης, συνεπώς υλοποιήσαμε μια στρατηγική αντικατάστασης $P2$ που δίνει προτεραιότητα στην αντικατάσταση των σελίδων που βρίσκονται σε χαμηλά επίπεδα. Τα Σχήματα 5.10(α′) και 5.10(β′) δείχνουν τα μετρούμενα I/Os για σύνολα δεδομένων με uniform και anticorrelated κατανομές αντιστοίχως.



(α′) Πλήθος I/Os σε σχέση με το πλήθος ερωτημάτων ανά ομάδα (uniform dataset)

(β′) Συνολικό κόστος επεξεργασίας σε σχέση με το πλήθος ερωτημάτων ανά ομάδα (uniform dataset)



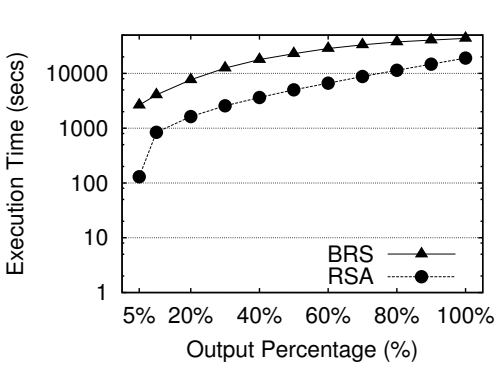
(γ′) Πλήθος I/Os σε σχέση με το πλήθος ερωτημάτων ανά ομάδα (anticorrelated dataset)

(δ′) Συνολικό κόστος επεξεργασίας σε σχέση με το πλήθος ερωτημάτων ανά ομάδα (anticorrelated dataset)

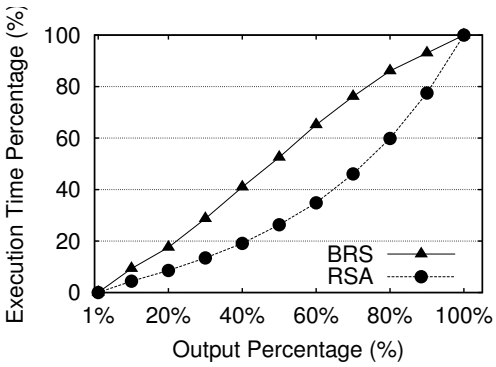
Σχήμα 5.11: *Επίδοση αλγορίθμων σε σχέση με το πλήθος ερωτημάτων ανά ομάδα*

Όπως παρατηρούμε, η στρατηγική LRU είναι ελαφρά πιο αποδοτική για το μέγεθος μνήμης cache που χρησιμοποιήσαμε στο βασικό μας σενάριο. Επίσης, ασχέτως της στρατηγικής αντικατάστασης ή του μεγέθους της μνήμης cache οι αλγόριθμοι RSA και bRSA είναι οι πιο αποδοτικοί.

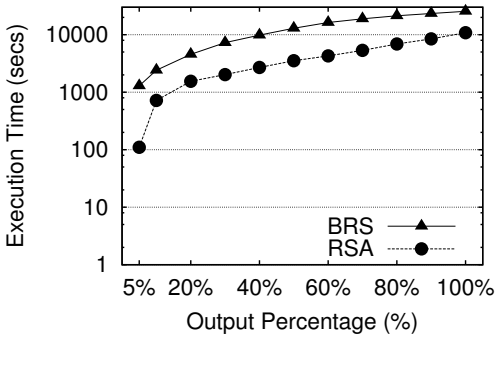
Επίδοση σε σχέση με το πλήθος ερωτημάτων ανά ομάδα. Στη συνέχεια αξιολογούμε πειραματικά την επίδοση του αλγορίθμου bRSA σε σχέση με τον αριθμό ερωτημάτων G που εκτελούνται παράλληλα σε μια ομάδα. Μεταβάλαμε την τιμή του G από 5 έως 100 ερωτήματα. Τα Σχήματα 5.11(α′)-5.11(β′) και 5.11(γ′)-5.11(δ′) δείχνουν τα πειραματικά αποτελέσματα όσον αφορά τον αριθμό I/Os και τον χρόνο επεξεργασίας στην CPU για τα σύνολα δεδομένων με uniform και anticorrelated κατανομές αντιστοίχως. Όπως αναμενόταν, όσο περισσότερα ερωτήματα εκτελούνται παράλληλα, τόσο λιγότερες προσπελάσεις στον δίσκο απαιτούνται, καθώς μία προσπέλαση χρησιμοποιείται για το κλάδεμα κόμβων σε πολλά ερωτήματα ταυτόχρονα. Όμως, όπως μπορούμε να παρατηρήσουμε από κάποιο αριθμό ερωτημάτων και πάνω, το συνολικό κόστος επεξεργασίας αυξάνεται σημαντικά, εξαιτίας του υψηλότερου χρόνου που απαιτείται για τη διαχείριση των ουρών προτεραιότητας και συνακόλουθα των περισσότερων ελέγχων κυριαρχίας που χρειάζονται. Όπως δείχνουν τα πειράματά μας, ο αλγόριθμος bRSA εμφανίζει τη βέλτιστη επίδοση για σχετικά μικρό αριθμό ερωτημάτων ανά ομάδα (περίπου 10)



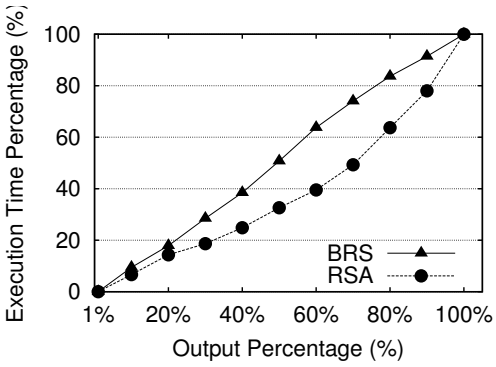
(α′) Χρόνος επεξεργασίας σε σχέση με τα παραγόμενα αποτελέσματα (uniform dataset)



(β′) Χρόνος επεξεργασίας/Συνολικός χρόνος (%) σε σχέση με τα παραγόμενα αποτελέσματα (uniform dataset)



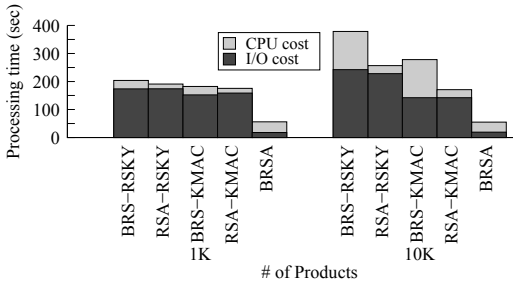
(γ′) Χρόνος επεξεργασίας σε σχέση με τα παραγόμενα αποτελέσματα (anticorrelated dataset)



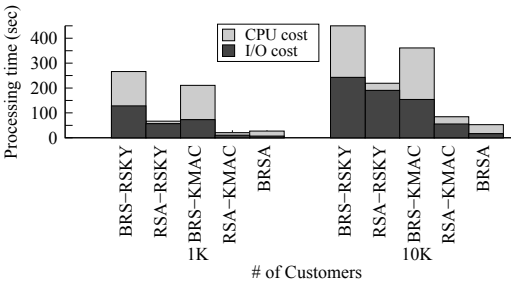
(δ′) Χρόνος επεξεργασίας/Συνολικός χρόνος (%) σε σχέση με τα παραγόμενα αποτελέσματα (anticorrelated dataset)

Σχήμα 5.12: *Προοδευτική παραγωγή αποτελεσμάτων*

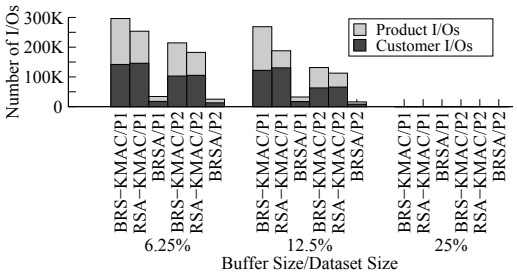
Προοδευτική παραγωγή αποτελεσμάτων. Στο συγκεκριμένο πείραμα εξετάζουμε την προοδευτική παραγωγή αποτελεσμάτων των αλγορίθμων BRS και RSA κατά την εκτέλεση ενός συνόλου $|Q|$ αντίστροφων ερωτημάτων κορυφογραμμής. Ο άξονας x αντιπροσωπεύει το ποσοστό αποτελεσμάτων που έχουν προσδιοριστεί σε σχέση με το τελικό σκορ επιρροής. Ο άξονας y δείχνει τον αντίστοιχο χρόνο εκτέλεσης που απαιτήθηκε σε απόλυτες τιμές (Σχήματα 5.12(α′) και 5.12(γ′)) και ως ποσοστό του συνολικού χρόνου αποτίμησης των ερωτημάτων (Σχήματα 5.12(β′) και 5.12(δ′)) για τα σύνολα δεδομένων που ακολουθούν την uniform και την anticorrelated κατανομή αντιστοίχως. Και τα δύο σχήματα δείχνουν ότι ο αλγόριθμος RSA έχει πολύ καλύτερη συμπεριφορά όσον αφορά την προοδευτική παραγωγή αποτελεσμάτων σε σχέση με τον BRS, ιδιαιτέρως για τον προσδιορισμό των πρώτων αποτελεσμάτων. Συγκεκριμένα, ο αλγόριθμος RSA επιστρέφει το 5% των αποτελεσμάτων στο ένα δέκατο του χρόνου που απαιτεί ο αλγόριθμος BRS. Το πλεονέκτημα αυτό είναι ιδιαίτερα σημαντικό για εφαρμογές που απαιτούν γρήγορη απόκριση ενώ δεν χρειάζονται το πλήρες σύνολο αποτελεσμάτων.



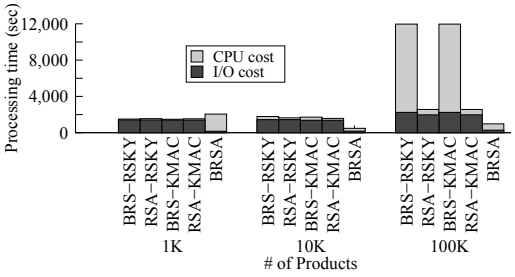
(α′) Συνολικό κόστος επεξεργασίας σε σχέση με το μέγεθος του συνόλου δεδομένων P (NBA dataset)



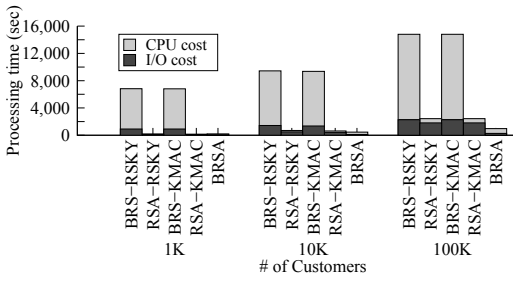
(β′) Συνολικό κόστος επεξεργασίας σε σχέση με το μέγεθος του συνόλου δεδομένων C (NBA dataset)



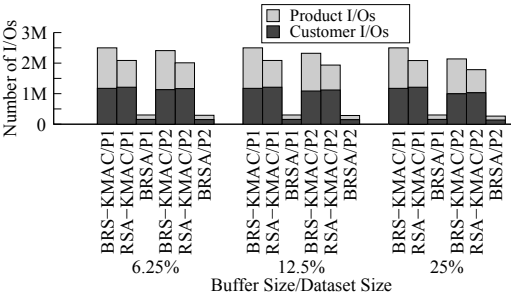
(γ′) Πλήθος I/Os σε σχέση με το μέγεθος της κρυφής μνήμης (NBA dataset)



(δ′) Συνολικό κόστος επεξεργασίας σε σχέση με το μέγεθος του συνόλου δεδομένων P (HOUSEHOLD dataset)



(ε′) Συνολικό κόστος επεξεργασίας σε σχέση με το μέγεθος του συνόλου δεδομένων C (HOUSEHOLD dataset)



(ϛ′) Πλήθος I/Os σε σχέση με το μέγεθος της κρυφής μνήμης (HOUSEHOLD dataset)

Σχήμα 5.13: *Πειράματα με πραγματικά σύνολα δεδομένων*

Πειραματικά αποτελέσματα για τα πραγματικά σύνολα δεδομένων. Τέλος, τα Σχήματα 5.13(α′)-5.13(γ′) και 5.13(δ′)-5.13(ϛ′) απεικονίζουν τα πειραματικά αποτελέσματα για τα πραγματικά σύνολα δεδομένων που εξετάσαμε, τα σύνολα NBA και HOUSEHOLD αντιστοίχως. Τα αποτελέσματα συμφωνούν με τα αποτελέσματα που είχαν προκύψει για συνθετικά δεδομένα. Όμως η διαφορά στην επίδοση του αλγορίθμου RSA σε σχέση με τον αλγόριθμο BRS είναι μεγαλύτερη στα πραγματικά σύνολα δεδομένων (ιδιαιτέρως σε σχέση με το μέγεθος του συνόλου C), κυρίως εξαιτίας του μεγαλύτερου αριθμού διαστάσεων (5 και 6 διαστάσεις αντιστοίχως). Σε αυτό

τον αριθμό διαστάσεων το μέγεθος των συνόλων επιρροής είναι αρκετά μεγαλύτερο στα πραγματικά σύνολα δεδομένων. Για παράδειγμα το μέσο σκορ επιρροής ήταν 196 στο σύνολο δεδομένων HOUSEHOLD έναντι 11 αποτελεσμάτων στο σύνολο δεδομένων που ακολουθεί την ομοιόμορφη κατανομή.

Κεφάλαιο 6

Επίλογος και Μελλοντικές Επεκτάσεις

6.1 Επίλογος

Στην παρούσα διατριβή εστιάσαμε στο αντικείμενο της εξατομίκευσης στη διαχείριση δεδομένων. Πιο συγκεκριμένα, ασχοληθήκαμε με θέματα που αφορούν την εξατομίκευση σε επίπεδο συστήματος διαχείρισης δεδομένων, αλλά και σε επίπεδο εφαρμογών (αλγόριθμοι, τεχνικές) προσεγγίζοντας το πρόβλημα τόσο από την πλευρά των χρηστών όσο και από την πλευρά των παρόχων (επιχειρήσεις). Πιο αναλυτικά, η συνεισφορά της παρούσας διατριβής μπορεί να συνοψιστεί στα παρακάτω σημεία.

Σε σχέση με το πρόβλημα της εξατομίκευσης στο επίπεδο ενός συστήματος βάσεων δεδομένων:

- Προτείναμε ένα νέο μοντέλο αναπαράστασης προτιμήσεων για σχεσιακά δεδομένα, το οποίο επιτρέπει τον ορισμό προτιμήσεων με βάση 3 άξονες: (α) μια συνθήκη προτίμησης (condition) με την οποία καθορίζονται οι εγγραφές που επηρεάζονται από μια προτίμηση, (β) μια συνάρτηση βαθμολόγησης των εγγραφών (ranking), και (γ) ένα βαθμό εμπιστοσύνης (confidence) που καθορίζει πόσο ισχυρή και βέβαιη είναι η συγκεκριμένη προτίμηση.

- Επεκτείναμε το σχεσιακό μοντέλο ώστε να μπορεί να διαχειρίζεται σχέσεις με προτιμήσεις (p-relations) εισάγοντας νέους τελεστές προτίμησης και επεκτείνοντας κατάλληλα τους υπάρχοντες σχεσιακούς τελεστές.

- Παρουσιάσαμε παραδείγματα σύνθετων ερωτημάτων προτίμησης που μπορούν να εκφραστούν με το προτεινόμενο μοντέλο καταδεικνύοντας τη δύναμη εκφραστικότητας και την ευελιξία της προσέγγισής μας.

- Υλοποιήσαμε ένα πρωτότυπο σύστημα διαχείρισης προτιμήσεων για το προτεινόμενο μοντέλο το οποίο προσφέρει τη δυνατότητα αποτίμησης διαφορετικών τύπων ερωτημάτων προτιμήσεων πάνω από μια τυπική σχεσιακή βάση δεδομένων.

- Βασισμένοι στις αλγεβρικές ιδιότητες του τελεστή προτίμησης, αρχικά προτείναμε ένα σύνολο ευριστικών κανόνων με στόχο να ελαχιστοποιήσουμε τον αριθ-

μό εγγραφών που επηρεάζονται από τους τελεστές προτίμησης. Στη συνέχεια προτείναμε μια μεθοδολογία βελτιστοποίησης βασισμένης στο κόστος. Χρησιμοποιώντας το πλάνο εκτέλεσης που παράγει το πρώτο βήμα βελτιστοποίησης ως είσοδο και ένα μοντέλο κόστους για την αποτίμηση προτιμήσεων, ο βελτιστοποιητής κάνει εκτίμηση του κόστους που θα έχει η αποτίμηση διαφορετικών εναλλακτικών πλάνων εκτέλεσης και επιλέγει αυτό με το ελάχιστο εκτιμώμενο κόστος.

- Παρουσιάσαμε νέες αποδοτικές μεθόδους αποτίμησης ερωτημάτων με προτιμήσεις σε μια σχεσιακή βάση δεδομένων. Οι μέθοδοι που προτείναμε αναμειγνύουν την αποτίμηση των προτιμήσεων με την εκτέλεση του ερωτήματος. Επιπλέον, ακολουθούν ένα σύνολο βελτιστοποιήσεων κατά την εκτέλεση του πλάνου ενώ παράλληλα χρησιμοποιούν κατά το δυνατόν την υποκείμενη μηχανή εκτέλεσης της βάσης δεδομένων για να επεξεργαστούν κομμάτια της ερώτησης που δεν σχετίζονται με προτιμήσεις.

- Αξιολογήσαμε πειραματικά τις προτεινόμενες μεθόδους για διάφορες κατηγορίες ερωτημάτων με τη χρήση πραγματικών δεδομένων και δείξαμε την αποδοτικότητα της προσέγγισής μας έναντι προσεγγίσεων που καταφεύγουν σε μετασχηματισμό των ερωτημάτων προτιμήσεων σε συμβατικά SQL ερωτήματα.

Σε σχέση με το πρόβλημα της εξατομίκευσης βάσει προτιμήσεων που εξαρτώνται από το τρέχον περιβάλλον χρήσης (context):

- Προτείναμε μια μεθοδολογία η οποία επιχειρεί να αντιμετωπίσει το πρόβλημα της απουσίας γνώσης των προτιμήσεων ενός χρήστη για το τρέχον περιβάλλον χρήσης. Για αυτό το σκοπό, εισάγαμε την έννοια των αβέβαιων προτιμήσεων και ορίσαμε τα πιθανοτικά ερωτήματα κορυφογραμμής που εξαρτώνται από το περιβάλλον χρήσης (Probabilistic Contextual Skylines - p-CSQ).

- Δοθέντος ενός συνόλου προτιμήσεων που ισχύουν για ένα σύνολο contexts και του τρέχοντος context, προτείναμε μια μέθοδο για την εξαγωγή πιθανοτήτων με τις οποίες ισχύει κάθε προτίμηση στο τρέχον context.

- Προτείναμε αλγορίθμους για την αποτίμηση p-CSQ οι οποίοι βασίζονται στην ύπαρξη ή όχι ευρετηρίων στα δεδομένα και είναι σημαντικά πιο αποδοτικοί από μια προσαρμοσμένη εκδοχή της μεθόδου εμφωλευμένων βρόχων. Η εκτενής πειραματική αξιολόγηση των μεθόδων αποδεικνύει την εγκυρότητα και την αποδοτικότητα των προτεινόμενων αλγορίθμων.

Σε σχέση με ερωτήματα ανάλυσης αγοράς που χρησιμοποιούν τις προτιμήσεις των καταναλωτών με στόχο την πιο αποτελεσματική προώθηση προϊόντων και υπηρεσιών:

- Αναπτύξαμε νέους αλγορίθμους για δύο προβλήματα που σχετίζονται με την ανάλυση μεγάλων όγκων καταναλωτικών προτιμήσεων, με πρακτικές εφαρμογές στην έρευνα αγοράς. Σχηματοποιήσαμε τα δύο προβλήματα ως παραλλαγές ενός και πολλαπλών αντίστροφων ερωτημάτων κορυφογραμμής αντιστοίχως.

- Προτείναμε έναν νέο αλγόριθμο, ονόματι RSA για την αποτίμηση αντίστροφων ερωτημάτων κορυφογραμμής. Ο αλγόριθμος RSA παρουσιάζει καλύτερη κλιμάκωση σε σύνολα δεδομένων που περιέχουν μεγάλο αριθμό αποτελεσμάτων που ανήκουν στην κορυφογραμμή (όπως π.χ. πολυδιάστατα δεδομένα), ενώ ταυτόχρονα παράγει τα πρώτα αποτελέσματα σημαντικά πιο γρήγορα από τον καλύτερο αλγόριθμο που έχει προταθεί στην έως σήμερα βιβλιογραφία.

- Αναπτύξαμε μια παραλλαγή του αλγορίθμου RSA για ομάδες ερωτημάτων ο οποίος μειώνει αισθητά τον απαιτούμενο χρόνο εκτέλεσης σε σχέση με το να επεξεργαζόμασταν κάθε επιμέρους ερώτημα ξεχωριστά, ομαδοποιώντας κατάλληλα παρόμοια υποψήφια προϊόντα, εκτελώντας κοινές προσπελάσεις στον δίσκο, και επιτρέποντας την ταυτόχρονη επεξεργασία πολλών ερωτημάτων. Στη συνέχεια εφαρμόσαμε τον νέο αυτό αλγόριθμο για την αποτίμηση ερωτημάτων k-MAC. Το ερώτημα k-MAC γενικεύει παρόμοια ερωτήματα που έχουν προταθεί σε προηγούμενες εργασίες [49, 57] για περιπτώσεις όπου οι προτιμήσεις των καταναλωτών συμπεριλαμβάνουν γνωρίσματα χωρίς αντικειμενικά βέλτιστη τιμή.

- Διεξάγαμε εκτενή πειραματική αξιολόγηση των προτεινόμενων αλγορίθμων τόσο σε πραγματικά δεδομένα όσο και σε δεδομένα που έχουν παραχθεί συνθετικά. Η πειραματική μελέτη μας καταδεικνύει ότι (α) ο αλγόριθμος RSA υπερτερεί αισθητά του αλγορίθμου BRS για την περίπτωση ενός αντίστροφου ερωτήματος κορυφογραμμής σε σχέση με την ταχύτητα εκτέλεσης (performance), τις δυνατότητες κλιμάκωσης (scalability), και την προοδευτική παραγωγή αποτελεσμάτων (progressiveness), ιδιαιτέρως για πολυδιάστατα δεδομένα ή όταν το μέγεθος του συνόλου προϊόντων είναι μεγαλύτερο από το μέγεθος του συνόλου των προτιμήσεων των καταναλωτών, και (β) ο αλγόριθμος που προτείνουμε για την ταυτόχρονη εκτέλεση πολλαπλών ερωτημάτων υπερτερεί έναντι μεθόδων που επεξεργάζονται κάθε ερώτημα ξεχωριστά.

6.2 Μελλοντικές επεκτάσεις

Όπως οι περισσότερες ερευνητικές εργασίες, η συγκεκριμένη διατριβή αφήνει ανοικτά εξίσου πολλά ερωτήματα με αυτά που επιχειρεί να επιλύσει. Παρακάτω αναφέρουμε επιγραμματικά ορισμένες ερευνητικές κατευθύνσεις προς τις οποίες μπορούν να επεκταθούν οι εργασίες που περιγράφουμε στην παρούσα διατριβή.

Σε σχέση με το πρόβλημα της εξατομίκευσης στο επίπεδο ενός συστήματος βάσεων δεδομένων ενδιαφέρουσες είναι οι επεκτάσεις στα παρακάτω αντικείμενα:

- Υλοποίηση του συστήματος PrefDB εντός του πυρήνα μιας βάσης δεδομένων και συνδυασμός των προτεινόμενων τεχνικών βελτιστοποίησης με άλλες τεχνικές όπως για παράδειγμα αυτές που έχουν προταθεί στις εργασίες [45, 46].

- Εφαρμογή των προτεινόμενων μεθόδων εξατομίκευσης για νέες κατηγορίες δεδομένων όπως για παράδειγμα RDF triples, ή streaming data προερχόμενα από κοινωνικά δίκτυα. Επίσης, προσαρμογή του συστήματος PrefDB σε νέου τύπου

αρχιτεκτονικές συστημάτων διαχείρισης δεδομένων όπως είναι οι NoSQL βάσεις δεδομένων.

- Σύνθετες εφαρμογές που συνδυάζουν τεχνικές εξατομίκευσης με προβλήματα που απασχολούν την έρευνα των συστημάτων συστάσεων. Για παράδειγμα, ένα βασικό πρόβλημα που έχουν πολλά συστήματα εξατομίκευσης είναι ότι πολλές φορές κάνουν overfit στις προτιμήσεις ενός χρήστη, επιστρέφοντας του κάθε φορά παρόμοια αποτελέσματα, χώρις ενδεχομένως να έχουν εξετάσει αν ο συγκεκριμένος χρήστης θα έβρισκε ως ενδιαφέροντα και άλλα αποτελέσματα με διαφορετικά χαρακτηριστικά. Για το λόγο είναι σκόπιμο οι υπάρχουσες τεχνικές εξατομίκευσης να μπορούν να συνδυαστούν με παράλληλη βελτίωση της νεωτερικότας (novelty), της τυχαιότητας/έκπληξης (serendipity), ή της ποικιλίας χαρακτηριστικών (diversity) των προτεινόμενων αποτελεσμάτων.

- Συνδυασμός της εξατομίκευσης με παράλληλη προστασία της ιδιωτικότητας των χρηστών. Ένα ενδιαφέρον πρόβλημα που ανακύπτει είναι πώς μπορούν να δημοσιευτούν εξατομικευμένες προτάσεις για κάποιον χρήστη χωρίς παράλληλα να μπορεί κάποιος να εκμαιεύσει με μεγάλο βαθμό βεβαιότητας τα ενδιαφέροντα ή τις ανάγκες του συγκεκριμένου χρήστη, ιδιαιτέρως αν αφορούν ευαίσθητα δεδομένα. Για παράδειγμα πρόσφατα[1] έγινε γνώστο ένα περιστατικό όπου μεγάλη αλυσίδα σούπερ μάρκετ παρείχε εξατομικευμένες προσφορές προϊόντων φροντίδας βρεφών βασισμένη στην ανάλυση της αγοραστικής συμπεριφοράς μιας οικογένειας. Ο πατέρας διαμαρτυρήθηκε στην εν λόγω εταιρία ότι κανένα μέλος της οικογένειας δεν ήταν έγκυος, ώσπου στη συνέχεια ανακάλυψε ότι η κόρη του ηλικίας 15 χρονών ήταν τελικά έγκυος χωρίς να το γνωρίζει. Συνεπώς, απαιτούνται τρόποι εξισορρόπησης της εξατομίκευσης με την προστασία της ιδιωτικότητας, ιδιαιτέρως για εφαρμογές που λαμβάνουν υπόψη τους ευαίσθητα προσωπικά δεδομένα, όπως για παράδειγμα αναζήτηση φαρμάκων για συγκεκριμένες ασθένειες, δεδομένα οικονομικού χαρακτήρα, κ.α.

Σε σχέση με το πρόβλημα της εξατομίκευσης βάσει προτιμήσεων που εξαρτώνται από το τρέχον περιβάλλον χρήσης (context) δύο πιθανές κατευθύνσεις μελλοντικής έρευνας είναι οι εξής:

- Επέκταση των προτεινόμενων μεθόδων στην περίπτωση αποτίμησης ερωτημάτων top-k όπου οι εγγραφές κατατάσσονται με βάση την πιθανότητα να ανήκουν στην κορυφογραμμή για το τρέχον context.

- Ανάπτυξη μεθόδων οι οποίες κάνουν χρήση της μνήμης cache, δηλαδή χρησιμοποιούν αποτελέσματα που έχουν υπολογιστεί για παρελθόντα contexts με σκοπό τη μείωση του χρόνου αποτίμησης ερωτημάτων p-CSQ για το τρέχον context.

Τέλος, σε σχέση με ερωτήματα ανάλυσης αγοράς που χρησιμοποιούν τις προτιμήσεις των καταναλωτών κάποιες πιθανές επεκτάσεις είναι οι εξής:

[1] http://www.businessinsider.com/the-incredible-story-of-how-target-exposed-a-teen-girls-pregnancy-2012-2

- Ανάπτυξη τεχνικών που ακολουθούν τη μεθοδολογία map-reduce για την παράλληλη επεξεργασία πολλαπλών αντίστροφων ερωτημάτων κορυφογραμμής.
- Επέκταση των ερωτημάτων k-MAC στην περίπτωση που οι ακριβείς προδιαγραφές των υποψήφιων προϊόντων δεν είναι γνωστές ή οι πιθανές τιμές τους είναι αβέβαιες ή πρέπει να προέλθουν από μια περιοχή τιμών (range) που ικανοποιεί κάποια κριτήρια κόστους.

Bibliography

[1] *Query templates*, http://tinyurl.com/8zs3e77.

[2] Gediminas Adomavicius and Alexander Tuzhilin, *Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions*, TKDE **17** (2005), no. 6, 734–749.

[3] Rakesh Agrawal, Ralf Rantzau, and Evimaria Terzi, *Context-sensitive ranking*, SIGMOD, 2006, pp. 383–394.

[4] Rakesh Agrawal and Edward L. Wimmers, *A framework for expressing and combining preferences*, SIGMOD, 2000, pp. 297–306.

[5] Reza Akbarinia, Esther Pacitti, and Patrick Valduriez, *Best position algorithms for top-k queries*, VLDB, 2007, pp. 495–506.

[6] Anastasios Arvanitis and Georgia Koutrika, *PrefDB: Bringing preferences closer to the DBMS*, SIGMOD, 2012, pp. 665–668.

[7] Anastasios Arvanitis, Antonios Deligiannakis, and Yannis Vassiliou, *Efficient influence-based processing of market research queries*, CIKM, 2012, pp. 1193–1202.

[8] Anastasios Arvanitis and Georgia Koutrika, *PrefDB: Supporting preferences as first-class citizens in relational databases*, Technical Report (2012), http://tinyurl.com/8ob2d8j.

[9] Anastasios Arvanitis and Georgia Koutrika, *Towards preference-aware relational databases*, ICDE, 2012, pp. 426–437.

[10] Wolf-Tilo Balke, Ulrich Güntzer, and Werner Kießling, *On real-time top-k querying for mobile services*, CoopIS, 2002, pp. 125–143.

[11] Ilaria Bartolini, Paolo Ciaccia, and Marco Patella, *Efficient sort-based skyline evaluation*, TODS **33** (2008), no. 4, 1–45.

[12] Jon Bentley, Kenneth Clarkson, and David Levine, *Fast linear expected-time algorithms for computing maxima and convex hulls*, SODA, 1990, pp. 179–187.

[13] Thomas Bernecker, Tobias Emrich, Hans-Peter Kriegel, Nikos Mamoulis, Matthias Renz, Shiming Zhang, and Andreas Züfle, *Inverse queries for multi-dimensional spaces*, SSTD, 2011, pp. 330–347.

[14] Stephan Börzsönyi, Donald Kossmann, and Konrad Stocker, *The skyline operator*, ICDE, 2001, pp. 421–430.

[15] Nicolas Bruno, Luis Gravano, and Amélie Marian, *Evaluating top-k queries over web-accessible databases*, ICDE, 2002, pp. 369–380.

[16] Chee Yong Chan, Pin-Kwang Eng, and Kian-Lee Tan, *Stratified computation of skylines with partially-ordered domains*, SIGMOD, 2005, pp. 203–214.

[17] Chee Yong Chan, H. V. Jagadish, Kian-Lee Tan, Anthony K. H. Tung, and Zhenjie Zhang, *Finding k-dominant skylines in high dimensional space*, SIGMOD, 2006, pp. 503–514.

[18] Kevin Chen-Chuan Chang and Seung won Hwang, *Minimal probing: supporting expensive predicates for top-k queries*, SIGMOD, 2002, pp. 346–357.

[19] Yuan-Chi Chang, Lawrence D. Bergman, Vittorio Castelli, Chung-Sheng Li, Ming-Ling Lo, and John R. Smith, *The onion technique: Indexing for linear optimization queries*, SIGMOD, 2000, pp. 391–402.

[20] Jan Chomicki, *Preference formulas in relational queries*, TODS **28** (2003), no. 4, 427–466.

[21] Jan Chomicki, Parke Godfrey, Jarek Gryz, and Dongming Liang, *Skyline with presorting*, ICDE, 2003, pp. 717–816.

[22] William W. Cohen, Robert E. Schapire, and Yoram Singer, *Learning to order things*, J. Artif. Intell. Res. (JAIR) **10** (1999), 243–270.

[23] Gautam Das, Dimitrios Gunopulos, Nick Koudas, and Dimitris Tsirogiannis, *Answering top-k queries using views*, VLDB, 2006, pp. 451–462.

[24] Evangelos Dellis and Bernhard Seeger, *Efficient computation of reverse skyline queries*, VLDB, 2007, pp. 291–302.

[25] Ke Deng, Xiaofang Zhou, and Heng Tao Shen, *Multi-source skyline query processing in road networks*, ICDE, 2007, pp. 796–805.

[26] Prasad Deshpande and Deepak P, *Efficient reverse skyline retrieval with arbitrary non-metric similarity measures*, EDBT, 2011, pp. 319–330.

[27] Ronald Fagin, *Combining fuzzy information from multiple systems*, PODS, 1996, pp. 216–226.

[28] Ronald Fagin, Amnon Lotem, and Moni Naor, *Optimal aggregation algorithms for middleware*, PODS, 2001, pp. 102–113.

[29] Parke Godfrey, Ryan Shipley, and Jarek Gryz, *Algorithms and analyses for maximal vector computation*, VLDBJ **16** (2007), no. 1, 5–28.

[30] Ulrich Güntzer, Wolf-Tilo Balke, and Werner Kießling, *Optimizing multi-feature queries for image databases*, VLDB, 2000, pp. 419–428.

[31] Dorit Hochbaum and Anu Pathria, *Analysis of the greedy approach in problems of maximum k-coverage*, NRL **45** (1998).

[32] Stefan Holland, Martin Ester, and Werner Kießling, *Preference mining: A novel approach on mining user preferences for personalized applications*, PKDD, 2003, pp. 204–216.

[33] Vagelis Hristidis, Nick Koudas, and Yannis Papakonstantinou, *Prefer: A system for the efficient execution of multi-parametric ranked queries*, SIGMOD, 2001, pp. 259–270.

[34] Ihab F. Ilyas, Walid G. Aref, and Ahmed K. Elmagarmid, *Supporting top-k join queries in relational databases*, VLDB, 2003, pp. 754–765.

[35] Thorsten Joachims, *Optimizing search engines using clickthrough data*, KDD, 2002, pp. 133–142.

[36] Werner Kießling, *Foundations of preferences in database systems*, VLDB, 2002, pp. 311–322.

[37] Jon M. Kleinberg, Christos H. Papadimitriou, and Prabhakar Raghavan, *A microeconomic view of data mining*, Journal of Data Mining and Knowledge Discovery **2** (1998), no. 4, 311–324.

[38] Flip Korn and S. Muthukrishnan, *Influence sets based on reverse nearest neighbor queries*, SIGMOD, 2000, pp. 201–212.

[39] Donald Kossmann, Frank Ramsak, and Steffen Rost, *Shooting stars in the sky: An online algorithm for skyline queries*, VLDB, 2002, pp. 275–286.

[40] Georgia Koutrika and Yannis E. Ioannidis, *Personalization of queries in database systems*, ICDE, 2004, pp. 597–608.

[41] Georgia Koutrika and Yannis E. Ioannidis, *Personalized queries under a generalized preference model*, ICDE, 2005, pp. 841–852.

[42] H. T. Kung, Fabrizio Luccio, and Franco P. Preparata, *On finding the maxima of a set of vectors*, Journal of the ACM **22** (1975), no. 4, 469–476.

[43] Michel Lacroix and Pierre Lavency, *Preferences: Putting more knowledge into queries*, VLDB, 1987, pp. 217–225.

[44] Ken C. K. Lee, Baihua Zheng, Huajing Li, and Wang-Chien Lee, *Approaching the skyline in z order*, VLDB, 2007, pp. 279–290.

[45] Justin Levandoski, Mohamed Mokbel, and Mohamed Khalefa, *FlexPref: A framework for extensible preference evaluation in database systems*, ICDE, 2010, pp. 828–839.

[46] Chengkai Li, Kevin Chen-Chuan Chang, Ihab F. Ilyas, and Sumin Song, *RankSQL: Query algebra and optimization for relational top-k queries*, SIGMOD, 2005, pp. 131–142.

[47] Cuiping Li, Beng Chin Ooi, Anthony K. H. Tung, and Shan Wang, *Dada: a data cube for dominant relationship analysis*, SIGMOD, 2006, pp. 659–670.

[48] Xiang Lian and Lei Chen, *Monochromatic and bichromatic reverse skyline search over uncertain databases*, SIGMOD, 2008, pp. 213–226.

[49] Chen-Yi Lin, Jia-Ling Koh, and Arbee L.P. Chen, *Determining k-most demanding products with maximum expected number of total customers*, TKDE (2012).

[50] Xuemin Lin, Yidong Yuan, Qing Zhang, and Ying Zhang, *Selecting stars: The k most representative skyline operator*, ICDE, 2007, pp. 86–95.

[51] Muhammed Miah, Gautam Das, Vagelis Hristidis, and Heikki Mannila, *Standing out in a crowd: Selecting attributes for maximum visibility*, ICDE, 2008, pp. 356–365.

[52] Chaitanya Mishra and Nick Koudas, *Interactive query refinement*, EDBT, 2009, pp. 862–873.

[53] Apostol Natsev, Yuan-Chi Chang, John R. Smith, Chung-Sheng Li, and Jeffrey Scott Vitter, *Supporting incremental join queries on ranked inputs*, VLDB, 2001, pp. 281–290.

[54] Surya Nepal and M. V. Ramakrishna, *Query processing issues in image (multimedia) databases*, ICDE, 1999, pp. 22–29.

[55] Dimitris Papadias, Yufei Tao, Greg Fu, and Bernhard Seeger, *Progressive skyline computation in database systems*, TODS **30** (2005), no. 1, 41–82.

[56] Jian Pei, Bin Jiang, Xuemin Lin, and Yidong Yuan, *Probabilistic skylines on uncertain data*, VLDB, 2007, pp. 15–26.

[57] Yu Peng, Raymond Chi-Wing Wong, and Qian Wan, *Finding top-k preferable products*, TKDE **24** (2012), no. 10, 1774–1788.

[58] Franco P. Preparata and Michael Ian Shamos, *Computational geometry: An introduction*, Springer, 1985.

[59] Nick Roussopoulos, Stephen Kelley, and Frédéic Vincent, *Nearest neighbor queries*, SIGMOD, 1995, pp. 71–79.

[60] Dimitris Sacharidis, Anastasios Arvanitis, and Timos Sellis, *Probabilistic contextual skylines*, ICDE, 2010, pp. 273–284.

[61] Dimitris Sacharidis, Panagiotis Bouros, and Timos Sellis, *Caching dynamic skyline queries*, SSDBM, 2008, pp. 455–472.

[62] Dimitris Sacharidis, Stavros Papadopoulos, and Dimitris Papadias, *Topologically sorted skylines for partially ordered domains*, ICDE, 2009, pp. 1072–1083.

[63] Patricia G. Selinger, Morton M. Astrahan, Donald D. Chamberlin, Raymond A. Lorie, and Thomas G. Price, *Access path selection in a relational database management system*, SIGMOD, 1979, pp. 23–34.

[64] Mehdi Sharifzadeh and Cyrus Shahabi, *The spatial skyline queries*, VLDB, 2006, pp. 751–762.

[65] Kostas Stefanidis, Marina Drosou, and Evaggelia Pitoura, *Perk: personalized keyword search in relational databases through preferences*, EDBT, 2010, pp. 585–596.

[66] Kostas Stefanidis, Evaggelia Pitoura, and Panos Vassiliadis, *Adding context to preferences*, ICDE, 2007, pp. 846–855.

[67] Kian-Lee Tan, Pin-Kwang Eng, and Beng Chin Ooi, *Efficient progressive skyline computation*, VLDB, 2001, pp. 301–310.

[68] Yufei Tao, Ling Ding, Xuemin Lin, and Jian Pei, *Distance-based representative skyline*, ICDE, 2009, pp. 892–903.

[69] Yufei Tao, Xiaokui Xiao, and Jian Pei, *Subsky: Efficient computation of skylines in subspaces*, ICDE, 2006, pp. 65–74.

[70] Ulrich Güntzer, Wolf-Tilo Balke, and Werner Kießling, *Towards efficient multifeature queries in heterogeneous environments*, ITCC, 2001, pp. 622–628.

[71] Akrivi Vlachou, Christos Doulkeridis, Yannis Kotidis, and Kjetil Nørvåg, *Reverse top-k queries*, ICDE, 2010, pp. 365–376.

[72] Akrivi Vlachou, Christos Doulkeridis, Kjetil Nørvåg, and Yannis Kotidis, *Identifying the most influential data objects with reverse top-k queries*, PVLDB **3** (2010), no. 1, 364–372.

[73] Qian Wan, Raymond Chi-Wing Wong, Ihab F. Ilyas, M. Tamer Özsu, and Yu Peng, *Creating competitive products*, PVLDB **2** (2009), no. 1, 898–909.

[74] Qian Wan, Raymond Chi-Wing Wong, and Yu Peng, *Finding top-k profitable products*, ICDE, 2011, pp. 1055–1066.

[75] Guoren Wang, Junchang Xin, Lei Chen, and Yunhao Liu, *Energy-efficient reverse skyline query processing over wireless sensor networks*, TKDE **24** (2011), no. 7, 1259–1275.

[76] Raymond Chi-Wing Wong, Ada Wai-Chee Fu, Jian Pei, Yip Sing Ho, Tai Wong, and Yubao Liu, *Efficient skyline querying with variable user preferences on nominal attributes*, PVLDB **1** (2008), no. 1, 1032–1043.

[77] Raymond Chi-Wing Wong, M. Tamer Özsu, Philip S. Yu, Ada Wai-Chee Fu, and Lian Liu, *Efficient method for maximizing bichromatic reverse nearest neighbor*, PVLDB **2** (2009), no. 1, 1126–1137.

[78] Raymond Chi-Wing Wong, Jian Pei, Ada Wai-Chee Fu, and Ke Wang, *Mining favorable facets*, KDD, 2007, pp. 804–813.

[79] Tianyi Wu, Yizhou Sun, Cuiping Li, and Jiawei Han, *Region-based online promotion analysis*, EDBT, 2010, pp. 63–74.

[80] Tianyi Wu, Dong Xin, Qiaozhu Mei, and Jiawei Han, *Promotion analysis in multi-dimensional space*, PVLDB **2** (2009), no. 1, 109–120.

[81] Xiaobing Wu, Yufei Tao, Raymond Chi-Wing Wong, Ling Ding, and Jeffrey Xu Yu, *Finding the influence set through skylines*, EDBT, 2009, pp. 1030–1041.

[82] Tian Xia, Donghui Zhang, Evangelos Kanoulas, and Yang Du, *On computing top-t most influential spatial sites*, VLDB, 2005, pp. 946–957.

[83] Dong Xin, Chen Chen, and Jiawei Han, *Towards robust indexing for ranked queries*, VLDB, 2006, pp. 235–246.

[84] Dong Xin and Jiawei Han, *P-cube: Answering preference queries in multi-dimensional space*, ICDE, 2008, pp. 1092–1100.

[85] Dong Xin, Jiawei Han, Hong Cheng, and Xiaolei Li, *Answering top-k queries with multi-dimensional selections: The ranking cube approach*, VLDB, 2006, pp. 463–475.

[86] Man Lung Yiu and Nikos Mamoulis, *Efficient processing of top-k dominating queries on multi-dimensional data*, VLDB, 2007, pp. 483–494.

[87] Yidong Yuan, Xuemin Lin, Qing Liu, Wei Wang, Jeffrey Xu Yu, and Qing Zhang, *Efficient computation of the skyline cube*, VLDB, 2005, pp. 241–252.

[88] Lei Zou and Lei Chen, *Dominant graph: An efficient indexing structure to answer top-k queries*, ICDE, 2008, pp. 536–545.

Παράρτημα Α΄

Μεταφράσεις Ξένων Όρων

Μετάφραση	Αγγλικός όρος
αλγόριθμος Fagin	Fagin's Algorithm - FA
αλγόριθμος διαίρει και βασίλευε	Divide and Conquer - DC
αληλεπίδραση ανθρώπου-μηχανής	Human-Computer Interaction
ανάκτηση πληροφοριών	Information Retrieval
αντίστροφο ερώτημα top-k	reverse top-k query
αντίστροφο ερώτημα εγγύτερου γείτονα	reverse nearest neighbor query - RNN
αντίστροφο ερώτημα κορυφογραμμής	reverse skyline query
αλγόριθμος κατωφλίου	Threshold Algorithm - TA
αλγόριθμος 'κλαδέματος - περίφραξης '	Branch and Bound Skyline - BBS
αλγόριθμος προτίμησης	preference algorithm
αποθηκευμένη διαδικασία	stored procedure
βέλτιστη χωροθέτηση	facility location planning
βέλτιστο σύνολο κατά Pareto	Pareto optimal set
βελτιστοποιητής ερωτημάτων	query optimizer
δεικτοδότηση	indexing
διασωλήνωση	pipelining
δυναμική κυριαρχία	dynamic dominance
δυναμικό ερώτημα κορυφογραμμής	dynamic skyline query
εγγραφή	tuple
ελάχιστος περίεχων κύβος	minimum bounding box - MBB
έλεγχος κυριαρχίας	dominance check
ενδιάμεσο λογισμικό	middleware
εξατομίκευση	personalization
εξαγωγή/εκμάθηση προτιμήσεων	preference elicitation/learning
εξόρυξη δεδομένων	Data Mining
επιλεκτικότητα	selectivity
επιχειρησιακή έρευνα	Operational Research
ερώτημα ανάκτησης των κορυφαίων k	top-k query
ερώτημα εγγύτερου γείτονα	nearest neighbor query - NN
ερώτημα κορυφογραμμής	skyline query

ερώτημα περιοχής	range query
ευρετήριο	index
ζώνη επιρροής	influence region
καμπύλη γεμίσματος χώρου	space filling curve
κατευθυνόμενος ακυκλικός γράφος	directed acyclic graph - DAG
κοινωνικός ιστός	social web
λήψη αποφάσεων με πολλαπλά κριτήρια	multi-criteria decision making
μέθοδος εμφωλευμένων βρόχων	Block Nested Loops - BNL
μέθοδος επιπέδου εφαρμογής	plug-in method
μέθοδος επιπέδου συστήματος	native/built-in method
μηχανική μάθηση	Machine Learning
ουρά προτεραιότητας	priority heap
περιβάλλον χρήσης	context
πλαίσιο	framework
πλάνο εκτέλεσης ερωτήματος	query execution plan
προτίμηση	preference
σειριακή προσπέλαση	sorted access
σκορ επιρροής	influence score
συναθροιστική συνάρτηση	aggregate function
συνάρτηση βαθμολόγησης	scoring function
συνάρτηση ωφέλειας	utility function
σύνολο δεδομένων	dataset
σύνολο επιρροής	influence set
σύστημα εξατομίκευσης	personalization system
σχέση κυριαρχίας	dominance relationship
σωρός	heap
τελεστής ένωσης	union operator
τελεστής επιλογής	select operator
τελεστής προβολής	project operator
τελεστής προτίμησης	prefer operator
τελεστής σύζευξης	join operator
τελεστής τομής	intersection operator
τεχνητή νοημοσύνη	Artificial Intelligence
τυχαία προσπέλαση	random access
υλοποιημένη όψη	materialized view
υπηρεσίες βασισμένες στη θέση	Location-Based Services - LBS

Παράρτημα Β′

Βιογραφικό Σημείωμα

Στοιχεία Επικοινωνίας

Εργαστήριο Συστημάτων Βάσεων Γνώσεων και Δεδομένων
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Εθνικό Μετσόβιο Πολυτεχνείο
Ηρώων Πολυτεχνείου 9. Ζωγράφου
157 80 Αθήνα, Ελλάδα

Τηλέφωνο: (+30) 210 772 1402
Fax: (+30) 210 772 1442
Ηλεκτρονικό ταχυδρομείο (e-mail): anarv-at-dblab.ece.ntua.gr
Προσωπική Ιστοσελίδα: http://www.dblab.ece.ntua.gr/~tasosarvanitis

Σπουδές

- **Εθνικό Μετσόβιο Πολυτεχνείο**, Ελλάδα (2006 - σήμερα)
  Υποψήφιος Διδάκτωρ Σχολής Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
  Περιοχή έρευνας: Εξατομικευμένα Συστήματα Διαχείρισης Δεδομένων
  Επιβλέπων: καθ. Ιωάννης Βασιλείου

- **Εθνικό Μετσόβιο Πολυτεχνείο**, Ελλάδα (2000 - 2005)
  Διπλωματούχος Σχολής Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
  Βαθμός: 8.70/10
  Διπλωματική εργασία: Ολοκλήρωση Συστημάτων και Εφαρμογών μέσω Web Services
  Επιβλέπων: καθ. Τιμολέων Σελλής

Ερευνητικά Ενδιαφέροντα

- Εξατομικευμένα Συστήματα Διαχείρισης Δεδομένων (Personalization in Data Management Systems)

- Διαχείριση Δεδομένων Εξαρτώμενων από το Ερμηνευτικό Περιβάλλον (Context-aware Data Management)
- Αποτίμηση Ερωτημάτων Κορυφογραμμής και Ανάκτησης των Κορυφαίων k Αποτελεσμάτων (Skyline and top-k query processing)
- Συστήματα Συστάσεων (Recommender Systems)

Διακρίσεις

- **Εθνικό Μετσόβιο Πολυτεχνείο** (2010 - σήμερα)
  Υπότροφος - Ηράκλειτος ΙΙ
- **Εθνικό Μετσόβιο Πολυτεχνείο** (2007 - 2010)
  Υπότροφος - Ειδικός Λογαριασμός Έρευνας (ΕΛΕ)
- **Θωμαϊδιο Βραβείο για την Πρόοδο των Επιστημών** (2009)
  για την εργασία Towards Preference-aware Relational Databases
- **ACM SIGIR, SIGWEB, Google** (2012)
  CIKM'12 Student Travel Grant

Δημοσιεύσεις

1. Anastasios Arvanitis, Georgia Koutrika, **PrefDB: Supporting Preferences as First-Class Citizens in Relational Databases**, in IEEE Transactions on Knowledge and Data Engineering (TKDE) vol. pages

2. Anastasios Arvanitis, Antonios Deligiannakis, Yannis Vassiliou, **Efficient Influence-based Processing of Market Research Queries**, in Proceedings of the 21st ACM International Conference on Information and Knowledge Management (CIKM '12), Maui, Hawaii, USA, October 29-November 2, 2012

3. Vassiliki Pouli, John Baras, Anastasios Arvanitis, **Increasing Message Relevance in Social Networks via Context-based Routing**, in Proceedings of the 3rd International Workshop on Modeling Social Media (MSM '12), Milwaukee, Wisconsin, USA, June 25, 2012

4. Anastasios Arvanitis, Georgia Koutrika, **PrefDB: Bringing Preferences closer to the DBMS (demo paper)**, in Proceedings of the 38th ACM SIGMOD Conference (SIGMOD '12), Scottsdale, Arizona, USA, May 20-24, 2012

5. Anastasios Arvanitis, Georgia Koutrika, **Towards Preference-aware Relational Databases**, in Proceedings of the 28th IEEE International Conference on Data Engineering (ICDE '12), Washington D.C., USA, April 1-5, 2012

6. Dimitris Sacharidis, Anastasios Arvanitis, Timos Sellis, **Probabilistic Contextual Skylines**, in Proceedings of the 26th IEEE International Conference on Data Engineering (ICDE '10), Long Beach, California, USA, March 1-6, 2010

Παρουσιάσεις

1. **Efficient Influence-based Processing of Market Research Queries**, presented in the 21st ACM International Conference on Information and Knowledge Management (CIKM '12), Maui, Hawaii, USA, November 1st, 2012

2. **User-Centered Data Management**, Invited Talk, University of California, Riverside, California, USA, May 29th, 2012

3. **PrefDB: Supporting Preferences as First-Class Citizens in Relational Databases**, Invited Talk, Institute for the Management of Information Systems, Athena R.C., Athens, Greece, May 17th, 2012

4. **Towards Preference-aware Relational Databases**, presented in the 28th IEEE International Conference on Data Engineering (ICDE '12), Washington D.C., USA, April 2nd, 2012

5. **Personalized Data Management Systems**, Thesis Proposal, National Technical University of Athens (NTUA), Greece, February 8th, 2011

6. **Towards Preference-aware Relational Databases**, Invited Talk, Institute for the Management of Information Systems, Athena R.C., Athens, Greece, November 26th, 2010

7. **Probabilistic Contextual Skylines**, DB Seminars Tutorial, National Technical University of Athens (NTUA), Greece, January 11th, 2010

8. **Finding the top-k Influential Points through Reverse Skylines**, Hong Kong University of Science and Technology (HKUST), Hong Kong, China, May 20th, 2009

9. **A Survey of Context and Context-aware Data Management**, DB Seminars Tutorial, National Technical University of Athens (NTUA), Greece, December 10th, 2007

Ακαδημαϊκή Εμπειρία

- **Εξωτερικός Κριτής**
  - SIGMOD (2013)
  - VLDB (2010)
  - EDBT (2012)

- CIKM (2009)
- WSDM (2013)
- ADBIS (2012)
- ICE-B (2009)
- TKDE
- DKE
- JWS

- **Οργάνωση Επιστημονικών Συνεδρίων**
  - Web Administrator στο PersDB'11 workshop (2011)
  - Web Administrator στο PersDB'10 workshop (2010)
  - Student Staff Member στο CIKM'12 (2012)
  - Student Staff Member στο SIGMOD'12 (2012)

- **Visiting Research Scholar**
  - University of California, Riverside (host Associate Prof. Vagelis Hristidis) (Ιούνιος 2012 - σήμερα)
  - Infolab at Stanford University (host Prof. Hector Garcia-Molina) (Φεβρουάριος - Μάρτιος 2010)
  - Hong Kong University of Science and Technology (HKUST) (host Prof. D. Papadias) (Μάιος - Ιούνιος 2009)

- **Βοηθός Διδασκαλίας**
  - Βάσεις Δεδομένων (Χειμερινό 2009, Χειμερινό 2010)
  - Ανάλυση και Σχεδιασμός Πληροφοριακών Συστημάτων (Χειμερινό 2007)

- **Συνεπιβλέπων Διπλωματικών Εργασιών**
  - *Εμμανουήλ Μαρούδας*, **Γραφικό Περιβάλλον Χρήστη για Εξατομικευμένο Σύστημα Διαχείρισης Δεδομένων** (2012)

Ακαδημαϊκά Έργα

- **Ινστιτούτο Πληροφοριακών Συστημάτων (ΙΠΣΥ) - Ερευνητικό Κέντρο Αθηνά**, Ελλάδα
  - TALOS (2009 - 2010)

- **Εθνικό Μετσόβιο Πολυτεχνείο - Σχολή Αγρονόμων Τοπογράφων Μηχανικών**, Ελλάδα
  - Σχεδιασμός και ανάπτυξη θεματικού portal και ψηφιακής βάσης Γεωγνώσης (2006)

Εργασιακή Εμπειρία

- **Satways LTD**, Ελλάδα (2007 - 2008)
  Software Engineer (Java, SQL Server, JBoss, Apache Tomcat, Eclipse)
  - Εξωτερικός συνεργάτης της Siemens SBT/SES/CCS στο έργο Athens C4I (Command, Control, Communications, Computers and Intelligence)
  - Υπεύθυνος για την ανάλυση, σχεδιασμό, ανάπτυξη και εγκατάσταση του συστήματος AIS (Automatic Identification System). Το σύστημα AIS αφορά την παρακολούθηση σε πραγματικό χρόνο, εντοπισμό και οπτικοποίηση εμπορικών πλοίων και σκαφών του Λιμενικού Σώματος και διαλειτουργεί με τα υποσυστήματα ELS/GEOFIS (Σύστημα Διαχείρισης Κρίσης) και Asset Management (Σύστημα Διαχείρισης Πόρων). Το σύστημα AIS χρησιμοποιείται αυτή τη στιγμή επιχειρησιακά από το Λιμενικό Σώμα.
  - Σχεδίαση και ανάπτυξη ενός τρισδιάστατου συστήματος οπτικοποίησης πλοίων. Ολοκλήρωση του συστήματος με το Skyline Terra Explorer (3D GIS) και ενσωμάτωση σε αυτό ενός αλγορίθμου εύρεσης βέλτιστης διαδρομής για πλοία
  - Σχεδίαση και ανάπτυξη του προϊόντος AutoTrack AVL Client (Automatic Vehicle Location System)

- **Saicon LTD**, Ελλάδα (2008)
  Software Engineer (Java EE (EJB 3, JPA), JBoss Drools, SQL Server, Netbeans)
  - Σχεδίαση και ανάπτυξη μιας πλατφόρμας gaming για χρήστες κινητών συσκευών με υποστήριξη πολλών χρηστών

- **Exodus SA**, Ελλάδα (2006 - 2007)
  Software Engineer (Java, JBoss, JavaScript, ASP, SQL Server, Eclipse)
  Business Process Management Department
  - Ανάπτυξη του προϊόντος iPerform workflow engine v2.0
  - Μετάπτωση των εγκαταστάσεων του iPerform στις εταιρίες Τράπεζα Πειραιώς και Wind Hellas Telecommunications S.A.
  - Εγκατάσταση, παραμετροποίηση και testing του iPerform v2.0 στη Wind Hellas Telecommunications S.A.
  - Σχεδίαση και ανάπτυξη νέων διαδικασιών και φορμών για το iPerform

- Σχεδίαση και ανάπτυξη διαδικασιών και φορμών για φορείς της τοπικής αυτοδιοίκησης στα πλαίσια έργων της πρόσκλησης 114 της ΚτΠ (Ηλεκτρονική Εξυπηρέτηση των Πολιτών)
- Σχεδίαση, ανάπτυξη και υποστήριξη για ἤανγε Ρεχυεστς στις εγκαταστάσεις του προϊόντος iPerform στις εταιρίες Wind Hellas Telecommunications S.A., Τράπεζα Πειραιώς και ΕΤΒΑ
- Συγγραφή τεκμηρίωσης και εκπαιδευτικού υλικού για χρήστες και developers του προϊόντος iPerform
- Εκπαιδευτικά σεμινάρια για χρήστες και developers του προϊόντος iPerform
- Εσωτερικά σεμινάρια εκπαίδευσης στην BPMN (Business Process Modeling Notation), και σε workflow patterns and best practices

- **Telesto Technologies**, Ελλάδα (2006)
  Software Engineer (Java, Apache Tomcat, Oracle)
  - Εξωτερικός συνεργάτης σε έργο του ΕΔΕΤ (Εθνικό Δίκτυο Έρευνας και Τεχνολογίας) (Σχεδίαση και ανάπτυξη web εφαρμογής για τη διαχείριση στοιχείων ελληνικών ερευνητικών φορέων)

- **Γενικό Επιτελείο Στρατού/Κέντρο Πληροφοριακής Υποστήριξης Ελληνικού Στρατού (ΓΕΣ/ΚΕΠΥΕΣ)**, Ελλάδα (2005–2006)
  Software Engineer, Network/System Administrator (J2EE, EJB 2.1, BEA Weblogic, Oracle, Borland JBuilder)
  Σώμα Έρευνας Πληροφορικής, Ειδικότητα Αναλυτής - Προγραμματιστής
  - Ανάπτυξη και υποστήριξη του ενδιάμεσου επιπέδου της εφαρμογής ΣΔΕΠ (Σύστημα Διοίκησης Ελέγχου Πληροφοριών) του ΓΕΣ/ΚΕΠΥΕΣ
  - Network/System administrator της ΞΞΙΙΙ ΤΘΤ, Αλεξανδρούπολη

Τεχνικές Ικανότητες

- **Γλώσσες Προγραμματισμού**: Java, C/C++, PHP, JavaScript, Pascal, Fortran, Prolog
- **Εργαλεία και Τεχνολογίες Ανάπτυξης Λογισμικού**: Eclipse, NetBeans, Oracle JDeveloper, Oracle Developer, Borland JBuilder, Ant scripting, CVS, SVN, Unix scripting basics
- **Συστήματα Διαχείρισης Βάσεων Δεδομένων**: PostgreSQL/pgSQL, Microsoft SQL Server, Oracle, MySQL, Microsoft Access
- **Τεχνολογίες Web**: AJAX, jQuery, WebSockets, CSS, XHTML, XML, XML Schema, RDF/S, OWL, Web Services/SOA

- **Τεχνολογίες Java**: EJB 2.1 & 3.0, JPA, Hibernate, JDBC, Application Servers (JBoss, BEA WebLogic, Apache Tomcat), Web Frameworks (Struts, Seam, Oracle ADF Faces, Facelets, JSP, Servlets, JSF, Applets), AXIS, JAXB, JAXP, JNDI, JBoss Drools, JMF, Sockets, Swing
- **Web/Application Servers**: Apache HTTP Server, Microsoft IIS
- **Άλλα**: Latex, Microsoft Office Visio, Dreamweaver, UML, BPMN, ArcGIS 8.3 & 9.x

Ξένες Γλώσσες

- Αγγλικά (Certificate of Proficiency in English, University of Cambridge 1998, Certificate of Proficiency in English, University of Michigan 1999)
- Γερμανικά (Zertifikat Deutsch als Fremdsprache, Goethe Institut 1997)
- Γαλλικά (μέτρια κατανόηση)