Πανεπιστήμιο Πελοποννήσου
Σχολή Θετικών Επιστημών και Τεχνολογίας
Τμήμα Επιστήμης και Τεχνολογίας Υπολογιστών

Μεταπτυχιακή Εργασία

Πολυδιάστατα Ευρετήρια για Ερωτήσεις Πλησιέστερων Γειτόνων

Λυκουρέντζος Χρήστος
Α.Μ.: 2009017

Επιβλέπων Καθηγητής
κ. Βασιλάκης Κωνσταντίνος

Τρίπολη Απρίλιος 2013

Πολυδιάστατα Ευρετήρια για Ερωτήσεις Πλησιέστερων Γειτόνων

Λυκουρέντζος Χρήστος

15 Απριλίου 2013

Περίληψη

Η αναζήτηση πλησιέστερων γειτόνων είναι ένα κλασικό πρόβλημα με πολλές εφαρμογές σε τομείς όπως η τεχνητή νοημοσύνη, η αναγνώριση προτύπων, η ανάκτηση πληροφορίας και άλλα.

Η παρούσα πτυχιακή εργασία αποτελεί μια προσπάθεια παρουσίασης, βελτίωσης και επέκτασης διαφόρων τεχνικών πάνω στο πρόβλημα της αναζήτησης πλησιέστερων γειτόνων.

Παρουσιάζεται η μέθοδος iDistance, η οποία είναι μια μέθοδος δεικτοδότησης για την αναζήτηση του K- πλησιέστερου γείτονα σε πολυδιάστατο μετρικό χώρο. Η μέθοδος iDistance είναι βασισμένη σε ένα αποτελεσματικό δένδρο B+. Στη συνέχεια έγινε επιλογή των σημείων αναφοράς και διαμέρισης του χώρου δεδομένων και παρουσιάζονται διάφορα συμπεράσματα που προέκυψαν από τη μέθοδο αυτή.

Στη συνέχεια, μελετάμε μια άλλη μέθοδο για την αναζήτηση πλησιέστερων γειτόνων, την LSH. Παρουσιάζονται και άλλες τεχνικές, όπως η μέθοδος Adhoc-LSH, η Rigorous LSH που παρουσιάζουν, αναλύουν και προτείνουν λύσεις για το εν λόγω πρόβλημα. Μελετούμε τη δομή Δένδρο LSB καθώς και τον αλγόριθμο NN, παραθέτοντας μια ανάλυσή του καθώς και των επεκτάσεων που προκύπτουν.

Τέλος παρουσιάζεται και αναλύεται μια πρωτότυπη προσέγγιση για την αποδοτική αναζήτηση ομοιότητας και ταξινόμησης σε πολυδιάστατα δεδομένα, η μέθοδος ευρετηρίου MedRank. Στο οικείο κεφάλαιο παρουσιάζεται αναλυτικά η εν λόγω μέθοδος και ο αλγόριθμος MedRank καθώς και διάφορες παραλλαγές του.

Abstract

The query for nearest neighbour is a well known, problem with many implementations on different sectors, such as artificial intelligence, pattern recognition and information retrieval. The present diploma thesis is an effort to presentat, improve and expand techniques coping with the problem of nearest neighbour query.

Method iDistance, which is an indexing method for K-nearest neighbour query in a multi-dimensional metric space, was studied and is presented. iDistance is based on an eficient B+ Tree structure. Reference points selection and data space partitioning were made.

Afterwards, we study the LSH method, considering extensions such as Adhoc-LSH and Rigorous LSH. These methods present, analyse and try to give a solution on the existing problem, while details on hash functions are also given. The LSB Tree and the NN algorithm are presented.

Finally, an analysis of a novel approach, tackling effective similarity search and classification regarding multi-dimensional data, namely MedRank is presented. We also consider its variations, OmedRank and L2TA.

Ευχαριστίες

Θα ήθελα να ευχαριστήσω θερμά τον Αναπληρωτή Καθηγητή κ. Βασιλάκη Κωνσταντίνο για την ακούραστη υποστήριξη του καθ' όλη την διάρκεια της εργασίας, για τις εύστοχες παρατηρήσεις και σχόλια του, που βοήθησαν να ξεπεράσω τα οποιαδήποτε εμπόδια προέκυπταν και να διευκρινιστούν αρκετά σημεία της εργασίας αυτής.

Επίσης, θα ήθελα να ευχαριστήσω τον Επίκουρο Καθηγητή κ. Θεοχάρη Μαλαμάτο για την καθοδήγηση και τις συμβουλές του.

Περιεχόμενα

Κατάλογος σχημάτων

Κεφάλαιο 1

Εισαγωγή

1.1 Δομή Εργασίας

Πολλές είναι οι εφαρμογές οι οποίες διαχειρίζονται πολυδιάστατα δεδομένα. Μια από τις πιο συχνές και ακριβές λειτουργίες είναι να βρεθούν αντικείμενα σε πολυδιάστατη βάση δεδομένων τα οποία να είναι παρόμοια με ένα δοσμένο αντικείμενο αναζήτησης. Σε τέτοιες περιπτώσεις, η αναζήτηση του πλησιέστερου γείτονα αποτελεί βασική προϋπόθεση.

Με το πρόβλημα της αναζήτησης του πλησιέστερου γείτονα έχουν ασχοληθεί πολλές έρευνες, και γι' αυτό τον σκοπό έχουν αναπτυχθεί πολλοί τύποι ευρετηρίων για πολυδιάστατα δεδομένα, όπως είναι τα R-δένδρα, τα οποία έχουν όμως αποδειχθεί αναποτελεσματικά ακόμη και για ερωτήσεις περιοχής σε πολυδιάστατες βάσεις δεδομένων. Τα R-δένδρα ωστόσο, αποτελούν τη βάση για δείκτες που είναι σχεδιασμένοι για πολυδιάστατες βάσεις δεδομένων. Για να λυθεί το πρόβλημα των πολλών διαστάσεων, έχει προταθεί η χρήση μεγαλύτερων παραγόντων εξερχόμενης διακλάδωσης (fanout), τεχνικές μείωσης διαστάσεων και μέθοδοι φιλτραρίσματος-διύλισης (filter-refine). Ακόμη, έχουν σχεδιαστεί ειδικά ευρετήρια για να διευκολυνθεί η επεξεργασία ερωτήσεων βάσει μετρικών. Ωστόσο, η γραμμική σάρωση παραμένει μια αποτελεσματική μέθοδος αναζήτησης ομοιοτήτων, λόγω της τάσης των σημείων των δεδομένων να ισαπέχουν από τα σημεία

ερώτησης σε πολυδιάστατο χώρο. Όμως, ενώ η γραμμική σάρωση είναι αποτελεσματική όσον αφορά την γραμμική ανάγνωση, κάθε σημείο απαιτεί δαπανηρούς υπολογισμούς απόστασης.

Στην εργασία αυτή θα παρουσιάσουμε μερικές από αυτές τις τεχνικές. Θα εστιάσουμε στη μέθοδο LSH, κάνοντας μια πλήρη αναφορά στις παραλλαγές της Rigoro us LSH και Adhoc LSH. Μέσα από τις λεπτομέρειες των συναρτήσεων κερματισμού, θα παρουσιάσουμε το δένδρο LSB και τον αλγόριθμο NN. Η γραμμική σάρωση εκτελεί μια αναζήτηση πλησιέστερων γειτόνων (ΠΓ) εξετάζοντας ολόκληρο το σύνολο, όμως, το κόστος της αυξάνεται γραμμικά με την πληθικότητα του συνόλου.

Μια καλή λύση για το πρόβλημα της αναζήτησης θα πρέπει να ικανοποιεί δύο προϋποθέσεις: 1) να μπορεί να υλοποιηθεί εύκολα σε μια σχεσιακή βάση δεδομένων, και 2) το κόστος αναζήτησης να αυξάνεται υπο-γραμμικά σε σχέση με την πληθικότητα για όλες τις κατανομές δεδομένων.

Παρά την εκτενή βιβλιογραφία που υπάρχει σχετικά με την αναζήτηση ΠΓ, δεν γνωρίζουμε να υπάρχει καμία λύση που να ικανοποιεί και τις δύο προϋποθέσεις ταυτόχρονα. Συγκεκριμένα, πολλές από τις λύσεις απαιτούν μη σχεσιακά χαρακτηριστικά, και επομένως δεν μπορούν να ενσωματωθούν σε εμπορικά συστήματα. Επίσης, υπάρχουν σχεσιακές λύσεις οι οποίες εμπειρικά φαίνεται να εκτελούνται καλά για κάποια σύνολα δεδομένων και για ορισμένες ερωτήσεις. Το μειονέκτημά τους είναι ότι, σε κάποια άλλα σύνολα δεδομένων, το κόστος αναζήτησης ίσως είναι μεγαλύτερο.

Στα επόμενα κεφάλαια γίνεται αξιολόγηση της μεθόδου δεικτοδότησης iDistan '1ce και της μεθόδου MedRank που μας οδηγούν σε τεχνικές υπολογισμού πλησιέστερων γειτόνων κατά προσέγγιση και μίας εννοιολογικά πλούσιας εναλλακτικής για εντοπισμό πλησιέστερων γειτόνων.

Έστω P ένα σύνολο σημείων δεδομένων σε μετρικό χώρο DS d-διαστάσεων, το οποίο είναι σύνολο σημείων με συνάρτηση απόστασης $dist$. Έστω $p_1 : (x_0, x_1, ..., x_{d-1})$ $p_2 : (y_0, y_1, ..., y_{d-1})$ και $p_3 : (z_0, z_1, ..., z_{d-1})$ τρία σημεία δεδομένων του DS. Η συνάρ-

τηση απόστασης dist έχει τις εξής ιδιότητες:

$$dist(p_1, p_2) = dist(p_2, p_1) \forall p_1, p_2 \in P$$
$$dist(p_1, p_1) = 0 \forall p_1 \in P$$
$$dist(p_1, p_2) > 0 \forall p_1, p_2 \in P, p_1 \neq p_2$$
$$dist(p_1, p_3) \leq dist(p_1, p_2) + dist(p_2, p_3) \forall p_1, p_2, p_3 \in P$$

Ο τελευταίος τύπος αποτελεί την τριγωνική ανισότητα, και εξασφαλίζει μία συνθήκη επιλογής υποψηφίων σημείων βάσει μετρικής σχέσης. Χωρίς βλάβη της γενικότητας, χρησιμοποιούμε την ευκλείδεια απόσταση, αν και στην iDistance μπορούν να χρησιμοποιηθούν και άλλες συναρτήσεις απόστασης. Στην ευκλείδεια απόσταση, ορίζουμε την απόσταση μεταξύ του p_1 και του p_2:

$$dist(p_1, p_2) = \sqrt{(x_0 - y_0)^2 + (x_1 - y_1)^2 + ... + (x_{d-1} - y_{d-1})^2}$$

Σημειώνεται ότι για την απόσταση δύο σημείων εκτός από τον συμβολισμό $dist$, μπορούμε να χρησιμοποιήσουμε και τον συμβολισμό $\| \|$.

Το πρόβλημα της αναζήτησης του πλησιέστερου γείτονα διατυπώνεται ως εξής: Αν δίνονται μια βάση δεδομένων D με n σημεία σε έναν μετρικό χώρο, και μια ερώτηση q στον ίδιο χώρο, να βρεθεί το σημείο (ή τα k σημεία) της D που είναι πλησιέστερα στο q. Μερικές περιοχές όπου εμφανίζεται το πρόβλημα αυτό, είναι η ανάκτηση πληροφορίας, η αναγνώριση προτύπων, η ανάλυση δεδομένων κ.ά.

Τέλος επισημαίνεται ότι το πρόβλημα αυτό είναι τόσο δημοφιλές, επειδή συχνά τα χαρακτηριστικά πραγματικών αντικειμένων να απεικονίζονται με άμεσο και φυσικό τρόπο χρησιμοποιώντας διανύσματα ενός μετρικού χώρου. Έτσι, τα προβλήματα ομοιότητας και ταξινόμησης μετατρέπονται σε προβλήματα αναζήτησης πλησιέστερου γείτονα.

Καθώς η απεικόνιση των αντικειμένων με διανύσματα είναι συχνά ευρετική, σε πολλές εφαρμογές αρκεί να βρεθεί ένα σημείο της βάσης δεδομένων το οποίο να είναι περίπου/προσεγγιστικά ο πλησιέστερος γείτονας. Προς την κατεύθυνση αυτή κινείται ο αλγόριθμος MedRank και δυο παραλλαγές του, ο αλγόριθμος Omed

Rank και ο αλγόριθμος L2TA, τα οποία παρουσιάζονται επίσης στην παρούσα εργασία.

Κεφάλαιο 2

Η μέθοδος δεικτοδότησης iDistance

2.1 Εισαγωγικά

Η iDistance είναι μία μέθοδος δεικτοδότησης για την αναζήτηση των K πλησιέστερων γειτόνων (KNN) σε πολυδιάστατο μετρικό χώρο, η οποία είναι βασισμένη σε ένα αποδοτικό B^+- δένδρο. Με την iDistance, τα δεδομένα χωρίζονται σε ομάδες με μια τεχνική διαμέρισης βάση των δεδομένων ή του χώρου, και επιλέγεται ένα σημείο αναφοράς για κάθε υποσύνολο της διαμέρισης. Τα σημεία δεδομένων σε κάθε υποσύνολο αντιστοιχίζονται σε μια μονοδιάστατη τιμή, ανάλογα με την ομοιότητά τους σε σχέση με το σημείο αναφοράς. Τα σημεία δεικτοδοτούνται με ένα B^+- δένδρο και η αναζήτηση KNN εκτελείται σε μια μονοδιάστατη περιοχή [8].

Η τεχνική iDistance για την αναζήτηση K πλησιέστερων γειτόνων μπορεί να προσαρμόζεται σε διαφορετικές κατανομές δεδομένων. Αρχικά, γίνεται διαμέριση των δεδομένων σε τμήματα και για κάθε τμήμα ορίζεται ένα σημείο αναφοράς. Κατόπιν, δεικτοδοτείται η απόσταση κάθε σημείου δεδομένων από το σημείο αναφοράς του υποσυνόλου.

Καθώς η απόσταση αυτή είναι απλό βαθμωτό μέγεθος, μπορούμε να δεικτοδοτήσουμε την απόσταση με ένα B^+- δένδρο. Έτσι, είναι εύκολο να χρησιμοποιήσουμε την τεχνική iDistance μαζί με μια υπάρχουσα εμπορική σχεσιακή βάση δεδομένων, γεγονός σημαντικό αφού τα περισσότερα εμπορικά DBMS σήμερα δεν

υποστηρίζουν άλλους δείκτες εκτός από τα B^+- δένδρα και τα R δένδρα ή κάποια από τις παραλλαγές τους. Επομένως, η iDistance είναι κατάλληλη να ενσωματωθεί στα υπάρχοντα ΣΔΒΔ [20].

Η αποτελεσματικότητα της iDistance εξαρτάται από τον τρόπο που διαμερίζονται τα δεδομένα, και από τον τρόπο που επιλέγονται τα σημεία αναφοράς. Η επιλογή των υποσυνόλων και των σημείων αναφοράς εξασφαλίζει στη μέθοδο μεγαλύτερο βαθμό ελευθερίας σε σχέση με άλλες μεθόδους. Έτσι, σε σύγκριση με δομές, όπως τη γραμμική σάρωση, το M-δένδρο και το BD-δένδρο, η iDistance υπερτερεί.

Για μία ερώτηση KNN με κέντρο το σημείο q, έχουμε μια ερώτηση περιοχής με ακτίνα r. Ο αλγόριθμος iDistance πραγματοποιεί αναζήτηση από το σημείο ερώτησης προς τα έξω, και για κάθε τμήμα που τέμνει την σφαίρα ερώτησης, έχουμε μία ερώτηση περιοχής. Αν ο αλγόριθμος βρει μέσω της αναζήτησης K στοιχεία που απέχουν από το q απόσταση μικρότερη από r, τότε ο αλγόριθμος τερματίζει. Διαφορετικά, αυξάνει την ακτίνα αναζήτησης κατά r, και η αναζήτηση συνεχίζεται στην περιοχή που δεν έχει εξεταστεί εντός των τμημάτων που τέμνουν την σφαίρα της ερώτησης. Η ανωτέρω διαδικασία επαναλαμβάνεται μέχρι να ικανοποιηθεί η συνθήκη τερματισμού [15].

2.2 Περιγραφή

Ο σχεδιασμός της μεθόδου iDistance βασίστηκε στις εξής παρατηρήσεις:

1. Η ομοιότητα ή μη μεταξύ των σημείων δεδομένων βασίζεται σε ένα επιλεγμένο σημείο αναφοράς ή σε ένα αντιπροσωπευτικό σημείο.

2. Τα σημεία δεδομένων μπορούν να ταξινομούνται βάσει των αποστάσεων τους από ένα σημείο αναφοράς.

3. Η απόσταση είναι μία μονοδιάστατη τιμή, επομένως, μπορούμε να αναπαραστήσουμε πολυδιάστατα δεδομένα σε χώρο μιας διάστασης, με αποτέλε-

σμα να μπορούμε να χρησιμοποιήσουμε υπάρχοντες μονοδιάστατα ευρετήρια, όπως τα B^+- δένδρα, τα οποία είναι αποτελεσματικά αλλά και υπάρχουν υλοποιημένα σε εμπορικά ΣΔΒΔ.

Επιπλέον, οι λανθασμένοι υποψήφιοι πλησιέστεροι γείτονες (false drops) μπορούν να φιλτραριστούν αποτελεσματικά χωρίς δαπανηρούς υπολογισμούς απόστασης.

Όπως συμβαίνει με τις μονοδιάστατες βάσεις δεδομένων, έτσι και μία πολυδιάστατη βάση δεδομένων μπορεί να χωριστεί σε υποσύνολα. Έστω ότι ένα σημείο, το οποίο συμβολίζεται με O_i, επιλέγεται ως σημείο αναφοράς για ένα τμήμα P_i. Το O_i δεν είναι απαραίτητο να είναι σημείο δεδομένων. Η απόσταση ενός σημείου δεδομένων p από το O_i είναι ίση με $dist(O_i, P)$. Με βάση την τριγωνική ανισότητα, προκύπτει ότι

$$dist(O_i, q) - dist(p, q) \leq dist(O_i, p) \leq dist(O_i, q) + dist(p, q)$$

Όταν εργαζόμαστε με ακτίνα αναζήτησης την querydist(q), θέλουμε να βρούμε όλα τα σημεία p για τα οποία ισχύει $dist(p, q) \leq querydist(q)$. Για κάθε τέτοιο σημείο p, έχουμε ότι

$$dist(O_i, q) - querydist(q) \leq dist(O_i, p) \leq dist(O_i, q) + querydist(q)$$

Δηλαδή, στο τμήμα P_i, χρειάζεται να εξετάσουμε μόνο τα υποψήφια σημεία, των οποίων η απόσταση $dist(O_i, p)$ από το σημείο αναφοράς ικανοποιεί αυτή την ανισότητα, η οποία γενικά καθορίζει έναν δακτύλιο γύρω από το σημείο αναφοράς [21].

Έστω $dist\_max_i$ η απόσταση του O_i από το πιο απομακρυσμένο σημείο στο τμήμα P_i, δηλαδή το P_i έχει ακτίνα ίση με $dist\_max_i$. Αν $dist(O_i, q) - querydist(q) \leq dist\_max_i$, τότε πρέπει να αναζητήσουμε στο P_i NN σημεία, διαφορετικά δεν λαμβάνουμε πλέον υπόψη το υποσύνολο αυτό. Το εύρος της αναζήτησης σε μονοδιάστατο χώρο είναι

$$[dist(O_i, q) - querydist(q), min(dist_max_i, dist(O_i, q)) + querydist(q))]$$

Για να γίνει αποτελεσματικότερη η αναζήτηση Κ πλησιέστερων γειτόνων βασισμένη σε μετρικές, πρέπει να εξετάσουμε δύο σημαντικά θέματα:

1. Ποιοί δείκτες μπορούν να χρησιμοποιηθούν για την αναζήτηση ομοιότητας με βάση μετρικές;

2. Πώς πρέπει να διαμεριστεί ο χώρος δεδομένων σε υποσύνολα και πιο σημείο πρέπει να επιλεχθεί ως σημείο αναφοράς ενός υποσυνόλου;

2.3 Δομή Δεδομένων

Στη μέθοδο iDistance, τα σημεία του πολυδιάστατου χώρου μετασχηματίζονται σε σημεία μονοδιάστατου χώρου. Αυτό επιτυγχάνεται με έναν αλγόριθμο τριών βημάτων.

Στο *πρώτο βήμα*, ο πολυδιάστατος χώρος δεδομένων χωρίζεται σε ένα σύνολο υποσυνόλων. Στο *δεύτερο βήμα*, ορίζεται ένα σημείο αναφοράς για κάθε υποσύνολο. Έστω ότι έχουμε m υποσύνολα, $P_0, P_1, ..., P_{m-1}$, και τα αντίστοιχα σημεία αναφοράς, $O_0, O_1, ..., O_{m-1}$. Στο *τρίτο βήμα*, όλα τα σημεία δεδομένων αναπαριστώνται σε μονοδιάστατο χώρο με τον εξής τρόπο: Για κάθε σημείο δεδομένων $p : (x_0, x_1, ..., x_{d-1}), 0 \leq x_j \leq 1, 0 \leq j \leq d-1$ υπολογίζεται ένα κλειδί δεικτοδότησης y, βασισμένο στην απόσταση από το πλησιέστερο σημείο αναφοράς O_i [1]

$$y = i \times c + dist(p, O_i)$$

όπου c μια σταθερά που χρησιμοποιείται για την επέκταση των περιοχών δεδομένων. Στην ουσία, το c χρησιμεύει για την διαμέριση του μονοδιάστατου χώρου σε περιοχές, έτσι ώστε όλα τα σημεία στο υποσύνολο P_i να απεικονιστούν στο διάστημα $[i \times c, (i+1) \times c)$. Το c πρέπει να είναι αρκετά μεγάλο έτσι ώστε να μην οδηγηθούμε σε επικάλυψη των πεδίων τιμών των κλειδιών δεικτοδότησης διαφορετικών υποσυνόλων. Θεωρητικά, θα πρέπει να είναι μεγαλύτερο από το μήκος της διαγωνίου του υπερκύβου.

Στην iDistance, χρησιμοποιούνται δύο δομές δεδομένων:

1. Ένα B^+- δένδρο για την δεικτοδότηση των μετασχηματισμένων σημείων για ταχύτερη ανάκτηση. Επιλέγουμε το B^+- δένδρο επειδή είναι αποτελεσματικό για τη δεικτοδότηση μονοδιάστατων δεδομένων και επίσης είναι διαθέσιμο στα περισσότερα εμπορικά ΣΔΒΔ. Στην εκδοχή του B^+- δένδρου που χρησιμοποιήθηκε για την αποτίμηση της αποδοτικότητας της μεθόδου iDistance, κάθε κόμβος-φύλλο συνδέεται τόσο με τον αμέσως προηγούμενο κόμβο στα αριστερά του όσο και με τον αμέσως επόμενο κόμβο, στα δεξιά του. Αυτό διευκολύνει την αναζήτηση σε γειτονικούς κόμβους όταν η περιοχή αναζήτησης επεκτείνεται σταδιακά.

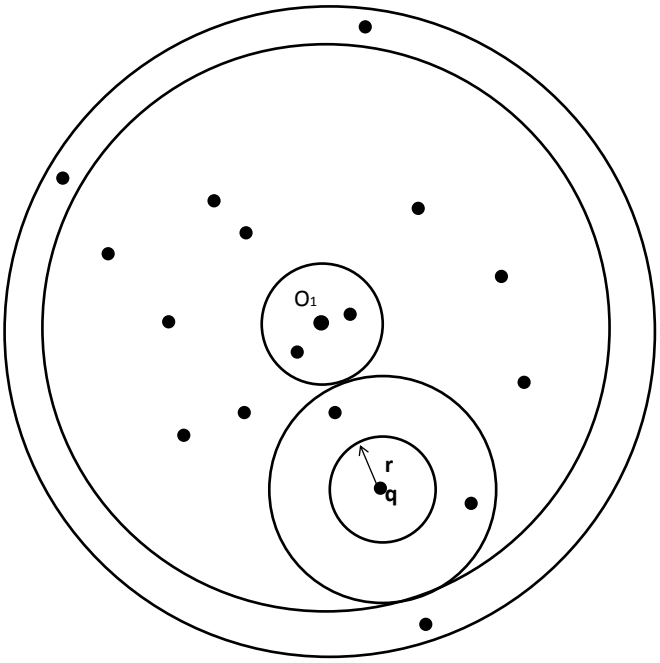
2. Έναν πίνακα πού αποθηκεύει τα m υποσύνολα της διαμέρισης και τα αντίστοιχα σημεία αναφοράς. Ο πίνακας χρησιμοποιείται για να καθοριστούν τα υποσύνολα στα οποία πρέπει να γίνει η αναζήτηση κατά την επεξεργασία των ερωτήσεων [11].

2.4 Αναζήτηση με την iDistance

Το σχήμα 2.1 δείχνει τον αλγόριθμο για την αναζήτηση KNN με τη μέθοδο iDistance. Ο αλγόριθμος ξεκινά με την αναζήτηση σε μία μικρή σφαίρα και συνεχίζει αυξάνοντας την μέχρι να βρεθούν όλοι οι K-πλησιέστεροι γείτονες. Η αναζήτηση σταματά μόλις η απόσταση του πιο απομακρυσμένου αντικειμένου του συνόλου απαντήσεων S από το σημείο ερώτησης q είναι μικρότερη ή ίση της ακτίνας r.

Αρχικά ο αλγόριθμος σαρώνει τη βοηθητική δομή πίνακα για να εντοπίσει τα σημεία αναφοράς, O_i, που τέμνουν την ερώτηση περιοχής.

Για να γίνει η αναζήτηση σε ένα υποσύνολο της διαμέρισης, πρέπει να εντοπιστεί ένα αρχικό σημείο αναζήτησης. Αν το q περιέχεται στη σφαίρα δεδομένων, χρησιμοποιείται άμεσα η τιμή iDistance του q, διαφορετικά χρησιμοποιείται η $dist_{maxi}$.



Σχήμα 2.1: Αλγόριθμος αναζήτησης KNN με την μέθοδο iDistance

Η αναζήτηση ξεκινά με μία μικρή ακτίνα. Στην βελτιστοποίησης, χρησιμοποιούμε ως αρχική ακτίνα αναζήτησης το Δ_r. Ύστερα, η ακτίνα αναζήτησης αυξάνεται κατά Δ_r, βήμα-βήμα, ώστε να σχηματιστεί μεγαλύτερη σφαίρα αναζήτησης. Για κάθε επέκταση, λαμβάνουμε υπόψη τρεις περιπτώσεις:

1. Το υποσύνολο διαμέρισης περιέχει το σημείο ερώτησης q. Σε αυτή την περίπτωση, θέλουμε να διατρέξουμε το υποσύνολο για να προσδιορίσουμε τους K-πλησιέστερους γείτονες. Πρώτα εντοπίζουμε τον κόμβο φύλλο που αποθηκεύεται το q και γίνεται αναζήτηση προς την κατεύθυνση του σημείου αναφοράς ή προς την αντίθετή της, ανάλογα με τις παραμέτρους της ερώτησης.

2. Το σημείο ερώτησης q βρίσκεται έξω από το υποσύνολο διαμέρισης αλλά η σφαίρα της ερώτησης τέμνει το υποσύνολο. Σε αυτή την περίπτωση, χρειάζεται να γίνει αναζήτηση μόνο προς την κατεύθυνση του σημείου αναφοράς.

3. Το υποσύνολο διαμέρισης δεν τέμνει τη σφαίρα της ερώτησης. Σε αυτή την περίπτωση, δεν χρειάζεται να εξετάσουμε το συγκεκριμένο υποσύνολο.

Η αναζήτηση σταματά όταν οι K πλησιέστεροι γείτονες έχουν βρεθεί στα υποσύνολα διαμέρισης που τέμνουν τη σφαίρα ερώτησης καθώς και όταν περαιτέρω επέκταση δεν αλλάζει τη λίστα με τους K γείτονες. Με άλλα λόγια, η αναζήτηση τερματίζεται όταν όλα τα σημεία που βρίσκονται έξω από τα υποσύνολα που τέμνουν την σφαίρα ερώτησης είναι βέβαιο ότι θα απέχουν απόσταση D από το σημείο ερώτησης τέτοια ώστε $D > querydist$. Αυτό συμβαίνει μετά από έναν αριθμό επαναλήψεων, όταν η απόσταση του πιο απομακρυσμένου σημείου στο σύνολο απαντήσεων S από το σημείο ερώτησης q είναι μικρότερη ή ίση από την ακτίνα αναζήτησης r. Τότε, όλα τα σημεία που βρίσκονται έξω από την σφαίρα ερώτησης απέχουν απόσταση μεγαλύτερη από $querydist$, ενώ όλα τα υποψήφια σημεία στο σύνολο απαντήσεων S απέχουν απόσταση μικρότερη από την $querydist$. Συνακόλουθα, περαιτέρω επέκταση της σφαίρας ερώτησης δεν θα αλλάξει το σύνολο των απαντήσεων. Από τη μελέτη του ανωτέρω αλγορίθμου, προκύπτει ότι οι απαντήσεις που επιστρέφει η iDistance είναι πάντοτε ορθές [15].

2.5 Διαμέριση του χώρου δεδομένων και επιλογή των σημείων αναφοράς

Ο διαμερισμός των δεδομένων μπορεί να πραγματοποιηθεί με τους εξής τρόπους:

1) Διαμερισμός με βάση τον χώρο

Μια άμεση προσέγγιση για τον διαμερισμό του χώρου δεδομένων είναι η διαίρεση του χώρου σε ίσα μέρη. Σε έναν d-διάστατο χώρο, έχουμε $2d$ υπερεπίπεδα. Χωρίζουμε τον χώρο σε $2d$ πυραμίδες που έχουν ως κορυφή το κέντρο του κύβου, ενώ κάθε υπερεπίπεδο αποτελεί τη βάση κάθε πυραμίδας. Στη μέθοδο iDistance μελετώνται οι κάτωθι στρατηγικές επιλογής σημείων αναφοράς και διαμερισμού:

(α) Κέντρο του υπερεπίπεδου, Κοντινότερη απόσταση: Ως σημείο αναφοράς επιλέγεται το κέντρο κάθε υπερεπίπεδου, και το αντίστοιχο υποσύνολο περιέχει

τα πλησιέστερα σημεία στο σημείο αναφοράς.

(β) Κέντρο του υπερεπίπεδου, Μεγαλύτερη απόσταση: Ως σημείο αναφοράς επιλέγεται το κέντρο κάθε υπερεπίπεδου, και το αντίστοιχο υποσύνολο περιέχει τα πιο απομακρυσμένα σημεία από το σημείο αναφοράς.

(γ) Εξωτερικό σημείο: Κάθε σημείο που βρίσκεται στην ευθεία που διέρχεται από το κέντρο του υπερεπιπέδου και από το κέντρο του αντίστοιχου χώρου δεδομένων μπορεί επίσης να χρησιμοποιηθεί ως σημείο αναφοράς [19].

2) Διαμέριση με βάση τα δεδομένα

Η διαμέριση σε υποσύνολα ίσου όγκου είναι ελκυστική για ομοιόμορφα κατανεμημένα δεδομένα, όμως, στην πράξη συχνά τα δεδομένα είναι σε συστάδες ή συσχετισμένα. Ακόμη και αν δεν υπάρχει συσχέτιση σε όλες τις διαστάσεις, υπάρχουν συνήθως υποσύνολα των δεδομένων που είναι συσχετισμένα τοπικά. Σε αυτές τις περιπτώσεις, είναι αναγκαία μία πιο κατάλληλη μέθοδος διαμερισμού για να ομαδοποιήσουμε τα δεδομένα. Στη μέθοδο iDistance χρησιμοποιείται ο αλγόριθμος ομαδοποίησης k-means.

Ο αλγόριθμος k-means (k-μέσων)είναι ένας αλγόριθμος που ομαδοποιεί αντικείμενα σε k ομάδες, βάσει των χαρακτηριστικών των k διαμερίσεων. Ο αλγόριθμος υποθέτει ότι τα χαρακτηριστικά του αντικειμένου δημιουργούν ένα διανυσματικό χώρο και ο σκοπός του είναι να ελαχιστοποιήσει τη συνολική απόκλιση εντός της κάθε ομάδας ή τη συνάρτηση τετραγωνικού σφάλματος. Τα βασικά βήματα του αλγόριθμου είναι τα εξής:

1. Επιλογή του αριθμού των ομάδων.
2. Διάλεξε k τυχαία βαρύκεντρα (centroids).
3. Ανάθεσε κάθε αντικείμενο στο πλησιέστερο προς αυτό βαρύκεντρο.
4. Για κάθε μία από τις k ομάδες, υπολόγισε το νέο βαρύκεντρο.

5. Αν στο βήμα 4 δεν υπήρξε μεταβολή, ο αλγόριθμος τερματίζει, αλλιώς επαναλαμβάνουμε από το βήμα 3.

Ο αλγόριθμος ξεκινά διαχωρίζοντας τα αρχικά σημεία σε k αρχικά σύνολα είτε τυχαία είτε χρησιμοποιώντας ευρετική προσέγγιση. Στη συνέχεια υπολογίζει το βαρύκεντρο του κάθε συνόλου, υλοποιεί νέο διαχωρισμό ώστε το κάθε σημείο να σχετίζεται με το κοντινότερο βαρύκεντρο. Έπειτα τα βαρύκεντρο ξανά υπολογίζονται για τις νέες ομάδες, ο αλγόριθμος επαναλαμβάνει τα δυο βήματα ωσότου τα σημεία δεν μπορούν να αλλάξουν ομάδες (ή εναλλακτικά τα βαρύκεντρa παραμένουν αμετάβλητα).

Ο αλγόριθμος αυτός παραμένει διάσημος επειδή συγκλίνει πολύ γρήγορα. Όσον αφορά την απόδοση, ο αλγόριθμος δεν εγγυάται ότι παράξει βέλτιστη λύση. Η ποιότητα της τελική λύσης εξαρτάται πολύ από το αρχικό σύνολο ομάδων και μπορεί να είναι πολύ χαμηλότερη από το ολικό βέλτιστο.

Επίσης ένα άλλο μειονέκτημα του αλγόριθμου είναι ότι ο αριθμός των ομάδων πρέπει να οριστεί εξαρχής. Το πλήθος των ομάδων επηρεάζει την περιοχή αναζήτησης και το πλήθος των διαδρομών από τη ρίζα στους κόμβους-φύλλα. Αφού δημιουργηθούν οι ομάδες, πρέπει να επιλέξουμε σημείο αναφοράς, το οποίο μπορεί να είναι α) το βαρύκεντρο της ομάδας, β) οι κορυφές.

Η επιλογή του τρόπου διαμέρισης είναι σημαντική ώστε να αποφεύγονται ή να μειώνονται όσο το δυνατόν περισσότερο οι επικαλύψεις των σφαιρών και να μην έχουμε πολλά σημεία με την ίδια απόσταση (ομοιότητα) από το σημείο αναφοράς.

2.6 Πειράματα και συμπεράσματα

Η τεχνική iDistance διαχωρίζει τα δεδομένα και επιλέγει ένα σημείο αναφοράς. Τα δεδομένα σε κάθε ομάδα μπορούν να περιγραφούν βασιζόμενα στην ομοιότητα τους, σε σχέση με ένα σημείο αναφοράς και για αυτό τον λόγο μπορούν να μετασχηματισθούν σε έναν μονοδιάστατο χώρο, με βάση αυτή την ομοιότητα. Αυτό μας επιτρέπει να ταξινομήσουμε τα σημεία δεδομένων χρησιμοποιώντας ένα B^+-

δένδρο, και εκτελώντας μια αναζήτηση KNN, χρησιμοποιώντας μια απλή μονοδιάστατη αναζήτηση περιοχής.

Ο αριθμός των σημείων αναφοράς είναι μια σημαντική παράμετρος ρύθμισης για τη μέθοδο iDistance. Γενικά, όσο μεγαλύτερος ο αριθμός των σημείων αναφοράς, τόσο καλύτερη η απόδοση, και παράλληλα, τόσο περισσότερος χρόνος χρειάζεται για να προσδιοριστούν τα σημεία αναφοράς των ομάδων.

Η χρήση υπερβολικού αριθμού σημείων αναφοράς υποβαθμίζει την απόδοση, εισάγοντας υψηλότερο υπολογιστικό φόρτο και για αυτό χρησιμοποιείται ένας μετριοπαθής αριθμός. Στα πειράματα αξιολόγησης της μεθόδου, έχει χρησιμοποιηθεί το 64 ως αριθμός των σημείων αναφοράς στα περισσότερα από τα πειράματα μας, και η μέθοδος iDistance αποδίδει καλύτερα από την ακολουθιακή σάρωση καθώς και άλλες τεχνικές δεικτοδότησης.

Συνήθως, η μέθοδος iDistance εκτελείται από 2 έως 6 φορές ταχύτερα από τις άλλες τεχνικές.

Μπορούμε να χρησιμοποιήσουμε ένα μοντέλο κόστους βασισμένο σε ιστόγραμμα στη βελτιστοποίηση των ερωτήσεων για να υπολογίσουμε το κόστος της μεθόδου iDistance σε πλήθος προσβάσεων σε σελίδες, και ο υπολογισμός αυτός έχει ένα σχετικά περιθώριο λάθους κάτω του 20%.

Τα πειραματικά αποτελέσματα έδειξαν ότι η μέθοδος iDistance υπερείχε σε απόδοση των άλλων τεχνικών στις περισσότερες των περιπτώσεων. Επιπλέον, η μέθοδος iDistance μπορεί να ενσωματωθεί σε υπάρχοντα συστήματα βάσεων δεδομένων χωρίς μεγάλο κόστος. Συνολικά, ότι η μέθοδος iDistance αποτιμάται ως μια πρακτική και αποτελεσματική μέθοδος στην αναζήτηση N κοντινότερων γειτόνων.

Κεφάλαιο 3

Η μέθοδος δεικτοδότησης LSH

3.1 Εισαγωγή

Η αναζήτηση πλησιέστερου γείτονα σε πολυδιάστατο χώρο είναι ένα σημαντικό πρόβλημα με πολλές εφαρμογές. Μια καλή λύση, όσον αφορά τη βάση δεδομένων, πρέπει να έχει δύο ιδιότητες: να μπορεί να ενσωματωθεί εύκολα σε μια σχεσιακή βάση δεδομένων και το κόστος αναζήτησης θα πρέπει να αυξάνεται υπογραμμικά σε σχέση με το μέγεθος του συνόλου δεδομένων, ανεξάρτητα από τις κατανομές δεδομένων και ερώτησης.

Η μέθοδος κερματισμού με ευαισθησία τοπικότητας (Locality Sensitive Hashing-LSH) είναι μια γνωστή μέθοδος που ικανοποιεί και τις δύο προϋποθέσεις, όμως, οι τρέχουσες υλοποιήσεις της είτε έχουν υψηλό κόστος χώρου (απαιτούν δηλαδή αυξημένο αποθηκευτικό χώρο) και ερώτησης, είτε δεν διατηρούν τη θεωρητική της εγγύηση ποιότητας αποτελεσμάτων.

Η μέθοδος LSH υποστηρίζει την αναζήτηση ΠΓ με προσέγγιση c. Ένα σημείο o είναι ΠΓ του q με προσέγγιση c αν η μεταξύ τους απόσταση είναι το πολύ c φορές μεγαλύτερη από την απόσταση του q και του πραγματικού του ΠΓ o^*, δηλαδή $\| oq \| \leq c \| o^*q \|$, όπου $c \geq 1$ είναι ο λόγος προσέγγισης. Η LSH αρχικά είχε προταθεί ως θεωρητική μέθοδος με ελκυστική απόδοση, τόσο σε σχέση με τον χώρο όσο και σε σχέση με τις επιδόσεις [16].

Στην πράξη, η μέθοδος μπορεί να είναι είτε αυστηρή (rigorous) είτε κατά περιπτωση (adhoc). Συγκεκριμένα, η rigorous-LSH εξασφαλίζει καλή ποιότητα αποτελεσμάτων (π.χ. μικρό λόγο προσέγγισης) αλλά έχει μεγάλο κόστος. Αν και η adhoc-LSH είναι πιο αποδοτική, υποβαθμίζει την ποιότητα των αποτελεσμάτων (π.χ. ο γείτονας που επιστρέφει μπορεί να είναι αυθαίρετα μακρυά). Με άλλα λόγια, καμία υλοποίηση της LSH δεν μπορεί να εξασφαλίσει ταυτόχρονα και την ποιότητα και την απόδοση, επομένως μειώνεται σημαντικά η δυνατότητα εφαρμογής της.

Για τον λόγο αυτό, έχει προταθεί μια μέθοδος προσπέλασης που καλείται LSB-δένδρο (locality sensitive B-tree) η οποία επιτρέπει γρήγορη αναζήτηση ΠΓ σε πολλές διαστάσεις με καλή ποιότητα. Ο συνδυασμός αρκετών LSB-δέντρων οδηγεί σε μια δομή, το δάσος-LSB, η οποία συνδυάζει τα πλεονεκτήματα των rigorous και του adhoc-LSH, χωρίς τα μειονεκτήματά τους. Συγκεκριμένα, το δάσος-LSB έχει τα εξής χαρακτηριστικά: Πρώτον, έχει την ίδια κατανάλωση χώρου με την adhoc-LSH, και σημαντικά μικρότερη από τη rigorous-LSH. Δεύτερον, διατηρεί την εγγύηση προσέγγισης της rigorous-LSH. Τρίτον, το κόστος ερώτησης είναι ουσιαστικά μικρότερο από την adhoc-LSH, και, συνεπώς, υπο-γραμμικό σε σχέση με το μέγεθος του συνόλου δεδομένων. Τέλος, το δάσος-LSB χρησιμοποιεί καθαρά σχεσιακή τεχνολογία, και επομένως, μπορεί εύκολα να ενσωματωθεί σε ένα εμπορικό σύστημα[7].

Όλες οι υλοποιήσεις της LSH απαιτούν να γίνει αντιγραφή της βάσης δεδομένων πολλές φορές και αυτό συνεπάγεται μεγάλη κατανάλωση χώρου και μεγάλο κόστος ενημερώσεων (update overhead). Πολλές εφαρμογές προτιμούν ένα ευρετήριο που να καταναλώνει γραμμικό χώρο, και να υποστηρίζει αποδοτικά εισαγωγές και διαγραφές.

Το δένδρο-LSB ικανοποιεί όλες αυτές τις προϋποθέσεις, αφού αποθηκεύει κάθε σημείο δεδομένων μία φορά σε ένα συμβατικό B-δένδρο. Έχουν διενεργηθεί συγκρίσεις μεταξύ του δένδρου-LSB και της iDistance, της MedRank, τόσο για ακριβείς όσο και για προσεγγιστικές ερωτήσεις πλησιεστέρων γειρόνων. Το δένδρο-

LSB ξεπερνά σε ό,τι αφορά τις επιδόσεις την iDistance κατά δύο τάξεις μεγέθους, επιβεβαιώνοντας την υπεροχή της προσεγγιστικής ανάκτησης. Σε σχέση με τη MedRank, η τεχνική του LSB-tree υπερέχει τόσο σε χρονικές επιδόσεις (περίπου μία τάξη μεγέθους), όσο και σε ποιότητα αποτελεσμάτων.

Η μέθοδος locality sensitive B-tree (LSB-tree) αποτελεί μία επέκταση της LSH, προκειμένου να επιτευχθεί γρήγορη και ακριβής πολυδιάστατη αναζήτηση πλησιέστερου γείτονα, που να μπορεί να ενσωματωθεί σε σχεσιακές βάσεις δεδομένων. Ο συνδυασμός αρκετών δέντρων-LSB δημιουργεί ένα δάσος-LSB το οποίο παρέχει ισχυρές εγγυήσεις ποιότητας. Στην πράξη, το δένδρο-LSB είναι αποδοτικός δείκτης, που καταναλώνει γραμμικό χώρο, υποστηρίζει αποδοτικές ενημερώσεις, και παρέχει ακριβή αποτελέσματα. Συγκριτικά, το δένδρο-LSB έχει αποδειχθεί ταχύτερο από (1) την iDistance κατά δύο τάξεις μεγέθους, και (2) από την MedRank κατά μία τάξη μεγέθους, επιστρέφοντας μάλιστα πολύ καλύτερα αποτελέσματα [12].

Θεωρούμε ότι το σύνολο δεδομένων D βρίσκεται σε εξωτερική μνήμη (π.χ. δίσκο) όπου κάθε σελίδα έχει B λέξεις. Επίσης, κάθε ακέραιος ή πραγματικός αριθμός αναπαριστάται με μία λέξη. Εφόσον κάθε σημείο έχει d συντεταγμένες, ολόκληρο το σύνολο D χρειάζεται συνολικά $d \times n/B$ σελίδες, όπου n η πληθικότητα του D. Με άλλα λόγια, όλοι οι αλγόριθμοι, που δεν έχουν αύξηση κόστους υπογραμμικά σε σχέση με το n, έχουν πολυπλοκότητα εισόδου-εξόδου (dn/B). Στόχος μας είναι να κατασκευάσουμε μια σχεσιακή λύση με μικρότερη πολυπλοκότητα.

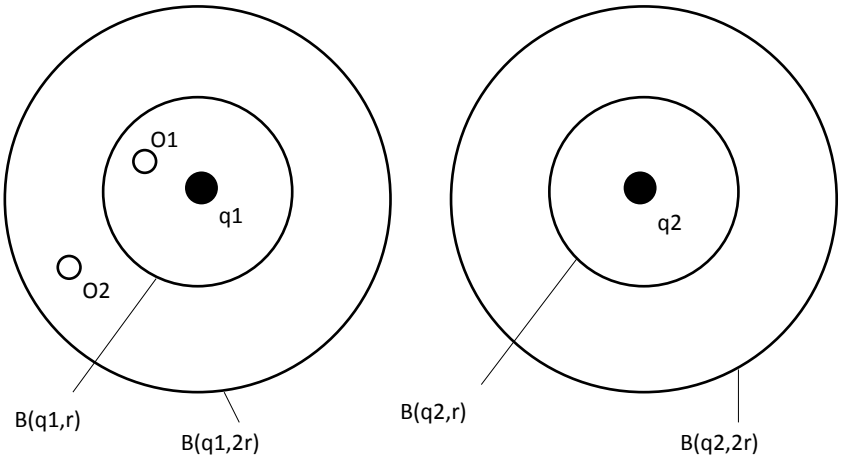
Συμβολίζουμε με M την διαθέσιμη μνήμη, με μονάδα μέτρησης τη λέξη. Αν δεν δηλώνεται ρητώς διαφορετικά, το M μπορεί να είναι ίσο με $3B$ για να μπορεί να εκτελεστεί η προτεινόμενη μέθοδος. Ωστόσο, δεν συμπεριλαμβάνεται η μνήμη που απαιτείται για την αποθήκευση των αποτελεσμάτων. Συγκεκριμένα, ένα σύνολο αποτελεσμάτων της kNN ή της kCP απαιτεί $O(kd)$ επιπλέον λέξεις στη μνήμη, κάτι που υποθέτουμε ότι είναι εφικτό.

3.2 Παραδοχές Προβλήματος

Χωρίς βλάβη της γενικότητας, υποθέτουμε ότι όλες οι διαστάσεις έχουν πεδίο τιμών το διάστημα $[0, t]$, όπου t ακέραιος. Σύμφωνα με τη βιβλιογραφία σχετικά με την LSH, για την ανάλυση της ποιότητας των αποτελεσμάτων, θεωρούμε όλες τις συντεταγμένες ακέραιους αριθμούς, έτσι ώστε να μπορούμε να θέσουμε το 1 ως κάτω φράγμα για την απόσταση μεταξύ δύο διαφορετικών σημείων. Πράγματι, με εφαρμογή κατάλληλης κλιμάκωσης (scaling), μπορούμε να μετατρέψουμε τους πραγματικούς αριθμούς σε ακέραιους στις περισσότερες εφαρμογές. Σε κάθε περίπτωση, η υπόθεση αυτή είναι αναγκαία μόνο σε θεωρητική ανάλυση. Η δομή που παρουσιάζεται στη συνέχεια και οι σχετικοί αλγόριθμοι δεν βασίζονται στην υπόθεση αυτή.

Θεωρούμε ότι οι αποστάσεις μετρώνται με την νόρμα l_p. Καθώς η νόρμα l_p γενικεύει ή προσεγγίζει και άλλες μετρικές, η τεχνική μας είναι άμεσα εφαρμόσιμη και σε αυτές τις μετρικές. Για παράδειγμα, στην περίπτωση που όλες οι διαστάσεις είναι δυαδικές (δηλ. δέχονται μόνο δυαδικές τιμές), η νόρμα l_1 είναι ακριβώς η απόσταση Hamming, η οποία χρησιμοποιείται ευρέως στην ανάκτηση κειμένου, στις χρονοσειρές κ.λπ. Επομένως, η τεχνική μας μπορεί να εφαρμοστεί και σε αυτές τις εφαρμογές [3].

Θα εξετάσουμε το πρόβλημα της αναζήτησης ΠΓ με προσέγγιση c, όπου c θετικός ακέραιος: δεδομένου ενός σημείου q, η ερώτηση επιστρέφει το σημεία o του συνόλου D, τέτοιο ώστε η απόσταση $\parallel o, q \parallel$ μεταξύ του o και του q να είναι το πολύ c φορές μεγαλύτερη από την απόσταση μεταξύ του q και του πραγματικού ΠΓ του q, ο οποίος είναι ο o^*. Υποθέτουμε ότι το q δεν ανήκει στο D, διαφορετικά, το πρόβλημα του πλησιέστερου γείτονα γίνεται μια απλή αναζήτηση, η οποία λύνεται εύκολα με έναν τυπικό κερματισμό. Άμεση επέκταση της αναζήτησης ΠΓ είναι η kNN αναζήτηση, η οποία βρίσκει τα k σημεία του D που βρίσκονται πλησιέστερα στο q. Η εκδοχή της kNN με προσέγγιση c έχει στόχο να επιστρέφει k σημεία, από τα οποία το i-οστό σημείο $(1 \leq i \leq k)$ είναι μια c-προσέγγιση του πραγματι-



Σχήμα 3.1: Απεικόνιση ερωτήσεων κάλυψης σφαίρας

κού i-οστού πλησιέστερου γείτονα. Έστω $o_1^*, o_2^*, ..., o_k^*$ οι πραγματικοί k πλησιέστεροι γείτονες με αύξουσα σειρά των αποστάσεων τους από το q. Τότε, ένα σύνολο σημείων $o_1, o_2, ..., o_k$ (σε αύξουσα σειρά) αποτελεί λύση με προσέγγιση c αν ισχύει $\parallel o_i, q \parallel \leq c \times \parallel o_i^*, q \parallel$ για κάθε $i \in [1, k]$.

Σύμφωνα με τη θεωρητική ανάλυση, ένα σημείο μπορεί να χωρέσει σε σταθερό αριθμό σελίδων δίσκου (δηλ. $d = O(B)$), κάτι που ισχύει σχεδόν πάντα στην πραγματικότητα. Για παράδειγμα, μπορούμε να θέσουμε τη σταθερά ίση με 10, και έτσι υποστηρίζονται έτσι διαστάσεις μέχρι και $10B$. Επίσης, υποθέτουμε ότι το πλήθος των διαστάσεων d είναι τουλάχιστον ίσο με $log(n/B)$ (όλοι οι λογάριθμοι στο κεφάλαιο αυτό έχουν βάση 2, εκτός αν δηλωθεί διαφορετικά). Αυτό είναι λογικό, αφού, για πρακτικές τιμές του n και του B, ο $log(n/B)$ σπάνια υπερβαίνει το 20, ενώ το $d = 20$ θεωρείται οριακά ως «πολυδιάστατο».

3.3 Rigorous-LSH και κάλυψη σφαίρας

Οι λύσεις που παρουσιάζονται στη συνέχεια είναι βασισμένες πάνω στην LSH. Θα δούμε τα μειονεκτήματα των υλοποιήσεων της LSH που ήδη υπάρχουν και θα παρουσιάσουμε τις τεχνικές λεπτομέρειες της LSH που χρειαζόμαστε.

Η LSH δεν επιλύει άμεσα αναζητήσεις ΠΓ με προσέγγιση c. Στην πραγματικότητα, είναι σχεδιασμένη για ένα διαφορετικό πρόβλημα, που λέγεται κάλυψη σφαίρας (ΚΣ). Έστω D ένα σύνολο σημείων σε d-διάστατο χώρο. Συμβολίζουμε με $B(q, r)$ μια σφαίρα με κέντρο το σημείο ερώτησης q και ακτίνα r. Μια ερώτηση ΚΣ με προσέγγιση c επιστρέφει το εξής αποτέλεσμα:

1) Αν η $B(q, r)$ καλύπτει τουλάχιστον ένα σημείο στο D, επιστρέφει ένα σημείο το οποίο απέχει από το q το πολύ cr.

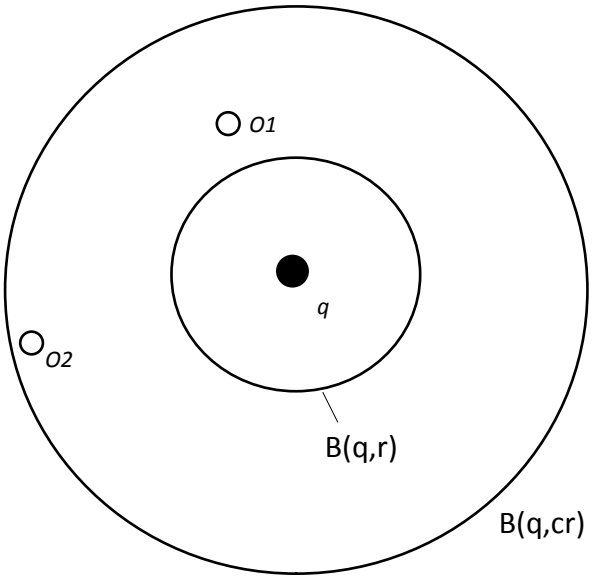
2) Αν η $B(q, cr)$ δεν καλύπτει κανένα σημείο στο D, δεν επιστρέφει τίποτα.

3) Διαφορετικά, το αποτέλεσμα δεν ορίζεται.

Το σχήμα 3.1 δείχνει ένα παράδειγμα όπου το D έχει δύο σημεία o_1 και o_2. Έχουμε δύο ερωτήσεις ΚΣ με προσέγγιση 2, έστω q_1 και q_2. Ας δούμε πρώτα την q_1. Οι δύο κύκλοι με κέντρο q_1 αναπαριστούν τις σφαίρες $B(q_1, r)$ και $B(q_1, 2r)$ αντίστοιχα. Αφού η $B(q_1, r)$ καλύπτει ένα σημείο o_1, η ερώτηση θα πρέπει να επιστρέψει ένα σημείο, το οποίο μπορεί να είναι είτε το o_1 είτε το o_2, αφού και τα δύο βρίσκονται μέσα στην $B(q_1, 2r)$. Ας δούμε τώρα την ερώτηση q_2. Η $B(q_1, 2r)$ δεν καλύπτει κανένα σημείο, άρα η ερώτηση δεν επιστρέφει τίποτα.

Είναι ενδιαφέρον το γεγονός ότι μια ερώτηση ΠΓ με προσέγγιση μπορεί να αναχθεί σε ερωτήσεις ΚΣ με προσέγγιση με διαφορετικές ακτίνες. Η λογική είναι η εξής: Αν η σφαίρα $B(q, r)$ είναι κενή αλλά η $B(q, cr)$ όχι, τότε κάθε σημείο που βρίσκεται μέσα στη $B(q, cr)$ είναι ΠΓ του q με προσέγγιση c. Ας δούμε την ερώτηση q στο σχήμα 3.2. Η σφαίρα $B(q, r)$ είναι κενή, όμως η $B(q, cr)$ δεν είναι. Ο πλησιέστερος γείτονας του q πρέπει να απέχει από το q απόσταση μεταξύ r και cr. Άρα, κάθε σημείο της $B(q, cr)$ (π.χ. το o_1 ή το o_2) είναι ΠΓ του q με προσέγγιση c [13].

Βασισμένοι σε αυτή την ιδέα, οι Indyk και Motwani [14] προτείνουν μια δομή



Σχήμα 3.2: Η λογική της μείωσης

που υποστηρίζει ερωτήσεις ΚΣ με προσέγγιση c με $r = 1, c, c^2, c^3, ..., x$ αντίστοιχα, όπου x είναι η μικρότερη δύναμη του c που είναι μεγαλύτερη ή ίση με $t \times d$ (υπενθυμίζεται ότι t είναι η μεγαλύτερη τιμή συντεταγμένη κάθε διάστασης). Δίνουν έναν αλγόριθμο που εγγυάται λόγο προσέγγισης c^2 για την αναζήτηση ΠΓ (δηλαδή, για ερωτήσεις ΠΓ με προσέγγιση c χρειαζόμαστε μια δομή για ερωτήσεις ΚΣ με προσέγγιση $\sqrt{c}$). Η μέθοδος τους, την οποία ονομάζουμε rigorous-LSH, καταναλώνει χώρο $O((log_c t + log_c d) * (dn/B)^{1+1/c})$ και απαντά μια ερώτηση σε χρόνο $O((log_c t + log_c d) * (dn/B)^(1/c)$. Ας σημειώσουμε ότι το t μπορεί να είναι μεγάλο, και συνεπώς το κόστος χώρου και χρόνου μπορεί να γίνει πολύ μεγάλο. Το δένδρο-LSB απαλείφει πλήρως τον παράγοντα $log_c t + log_c d$.

Τέλος, αξίζει να αναφέρουμε ότι υπάρχουν πολύπλοκες αναγωγές ΠΓ σε ΚΣ με καλύτερες πολυπλοκότητες. Ωστόσο, οι αναγωγές αυτές είναι πολύ θεωρητικές και είναι δύσκολο να υλοποιηθούν σε σχεσιακές βάσεις δεδομένων.

3.4 Adhoc-LSH

Αν και η rigorous-LSH είναι θεωρητικά ασφαλής, το κόστος είναι υπερβολικά μεγάλο στην πράξη. Η αιτία του προβλήματος είναι ότι πρέπει να υποστηρίζει ερωτήσεις ΚΣ για πάρα πολλές ακτίνες. Αυτό το μειονέκτημα διορθώνεται με μια ευρετική προσέγγιση, την οποία ονομάζουμε adhoc-LSH. Δεδομένης μιας ερώτησης ΠΓ q, επιστρέφεται άμεσα το αποτέλεσμα της ερώτησης ΚΣ που είναι στη θέση q και έχει ακτίνα r_m, όπου r_m είναι μια «μαγική» ακτίνα που προκαθορίζεται από το σύστημα. Καθώς πρέπει να υποστηριχθεί μόνο μία ακτίνα, η adhoc-LSH βελτιώνει την rigorous-LSH και απαιτεί χώρο μόνο $O((dn/B)^{1+1/c})$ και χρόνο $O((dn/B)^{1/c})$ χρόνο. Δυστυχώς, η ωφέλεια σε χρόνο έρχεται με σημαντική απώλεια στην ποιότητα των αποτελεσμάτων.

Η adhoc-LSH εκτελείται καλά αν η r_m είναι περίπου ίση με την απόσταση μεταξύ του q και του πραγματικού του πλησιέστερου γείτονα, και γι' αυτό η adhoc-LSH μπορεί να είναι αποδοτική όταν δίνεται η σωστή r_m. Δυστυχώς, το να βρεθεί μια τέτοια ακτίνα είναι πολύ δύσκολο. Ακόμη χειρότερα, μια τέτοια ακτίνα μπορεί να μην υπάρχει καθόλου επειδή μια ακτίνα r_m μπορεί να είναι καλή για κάποιες ερωτήσεις αλλά να μην είναι καλή για κάποιες άλλες [10].

Συνοπτικά, αν θέλουμε να εφαρμόσουμε την LSH, αντιμετωπίζουμε το δίλημμα ανάμεσα στην αποδοτικότητα χώρου/χρόνου και στην εγγύηση της ποιότητας προσέγγισης. Αν είναι σημαντική η ποιότητα του γείτονα που ανακτάμε (π.χ. σε συστήματα ασφαλείας όπως η αναγνώριση δακτυλικών αποτυπωμάτων), χρειάζεται τεράστιος χώρος, και το κόστος ερώτησης είναι πολύ μεγάλο. Από την άλλη πλευρά, αν απαιτούμε μικρό χώρο και χρόνο, θα πρέπει να θυσιάσουμε την εγγύηση ποιότητας της LSH, η οποία κατά κάποιο τρόπο είναι αυτό ακριβώς που αποτελεί την κύρια δύναμη της LSH.

3.5 Συναρτήσεις κερματισμού

Έστω $h(o)$ μια συνάρτηση κερματισμού που αντιστοιχίζει ένα d-διάστατο σημείο o σε μια μονοδιάστατη τιμή. Η συνάρτηση διαθέτει ευαισθησία τοπικότητας αν η πιθανότητα αντιστοίχισης δύο σημείων o_1, o_2 στην ίδια τιμή αυξάνεται όσο η απόσταση τους $\parallel o_1, o_2 \parallel$ ελαττώνεται.

ΟΡΙΣΜΟΣ 1 (LSH). Δίνονται απόσταση r, λόγος προσέγγισης c, πιθανότητες p_1 και p_2 έτσι ώστε $p_1 > p_2$. Μια συνάρτηση κερματισμού $h(.)$ διαθέτει ευαισθησία τοπικότητας σε σχέση με τα (r, cr, p_1, p_2) αν ικανοποιεί τις συνθήκες:

1) Αν $\parallel o_1, o_2 \parallel \leq r$, τότε $P[h(o_1) = h(o_2)] \geq p_1$.

2) Αν $\parallel o_1, o_2 \parallel > cr$, τότε $P[h(o_1) = h(o_2)] \leq p_2$.

Γνωρίζουμε συναρτήσεις LSH για πολλές μετρικές. Για τη νόρμα l_p, μια πολύ γνωστή συνάρτηση LSH είναι η εξής:

$$h(o) = \lfloor \frac{\overrightarrow{a} * \overrightarrow{o} + b}{w} \rfloor$$

όπου το $\overrightarrow{o}$ είναι το d-διάστατο διάνυσμα αναπαράστασης του σημείου o, το $\overrightarrow{a}$ είναι ένα d-διάστατο διάνυσμα του οποίου οι συνιστώσες επιλέγονται ανεξάρτητα από μια κατανομή p-stable, το $\overrightarrow{a}$*$\overrightarrow{o}$ είναι το εσωτερικό γινόμενο των δύο διανυσμάτων, w είναι μια αρκετά μεγάλη σταθερά και το b επιλέγεται ομοιόμορφα από το διάστημα $[0, w)$.

Η ανωτέρω εξίσωση έχει μία απλή γεωμετρική ερμηνεία. Έστω $p = 2$, οπότε το l_p είναι η Ευκλείδεια απόσταση. Σε αυτή την περίπτωση, μια κατανομή 2-stable μπορεί να είναι η κανονική κατανομή (μέση τιμή 0, διασπορά 1), και αρκεί να θέσουμε $w = 4$. Θεωρώντας $d = 2$, το σχήμα 3,3 απεικονίζει την ευθεία που τέμνει την αρχή των αξόνων και η κλίση της συμπίπτει με την κατεύθυνση του $\overrightarrow{a}$. Για ευκολία, υποθέτουμε ότι το $\overrightarrow{a}$ είναι μοναδιαίο, ώστε το εσωτερικό γινόμενο $\overrightarrow{a}$*$\overrightarrow{o}$ να είναι η προβολή του σημείου o στο $\overrightarrow{a}$, δηλαδή το σημείο A. Το $\overrightarrow{a} * \overrightarrow{o} + b$ σημαίνει ότι μεταφέρεται το σημείο A κατά απόσταση b, στο σημείο B. Τέλος, αν χωρίσουμε την ευθεία σε διαστήματα μήκους w, τότε η τιμή $h(o)$ είναι ο αριθμός του διαστήματος στο οποίο βρίσκεται το B [10].

Η ιδέα πίσω από μια τέτοια συνάρτηση κερματισμού είναι ότι, αν δύο σημεία είναι κοντά το ένα στο άλλο, τότε είναι μεγάλη η πιθανότητα οι προβολές τους να ανήκουν στο ίδιο διάστημα, ενώ, δύο απομακρυσμένα σημεία είναι πολύ πιθανό να προβληθούν σε διαφορετικά διαστήματα, σύμφωνα με το ακόλουθο λήμμα:

Λήμμα 3.5.1 *Η εξίσωση* $h(o) = \lfloor \frac{\overrightarrow{a}\,\overrightarrow{o}+b}{w} \rfloor$ *διαθέτει ευαισθησία τοπικότητας σε σχέση με το* $(1, c, p_1, p_2)$*, όπου* p_1 *και* p_2 *είναι σταθερές έτσι ώστε* $\frac{\ln 1/p_1}{\ln 1/p_2} \leq \frac{1}{c}$.

3.6 LSB-δένδρο

Η κατασκευή ενός δένδρου-LSB είναι πολύ απλή. Δεδομένου ενός d-διάστατου συνόλου D, στην αρχή μετατρέπουμε κάθε σημείο $o \in D$ σε ένα m-διάστατο σημείο $G(o)$, και ακολούθως λαμβάνουμε την τιμή Z-τάξης του $G(o)$. Ας σημειωθεί ότι το $z(o)$ είναι απλώς ένας αριθμός. Επομένως, μπορούμε να δεικτοδοτήσουμε όλες τις τιμές Z-τάξης που προκύπτουν με ένα συμβατικό B-δένδρο, το οποίο είναι το LSB-δένδρο. Οι συντεταγμένες του αποθηκεύονται μαζί με τον αντίστοιχο κόμβο-φύλλο.

Από το o **στο** $G(o)$: Για πλήθος διαστάσεων m του $G(o)$ έχουμε:

$$m = log_{1/p_2}(dn/B)$$

όπου p_2 είναι η σταθερά του παραπάνω λήμματος για $c = 2$, n είναι το μέγεθος του συνόλου D, και B το μέγεθος της σελίδας. Με την συγκεκριμένη επιλογή του m, είναι απίθανο οι τιμές $G(o_1)$ και $G(o_2)$ δύο απομακρυσμένων σημείων o_1, o_2 να είναι ίδιες και στις m διαστάσεις. Η επιλογή $c = 2$ δεν είναι υποχρεωτική, και η τεχνική μας μπορεί να προσαρμοστεί σε οποιοδήποτε ακέραιο $c \geq 2$.

Η $G(o)$ προκύπτει από μια οικογένεια συναρτήσεων κερματισμού:

$$H(o) = \overrightarrow{a} * \overrightarrow{o} + b^*$$

όπου το $\overrightarrow{a}$ είναι ένα d-διάστατο διάνυσμα του οποίου οι συνιστώσες επιλέγονται ανεξάρτητα από την κανονική κατανομή (μέση τιμή 0, διασπορά 1). Η τιμή b^* κατανέμεται ομοιόμορφα στο $[0, 2^f w]$, όπου w μια σταθερά με $w \geq 4$ και

$$f = \lceil logd + logt \rceil$$

(t είναι η μεγαλύτερη συντεταγμένη σε κάθε διάσταση)

Παρατηρούμε ότι, ενώ το $\overrightarrow{a}$ και το w είναι τα ίδια όπως και στην εξίσωση (1), το b^* διαφέρει, κάτι που είναι σημαντικό για την αποδοτικότητα του δένδρου-LSB.

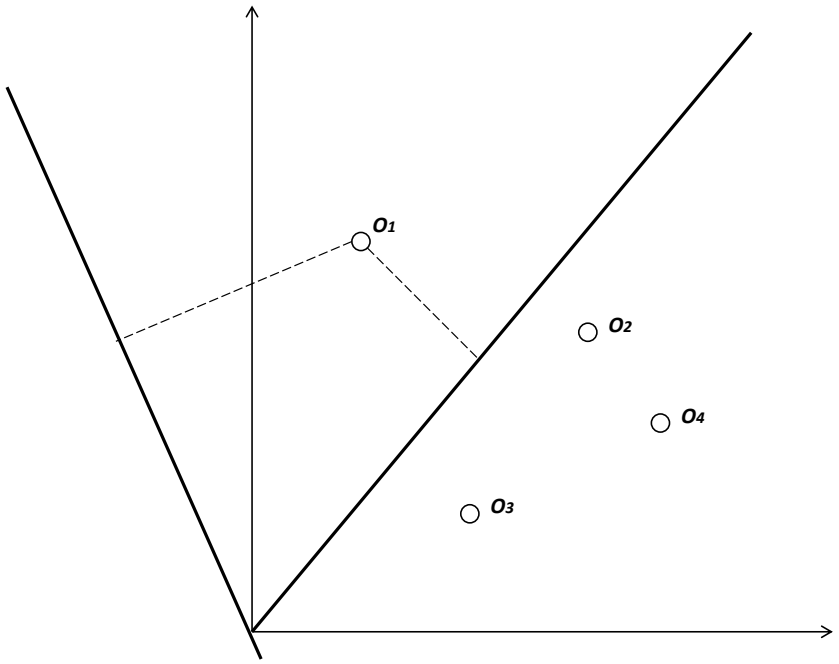
Επιλέγουμε τυχαία m συναρτήσεις $H_1(.), H_2(.), ..., H_m(.)$ από την οικογένεια H. Τότε, $G(o)$ είναι το m-διάστατο διάνυσμα $G(o) = \langle H_1(o), H_2(o), ..., H_m(o) \rangle$ (5).

Από το $G(o)$ στο $z(o)$. Έστω U το μήκος του άξονα του m-διάστατου χώρου στο οποίο ανήκει το $G(o)$. Θα επιλέξουμε μια τιμή του U έτσι ώστε το U/w να είναι δύναμη του 2. Ο υπολογισμός μιας καμπύλης Z-τάξης απαιτεί μία διαμέριση υπερπλέγματος (hyper-grid) του χώρου. Κάθε κελί του υπερπλέγματος είναι ένα υπερτετράγωνο πλευράς w. Έτσι, υπάρχουν U/w κελιά ανά διάσταση, και συνολικά $(U/w)^m$ κελιά σε ολόκληρο το πλέγμα. Δεδομένου του πλέγματος, ο υπολογισμός της τιμής $z(o)$ του $G(o)$ είναι μια τυποποιημένη διαδικασία. Έστω $u = log(U/W)$. Κάθε $z(o)$ είναι μια δυαδική συμβολοσειρά με um bits.

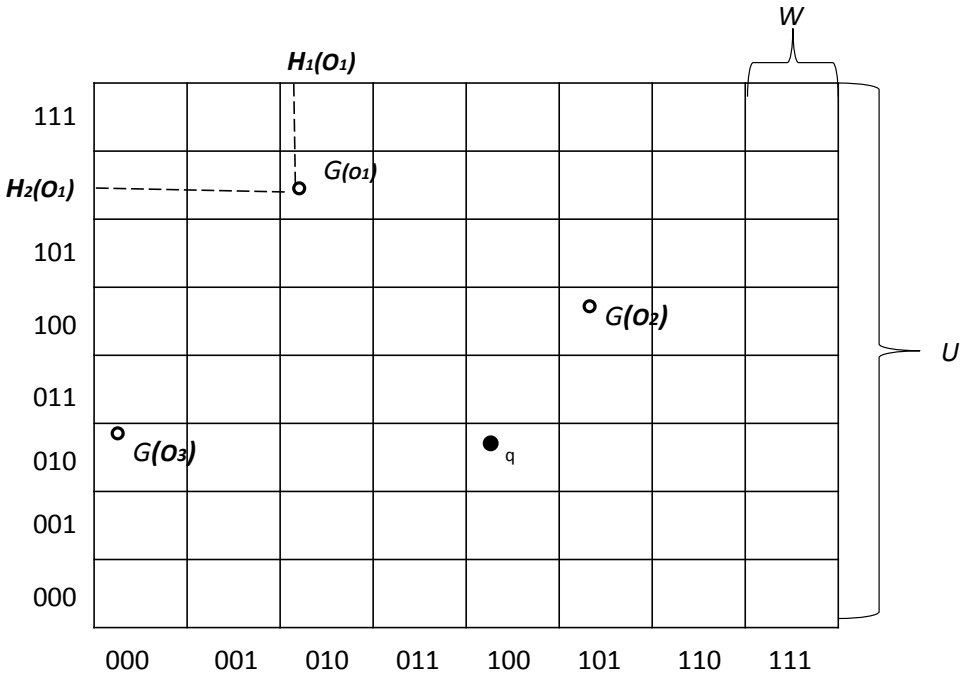
Παράδειγμα. Ας υποθέσουμε ότι το σύνολο D αποτελείται από 4 δισδιάστατα σημεία o_1, o_2, o_3, o_4 όπως φαίνεται στο σχήμα 3.4. Έστω ότι επιλέγουμε $m = 2$ συναρτήσεις κερματισμού $H_1(o_1)$ και $H_2(o_2)$, και $\overrightarrow{a_1}(\overrightarrow{a_2})$ είναι το «διάνυσμα $\overrightarrow{a}$ » στη συνάρτηση $H_1(o_1)H_2(o_2)$. Ας υποθέσουμε ότι και τα δύο διανύσματα έχουν νόρμα 1.

Ας πάρουμε για παράδειγμα το σημείο o_1. Το πρώτο βήμα της μετατροπής είναι να δημιουργήσουμε το $G(o_1)$, το οποίο είναι ένα 2-διάστατο διάνυσμα με συνιστώσες $H_1(o_1), H_2(o_2)$. Πρώτα προβάλλουμε το o_1 πάνω στην ευθεία $\overrightarrow{a}_1$ και στη συνέχεια μεταφέρουμε το προβαλλόμενο σημείο A (κατά μήκος της ευθείας) κατά b_1^* στο σημείο B. $H_1(o_1)$ είναι η απόσταση του B από την αρχή. Το $H_2(o_2$ υπολογίζεται με τον ίδιο τρόπο στην ευθεία $\overrightarrow{a}_2$.

Στο δεύτερο βήμα, αντιμετωπίζοντας τις $H_1(o_1), H_2(o_2)$ ως συντεταγμένες, θεωρούμε το $G(o_1)$ σημείο ενός χώρου δεδομένων όπως φαίνεται στο σχήμα 3.4, και καταλήγουμε στη $z(o_1)$. Στο σχήμα 3.4, ο υπολογισμός της $z(o_1)$ βασίζεται σε ένα πλέγμα 8×8. Αν το $G(o_1)$ ανήκει σε ένα κελί το οποίο έχει οριζόντια ετικέτα 010



Σχήμα 3.3: Σύστημα αξόνων



Σχήμα 3.4: Σύστημα πλέγματος

και κάθετη ετικέτα 110, η τιμή $z(o_1)$ είναι ίση με 011100.

Επιλογή του U. Στην πράξη, το U μπορεί να είναι οποιαδήποτε τιμή κάνει το U/w να είναι αρκετά μεγάλη δύναμη του 2. Επίσης, η επιλογή μας πρέπει να ικανοποιεί ακόμα δύο συνθήκες: 1) $U/w \geq 2^f$ και 2) η τιμή του $\mid H_i(o) \mid$ να είναι μικρότερη ή ίση από το $U/2$ για κάθε $i \in [1, m]$.

Για κάθε $i \in [1, m]$, γίνεται $H_i(o) = \overrightarrow{a}_i * \overrightarrow{o} + b_i^*$. Συμβολίζουμε με $\| \overrightarrow{a_i} \|_1$ τη νόρμα l_1 του $\overrightarrow{a}_i$. Έχουμε:

$$H_{max} = max_{i=1}^{m}(\|\overrightarrow{a}_i\|_1 * t + b_i^*)$$

Έτσι προσδιορίζουμε το U θέτοντας ως U/w τη μικρότερη δύναμη του 2 που είναι μεγαλύτερη ή ίση από 2^f και από $2H_{max}/w$.

3.7 Αλγόριθμος NN

Στην πράξη, ακόμα και ένα μόνο δένδρο-LSB μπορεί να δώσει αποτελέσματα με πολύ καλή ποιότητα. Για να αυξήσουμε την ποιότητα σε θεωρητικό επίπεδο, μπορούμε ανεξάρτητα να κατασκευάσουμε l δένδρα. Επιλέγουμε

$$l = \sqrt{dn/B}$$

Η τιμή αυτή εξασφαλίζει μεγάλη πιθανότητα τα κοντινά σημεία o_1, o_2 να έχουν κοντινές τιμές Z-τάξης σε ένα τουλάχιστον δένδρο.

Αν συμβολίσουμε $T_1, T_2, ..., T_l$ τα l δένδρα, τότε έχουμε ένα δάσος-LSB. Συμβολίζουμε με $z_j(o)$ την τιμή Z-τάξης του o στο δένδρο $T_j (1 \leq j \leq l)$. Επίσης η $z_j(o)$ θα αναφέρεται και στην καταχώρηση-φύλλο του o στο δένδρο T_j.

Δεδομένης μιας ερώτησης q, πρώτα παίρνουμε τις τιμές $z_j(q)$ σε κάθε δένδρο $T_j (1 \leq j \leq l)$. Η $z_j(q)$ είναι μια δυαδική συμβολοσειρά με um bits. Συμβολίζουμε με $LLCP(z_j(o), z_j(q))$ το μήκος του μακρύτερου κοινού προθέματος (length of the longest common prefix) των $z_j(o)$ και $z_j(q)$. Για παράδειγμα, έστω $z_j(o) = 100101$ και $z_j(q) = 100001$, τότε $LLCP(z_j(o), z_j(q)) = 3$.

Η κύρια ιδέα είναι να επισκεφτεί τα φύλλα όλων των l δέντρων με φθίνουσα σειρά των LLCP, μέχρι να έχουν βρεθεί αρκετά σημεία ή μέχρι να έχουμε βρει ένα σημείο που είναι αρκετά κοντά. Πρέπει όμως να βρεθεί το επόμενο μεγαλύτερο LLCP. Αυτό μπορεί να γίνει με μια σύγχρονη αμφίδρομη επέκταση στα επίπεδα φύλλων όλων των δέντρων. Συγκεκριμένα, έχουμε ήδη τη τιμή Z-τάξης $z_j(q)$ σε κάθε δένδρο $T_j \leq 1 \leq j \leq l$). Αναζητούμε στο T_j την καταχώρηση φύλλου e_j με την μικρότερη τιμή Z-τάξης. Έστω e_j το αμέσως επόμενο καταχώρησης φύλλο [9].

Η καταχώρηση φύλλου με το μεγαλύτερο LLCP πρέπει να βρίσκεται στο σύνολο $S = \{e_1, e_1, ..., e_l, e_l\}$, και έστω ότι αυτό είναι το $e \in S$. Για να προσδιορίσουμε την καταχώρηση φύλλου με το αμέσως επόμενο μεγαλύτερο LLCP, απομακρύνουμε το e από το q κατά μία θέση στο αντίστοιχο δένδρο, και στη συνέχεια επαναλαμβάνουμε την ίδια διαδικασία.

Συνθήκη τερματισμού. Ο αλγόριθμος NN1 τερματίζει όταν ένα από τα δύο ενδεχόμενα E_1, E_2 πραγματοποιείται.

E_1 : ο συνολικός αριθμός των καταχωρημένων φύλλων από όλα τα δένδρα-LSB έχει γίνει ίσος με $4Bl/d$.

E_2 : το πλησιέστερο σημείο που έχει βρεθεί ως τώρα απέχει από το q απόσταση το πολύ έως $2^{u-\lfloor u/m \rfloor+1}$.

Ας θεωρήσουμε το σύνολο που αποτελείται από τα σημεία o_1, o_2, o_3, o_4 στο σχήμα 3.3, και ότι η ερώτηση είναι το μαύρο σημείο q. Οι τιμές Z-τάξης στο δένδρο T_1 προκύπτουν από τον μετασχηματισμό του σχήματος 3.4 με $u = 3$ και $m = 2$. Έστω $\|o_3, q\| = 3$ και $\|o_4, q\| = 5$.

Η είσοδος $z_2(o_4)$ έχει τη μεγαλύτερη τιμή $LLCP = 5$. Ο αλγόριθμος βρίσκει το αντικείμενο o_4 και υπολογίζει την απόστασή του από το q. Αφού ισχύει $\|o_4, q\| = 5 > 2^{u-\lfloor u/m \rfloor+1} = 4$, η υπόθεση E_2 δεν ισχύει. Αν υποθέσουμε ότι ούτε η E_1 ισχύει (π.χ. έστω $4Bl/d > 1$), τότε ο αλγόριθμος επεξεργάζεται την είσοδο με το αμέσως μεγαλύτερο LLCP, δηλαδή το $z_1(o_3)$ το οποίο έχει $LLCP = u = 4$. Σε αυτή την είσοδο, ο αλγόριθμος βρίσκει το o_3 το οποίο αντικαθιστά το o_4 ως το πλησιέστερο σημείο μέχρι στιγμής. Αφού τώρα ισχύει $\|o_3, q\| = 3 \leq 2^{u-\lfloor u/m \rfloor+1} = 4$, η E_2 ισχύει,

άρα ο αλγόριθμος τερματίζει και επιστρέφει το o_3.

Ανάκτηση k γειτόνων. Ο αλγόριθμος NN1 μπορεί εύκολα να προσαρμοστεί για kNN αναζητήσεις. Συγκεκριμένα, αρκεί να τροποποιήσουμε την E_1 ως εξής: «ο συνολικός αριθμός των καταχώρησης φύλλων από όλα τα δένδρα-LSB να έχει γίνει ίσος με $(4Bl/d) + (k-1)l$» και την E_2 ως εξής: «το q βρίσκεται σε απόσταση εντός του $2^{u-\lfloor u/m \rfloor+1}$ από τα k πλησιέστερα σημεία που έχουν βρεθεί ως τώρα». Προφανώς η σειρά 4 του αλγορίθμου πρέπει να επιστρέψει τα k πλησιέστερα σημεία. Τέλος, η τιμή του l στην εξίσωση $l = \sqrt{dn/B}$ πρέπει να αυξηθεί $O(log(n))$ φορές. Αυτές οι αλλαγές γίνονται για να εξασφαλιστούν δυνατές εγγυήσεις ποιότητας στη θεωρία για κάθε k. Στην πράξη, όσο το k είναι μικρό, μόνο η αλλαγή στο E_2 χρειάζεται, ενώ το E_1 και το l μπορούν να παραμείνουν όπως είναι για k=1.

αναζήτηση kNN με ένα μόνο δένδρο. Ένα δάσος με l δένδρα-LSB απαιτεί μεγάλη κατανάλωση χώρου και επιφέρει μεγάλο κόστος ενημερώσεων. Στην πράξη, ίσως προτιμάμε ένα δείκτη που καταλαμβάνει γραμμικό χώρο και υποστηρίζει γρήγορα δεδομένα εισαγωγής/διαγραφής. Σε αυτή την περίπτωση, μπορούμε να κατασκευάσουμε μόνο ένα LSB-δένδρο, και να το χρησιμοποιήσουμε για να εκτελέσουμε αναζητήσεις kNN. Τροποποιούμε τον αλγόριθμο NN1 απλά παραλείποντας το ενδεχόμενο E_1. Ωστόσο, η συνθήκη E_2 παραμένει. Η αναζήτηση σε ένα μόνο δένδρο μπορεί να αυξάνει την αποδοτικότητα, μειώνει όμως τις θεωρητικές εγγυήσεις του δάσους-LSB. Ωστόσο, αυτή η προσέγγιση αναμένεται να επιστρέψει γείτονες με πολύ καλή ποιότητα, επειδή οι τιμές Z-τάξης διατηρούν επαρκώς την εγγύτητα των σημείων δεδομένων στον αρχικό χώρο δεδομένων [**?**].

3.8 Ανάλυση αλγορίθμου

Αποδεικνύεται ότι μπορούμε να κατασκευάσουμε ένα δάσος από l δένδρα LSB που καταναλώνουν συνολικά χώρο $O((dn/B)^{1.5})$. Δεδομένων αυτών των δένδρων, ο αλγόριθμος NN1 απαντά μία 4-προσεγγιστική ερώτηση πλησιέστερου γείτονα χρησιμοποιώντας $O(E\sqrt{dn/B})$ πράξεις εισόδου-εξόδου, όπου E είναι το ύψος του

δένδρου LSB. . **Απόδειξη**

Κάθε καταχώρηση-φύλλο από ένα LSB δένδρο αποθηκεύει μια τιμή Z-order $z(o)$ και τις συντεταγμένες των o. Η τιμή $z(o)$ έχει um bits όπου $u = O(f) = O(\log d + \log t)$ και$m = O(log(dn/B))$. Δεδομένου ότι $logd + logt$ bit χωράνε σε 2 λέξεις, η τιμή $z(o)$ καταλαμβάνει $O(log(dn/B))$ λέξεις. Χρειάζονται d λέξεις για να αποθηκευθούν οι συντεταγμένες του o. Ως εκ τούτου, συνολικά μια καταχώρηση-φύλλο έχει μέγεθος $O(d)$ και συνακόλουθα ένα δένδρο-LSB καταναλώνει $O((dn/B))$ σελίδες και $l = \sqrt{dn/B}$ από αυτές απαιτούν χώρο $O((dn/B)^{1.5})$.

Ο αλγόριθμος $NN1$ (1) αρχικά προσπελαύνει μία μόνο διαδρομή σε κάθε δένδρο-LSB, και στη συνέχεια (2) προσκομίζει το πολύ $4Bl/d$ καταχωρήσεις-φύλλα. Το κόστος του (1) φράσσεται από $O(lE)$. Δεδομένου ότι μια καταχώρηση σε φύλλα καταναλώνει $O(d)$ λέξεις, τα $4Bl/d$ τους καταλαμβάνουν το πολύ $O(l)$ σελίδες [16].

Υλοποιώντας κάθε δένδρο-LSB ως B-δένδρο συμβολοσειρών, το ύψος E φράσσεται από το $O(log_B n)$, οδηγώντας σε πολυπλοκότητα $O(\sqrt{dn/B} \log_B n)$.

Θεώρημα

Μπορούμε να χτίσουμε ένα δάσος από I LSB δένδρα που καταναλώνουν χώρο $O((dn/B)^{1.5})$. Δίνοντας σε αυτά τα δένδρα το αλγόριθμο $NN1$ που απαντά σε ένα ερώτημα NN 4-προσέγγισης στην $O(E\sqrt{dn/B})$ όπου E είναι το ύψος του δένδρου LSB.

3.9 Επεκτάσεις

Το δεύτερο πρόβλημα με το οποίο ασχολούμαστε είναι η αναζήτηση πλησιέστερου ζεύγους (ΠΖ) με προσέγγιση c. Συγκεκριμένα, έστω ότι το πλησιέστερο ζεύγος του D είναι εκείνο το ζεύγος των σημείων (o_1^*, o_2^*) με τη μικρότερη μεταξύ τους απόσταση από όλα τα ζεύγη των σημείων του D. Στόχος της αναζήτησης του ΠΖ με προσέγγιση c είναι να επιστρέψει ένα ζεύγος σημείων (o_1, o_2) του D των οποίων η μεταξύ τους απόσταση είναι το πολύ c φορές μεγαλύτερη από την απόσταση του πλησιέστερου ζεύγους, δηλαδή, $\| o_1, o_2 \| \leq c \| o_1^*, o_2^* \|$. Μια απλοϊκή λύση εξετάζει

όλα τα ζεύγη, και επομένως η πολυπλοκότητα χρόνου είναι τετραγωνική του n.

Ας σημειωθεί το πρόβλημα του ΠΖ έχει ένα αντίστοιχο διχρωματικό (bichromatic counterpart), που περιλαμβάνει δύο σύνολα D_1 και D_2. Εδώ, η ακριβής απάντηση (o_1^*, o_2^*) είναι το ζεύγος των σημείων με τη μικρότερη μεταξύ τους απόσταση στο καρτεσιανό γινόμενο $D_1 \times D_2$, ενώ η προσεγγιστική απάντηση είναι ένα ζεύγος $(o_1, o_2) \in D_1 \times D_2$ για το οποίο ισχύει $\| o_1, o_2 \| \leq c \| o_1^*, o_2^* \|$. Τα δύο αυτά προβλήματα ΠΖ μπορούν να επεκταθούν σε kCP αναζήτηση (kΠΖ), και η εκδοχή της με προσέγγιση c μπορεί να οριστεί με τον ίδιο τρόπο όπως της kNN [13].

3.10 Συμπεράσματα

Η αναζήτηση του πλησιέστερου γείτονα βρίσκει πολλές εφαρμογές σε ένα μεγάλο αριθμό επιστημονικών κλάδων. Ασχοληθήκαμε με μια μέθοδο ανάκτησης που ονομάζεται δένδρο-LSB για να δώσουμε τη δυνατότητα για γρήγορη αναζήτηση NN με εξαιρετικό αποτέλεσμα ποιότητας. Στην πράξη, χρησιμοποιώντας ένα μόνο δένδρο-LSB παρέχουμε ένα αποδοτικό σχήμα δεικτοδότησης που μπορεί εύκολα να ενσωματωθεί σε μία σχεσιακή βάση, καταναλώνει γραμμικό χώρο, υποστηρίζει ενημερώσεις με λογαριθμική αύξηση του χώρου και μπορει να χρησιμοποιηθεί για την απάντηση ερωτημάτων NN αποτελεσματικά και με ακρίβεια.

Ειδικότερα, μπορεί να εφαρμοστεί αμέσως σε ένα εμπορικό σύστημα. Επιπλέον, αυτή η λύση μπορεί άμεσα να αξιοποιήσει το σχήμα ευρετηρίου που αναφέρθηκε παραπάνω για την αναζήτηση NN. Αυτό είναι ένα αρκετά ελκυστικό χαρακτηριστικό διότι, με μόνο μία ενιαία δομή, είναι σε θέση να αντιμετωπίσουν επαρκώς δύο δύσκολα προβλήματα ταυτόχρονα.

Κεφάλαιο 4

Μέθοδος ευρετηρίου MedRank

4.1 Πλησιέστεροι γείτονες και Ψηφοφόροι

Στο κεφάλαιο αυτό περιγράφεται μία πρωτότυπη προσέγγιση για αποδοτική αναζήτηση ομοιότητας και για κατηγοριοποίηση σε πολυδιάστατα δεδομένα. Τα στοιχεία της βάσης δεδομένων είναι διανύσματα στον ευκλείδειο χώρο. Δοθέντος ενός διανύσματος, ο στόχος είναι να βρεθούν στοιχεία της βάσης δεδομένων που είναι όμοια με το διάνυσμα. Στην προσέγγισή μας, ένας μικρός αριθμός ανεξάρτητων «ψηφοφόρων» ταξινομούν τα στοιχεία της βάσης δεδομένων ανάλογα με την ομοιότητά τους με την ερώτηση. Στη συνέχεια, αυτές οι διατάξεις ενώνονται μέσω ενός πολύ αποδοτικού αλγόριθμο συνάθροισης (aggregation). Η μέθοδός μας οδηγεί και σε τεχνικές υπολογισμού πλησιέστερων γειτόνων κατά προσέγγιση και σε μία εννοιολογικά πλούσια εναλλακτική μέθοδο για εύρεση των πλησιέστερων γειτόνων.

Κάθε ψηφοφόρος προβάλλει όλα τα διανύσματα (τα στοιχεία της βάσης δεδομένων και την ερώτηση) σε μια τυχαία ευθεία (διαφορετική για κάθε ψηφοφόρο), και ταξινομεί τα στοιχεία της βάσης δεδομένων βάσει της εγγύτητας των προβολών στην την προβολή της ερώτησης. Κατά τη συνάθροιση, επιλέγεται το στοιχείο της βάσης δεδομένων με την καλύτερη θέση διαμέσου (median rank). Η ένωση αυτή έχει διάφορα ελκυστικά χαρακτηριστικά [5].

Από θεωρητικής πλευράς, παράγει, με μεγάλη πιθανότητα, ένα αποτέλεσμα που είναι μία προσέγγιση (1 + e) του ευκλείδειου πλησιέστερου γείτονα. Από πρακτικής πλευράς, αποδεικνύεται ότι είναι εξαιρετικά αποδοτική, και συχνά, ερευνά λιγότερο από το 5% των δεδομένων για να καταλήξει σε αποτελέσματα πολύ υψηλής ποιότητας. Η μέθοδος αυτή, επίσης, είναι φιλική προς τη βάση δεδομένων, με την έννοια ότι η πρόσβαση στα δεδομένα γίνεται κυρίως με προκαθορισμένη σειρά χωρίς τυχαίες προσπελάσεις και, σε αντίθεση με άλλες μεθόδους για προσεγγιστική εύρεση γειτόνων, δεν απαιτεί σχεδόν καθόλου παραπάνω χώρο αποθήκευσης. Επίσης, η μέθοδός μας επεκτείνεται και για K- πλησιέστερους γείτονες.

Η MedRank είναι μία πρωτότυπη μέθοδος για την αναζήτηση ομοιότητας, για προβλήματα κατηγοριοποιήσης, και για άλλες εφαρμογές που βασίζονται στην αναζήτηση πλησιέστερου γείτονα. Η μέθοδός μας ικανοποιεί τα δύο ακόλουθα κριτήρια: είναι μία αυτοδύναμη γενίκευση του πλησιέστερου γείτονα, και ενσωματώνει αλγόριθμους οι οποίοι είναι εξαιρετικά αποδοτικοί και φιλικοί.

Έστω ότι εκτελούμε αναζήτηση πλησιέστερου γείτονα με μία βάση δεδομένων Dn σημείων στον χώρο d-διάστατο X^d, (όπου X είναι το υποκείμενο σύνολο των πραγματικών, το $\{0, 1\}$ κ.λπ.) και έστω ότι έχουμε μια ερώτηση $q \in {}^d$. Θεωρούμε ότι κάθε συντεταγμένη του d-διάστατου χώρου ότι είναι «ψηφοφόρος» (“voter”) και τα n σημεία ότι είναι οι «υποψήφιοι» στις εκλογές. Ο ψηφοφόρος j, με $1 \leq j \leq d$, ταξινομεί όλους τους n υποψήφιους ανάλογα με το πόσο κοντά βρίσκονται στην ερώτηση στη j-οστή συντεταγμένη. Έτσι έχουμε d ταξινομημένες λίστες υποψηφίων και στόχος μας είναι να συνθέσουμε από αυτές μία μόνο διάταξη υποψηφίων. Μας ενδιαφέρουν οι υποψήφιοι που βρίσκονται στις πρώτες θέσεις σε αυτή την ενοποιημένη διάταξη [6].

4.2 Το πρόβλημα της συνάθροισης διατάξεων

Το πρόβλημα της συνάθροισης διατάξεων (rank aggregation) διατυπώνεται εξής: πώς να συνδυάσουμε τις d ταξινομημένες λίστες που δημιουργήθηκαν από τις d

συντεταγμένες. Το πρόβλημα αυτό έχει ιστορία δύο αιώνων, όμως από μαθηματικής πλευράς έγινε κατανοητό τα τελευταία 60 χρόνια, και υπάρχει ακόμη μεγάλο ερευνητικό ενδιαφέρον για τα υπολογιστικά προβλήματα που προκύπτουν.

Τα πιο σημαντικά μαθηματικά θέματα σχετικά με την ενοποίηση διατάξεων αφορούν στο να βρεθούν αυτοδύναμοι μηχανισμοί ενοποίησης. Αξιοσημείωτες είναι οι μελέτες των Young και Levenglick που έδειξαν ότι η πρόταση του Kemeny οδηγεί σε έναν μηχανισμό ενοποίησης με πολλά επιθυμητά χαρακτηριστικά. Για παράδειγμα, ικανοποιεί το κριτήριο Condorcet, σύμφωνα με το οποίο αν υπάρχει υποψήφιος c έτσι ώστε για οποιονδήποτε άλλον υποψήφιο c, η πλειοψηφία των ψηφοφόρων προτιμούν τον c αντί του c, τότε ο c πρέπει να είναι αυτός που θα κερδίσει τις εκλογές. Οι μηχανισμοί ενοποίησης που ικανοποιούν το κριτήριο Condorcet έχουν παράγει αυτοδύναμα αποτελέσματα που δεν αλλοιώνονται από λίγους κακούς ψηφοφόρους.

Η πρόταση του Kemeny [6] είναι η εξής: αν έχουμε n υποψηφίους και d μεταθέσεις $_1, _2, ..., _d$ αυτών των υποψηφίων, φτιάχνουμε τον συνδυασμό σ που ελαχιστοποιεί το $\sum_{i=1}^{d} K(T_i, \sigma)$, όπου με $K(\tau, \sigma)$ συμβολίζεται η απόσταση Kendall tau, δηλαδή το πλήθος των ζευγών $(c, c^{'})$ των υποψηφίων για τα οποία οι διατάξεις τ και σ διαφωνούν (η μία από τις δύο κατατάσσει τον c πριν από τον $c^{'}$, ενώ η άλλη κατατάσσει τον $c^{'}$ πριν από τον c). Την προσέγγιση αυτή θα την καλούμε βέλτιστη συνάθροιση. Δυστυχώς, μια τέτοια ενοποίηση ακόμη και 4 λιστών είναι NP-complete (NP-πλήρης), επομένως πρέπει να καταφύγουμε σε προσεγγιστικούς αλγόριθμους και ευρετικές μεθόδους.

Ας δούμε ποια η σχέση ανάμεσα στους πλησιέστερους γείτονες και στη συνάθροιση διατάξεων. Αν πάρουμε για παράδειγμα τον χώρο $\{0,1\}^d$ εφοδιασμένο με τη μετρική Hamming. Κάθε ψηφοφόρος παράγει μια μερική κατάταξη. Δεδομένης μιας ερώτησης q, ο i-οστός ψηφοφόρος χωρίζει τη βάση δεδομένων D σε δύο σύνολα $D_i^+ = \{x \in D \mid x_i = q_i\}$ και $D_i^- = \{x \in D \mid x_i \neq q_i\}$, κατατάσσοντας όλα τα D_i^+ πριν από τα D_i^-. Σε αυτή την περίπτωση, η βέλτιστη ενοποίηση των μερικών διατάξεων που παράγονται από τους ψηφοφόρους ταξινομεί επακριβώς τα

σημεία της βάσης δεδομένων με βάση την απόσταση τους κατά Hamming από το διάνυσμα q . Λαμβάνοντας υπόψη ότι τα προβλήματα πλησιέστερου γείτονα σε διάφορες ενδιαφέρουσες μετρικές μπορούν να αναχθούν στη μετρική Hamming, σημειώνουμε ότι η μέθοδος της συνάθροισης, γενικά, είναι τουλάχιστον εξίσου ισχυρή με τον πλησιέστερο γείτονα.

Από την άλλη πλευρά, έχουμε θεωρήσει ένα πρόβλημα (το πρόβλημα του πλησιέστερου γείτονα) το οποίο μπορεί να λυθεί με έναν απλό αλγόριθμο σε χρόνο $O(nd)$ και το μετατρέπουμε σε ένα πρόβλημα NP-πλήρες. Ακόμα και καλοί προσεγγιστικοί αλγόριθμοι για το πρόβλημα της συνάθροισης χρειάζονται χρόνο $\Omega(nd+n^2$ τουλάχιστον. Ωστόσο, ο συνδυασμός δύο βασικών παραγόντων μας απαλλάσσει από αυτό το δίλημμα. Πρώτον, μας ενδιαφέρουν μόνο τα λίγα στοιχεία που βρίσκονται πιο πάνω στην ενοποιημένη διάταξη (κορυφαία σημεία), και όχι όλα τα στοιχεία της διάταξης των σημείων της βάσης δεδομένων. Δεύτερον, όσον αφορά την εύρεση των k κορυφαίων νικητών στην ενοποίηση, μία ευρετική μέθοδος που είναι βασισμένη σε διατάξεις διαμέσου τελικά επιδέχεται εξαιρετικά αποδοτική υλοποίηση.

Ένα ακόμη πλεονέκτημα της ενοποιημένης διάταξης σε σύγκριση με τους πλησιέστερους γείτονες όσον αφορά τις βάσεις δεδομένων είναι το εξής: ας θεωρήσουμε ένα πρόβλημα αναζήτησης ομοιότητας όπου για τα αντικείμενα δεν υπάρχει φυσικό ταίριασμα με κανέναν μετρικό χώρο, όπως είναι για παράδειγμα ένας κατάλογος συσκευών, όπου τα «χαρακτηριστικά» μπορεί να είναι στοιχεία κατηγοριών (π.χ. χρώμα) ή αριθμητικά για τα οποία όμως διαφορετικές συντεταγμένες έχουν ασύμβατες μονάδες (π.χ. δολάρια και εκατοστά).

Σε αυτές τις περιπτώσεις, τα αντικείμενα δεν γίνεται να απεικονιστούν ως σημεία σε έναν μετρικό χώρο όπου όλες οι συντεταγμένες έχουν την ίδια σημασία. Το παράδειγμα της ενοποιημένης διάταξης, όταν αναζητά αντικείμενα όμοια με το αντικείμενο ερώτησης, απλά ταξινομεί τη βάση δεδομένων ανάλογα με κάθε χαρακτηριστικό (π.χ. προτίμηση χρώματος, κόστος, κ.λπ.) και συναθροίζει τις διατάξεις που δημιουργήθηκαν. Οι αναζητήσεις καταλόγου είναι πολύ συνηθισμένες

λειτουργίες στις βάσεις δεδομένων, και ο αλγόριθμός μας, MedRank, αν εφαρμοστεί κατάλληλα, δίνει μια αποδοτική λύση σε αυτό το πρόβλημα [4].

4.3 Συνάθροιση διαμέσου διατάξεων

Η Συνάθροιση διαμέσου διατάξεων (Median rank aggregation) ταξινομεί όλα τα σημεία της βάσης δεδομένων βάσει της διαμέσου των διατάξεων από τους d ψηφοφόρους. Αυτή η ευρετική μέθοδος έχει νόημα, αφού, αν η μεσαίες διατάξεις είναι όλες διακριτές, τότε αυτή η διαδικασία οδηγεί σε μια βέλτιστη ενοποίηση. Έτσι, το πρόβλημά μας ανάγεται στην εύρεση του σημείου της βάσης δεδομένων με την καλύτερη διάταξη διαμέσου.

Τα πλεονεκτήματα της median rank aggregation είναι τα εξής

1) *Είναι φιλική προς τη βάση δεδομένων*. Η μέθοδος αυτή χρησιμοποιεί την ταξινόμηση ως το μόνο βήμα προεπεξεργασίας, δεν χρειάζεται σχεδόν καθόλου παραπάνω χώρο αποθήκευσης και δεν πραγματοποιεί σχεδόν καθόλου τυχαίες προσπελάσεις. Χωρίς τις τυχαίες προσπελάσεις, η μέθοδός μας δεν χρειάζεται δείκτες για τον εντοπισμό της τιμής μιας συντεταγμένης ενός στοιχείου.

Ας θεωρήσουμε ότι προ-ταξινομούμε τα στοιχεία της βάση δεδομένων με βάση κάθε μία από τις d-συντεταγμένες. Δεδομένης μιας ερώτησης $q = (q_1, q_2, ..., q_d)$, μπορούμε εύκολα να εντοπίσουμε την τιμή q_i, για $1 \leq i \leq d$ στην i-οστή ταξινομημένη λίστα, και τοποθετούμε δύο δείκτες σε αυτή τη θέση. Όταν τοποθετηθούν οι $2d$ δείκτες (δύο για κάθε i), μετακινώντας τον έναν δείκτη «πάνω» και τον άλλον «κάτω», μπορούμε να δημιουργήσουμε τη διατεταγμένη λίστα του i-οστού ψηφοφόρου, ένα στοιχείο τη φορά και να το προσκομίζουμε όταν μας χρειάζεται.

Την πρώτη φορά που ζητείται ο i-οστός ψηφοφόρος, επιστρέφεται το στοιχείο της βάσης δεδομένων που βρίσκεται πιο κοντά στο q σε ό,τι αφορά τη συντεταγμένη i, τη δεύτερη φορά θα επιστρέψει το δεύτερο πλησιέστερο στοιχείο σε ό,τι αφορά συντεταγμένη i, κ.ο.κ. Έτσι, έχουμε μια online εκδοχή του προβλήματος[4].

Το γεγονός ότι μπορούμε εύκολα να έχουμε online πρόσβαση στους d ψηφοφό-

ρους, σε συνδυασμό με το γεγονός ότι θέλουμε να βρούμε τον υποψήφιο με την καλύτερη διάταξη διαμέσου, μας οδηγεί στο συμπέρασμα ότι ίσως μπορούμε να βρούμε τον νικητή χωρίς καν να χρειαστεί να διαβάσουμε ολόκληρες τις διατεταγμένες λίστες.

2) *Πλησιέστεροι γείτονες κατά προσέγγιση.* Η μέθοδος MedRank χρησιμοποιεί την ιδέα των προβολών σε τυχαίες ευθείες σε d-διάστατο χώρο. Αν προβάλλουμε τα n σημεία της βάσης δεδομένων σε m διαστάσεις, όπου $m = O(\varepsilon^{-2} \log n)$, και στη συνέχεια να εκτελέσουμε τον αλγόριθμο MedRank στα δεδομένα της προβολής, τότε, με μεγάλη πιθανότητα, ο νικητής είναι μια ε-προσέγγιση του πλησιέστερου γείτονα του σημείου ερώτησης, με βάση την Ευκλείδεια μετρική.

Άρα, τα δύο κύρια τεχνικά αποτελέσματα στα οποία καταλήγουμε είναι: **(1)** αναγωγή από το πρόβλημα του ε-προσεγγιστικού ευκλείδειου πλησιέστερου γείτονα στο πρόβλημα εύρεσης του υποψηφίου με την καλύτερη διάταξη διαμέσου σε μια εκλογή όπου υπάρχουν n υποψήφιοι και $O(\varepsilon^{-2} \log n)$ ψηφοφόροι, και **(2)** μία απόδειξη ότι το πλήθος των προσπελάσεων που πραγματοποιεί ο αλγόριθμος MedRank (ο οποίος εκτελεί μόνο γραμμικές προσπελάσεις στις d ταξινομημένες λίστες) υπερβαίνει το πολύ κατά έναν σταθερό παράγοντα το πλήθος των προσπελάσεων που απαιτεί οποιοσδήποτε άλλος αλγόριθμος που χρησιμοποιεί σειριακές και τυχαίες προσπελάσεις στις λίστες, για κάθε βάση δεδομένων και για κάθε ερώτηση [6].

4.4 Ο Αλγόριθμος MedRank

Υποθέτουμε ότι έχουμε μια βάση δεδομένων D με n σημεία στο χώρο R^m, όπου $m = d$ (ο ευκλείδειος χώρος) ή $m = O(\varepsilon^{-2} \log n)$ (ο χώρος μετά την προβολή όλων των δεδομένων στις m τυχαίες ευθείες). Για $c \in D$ και $1 \leq i \leq m$, συμβολίζουμε με c_1 την τιμή του c στην i -οστή συντεταγμένη.

Ο αλγόριθμος MedRank ανήκει σε μία οικογένεια συνάθροισης αλγορίθμων, όπου μπορούμε να ενισχύσουμε την έννοια της διαμέσου, θεωρώντας και άλλα

ποσοστημόρια, πέραν αυτού που αντιστοιχεί στο 50 %. Εισάγουμε την παράμετρο Minfreq στον MedRank για να μεταβάλλει την τιμή αυτή στα άλλα ποσοστημόρια. Η παράμετρος Minfreq είναι ένα αυστηρό κάτω όριο για τον αριθμό των λιστών στις οποίες πρέπει να εμφανιστεί ένα στοιχείο για να είναι ο νικητής. Ο διάμεσος αντιστοιχεί σε Minfreq=0,5. Αυξάνοντας την τιμή της παραμέτρου, μπορούμε να αναμένουμε βελτίωση της ποιότητας, όμως, θα γίνονται περισσότερες αναζητήσεις στη βάση δεδομένων, και επομένως θα έχουμε χειρότερο χρόνο εκτέλεσης [6].

Προεπεξεργασία ερώτησης

Δημιούργησε m λίστες $L_1, L_2, ..., L_m$, όπου η L_i αποτελείται από τα ζεύγη c, c_i για κάθε $c \in D$.

Για $1 \leq I \leq m$, ταξινόμησε την L_i σε αύξουσα σειρά της δεύτερης συνιστώσας. Τώρα κάθε L_i είναι της μορφής $(c_{i,1}, u_{i,1}), ..., (c_{i,n}, u_{i,n})$, όπου $c_{i,t}$ είναι τα n διακριτά αντικείμενα στη βάση δεδομένων, και $u_{i,1} \leq u_{i,2} \leq ... \leq u_{i,n}$.

Επεξεργασία ερώτησης

Αν δίνεται ερώτηση $q \in \Re^m$, για κάθε i, αρχικοποίησε δύο δείκτες h_i και l_i στην L_i, έτσι ώστε $u_{i,h_i} \leq q_i \leq u_{i,l_i}$.

Το S θα είναι ένα σύνολο που περιλαμβάνει τα στοιχεία που έχουν εξετασθεί $c \in D$ και τις συχνότητές τους f_c. Αρχικοποίησε το S να είναι το κενό σύνολο $\oslash$.

Ο αλγόριθμος αυτός μπορεί να περιγραφεί ως εξής: «Προσπέλασε τις διατεταγμένες λίστες, ένα στοιχείο κάθε λίστας τη φορά, μέχρι να έχουμε συναντήσει κάποιο στοιχείο περισσότερες από Minfreq φορές τότε αυτός θα είναι και ο νικητής» [6].

Ο MedRank είναι ο καλύτερος αλγόριθμος για κάθε στιγμιότυπο (instance) από όλους τους αλγόριθμους που εκτελούν γραμμικές προσπελάσεις. Ακόμα και αν γίνονται γραμμικές και τυχαίες προσπελάσεις, ο αλγόριθμος χρειάζεται χρόνο που είναι το πολύ μία σταθερά μεγαλύτερη από τον καλύτερο δυνατό χρόνο για κάθε στιγμιότυπο, δηλαδή ο αλγόριθμος είναι βέλτιστου στιγμιότυπου (instance optimal).

while το S δεν έχει στοιχείο c τέτοιο ώστε $f_c > Minfreq * m$ **do**:

for $1 \leq i \leq m$ do: **do**

if $\mid u_{i,h_i} - q_i \mid < \mid u_{i,l_i} - q_i \mid$ **then**

θέσε $c = c_{i,h_i}$ και μείωσε το h_i

elseθέσε $c = c_{i,l_i}$ και αύξησε το l_i

if $c \notin S$ **then**,

πρόσθεσε το c στο S και θέσε $f_c = 1$

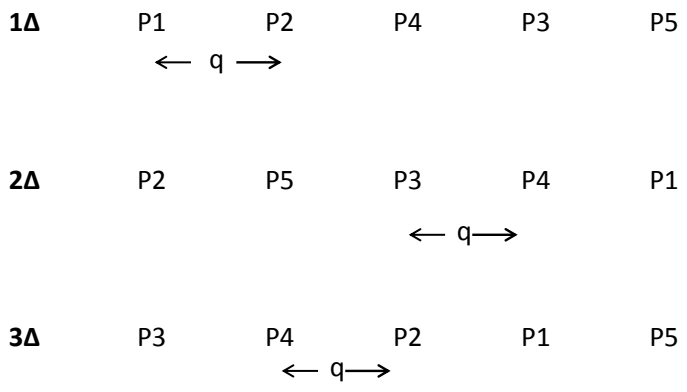
elseαύξησε το f_c

End-For

End-While

Επίστρεψε το στοιχείο $c \in S$ με τη μεγαλύτερη συχνότητα f_c

Σχήμα 4.1: Ο Ψευδοκώδικας MedRank



Σχήμα 4.2: Αλγόριθμος MedRank

while το S δεν έχει στοιχείο c τέτοιο ώστε $f_c > Minfreq * m$ **do**:

for $1 \leq i \leq m$ do: **do**

for **do**$c \in \{c_{i,h_i}, c_{i,l_i}\}$ **do**:

if $c \notin S$ **then** πρόσθεσε το c στο S και θέσε $f_c = 1$

elseαύξησε το f_c

End-For μείωσε το h_i και αύξησε το l_i

End-For

End-While

Επίστρεψε το στοιχείο $c \in S$ με τη μεγαλύτερη συχνότητα f_c

Σχήμα 4.3: Ο Ψευδοκώδικας OMedRank

4.5 Παραλλαγές του MedRank

Στο κεφάλαιο αυτό θα αναφερθούμε σε δυο αλγορίθμους που είναι παραλλαγές του Medrank, τον Omedrank και τον L2TA.

Ο αλγόριθμος Omedrank είναι ένας ευρετικός αλγόριθμος, που βασίζεται στην εξής συλλογιστική: Αντί να συγκρίνουμε τις τιμές u_{i,h_i} και u_{i,l_i} και να επιλέξουμε την πλησιέστερη στο q_i, εξετάζουμε τα δύο στοιχεία c_{i,h_i} και c_{i,l_i}. Αφού δεν εκτελούμε τυχαίες προσπελάσεις (π.χ. «βρες την διάταξη του c_{i,h_i} σε κάποια άλλη λίστα L_j»), θα αυξηθεί ο αριθμός των στοιχείων που μελετάμε αν ανήκουν στο S, αλλά εξοικονομούμε πολλές συγκρίσεις. Η Omedrank είναι μια ευρετική μέθοδος με στόχο να βελτιώσει περισσότερο τον χρόνο εκτέλεσης [17].

Προ επεξεργασία

Ίδια με τον MedRank

Επεξεργασία ερώτησης

Αν δίνεται ερώτηση $q \in \Re^m$, για κάθε i, αρχικοποίησε δύο δείκτες h_i και l_i στην L_i, έτσι ώστε $u_{i,h_i} \leq q_i \leq u_{i,l_i}$.

Το S θα είναι ένα σύνολο στοιχείων που έχουμε εξετάσει («seen elements») $c \in D$ και οι συχνότητές τους f_c. Αρχικοποίησε το S να είναι το κενό σύνολο $\oslash$.

Ο αλγόριθμος L2TA είναι ένας instance optimal αλγόριθμος για τον υπολογισμό ευκλείδειων πλησιέστερων γειτόνων. Ο αλγόριθμος αυτός είναι μια εφαρμογή του «αλγορίθμου κατωφλίου» (threshold algorithm, TA) στο πρόβλημα του υπολογισμού ευκλείδειων ($L2$) πλησιέστερων γειτόνων, στο μοντέλο όπου η πρόσβαση στα δεδομένα σε κάθε συντεταγμένη γίνεται μέσω σειριακών και τυχαίων προσπελάσεων. Μπορεί να χρησιμοποιηθεί στη θέση του απλοϊκού αλγόριθμου πλησιέστερου γείτονα και συχνά είναι πολύ πιο γρήγορος. Ονομάζουμε αυτόν τον αλγόριθμο L2TA. Συμβολίζουμε με L2NN τον απλοϊκό αλγόριθμο εύρεσης πλησιέστερων γειτόνων μέσω μιας σειριακής σάρωσης όλων των στοιχείων της βάσης δεδομένων [17].

Προεπεξεργασία

Δημιούργησε m λίστες $L_1, L_2, ..., L_m$, όπου η L_i αποτελείται από τα ζεύγη c, c_i για κάθε $c \in D$.

Για $1 \leq i \leq m$, ταξινόμησε την L_i σε αύξουσα σειρά της δεύτερης συνιστώσας. Τώρα κάθε L_i είναι της μορφής $(c_{i,l}, u_{i,l}), ..., (c_{i,n}, u_{i,n})$, όπου $c_{i,t}$ είναι τα n διακριτά αντικείμενα στη βάση δεδομένων, και$u_{i,1} \leq u_{i,2} \leq ... \leq u_{i,n}$.

Δημιούργησε τον δείκτη P έτσι ώστε $P(i, c)$ είναι ίσο με τη τιμή του j όπου $c_{i,j} = c$, για κάθε $c \in D$ και $1 \leq i \leq m$. Δηλαδή, $P(i, c)$ είναι η θέση του c στην ταξινομημένη λίστα L_i.

Επεξεργασία ερώτησης

Αν δίνεται ερώτηση $q \in \Re^m$, για κάθε i, αρχικοποίησε δύο δείκτες h_i και l_i στην L_i, έτσι ώστε $u_{i,h_i} \leq q_i \leq u_{i,l_i}$.

Το S θα είναι ένα σύνολο που αποτελείται από τα στοιχεία που έχουμε εξετάσει («seen elements») $c \in D$ και τις αποστάσεις τους d_c από το q. Αρχικοποίησε το S να είναι το κενό σύνολο $\oslash$.

Το T θα είναι μια τιμή κατωφλίου που εντοπίζει την ελάχιστη δυνατή απόσταση οποιουδήποτε μη εξετασθέντος στοιχείου $c \in S$ από το q. Αρχικοποίησε το $T = 0$.

while το S δεν έχει στοιχείο c τέτοιο ώστε $f_c \leq YT$ **do**:

for $1 \leq i \leq m$ do: **do**

for do$c \in \{c_{i,h_i}, c_{i,l_i}\}$ **do**:

θέσε $a_i =\mid v_{i,h_i} - q_i \mid b_i =\mid v_{i,l_i} - q_i \mid$

if $a_i < b_i$ **then**

θέσε $c = c_{i,h_i}$ και μείωσε το h_i

elseθέσε $c = c_{i,l_i}$ και αύξησε το l_i

if $c \notin S$ **then**

θέσε s=0

for $1 \leq j \leq m$ **do: do** θέσε$p = P(j, c;$ θέσε $s = s + u^{2j,p}$)

End-For πρόσθεσε το c στο S και θέσε $d_c = \sqrt{s}$

End-if

End-for$T = \sum_{i=1}^{m} \min(a_i, b_i)^2)^{1/2}$

End-for

Επίστρεψε το στοιχείο $c \in S$ με τη μικρότερη τιμή d_c

Σχήμα 4.4: Ο Ψευδοκώδικας L2TA

4.6 Πειράματα

Τα πειράματά που παρουσιάζονται στη συνέχεια αποτελούνται από δύο σύνολα, τα οποία ονομάζονται STOCK και HW αντίστοιχα [6]. Μελετώντας τα πειράματα αυτά, καταλήγουμε στα εξής συμπεράσματα όσον αφορά τους αλγόριθμους:

1) Και ο MedRank και ο Omedrank είναι εξαιρετικά γρήγοροι και σαρώνουν μόνο ένα μικρό μέρος της βάσης δεδομένων. Συνεπώς οι αλγόριθμοι αυτοί είναι φιλικοί προς τη βάση δεδομένων και αποτελούν μία πολύ αποδοτική και αποτελεσματική εναλλακτική της L2NN.

2) Η προβολή των δεδομένων σε λιγότερες διαστάσεις αποτελεί πάντα πλεονέκτημα, αν αυτό που μας ενδιαφέρει μόνο είναι οι πλησιέστεροι γείτονες κατά προσέγγιση. Στα δεδομένα της προβολής η ποιότητα αυτών των αλγορίθμων είναι περίπου ίδια με την ποιότητα του L2NN για τα ίδια δεδομένα, ενώ ο χρόνος εκτέλεσης είναι σημαντικά καλύτερος. Η προβολή μειώνει επίσης σημαντικά (κατά μία τουλάχιστον τάξη) το βάθος της αναζήτησης.

3) Συγκρίνοντας τον MedRank με τον Omedrank, σε πολλές περιπτώσεις ο Omed rank προσφέρει μέχρι και 20 φορές μεγαλύτερη ταχύτητα από τον MedRank, ενώ διατηρεί την υψηλή ποιότητα των αποτελεσμάτων.

4) Η παράμετρος Minfreq παίζει έναν ρόλο ο οποίος έχει διαφορετική σημασία για τον MedRank και τον Omedrank. Για το πείραμα STOCK, αυτή η παράμετρος δεν παίζει σημαντικό ρόλο, γι' αυτό αρκεί να την διατηρούμε χαμηλή (0,5), όπου και δίνει εξαιρετικούς χρόνους εκτέλεσης. Στο πείραμα HW, συμβάλλει στη μείωση του σφάλματος. Ωστόσο, η παράμετρος Minfreq επηρεάζει το βάθος αναζήτησης (και επομένως και τον χρόνο εκτέλεσης) αυτών των αλγορίθμων. Το βάθος αναζήτησης όμως παραμένει μία ή δύο τάξεις μικρότερο από το μέγεθος της βάσης δεδομένων, κάτι που δείχνει την αυτοδυναμία αυτών των αλγορίθμων.

5) Σχετικά με το πόσο μακριά πρέπει να πάει ο MedRank ώστε να ανακαλύψει καθένα από τα 10 κορυφαία αποτελέσματα, δεν υπάρχει μεγάλη διαφορά αν

ζητήσουμε να βρει μόνο το πρώτο ή τα πρώτα 10 αποτελέσματα.

6) Ο L2TA για το πρόβλημα του πλησιέστερου γείτονα προσφέρει μια βελτίωση στην ταχύτητα σε λίγες διαστάσεις. Πραγματοποιεί προσπελάσεις σε μεγάλο μέρος της βάσης δεδομένων, ενώ ο MedRank μόνο σε ένα πολύ μικρό κομμάτι.

4.7 Συμπεράσματα

Η συνάθροιση διατάξεων είναι μια διαφορετική προσέγγιση για την αναζήτηση ομοιότητας και την ταξινόμηση. Θεωρούμε ότι η ερώτηση και οι υποψήφιοι είναι σημεία σε έναν πολυδιάστατο χώρο. Κάθε συντεταγμένη θεωρείται ως ένας ψηφοφόρος, ο οποίος κατατάσσει τα σημεία ανάλογα με το πόσο κοντά βρίσκονται στην αντίστοιχη συντεταγμένη της ερώτησης.

Νικητές θεωρούνται τα σημεία εκείνα που έχουν την υψηλότερη συνάθροιση διατάξεων. Σε συνδυασμό με τεχνικές μείωσης των διαστάσεων, η προσέγγιση αυτή προσφέρει έναν απλό, φιλικό προς τη βάση δεδομένων αλγόριθμο, ο οποίος δίνει μια πολύ καλή προσεγγιστική απάντηση στο πρόβλημα αναζήτησης του πλησιέστερου γείτονα.

Ο αλγόριθμος είναι εξαιρετικά αποδοτικός, καθώς συχνά αρκεί να ερευνήσει μέχρι το 5% των δεδομένων για να έχει αποτελέσματα με πολύ υψηλή ποιότητα. Η μέθοδος αυτή έχει μεγάλο εννοιολογικό ενδιαφέρον, όχι μόνο ως προσέγγιση του πλησιέστερου γείτονα. Τα αποτελέσματά επίσης δείχνουν ότι η median rank aggregation είναι μία αποδοτική και χρήσιμη μορφή συνάθροισης διατάξεων.

Κεφάλαιο 5

Συμπεράσματα

5.1 Συμπερασματικά σχόλια

Στην εργασία αυτή μελετήσαμε τρεις μεθόδους για την αναζήτηση πλησιέστερων γειτόνων. Οι τρεις αυτές μέθοδοι μας παρουσίασαν το πρόβλημα όχι από την πλευρά της ομοιότητας και της ταξινόμησης, αλλά από την πλευρά της αναζήτησης των πλησιέστερων γειτόνων. Η προσέγγιση αυτή λαμβάνει υπόψιν ότι η απεικόνιση των αντικειμένων με διάνυσμα είναι συχνά ευρετική σε πολλές εφαρμογές και αρκεί μόνο να βρεθεί ένα σημείο της βάσης δεδομένων το οποίο να είναι προσαρμόστηκα ο πλησιέστερος γείτονας.

Μελετήσαμε την αναζήτηση πλησιέστερων γειτόνων σε πολλές διαστάσεις. Κάποιες μελέτες υποστηρίζουν ότι οι αναζητήσεις πλησιέστερων γειτόνων σε πολλές διαστάσεις δεν έχουν νόημα. Όμως, υπάρχουν και στοιχεία που δείχνουν πως αυτός ο ισχυρισμός βασίζεται σε περιοριστικές υποθέσεις. Μια αναζήτηση έχει νόημα όταν το σημείο ερώτησης βρίσκεται πολύ πιο κοντά στον πλησιέστερο γείτονα του, από ό,τι στα περισσότερα σημεία δεδομένων. Αυτό ισχύει σε πολλές εφαρμογές με δεδομένα πολλών διαστάσεων.

Και στις τρεις μεθόδους που μελετήσαμε διαπιστώσαμε σημεία που τις διαφοροποιούν και άλλα που τις ταυτίζουν. Τα σημεία που τις διαφοροποιούν είναι είτε στην ταχύτητα απόδοσης του αλγορίθμου, είτε την ορθότητα των αποτελεσμά-

των. Αξίζει εδώ να αναφέρουμε ότι η iDistance έχει σχετικά το μικρότερο ποσοστό λάθους, ενώ ο αλγόριθμος της MedRank είναι εξερευνητικά αποδοτικός, καθώς συχνά αρκεί να εξερευνήσει το 5% των δεδομένων για να έχει αποτελέσματα με πολύ υψηλή ποιότητα.Κοινό σημείο και των τριών είναι ότι μπορούν να εφαρμοστούν αποτελεσματικά σε ένα εμπορικό σύστημα, κάτι που άλλωστε αποτελούσε και σχεδιαστικό στόχο και για τους τρεις αλγορίθμους.

Βιβλιογραφία

1. Berchtold, S.,Ertl, B.,Keim, D.,Kriegel, H.-P., and Seidl, T. 1998b. Fast nearest neighbor search in high-dimensional space. In Proceedings of the International Conference on Data Engineering.190–218.

2. Ciaccia, P., Patella, M., and Zezula, P. 1997. M-trees: An efficient access method for similarity search in metric space. In Proceedings of the International Conference on Very Large Data Bases. 386–435.

3. Datar, M., Immorlica, N., Indyk, P., and Mirrokni, V. S. 2004. Locality-sensitive hashing scheme based on p-stable distributions. In SoCG. 253–262.

4. P. Diaconis and R. Graham. Spearman's footrule as ameasure of disarray. Journal of the Royal Statistical Society,Series B, 39(2):262–268, 1977.

5. R. Fagin. Combining fuzzy information from multiplesystems. Journal of Computer and System Sciences, 58:83–99, 1999.

6. R. Fagin, A. Lotem, and M. Naor. Optimal aggregation algorithms for middleware. In Proceedings of the 20th ACM Symposium on Principles of Database Systems, pages 102–113, 2001.

7. Fagin, R., Kumar, R., and Sivakumar, D. 2003. Efficient similarity search and classification via rank aggregation. In SIGMOD. 301–312.

8. Filho, R. F. S., Traina, A., and Faloutsos, C. 2001. Similarity search without tears: The omnifamily of all-purpose access methods. In Proceedings of the International Conference on Data Engineering. 613–630.

9. Gaede, V. and Gunther, O. 1998. Multidimensional access methods. ACM Computing Surveys 30, 2, 170–231.

10. Gionis, A., Indyk, P., and Motwani, R. 1999. Similarity search in high dimensions via hashing. In VLDB. 518–529.

11. Goldstein, J. and Ramakrishnan, R. 2000. Contrast plots and p-sphere trees: space vs. time in nearest neighbor searches. In Proceedings of the International Conference on Very Large Databases. 410–440.

12. Har-Peled, S. 2001. A replacement for voronoi diagrams of near linear size. In FOCS. 94–103.

13. Houle, M. E. and Sakuma, J. 2005. Fast approximate similarity search in extremely highdimensional data sets. In ICDE. 609–630.

14. Indyk, P. and Motwani, R. 1998. Approximate nearest neighbors: Towards removing the curse of dimensionality. In STOC. 604–613.

15. Jagadish, Tan, Yu, and Zhang, R. 2004. iDistance: An adaptive B+-tree based indexing method for nearest neighbor search. Tech. Rep. www.comp.nus.edu.sg/◎ooibc,National University of Singapore.

16. Lv, Q., Josephson, W., Wang, Z., Charikar, M., and Li, K. 2007. Multi-probe lsh: Efficient indexing for high-dimensional similarity search. In VLDB. 940–961.

17. J. Kleinberg. Two algorithms for nearest-neighbor search in high dimensions. In Proceedings of the 27th Annual ACM Symposium on Theory of Computing, pages 599–608, 1997.

18. Ramakrishnan, R. and Gehrke, J. 2000. Database Management Systems. McGraw-Hill.

19. Sakurai, Y., Yoshikawa, M., and Uemura, S. 2000. The a-tree: An index structure for highdimensional spaces using relative approximation. In Proceedings of the International Conference on Very Large Data Bases. 516–526.

20. Weber, R., Schek, H., and Blott, S. 1998. A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces. In Proceedings of the International Conference on Very Large Data Bases. 184–205.

21. Yu, Tan and Jagadish, H. 2001. Indexing the distance: an efficient method to knn processing. In Proceedings of the International Conference on Very Large Data Bases. 411–430.