ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ
Τμημα Πληροφορικης και Τηλεπικοινωνιων

Γραφοθεωρητική αναζήτηση πλησιέστερων γειτόνων στη C/C++

Χατζησπύρου Μιχαήλ – sdi: 1115202000212
Καζάκος Παναγιώτης – sdi: 1115201900067

Κ23Γ: Ανάπτυξη Λογισμικού για Αλγοριθμικά Προβλήματα – Χειμερινό ΄23
Ημερ. Ανακοίνωσης: 7/11
Ημερ. Υποβολής: 1/12 – Ώρα 23:59

Περιεχόμενα

1. Γενική Περιγραφή

Κύριος στόχος της εργασίας είναι η ανάπτυξη λογισμικού για την αποτελεσματική αναζήτηση όμοιων διανυσμάτων σε χώρους μεγάλων διαστάσεων. Πιο συγκεκριμένα, το λογισμικό που αναπτύχθηκε βασίζεται σε προηγμένους αλγόριθμους που λαμβάνουν τα θεμέλια τους από την θεωρία γράφων, όπως η «Αναζήτηση Πλησιέστερου Γείτονα Γραφήματος» (Graph Nearest Neighbor Search - GNNS) και το «Μονοτονικό Γράφημα Σχετικής Γειτονιάς» (Monotonic Relative-Neighborhood Graph - MRNG). Το λογισμικό έχει σχεδιαστεί για να είναι συμβατό με το σύνολο δεδομένων MNIST, το οποίο αποτελείται από χειρόγραφες ψηφιακές εικόνες. Επιπλέον, παρέχεται η δυνατότητα προσέγγισης των αποστάσεων μεταξύ των διανυσμάτων, προσφέροντας μια ισορροπία μεταξύ ακρίβειας και υπολογιστικής απόδοσης.

2. Οδηγίες Μεταγλώττισης

Έχει δημιουργηθεί ένα Makefile για την διευκόλυνση της μεταγλώττισης και την εκτέλεση των προγραμμάτων με κάποιες προκαθορισμένες παραμέτρους. Η μεταγλώττιση όλων των αρχείων και γίνεται με την εντολή μακε. Χρησιμοποιείται η τεχνική του separate compilation. Επίσης δίνεται η δυνατότητα στον χρήστη να μεταγλωττίσει το πρόγραμμα με την εντολή:

- make graph – για την μεταγλώττιση του προγράμματος Graph

3. Οδηγίες Χρήσης του Προγράμματος

Αφού ο χρήστης δημιουργήσει το εκτελέσιμο αρχείο με όποια μέθοδο επιθυμεί μπορεί να το εκτελέσει με την βοήθεια του Makefile χρησιμοποιώντας την εξής εντολή:

- make run-graph – για την εκτέλεση του προγράμματος Graph με τα ορίσματα που δηλώνονται στην μεταβλητή ARGS_GRAPH

Η μεταβλητή ARGS_GRAPH είναι δηλωμένη εντός του Makefile και παίρνει τα κατάλληλα ορίσματα προκειμένου να εκτελεστεί το εκτελέσιμο αρχείο. Επιπλέον δίνεται στον χρήστη η δυνατότητα εκτέλεσης των προγραμμάτων μέσω terminal ως εξής:

- ./bin/graph_search –d <input file> –q <query file> –k <int>? -E <int>? -R <int> ? -N <int>? -l <int, only for Search-on-Graph>? -m <1 for GNNS, 2 for MRNG> -o <output file>

  Default arguments: k = 50, E = 30, R = 1, N = 1, l = 20

Όποια ορίσματα έχουν ερωτηματικό στο τέλος τους είναι προαιρετικά, ο αλγόριθμος θα χρησιμοποιήσει τα default.

Χατζησπύρου Μιχαήλ – sdi: 1115202000212
Καζάκος Παναγιώτης – sdi: 1115201900067

4. Δομή Φακέλων Κώδικα

root

- bin — Εκτελέσιμα αρχεία από main συναρτήσεις του src και tests
- build — Αντικειμενικά αρχεία που δεν έχουν συνδεθεί
- datasets — Σύνολα δεδομένων, εισόδου και αναζήτησης
- modules — Κομμάτια κώδικα οργανωμένα σε επιμέρους φακέλους, κλάσεις και συναρτήσεις που χρησιμοποιούνται στα προγράμματα για καλύτερη οργάνωση του κώδικα και επαναχρησιμοποίησή του.
  - Common — Φάκελος που περιέχει κοινές δομές/υλοποιήσεις για όλα τα προγράμματα
    - BruteForce — Υλοποίηση της εξαντλητικής αναζήτησης
    - FileParser — Υλοποίηση του parser που διαβάζει τα αρχεία εικόνων
    - HashFunction — Υλοποίηση της hash function h
    - ImageDistance — Υλοποίηση γενικευμένης απόστασης
    - Utils — Διάφορα utils που χρησιμοποιούνται από όλους τους αλγορίθμους
  - Cube — Υλοποίηση του κύβου
  - Graphs — Φάκελος που περιέχει τις υλοποιήσεις των γραφοθεωρητικών αλγορίθμων
    - GNNS — Υλοποίηση του GNNS
    - MRNG — Υλοποίηση του MRNG
  - LSH — Υλοποίηση του Lsh
    - HashTable — Υλοποίηση των Hash tables για των αλγόριθμο
- src — Περιέχει την main συνάρτηση για την υλοποίηση των γράφων
- tests — Περιέχει τα scripts που χρησιμοποιήθηκαν για τα tests μαζί με τα αποτελέσματα

4.1. Φάκελος Modules

Περιέχει διάφορες συναρτήσεις και κλάσεις που ομαδοποιούν σημαντικές λειτουργίες των προγραμμάτων. Κάθε φάκελος έχει όνομα που ταιριάζει στα τρία διαφορετικά προγράμματα

εκτός από τα Common modules που χρησιμοποιούνται σε όλα.

4.2. Φάκελος SRC

Περιέχει την main συνάρτηση για το πρόγραμμα των γράφων. Η main συνάρτηση είναι υπεύθυνη να λύσει το ζητούμενο πρόβλημα κάθε φορά με αφηρημένο τρόπο διαβάζοντας τα κατάλληλα αρχεία, συνδυάζοντας τα απαραίτητα modules και εκτυπώνοντας τα αποτελέσματα.

4.3. Σημαντικές Δομές, Κλάσεις και Συναρτήσεις

Image: Αναπαριστά ένα σημείο από ένα MNIST σύνολο δεδομένων αποθηκεύοντας ένα αναγνωριστικό (id) και το διάνυσμα (pixels).

Neighbor: Αναπαριστά έναν γείτονα ενός σημείου. Αποθηκεύει το ίδιο το σημείο (image) αλλά και την απόσταση (distance) από το σημείο για το οποίο είναι γείτονας.

DistanceMetric: Enum τύπος που περιέχει τις Manhattan και Ευκλείδια μετρικές.

ImageDistance: Αποθηκεύει ποιά μετρική θα χρησιμοποιηθεί για το κάθε πρόγραμμα. Η αρχικοποίηση γίνεται μία φορα σε κάθε main και μπορεί να χρησιμοποιηθεί δυναμικά οπουδήποτε έχει κληθεί όταν τρέχει η εκάστοτε main.

CmdArgs: Αναλύει τα ορίσματα κάθε main συνάρτησης και τα αποθηκεύει σε μία εύκολα προσβάσιμη δομή. Για κάθε main υπάρχει ξεχωριστή κλάση (**LshCmdArgs**, **CubeCmdArgs**, **GraphCmdArgs**).

FileParser: Αναλύει τα αρχεία εισόδου και αν ταιριάζουν στη μορφή ενός MNIST συνόλου δεδομένων, τότε αποθηκεύει τα μεταδεδομένα και τις εικόνες στην προσωρινή μνήμη του προγραμμάτος.

Utils: Περιέχει διάφορες σύντομες συναρτήσεις που χρησιμοποιούνται σε όλα τα προγράμματα.

BruteForce: Πραγματοποιεί αναζήτηση K πραγματικών πλησιέστερων γειτόνων ελέγχοντας για ένα σημείο αναζήτησης την απόσταση του με όλα τα σημεία.

5. Γράφοι

5.1. Graph Nearest Neighbor Search (GNNS)

Ο Graph Nearest Neighbor Search (GNNS) με χρήση Locality-Sensitive Hashing (LSH) είναι μια εξελιγμένη αλγοριθμική προσέγγιση που έχει σχεδιαστεί για την αποτελεσματική εύρεση κατά προσέγγιση πλησιέστερων γειτόνων σε χώρους μεγάλων διαστάσεων. Σε αυτό το πλαίσιο, ο GNNS αξιοποιεί τεχνικές LSH για να βελτιώσει τη διαδικασία αναζήτησης μέσα σε μια δομή γραφήματος.

5.1.1. Initialization: Ο γράφος αναπαρίσταται από έναν δυσδιάστατο βέκτορα με την πρώτη διάσταση να υποδεικνύει το σημείο που αναφερόμαστε ενώ η δεύτερη αφορά τους γείτονες του εκάστοτε σημείου. Οι γείτονες κάθε σημείου βρίσκονται με την μέθοδο LSH.

Εδώ πρέπει να σημειωθεί ότι επειδή το LSH αρχικοποιείται με το ίδιο σύνολο με το οποίο εκτελέσουμε και αναζήτηση, για την αρχικοποίηση του γράφου, ο πρώτος γείτονας κάθε σημείου πρέπει να παραληφθεί διότι είναι το ίδιο το σημείο. Η αναζήτηση των γειτόνων είναι αυξημένη κατά ένα σε σχέση με αυτό που δίνει ο χρήστης για τους λόγους που αναφέρθηκαν προηγουμένως.

5.1.2. Search: Η προσεγγιστική αναζήτηση, ξεκινάει αρχικοποιώντας ένα σύνολο γειτόνων με το κενό σύνολο. Στην συνέχεια για κάθε επανεκκίνηση παίρνουμε ένα τυχαίο σημείο στο σύνολο δεδομένων εκπαίδευσης ως «οδηγό» στοιχείο. Ύστερα, σε κάθε επανάληψη των greedy steps (T) προσθέτουμε τις επεκτάσεις, δηλαδή του γείτονες, του σημείο «οδηγού» στον γράφο. Για νέο σημείο «οδηγό» διαλέγουμε τον γείτονα που είναι πιο κοντά στο σημείο ερωτήματος. Επιπλέον, χρησιμοποιώντας σύνολο καταφέρνουμε να έχουμε ανά πάσα στιγμή ταξινομημένα τα σημεία, σε αύξουσα σειρά με βάση την απόσταση τους. Στο τέλος, επιστρέφουμε από το σύνολο τον αριθμό των ζητούμενων γειτόνων.

5.2. Monotonic Relative-Neighborhood Graph (MRNG)

Ο αλγόριθμος Monotonic Relative-Neighborhood Graph (MRNG) χρησιμοποιείται για να κατασκευαστεί ένας γράφος που δείχνει την εγγύτητα των διάφορων σημείων του συνόλου ακολουθώντας κατάλληλα τις ακμές του. Η ιδαιτερότητά του είναι ότι εφαρμόζει μια συνθήκη μονοτονίας για να εξασφαλίσει ένα αυξανόμενο μονοπάτι απόστασης των κόμβων.

5.2.1. Initialization: Ο γράφος περιέχει τους γείτονες για κάθε εικόνα του συνόλου δεδομένων οι οποίοι υπολογίζονται με brute force μεθοδο. Για κάθε εικόνα υπολογίζεται ένας πίνακας ταξινομημένος με βάση την απόσταση των υπόλοιπων εικόνων από αυτή. Αρχικά στη λίστα των γειτόνων αποθηκεύουμε τα σημεία που έχουν την ελάχιστη απόσταση. Στην συνέχεια, προσθέτουμε κι άλλους γείτονες με βάση τον ταξινομημένο πίνακα αλλα μόνο εάν ισχύει η συνθήκη μονοτονίας. Η συνθήκη αυτή είναι αληθής αν η απόσταση του επόμενου πιθανού γείτονα από την εικόνα που εξετάζουμε είναι μικρότερη ή ίση από την απόστασή του με τουλάχιστον έναν από τους γείτονες που έχουν βρεθεί μέχρι στιγμής ή από κάποια απόσταση μετάξύ των γειτόνων και της εικόνας που εξετάζουμε.

5.2.2. Search: Για την αναζήτηση χρησιμοποιώντας τον γράφο MRNG ακολουθείται ο αλγόριθμος Generic Search On Graph ο οποίος λαμβάνει τον αριθμό αριθμό των μέγιστων υποψήφιων μαζί με τον αριθμό των πλησιέστερων γειτόνων που πρέπει να επιστρέψει και επιστρέφει προσεγγιστικά τους γείτονες μιας query εικόνας. Ξεκινώντας από την εικόνα που που βρίσκεται πιο κοντά στο μέσο όρο, έχοντας υπολογίσει μέσο όρο για κάθε διάσταση, εξετάζουμε τους γείτονες που προκύπτουν από τις εξωτερικές ακμές του κόμβου μέχρι να φτάσουμε τον αριθμό των μέγιστων υποψηφίων. Γίνεται χρήση ενός set έτσι ώστε να επισκεφθούν πρώτα οι κοντινότεροι γείτονες. Ο πίνακας των γειτόνων επιστρέφεται ταξινομημένος.

6. Υπερπαράμετροι

Οι υπερπαράμετροι είναι ρυθμίσεις εξωτερικών παραμέτρων για ένα μοντέλο που δεν μαθαίνονται από τα δεδομένα αλλά έχουν οριστεί πριν από τη διαδικασία εκπαίδευσης. Διέπουν τη συνολική συμπεριφορά ενός αλγορίθμου μηχανικής μάθησης και διαδραματίζουν κρίσιμο ρόλο στον προσδιορισμό της απόδοσης και των δυνατοτήτων γενίκευσης ενός μοντέλου. Το tuning των υπερπαραμέτρων περιλαμβάνει την εύρεση των βέλτιστων τιμών για αυτές

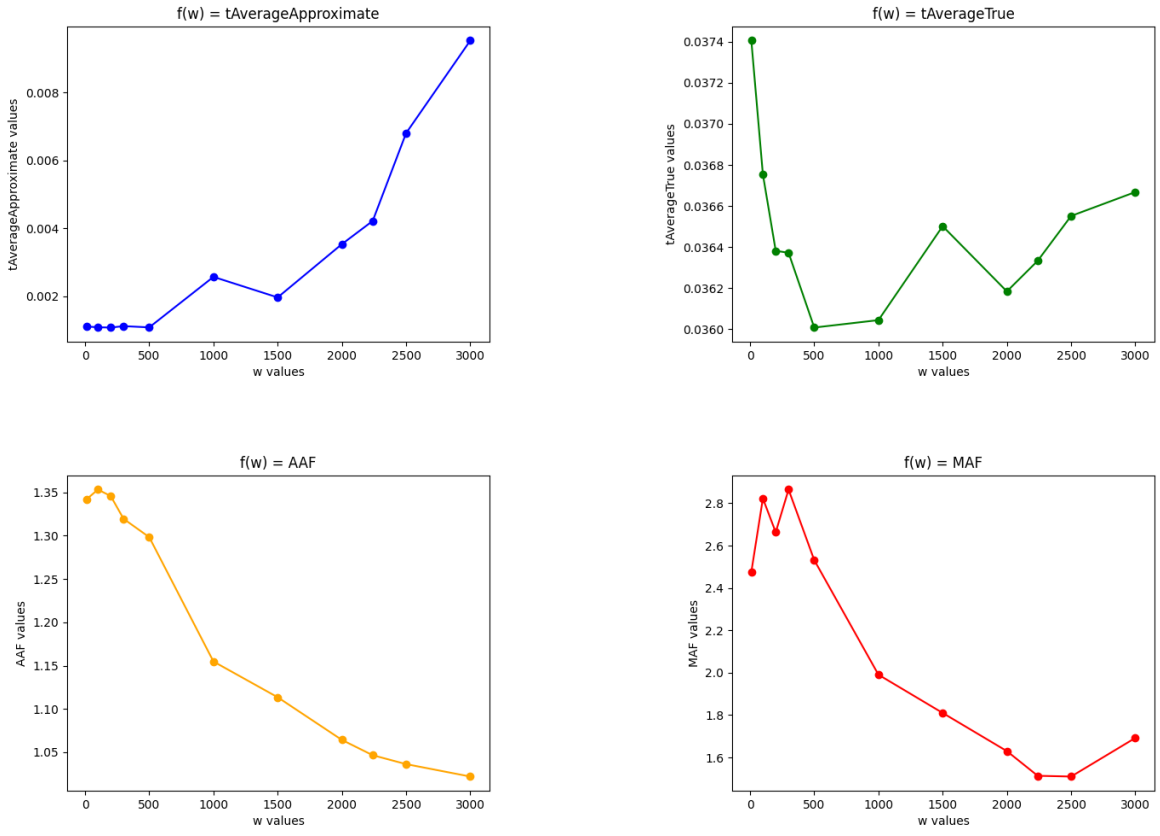
τις ρυθμίσεις για την επίτευξη της καλύτερης απόδοσης μοντέλου σε μια δεδομένη εργασία. Στην δική μας περίπτωση χρησιμοποιήθηκαν οι εξής μετρικές για την παρατήρηση αλλά και εύρεση αξιόλων υπερπαραμέτρων:

- tAverageApproximate – Μέσος χρόνος προσεγγιστικής αναζήτησης
- tAverageTrue – Μέσος χρόνος πραγματικής αναζήτησης
- AAF – Average Approximation Factor
- MAF – Maximum Approximation Factor

Επιπλέον, για τους πειραματισμούς με τους αλγορίθμους LSH, CUBE, GNNS χρησιμοποιήθηκε σύνολο εκπαίδευσης με 60000 στοιχεία ενώ για τον MRNG λόγω της αργής αρχικοποιήσης καταστήθηκε εφικτό να κάνουμε πειράματα σε σύνολο δεδομένων με 20000 στοιχεία. Τέλος, όλοι οι αλγόριθμοι εκτελούσαν 1000 queries ψάχνοντας τους 3 κοντινότερους γείτονες.

6.1. LSH

Τα πειράματα ΛΣΗ διεξήχθησαν χρησιμοποιώντας μια διαμόρφωση που χρησιμοποιούσε τέσσερις συναρτήσεις κατακερματισμού και πέντε πίνακες κατακερματισμού. Αυτή η ρύθμιση επέτρεψε μια αποτελεσματική και λεπτή εξερεύνηση του κατακερματισμού ευαίσθητου στην τοποθεσία, παρέχοντας ένα ισχυρό πλαίσιο για την ανάλυση και την αξιολόγηση των πειραματικών αποτελεσμάτων.



Σχήμα 1: LSH Plot Results

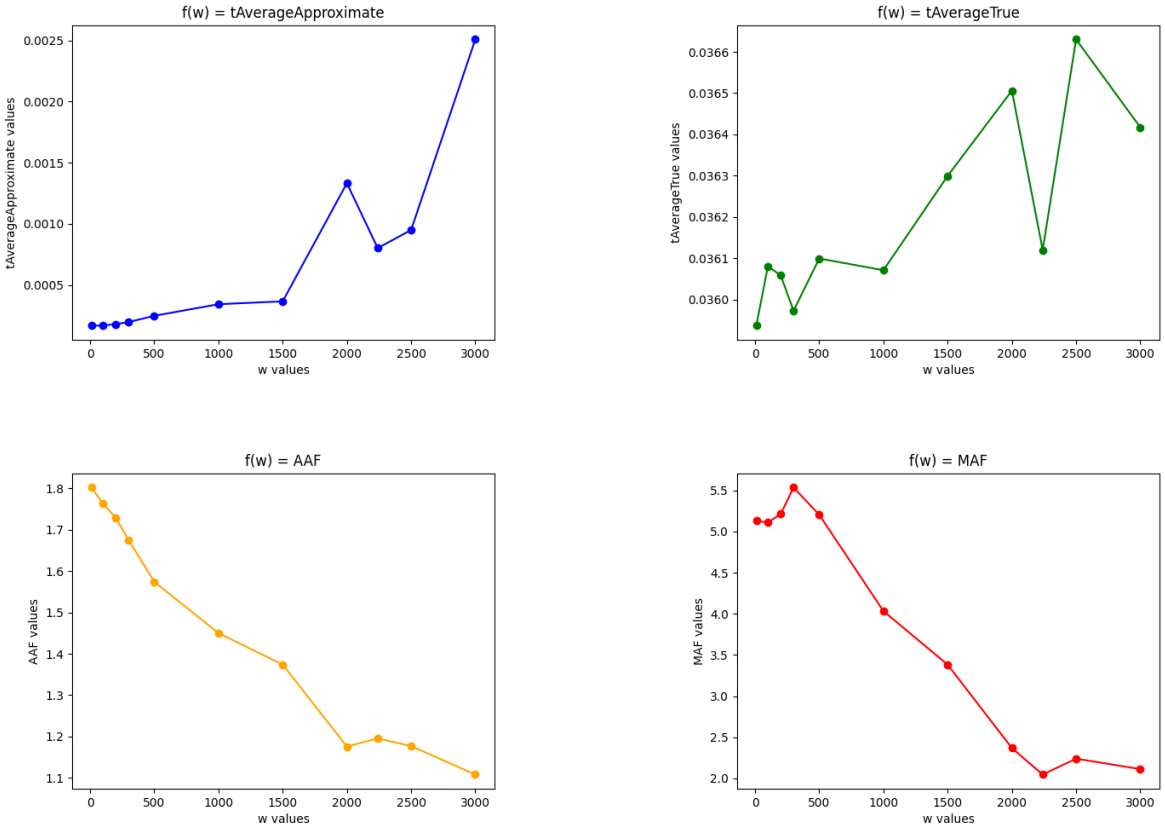
Πίνακας 1: LSH Expriments Table

w	tAverageApproximate	tAverageTrue	AAF	MAF
10	0.001103	0.037406	1.34182	2.47560
100	0.001080	0.036754	1.35334	2.82273
200	0.001071	0.036382	1.34583	2.66324
300	0.001113	0.036372	1.31953	2.86478
500	0.001077	0.036008	1.29834	2.53172
1000	0.002565	0.036044	1.15473	1.99075
1500	0.001960	0.036502	1.11347	1.81031
2000	0.003530	0.036184	1.06421	1.63019
2240	0.004214	0.036333	1.04653	1.51368
2500	0.006801	0.036552	1.03625	1.51008
3000	0.009531	0.036668	1.02189	1.69220

Αναλύοντας τα διαγράμματα (1) και τα δεδομένα του πίνακα (1), είναι προφανές ότι η υπερπαράμετρος πλάτους (w) επηρεάζει σημαντικά τόσο τον μέσο χρόνο προσέγγισης όσο και την ακρίβεια του μοντέλου. Αξίζει να σημειωθεί ότι καθώς αυξάνεται το πλάτος, παρατηρείται βελτίωση του μέσου παράγοντα προσέγγισης (AAF). Ωστόσο, η προσεκτική παρατήρηση αποκαλύπτει σποραδικές αιχμές στις τιμές μέγιστου παράγοντα προσέγγισης (MAF). Επιπλέον, ένα πλάτος 3000 δείχνει σχεδόν τετραπλάσια αύξηση στην ταχύτητα προσέγγισης αλλά, είναι σημαντικό να ληφθεί υπόψη ότι η υπερβολική αύξηση του πλάτους μπορεί να οδηγήσει σε μεγαλύτερους χρόνους προσέγγισης από τις πραγματικές τιμές. Αυτό θα μπορούσε ενδεχομένως να καταστήσει τον αλγόριθμο αναποτελεσματικό, καθώς τα κέρδη στην ταχύτητα μπορεί να αντισταθμιστούν από τις φθίνουσες αποδόσεις και τις περιστασιακές αιχμές στην ακρίβεια.

6.2. CUBE

Τα πειράματα του κύβου, που διεξήχθη με μια αυστηρή εξερεύνηση 14 διαστάσεων, περιελάμβανε την εξέταση 6000 υποψήφιων σημείων και χρησιμοποίησε 15 κορυφές για να αξιολογήσει διεξοδικά την απόδοσή του.



Σχήμα 2: CUBE Plot Results

Πίνακας 2: CUBE Experiments Table

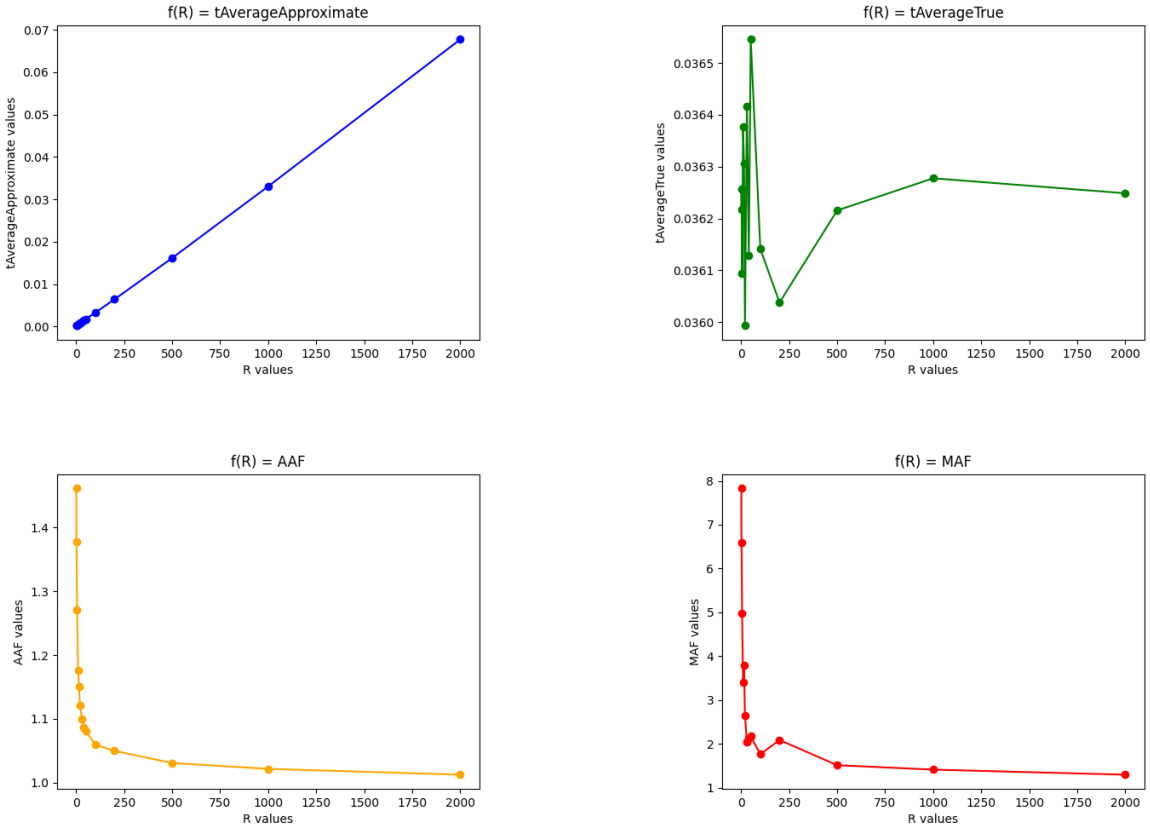
w	tAverageApproximate	tAverageTrue	AAF	MAF
10	0.000168634	0.0359368	1.80225	5.13259
100	0.000170431	0.0360809	1.76384	5.10976
200	0.000181505	0.0360586	1.72922	5.21417
300	0.000198396	0.0359730	1.67565	5.53558
500	0.000249213	0.0360995	1.57420	5.20435
1000	0.000344184	0.0360712	1.45005	4.03388
1500	0.000367387	0.0362996	1.37388	3.38219
2000	0.001332860	0.0365061	1.17557	2.36999
2240	0.000802734	0.0361202	1.19511	2.04738
2500	0.000950239	0.0366302	1.17678	2.24048
3000	0.002508000	0.0364173	1.10772	2.11349

Όπως φαίνεται τόσο από τα διαγράμματα (2) όσο και από τον πειραματικό πίνακα (2), είναι σαφές ότι η υπερπαράμετρος "w" επηρεάζει σημαντικά τον μέσο χρόνο προσέγγισης. Η σχέση μεταξύ "w" και του μέσου χρόνου προσέγγισης είναι εμφανής, δείχνοντας την ευαισθησία της απόδοσης του αλγορίθμου στη συγκεκριμένη υπερπαράμετρο. Είναι ενδιαφέρον ότι το γράφημα AAF (Average Approximation Factor) παρουσιάζει ένα σχεδόν αντίστροφο μοτίβο σε σύγκριση με τον μέσο χρόνο προσέγγισης. Αυτό υποδηλώνει ότι οι προσαρμογές στην υπερπαράμετρο "w" οδηγούν σε βελτιώσεις στην ακρίβεια. Το διάγραμμα AAF

υποδηλώνει ότι οι αλλαγές στο "w" επηρεάζουν θετικά την ικανότητα του αλγορίθμου να παρέχει πιο ακριβείς προσεγγίσεις, δείχνοντας τη λεπτή ισορροπία μεταξύ ακρίβειας και χρόνου υπολογισμού. Το μέσο διάγραμμα πραγματικού χρόνου, από την άλλη πλευρά, μπορεί να παραλειφθεί στην ανάλυσή μας. Δεν πάρχει συσχέτιση μεταξύ της υπερπαράμετρου "w" και του αληθινού χρόνου, υποδεικνύοντας ότι οι αλλαγές στο ¨ω' δεν επηρεάζουν τον πραγματικό χρόνο του αλγορίθμου. Τέλος, παρά την επίτευξη αξιοπρεπών συντελεστών προσέγγισης, ο MAF (Maximum Approximation Factor) παραμένει υψηλός. Αυτό υποδηλώνει ότι, ακόμη και με αποδεκτές προσεγγίσεις, υπάρχουν περιπτώσεις όπου ο αλγόριθμος δυσκολεύεται να προβλέψει ορισμένους γείτονες με ακρίβεια.

6.3. GNNS

Ο πειραματισμός με το GNNS αποδείχθηκε μια σχετικά απλή διαδικασία, κυρίως λόγω των κανονικών χρόνων αρχικοποίησής τους. Μέσω της εμπειρικής εξερεύνησης, αποκτήσαμε γνώσεις που δείχνουν ότι οι πιο κρίσιμες υπερπαράμετροι ήταν οι επανεκκινήσεις (R). Κατά συνέπεια, επικεντρώσαμε τις πειραματικές μας προσπάθειες σε αυτή τη συγκεκριμένη παράμετρο, περιορίζοντας τις δοκιμές μας στις διακυμάνσεις των τιμών της.



Σχήμα 3: GNNS Plot Charts

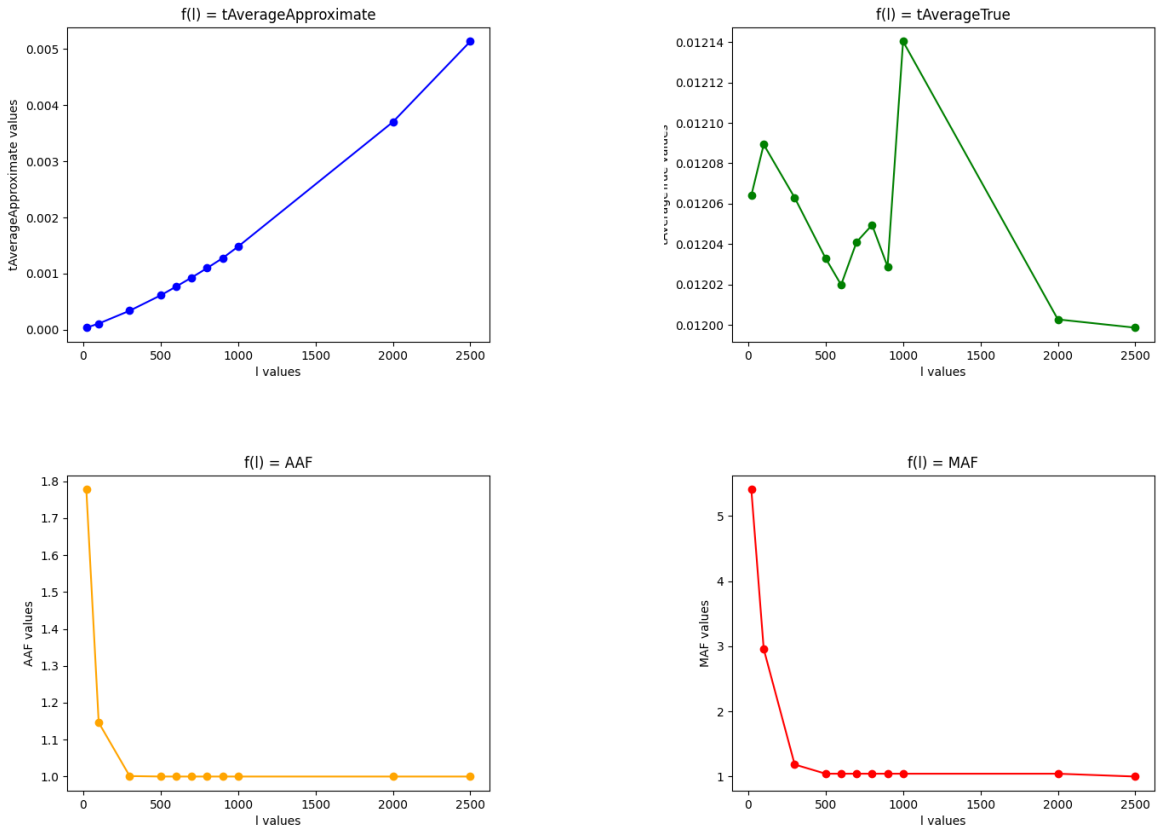
Πίνακας 3: GNNS Experiments Table

R	tAverageApproximate	tAverageTrue	AAF	MAF
1	0.000181	0.036218	1.46134	7.82793
2	0.000216	0.036257	1.37709	6.58248
5	0.000336	0.036093	1.27030	4.97808
10	0.000497	0.036376	1.17615	3.41054
15	0.000658	0.036305	1.15037	3.7879
20	0.000808	0.035993	1.12189	2.64309
30	0.001118	0.036416	1.10020	2.03744
40	0.001435	0.036129	1.08627	2.11885
50	0.001743	0.036546	1.08067	2.17398
100	0.003275	0.036141	1.05991	1.76417
200	0.006408	0.036038	1.04991	2.08777
500	0.016152	0.036215	1.03096	1.51368
1000	0.033054	0.036278	1.02177	1.41264
2000	0.067729	0.036249	1.01264	1.29703

Σύμφωνα με τα διαγράμματα (3) και τον πίνακα (3), φαίνεται να υπάρχει μια γραμμική σχέση μεταξύ της αύξησης της παραμέτρου R και του μέσου χρόνου προσέγγισης. Ωστόσο, είναι αξιοσημείωτο ότι η ακρίβεια συγκλίνει στο 1.1 αρκετά γρήγορα. Όπως αναφέρθηκε προηγουμένως, το γραμμικό διάγραμμα με τις τρέχουσες υπερπαραμέτρους και τον πραγματικό μέσο χρόνο φαίνεται να στερείται ουσιαστικής σχέσης. Συμπερασματικά, θα πρέπει να ληφθούν υπόψη δύο προσεκτικές παρατηρήσεις. Πρώτον, η αύξηση της τιμής του R, όπως αποδείχθηκε στο τελευταίο πείραμα (2000), έχει ως αποτέλεσμα ο κατά προσέγγιση χρόνος να διπλασιάζεται σχεδόν σε σύγκριση με τον πραγματικό χρόνο και δεν υπάρχει καμία εγγύηση ότι θα αναγνωρίσει με επιτυχία τους αληθινούς γείτονες. Δεύτερον, παρά το γεγονός ότι έχει έναν αρκετά αξιοπρεπή AAF, το MAF εμφανίζει υψηλές αιχμές στις τιμές του, καθιστώντας τον μια αναποτελεσματική μέτρηση για αξιολόγηση.

6.4. MRNG

Καθώς εμβαθύνουμε στον αλγόριθμο MRNG, έγινε προφανές ότι παρά την απλότητά του με μία μόνο υπερπαράμετρο, το 'l', εμφανίζεται ένα αξιοσημείωτο μειονέκτημα με τη μορφή αργών χρόνων αρχικοποίησης κατά τη διάρκεια του πειραματισμού.



Σχήμα 4: MRNG Plot Charts

Πίνακας 4: MRNG Experiments Table

l	tAverageApproximate	tAverageTrue	AAF	MAF
20	3.45975ε-05	0.0120641	1.77739	5.40391
100	0.000108033	0.0120893	1.14613	2.95789
300	0.000337579	0.0120630	1.00130	1.18480
500	0.000611109	0.0120330	1.00009	1.04364
600	0.000767632	0.0120199	1.00005	1.04364
700	0.000927916	0.0120411	1.00005	1.04364
800	0.001097120	0.0120495	1.00004	1.04364
900	0.001274050	0.0120289	1.00003	1.04364
1000	0.001479860	0.0121403	1.00002	1.04364
2000	0.003701590	0.0120028	1.00002	1.04364
2500	0.005138200	0.0119987	1	1

Σύμφωνα με τα διαγράμματα (4) και τον πίνακα (4) παρατηρούμε ότι με μια αύξηση στην παράμετρο 'l', ο κατά προσέγγιση μέσος χρόνος θα αυξανόταν επίσης. Ωστόσο, παρατηρήσαμε πιο ακριβή αποτελέσματα. Συγκεκριμένα, the Maximum Approximation Factor (MAF) πέφτει, αν και όχι τόσο γρήγορα όσο the Average Approximation Factor (AAF). Παραδόξως, μετά το τέταρτο πείραμα, το MAF σταθεροποιείται, υποδεικνύοντας ένα σταθερό επίπεδο ακρίβειας. Είναι σημαντικό να αναγνωρίσουμε ότι, σε αντίθεση με το AAF, η υπερπαράμετρος δεν επηρεάζει καθόλου τον υπολογισμό του πραγματικού χρόνου. Επομένως, το γράφημα εξυπηρετεί κυρίως αισθητικούς σκοπούς. Τέλος, είναι προφανές ότι η ουσιαστική

αύξηση της παραμέτρου 'l' θα είχε ως αποτέλεσμα βραδύτερο κατά προσέγγιση χρόνο σε σύγκριση με τον πραγματικό χρόνο. Αυτό είναι ιδιαίτερα αξιοσημείωτο όταν εξετάζουμε τον αργό χρόνο αρχικοποίησης του αλγορίθμου. Η σωρευτική επίδραση αυτών των παραγόντων θα μπορούσε να καταστήσει τον αλγόριθμο εξαιρετικά αναποτελεσματικό υπό εκτεταμένες αυξήσεις παραμέτρων.

7. Αξιολόγηση

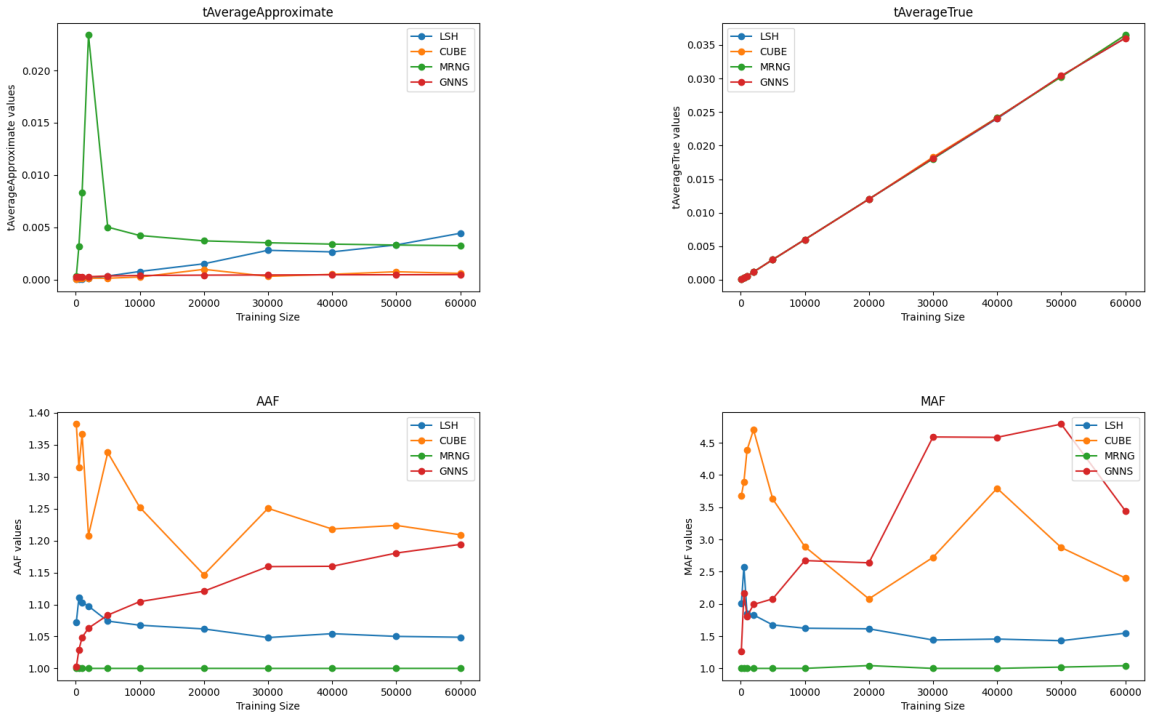
Μετά τη διεξαγωγή μιας ενδελεχούς ανάλυσης των υπερπαραμέτρων και τη λεπτομερή ρύθμιση των αλγορίθμων μας, το επόμενο κρίσιμο βήμα στην έρευνά μας είναι η αξιολόγηση της απόδοσής τους σε διάφορα σύνολα εκπαίδευσης. Αυτή η διαδικασία στοχεύει να παρέχει πληροφορίες σχετικά με το πόσο καλά οι αλγόριθμοι γενικεύονται σε διαφορετικούς όγκους δεδομένων και εάν η αποτελεσματικότητά τους κλιμακώνεται με αυξημένα δεδομένα εκπαίδευσης. Για την αξιολόγηση των αλγορίθμων χρησιμοποιήθηκαν οι ίδιες μετρικές που χρησιμοποιήθηκαν και στην ενότητα των υπερπαραμέτων:

- tAverageApproximate – Μέσος χρόνος προσεγγιστικής αναζήτησης
- tAverageTrue – Μέσος χρόνος πραγματικής αναζήτησης
- AAF – Average Approximation Factor
- MAF – Maximum Approximation Factor

Για τα πειράματα αξιολόγησης χρησιμοποιήθηκαν τα παρακάτω σύνολα δεδομένων:

100 500 1000 2000 5000 10000 20000 30000 40000 50000 60000

Και σε αυτά τα πειράματα οι αλγόριθμοι εκτέλεσαν 1000 queries ψάχνοντας τους 3 κοντινότερους γείτονες.



Σχήμα 5: Evaluation Plot Results

Πίνακας 5: Evaluation Experiments Table

Training Size=100	tAverageApproximate	tAverageTrue	AAF	MAF
LSH	4.06938e-05	5.8738e-05	1.07258	2.00541
CUBE	0.000102631	5.85538e-05	1.38238	3.67674
GNNS	0.000281131	5.83409e-05	1.00223	1.26557
MRNG	0.000376309	5.88235e-05	1	1

Training Size=500	tAverageApproximate	tAverageTrue	AAF	MAF
LSH	5.69329e-05	0.000287464	1.11036	2.57459
CUBE	0.000108226	0.000289334	1.31401	3.89483
GNNS	0.000273439	0.000287663	1.02902	2.16685
MRNG	0.00318457	0.000288363	1	1

Training Size=1000	tAverageApproximate	tAverageTrue	AAF	MAF
LSH	8.17232e-05	0.000574483	1.10323	1.84699
CUBE	0.000113303	0.000578318	1.36691	4.3905
GNNS	0.000277211	0.000574292	1.04852	1.80578
MRNG	0.00835453	0.000577587	1	1

Training Size=2000	tAverageApproximate	tAverageTrue	AAF	MAF
LSH	0.000128265	0.00115978	1.09749	1.82979
CUBE	0.000153441	0.00116089	1.20813	4.70234
GNNS	0.000295287	0.00116232	1.06312	1.99178
MRNG	0.0233754	0.0011797	1	1

Training Size=5000	tAverageApproximate	tAverageTrue	AAF	MAF
LSH	0.000363887	0.00301072	1.07404	1.67456
CUBE	0.00015144	0.00301414	1.33792	3.62819
GNNS	0.000372294	0.00300698	1.08338	2.07871
MRNG	0.00503224	0.0029895	1	1

Training Size=10000	tAverageApproximate	tAverageTrue	AAF	MAF
LSH	0.00079711	0.00600404	1.06759	1.62348
CUBE	0.000269949	0.00599462	1.25208	2.88964
GNNS	0.0004133	0.00599282	1.10451	2.67304
MRNG	0.00423272	0.00599832	1	1

Training Size=20000	tAverageApproximate	tAverageTrue	AAF	MAF
LSH	0.0015317	0.0120182	1.06176	1.61418
CUBE	0.00100175	0.0120162	1.14662	2.07844
GNNS	0.000446279	0.0120126	1.12081	2.63874
MRNG	0.00372303	0.0120354	1.00002	1.04364

Training Size=30000	tAverageApproximate	tAverageTrue	AAF	MAF
LSH	0.00282087	0.0180431	1.04827	1.44067
CUBE	0.000333802	0.0182794	1.25051	2.72259
GNNS	0.000461711	0.0181259	1.15928	4.59107
MRNG	0.00354003	0.0180301	1	1

Training Size=40000	tAverageApproximate	tAverageTrue	AAF	MAF
LSH	0.00266781	0.0240172	1.05436	1.45542
CUBE	0.000527932	0.024175	1.21815	3.79447
GNNS	0.000479647	0.0240971	1.15982	4.58524
MRNG	0.0034111	0.0241498	1	1

Training Size=50000	tAverageApproximate	tAverageTrue	AAF	MAF
LSH	0.0033333	0.0303707	1.05011	1.42991
CUBE	0.000772591	0.0302645	1.22383	2.87553
GNNS	0.000484446	0.0303996	1.18051	4.79063
MRNG	0.00332195	0.0302453	1.00001	1.02014

Training Size=60000	tAverageApproximate	tAverageTrue	AAF	MAF
LSH	0.00444907	0.0360411	1.04879	1.54656
CUBE	0.000615765	0.0364507	1.2091	2.40257
GNNS	0.000496924	0.0360072	1.19411	3.44551
MRNG	0.00326701	0.0364649	1.00002	1.04169

Για το σύνολο εκπαίδευσης των 100 στοιχείων, παρατηρείται ότι μόνο το LSH είναι ελαφρώς ταχύτερο από τον πραγματικό χρόνο, αλλά λόγω της έλλειψης 100% ακρίβειας, κανένας από τους αλγόριθμους δεν φαίνεται να αξίζει να χρησιμοποιηθεί για αυτό το σύνολο δεδομένων. Προχωρώντας σε μέγεθος δεδομένων 500, ο πιο αργός αλγόριθμος είναι ο MRNG, ακόμη πιο αργός από τη μέθοδο brute-force. Το GNNS έχει εξαιρετικό AAF 1.02. Το CUBE είναι πιο αργό από το LSH αλλά και πολύ λιγότερο ακριβές. Ως εκ τούτου, το LSH ξεχωρίζει ως η καλύτερη επιλογή λόγω της εξαιρετικής ταχύτητας και ακρίβειάς του. Για σύνολο δεδομένων 1000, το LSH παραμένει ο ταχύτερος αλγόριθμος με την καλύτερη ακρίβεια. Ο CUBE είναι σχεδόν δύο φορές πιο γρήγορος από το GNNS αλλά λιγότερο ακριβής. Το MRNG βρίσκεται και πάλι στην τελευταία θέση, όντας σχεδόν 14.5 φορές πιο αργό από τη μέθοδο brute-force, καθιστώντας το αναποτελεσματικό. Στο σύνολο δεδομένων του 2000, το MRNG παραμένει στην τελευταία θέση αλλά με την καλύτερη ακρίβεια. Το LSH παραμένει το ταχύτερο, με τον CUBE ελαφρώς πιο αργό στη δεύτερη θέση. Το GNNS, στην τρίτη θέση, είναι σχεδόν 2 φορές πιο αργό από τον κύβο αλλά πιο ακριβές. Το LSH παραμένει η ανώτερη επιλογή με μόνο 0.03% μικρότερη ακρίβεια από το GNNS. Για το σετ εκπαίδευσης 5000, ο CUBE γίνεται ο ταχύτερος αλγόριθμος αλλά με τη μικρότερη ακρίβεια. Το LSH και το GNNS έχουν αμελητέα διαφορά χρόνου και ακρίβειας, καθιστώντας τα και τα δύο εξαιρετικές επιλογές. Το MRNG, παρά το γεγονός ότι παρέχει 100% ακρίβεια, εξακολουθεί να είναι πιο αργό από τη μέθοδο brute-force. Στην περίπτωση ενός συνόλου εκπαίδευσης 10000, το MRNG νικάει την τη μέθοδο brute-force για πρώτη φορά, αλλά το LSH είναι το πιο αργό μεταξύ των LSH, GNNS και MRNG. Ο CUBE είναι 2 φορές πιο γρήγορος από το GNNS αλλά στερείται ακρίβειας κατά 0.15. Για το σύνολο εκπαίδευσης 20000, ο MRNG χάνει την ακρίβειά του 100% και παραμένει ο πιο αργός αλγόριθμος. Το GNNS παίρνει το προβάδισμα όσον αφορά το χρόνο και το LSH και ο CUBE είναι αρκετά κοντά χρονικά, με το LSH να έχει πλεονέκτημα στην ακρίβεια. Για ένα σύνολο εκπαίδευσης 30000, ο CUBE είναι ο πιο γρήγορος αλλά στερείται ακρίβειας κατά 0.10. Το GNNS είναι ταχύτερο από το LSH και το MRNG αλλά λιγότερο αποτελεσματικό και από τα δύο. Το LSH είναι ελαφρώς ταχύτερο από το MRNG αλλά υστερεί στην ακρίβεια κατά 0.04. Για το σύνολο εκπαίδευσης των 40000 το MRNG επιστρέφει με 100% ακρίβεια αλλά παραμένει το πιο αργό. Το GNNS έχει καλή απόδοση τόσο σε χρόνο όσο και σε ακρίβεια, με τον CUBE

να ακολουθεί στενά όσον αφορά την καθυστέρηση και μια μικρή μείωση της ακρίβειας. Για σύνολο εκπαίδευσης 50000, το MRNG βρίσκεται στην 3η θέση όσον αφορά τους χρόνους, αλλά διατηρεί την καλύτερη ακρίβεια συνολικά. Το GNNS έχει καλή απόδοση από άποψη χρόνου και ο CUBE ακολουθεί με μια μικρή καθυστέρηση αλλά λιγότερη ακρίβεια. Τέλος, για μέγεθος εκπαίδευσης 60000, όλοι οι αλγόριθμοι από τη μέθοδο brute-force. Το GNNS είναι το ταχύτερο, με το MRNG στην τρίτη θέση αλλά με σχεδόν 100% ακρίβεια. Η δεύτερη θέση πηγαίνει στον CUBE με μικρή καθυστέρηση σε σύγκριση με το GNNS ενώ το ΛΣΗ είναι το πιο αργό αλλά με ακρίβεια που μοιάζει με του MRNG.

Με βάση την ανάλυση των διαφόρων μεγεθών εκπαίδευσης για διαφορετικούς αλγόριθμους, μπορούμε να βγάλουμε τα ακόλουθα συμπεράσματα:

- Το LSH έχει σταθερά καλή απόδοση σε διαφορετικά μεγέθη προπόνησης. Είναι γρήγορο και ακριβές, καθιστώντας το κατάλληλη επιλογή για σύνολα δεδομένων διαφόρων μεγεθών. Υπερέχει ιδιαίτερα σε σύνολα δεδομένων με μικρότερα μεγέθη (100 έως 1000) όπου η ταχύτητα και η ακρίβεια είναι ζωτικής σημασίας.

- Ο αλγόριθμος CUBE τείνει να είναι γρήγορος αλλά λιγότερο ακριβής σε σύγκριση με το LSH και το GNNS. Μπορεί να είναι μια καλή επιλογή για σύνολα δεδομένων με μεγαλύτερα μεγέθη όπου η ταχύτητα είναι προτεραιότητα, αλλά η ακρίβεια δεν είναι κρίσιμη.

- Το GNNS αποδίδει καλά από άποψη χρόνου και ακρίβειας, ειδικά σε σύνολα δεδομένων με μεγαλύτερα μεγέθη (20000 έως 60000). Ωστόσο, το πλεονέκτημά του στην ταχύτητα είναι πιο αισθητό σε μεγαλύτερα σύνολα δεδομένων και μπορεί να μην είναι η ταχύτερη επιλογή για μικρότερα σύνολα δεδομένων.

- Το MRNG παρέχει σταθερά 100% ακρίβεια, αλλά είναι γενικά ο πιο αργός αλγόριθμος. Μπορεί να είναι κατάλληλο για σενάρια όπου η ακρίβεια είναι η ύψιστη προτεραιότητα και ο υπολογιστικός χρόνος δεν αποτελεί μεγάλη ανησυχία. Ωστόσο, γίνεται πιο πρακτικό καθώς αυξάνεται το μέγεθος δεδομένων.