Πειραματική συγκριτική μελέτη

Κυριακού Χρήστος - sdi2300096 ~ Πετρίδου Ελισάβετ - sdi2300170

1. LSH – MNIST

Παραδείγματα αποτελεσμάτων:

Παράμετροι LSH	Avg. AF	Recall@5	QPS	tApprox	tTrue	Σχόλια
k=16, L=25, w=230, R=1200	1.008	0.868	140.26	0.00713 s	0.0527 sec	Γρήγορο (7.4×) με υψηλό recall
k=10, L=40, w=200, R=1200	1.0002	0.994	27.13	0.03685 s	0.0568 s	Πολύ υψηλό recall (1.5× ταχύτερο)
k=20, L=10, w=250, R=1200	1.0634	0.520	528.13	0.00189 s	0.0542 s	Πολύ γρήγορο (28×), χαμηλό recall

Όταν αυξάνουμε τον αριθμό των πινάκων κατακερματισμού, δημιουργούνται περισσότεροι πίνακες για αναζήτηση, με αποτέλεσμα να αυξάνονται οι πιθανότητες να εντοπιστούν όλοι οι πραγματικοί γείτονες, δηλαδή το recall βελτιώνεται. Ωστόσο, η ύπαρξη περισσότερων hash tables αυξάνει και τον χρόνο υπολογισμού, κάνοντας την αναζήτηση πιο αργή και άρα πιο κοντά στο brute force . Από την άλλη πλευρά, η μείωση του K (αριθμός των hash functions), δηλαδή η χρήση λιγότερο αυστηρού hash, επιτρέπει να συμπεριλαμβάνονται περισσότεροι υποψήφιοι για επαλήθευση, γεγονός που αυξάνει το την ακρίβεια του αλγορίθμου, αλλά παράλληλα επιβαρύνει τον χρόνο εκτέλεσης, καθιστώντας την αναζήτηση πιο αργή. Αντίθετα, η αύξηση του K και η ταυτόχρονη μείωση του L μειώνουν τον αριθμό των υποψηφίων και πινάκων κατακερματισμού που εξετάζονται, με αποτέλεσμα η αναζήτηση να γίνεται πολύ γρήγορη, αλλά το recall να μειώνεται δραματικά, επειδή λιγότεροι γείτονες εξετάζονται.

2. LSH – SIFT

Παραδείγματα αποτελεσμάτων:

Παράμετροι LSH	Average AF	Recall@5	QPS	tApprox	tTrue	Σχόλια
k=16, L=30, w=5000,R=250	1.000	1.0	2.30	0.4356 s	0.3375 sec	Αργό (απόδοση περίπου ίδια με brute force)
k=16, L=2, w=800, R=50	1.001	0.97	2.92	0.342 s	0.345 sec	Λίγο γρηγορότερο, recall σχεδόν ίδιο
k=16, L=1, w=600, R=30	1.027	0.778	7.93	0.126 s	0.336 s	Γρήγορο, αλλά με χαμηλό recall
k=16, L=1, w=500, R=20	1.044	0.668	10.69	0.094 s	0.346 s	Πολύ γρήγορο, χαμηλό recall

Η μείωση του αριθμού L των hash tables συνεπάγεται ότι υπάρχουν λιγότεροι πίνακες για αναζήτηση, με αποτέλεσμα η διαδικασία να γίνεται ταχύτερη, αλλά ταυτόχρονα μειώνεται η πιθανότητα να εντοπιστούν όλοι οι πραγματικοί γείτονες, δηλαδή το recall μειώνεται. Παράλληλα, η μείωση του πλάτους των buckets W οδηγεί σε μικρότερα buckets, άρα λιγότερους υποψήφιους για επαλήθευση σε κάθε

bucket, γεγονός που αυξάνει το speedup, αλλά μπορεί να χαθούν ορισμένοι σωστοί γείτονες, μειώνοντας την ακρίβεια. Η μείωση του αριθμού των υποψηφίων που θα επαληθευτούν σημαίνει ότι εξετάζονται λιγότερα σημεία συνολικά, γεγονός που αυξάνει σημαντικά την ταχύτητα της αναζήτησης, αλλά ταυτόχρονα μειώνει το recall, καθώς πολλά πιθανά κοντινά σημεία δεν ελέγχονται καθόλου.

3. Hypercube – SIFT

Παραδείγματα αποτελεσμάτων:

Παράμετροι Hypercube	**Avg. AF**	**Recall@5**	**QPS**	**tApprox**	**tTrue**	**Σχόλια**
Kproj=14, M=10, probes =2	1.936	0.02	12591	0.000079 s	0.677 s	Ακραία ταχύτητα, πολύ χαμηλό recall
kproj=14, M=5000, probes =100	1.187	0.148	216	0.0046 s	0.6396 sec	Γρηγορότερο με υψηλότερο recall
kproj=12, M=500000, probes =1000	1.011	0.85	2.54	0.393 sec	0.638 sec	Πολύ υψηλό recall (~90%), αργό αλλά αποδεκτό για 1M vectors

Η αύξηση της παραμέτρου M, δηλαδή του αριθμού των υποψήφιων κορυφών που εξετάζονται στον αλγόριθμο Hypercube, οδηγεί σε μεγαλύτερο αριθμό πιθανών γειτόνων για κάθε εξεταζόμενο query, γεγονός που αυξάνει το recall, αλλά ταυτόχρονα αυξάνει και τον χρόνο εκτέλεσης. Παράλληλα, η αύξηση των probes σημαίνει ότι εξετάζονται περισσότερα γειτονικά cubes, με αποτέλεσμα να βελτιώνεται ακόμα περισσότερο η πιθανότητα να εντοπιστούν όλοι οι πραγματικοί γείτονες, αυξάνοντας το recall. Αντίθετα, η μείωση του αριθμού των προβολών Kproj αυξάνει την πιθανότητα οι σωστοί γείτονες να βρίσκονται στο ίδιο cube, βελτιώνοντας το recall χωρίς να επηρεάζεται σημαντικά η ταχύτητα της αναζήτησης.

4. hypercube – MNIST

Παραδείγματα αποτελεσμάτων:

Παράμετροι Hypercube	**Avg. AF**	**Recall@10**	**QPS**	**tApprox**	**tTrue**
kproj=14, M=2000, probes =20	1.691	0.234	10907	0.000092	0.023
kproj=6, M=20000, probes =150	1.000	1.0	87.65	0.011	0.025

Ο hypercube είναι πολύ γρήγορος όταν χρησιμοποιούνται λίγες probes και μικρό M, αλλά το recall είναι χαμηλό λόγω περιορισμένου αριθμού υποψηφίων. Αυξάνοντας το M και τις probes, η ακρίβεια βελτιώνεται σημαντικά, καθώς εξετάζονται περισσότερες υποψήφιες κορυφές και γειτονικά cubes. Ωστόσο, η αύξηση αυτή μειώνει την ταχύτητα και αυξάνει τον χρόνο ανά εξεταζόμενο διάνυσμα, δημιουργώντας το γνωστό trade off ταχύτητας – ακρίβειας.