



Πολυτεχνική Σχολή
Τμήμα Μηχανικών Η/Υ & Πληροφορικής

ΥΔΑ

Τεχνικές Διαχείρισης και Εξόρυξης για Δεδομένα Μεγάλου Όγκου

Scalable Supervised Discrete Hashing for Large-Scale Search

Πετράκης Κωνσταντίνος

A.M. 1041589

1. Εισαγωγή

Το πρόβλημα της αναζήτησης κοντινότερου γείτονα, γνωστό και ως αναζήτηση ομοιότητας, αντιστοιχεί στην εύρεση εκείνων των στοιχείων μιας μεγάλης Βάσης Δεδομένων των οποίων η απόσταση από ένα δοθέν στοιχείο αναζήτησης (query) είναι μικρότερη από ένα συγκεκριμένο κατώφλι. Έχουν αναπτυχθεί διάφορες μέθοδοι για τη αντιμετώπιση αυτού του προβλήματος αλλά η ακριβής εύρεση κοντινότερων γειτόνων είναι εξαιρετικά κοστοβόρα εξαιτίας του πλήθους των αντικειμένων που πρέπει να συγκριθούν αλλά και της ανάγκης υπολογισμού της απόστασης κάθε φορά μεταξύ query και αντικειμένων της βάσης. Εξ' αιτίας αυτού τα τελευταία χρόνια πολλές προσπάθειες έχουν καταβληθεί για την κατά προσέγγιση αναζήτηση κοντινότερων γειτόνων, τα αποτελέσματα της οποίας δείχνουν ικανοποιητικά για τις περισσότερες πρακτικές εφαρμογές.

Η πιο διαδεδομένη λύση στη προσεγγιστική αναζήτηση κοντινότερων γειτόνων είναι η μέθοδος του κατακερματισμού (hashing). Ο κατακερματισμός στοχεύει στο μετασχηματισμό των δεδομένων σε μια νέα αναπαράσταση σε ένα χώρο χαμηλότερης διάστασης, ή ισοδύναμα σε ένα μικρό κωδικό που αποτελείται από μια ακολουθία bits και καλείται hash code. Οι αλγόριθμοι κατακερματισμού διαιρούνται σε 2 μεγάλες κατηγορίες: στον ευαίσθητο στη τοπικότητα κατακερματισμό (Locality Sensitive hashing ή LSH) και στις μεθόδους learning to hash. Στους LSH αλγόριθμους κατασκευάζονται συναρτήσεις κατακερματισμού χωρίς να λαμβάνεται οιαδήποτε πληροφορία σχετικά με την κατανομή των δεδομένων. Η έρευνα σε αυτή την πρώτη κατηγορία εστιάζει στην εύρεση νέων τυχαίων hash συναρτήσεων οι οποίες να ικανοποιούν την ιδιότητα της 'ευαισθησίας στην τοπικότητα'.

Αντίθετα οι αλγόριθμοι learning to hash αντιστοιχούν σε προσέγγιση που εξαρτάται από την κατανομή των δεδομένων που έχουμε στη διάθεση μας. Πιο συγκεκριμένα στο learning to hash στόχος είναι η μάθηση μιας συνάρτησης κατακερματισμού με βάση ένα συγκεκριμένο σύνολο δεδομένων έτσι ώστε τα αποτελέσματα της αναζήτησης κοντινότερων γειτόνων στο χώρο των hash codes να είναι όσο το δυνατόν πιο κοντά στα πραγματικά αποτελέσματα, ενώ ταυτόχρονα να διατηρείται χαμηλά το κόστος σε χώρο και χρόνο. Η κύρια μεθοδολογία που ακολουθείται στους αλγόριθμους learning to hash είναι η διατήρηση της ομοιότητας, δηλαδή η ελαχιστοποίηση του χάσματος μεταξύ των ομοιοτήτων (ή της κατάταξης τους) όταν αυτές υπολογιστούν στον αρχικό χώρο και των ομοιοτήτων όταν υπολογίζονται στο χώρο των hash codes. Η ομοιότητα στον αρχικό χώρο προκύπτει είτε από τις κλάσεις των αντικειμένων (αναφέρεται και σαν σημασιολογική πληροφορία), ή από τις αποστάσεις (π.χ. Ευκλείδειες) μεταξύ όλων των ζευγών αντικειμένων.

Μια μέθοδος learning to hash πρέπει να απαντά σε 3 προβλήματα ώστε να υπολογίζει τα hash codes. Ποια συνάρτηση hash θα υιοθετηθεί που θα αντιστοιχίζει κάθε αντικείμενο από τον αρχικό χώρο σε έναν hash κωδικό, τι είδους ομοιότητα θα χρησιμοποιηθεί στον χώρο κωδικοποίησης (hash) και τι είδους ομοιότητα θα χρησιμοποιηθεί στον αρχικό χώρο, και τέλος τι συνάρτηση σφάλματος θα χρησιμοποιηθεί με βάση την οποία θα γίνει η βελτιστοποίηση.

2. Scalable Supervised Discrete Hashing for Large Scale Search

Πολλές σημαντικές εφαρμογές αξιοποιούν την τεχνική αναζήτησης με βάση την ομοιότητα για την επιστροφή εκείνων των δεδομένων που ομοιάζουν με το δοθέν query αναζήτησης. Παραδείγματα τέτοιων εφαρμογών αποτελούν η ανάκτηση εικόνων στο διαδίκτυο και η σύσταση αντικειμένων, μεταξύ άλλων. Όμως, όπως αναφέρθηκε και πιο πάνω, η ραγδαία αύξηση στο πλήθος των δεδομένων τα τελευταία χρόνια καθιστά την εξαντλητική σύγκριση μεταξύ αντικειμένων σε χώρους πολύ υψηλών διαστάσεων απαγορευτική αν θέλουμε να έχουμε κλιμακώσιμη αναζήτηση. Τη λύση ήρθαν να δώσουν οι διάφορες μέθοδοι κατακερματισμού, οι οποίες δείχνουν να πετυχαίνουν πολλά υποσχόμενα αποτελέσματα.

Μια μέθοδος κατακερματισμού μετασχηματίζει τα αρχικά δεδομένα σε μια συμπαγή δυαδική αναπαράσταση (κωδικολέξη ή hash code) σταθερού αριθμού bits, μειώνοντας έτσι το κόστος που απαιτείται σε χώρο για την αποθήκευση των δεδομένων. Κάθε query που θέλουμε να απαντήσουμε μετατρέπεται επίσης σε μια δυαδική αναπαράσταση, χρησιμοποιώντας την ίδια συνάρτηση που μετέτρεψε και τα αρχικά δεδομένα, και η αναζήτηση ομοιότητας εκτελείται υπολογίζοντας απλά την Hamming απόσταση μεταξύ της hash code αναπαράστασης του query και των δεδομένων. Η Hamming απόσταση μεταξύ 2 κωδικολέξεων αντιστοιχεί στο πλήθος των bits που διαφέρουν (XOR). Ο λόγος που οι hash μέθοδοι δείχνουν να λύνουν το πρόβλημα της κλιμακωσιμότητας είναι ακριβώς το ότι επιτρέπουν την αποδοτική από αποψη χώρου και χρόνου σύγκριση μεταξύ τεράστιου πλήθους αντικειμένων.

Οι μέθοδοι κατακερματισμού τώρα χωρίζονται σε 2 μεγάλες κατηγορίες. Στις ανεξάρτητες από τα δεδομένα (data-independent) και σε αυτές που βασίζονται στα δεδομένα που έχουμε στην διάθεση μας (data-dependent). Όπως προδίδει η ονομασία, και αναφέραμε και στην εισαγωγή, οι ανεξάρτητες των δεδομένων μέθοδοι δεν αξιοποιούν την παραμικρή πληροφορία από τα δεδομένα που έχουμε, και μαθαίνουν τις συναρτήσεις κατακερματισμού χρησιμοποιώντας συνήθως τυχαίες προβολές. Κύριος εκπρόσωπος αυτής της κατηγορίας είναι η μέθοδος LSH, η επιτυχία της οποίας έχει οδηγήσει στην επέκταση της με την αξιοποίηση κι άλλων μετρικών απόσταση (π.χ. p-νόρμα, απόσταση Mahalanobis). Παρόλη την ταχύτητα τους το σημαντικότερο μειονέκτημα αυτών των μεθόδων παραμένει η ανάγκη χρήσης μεγάλων κωδικολέξεων που με την σειρά του επιφέρει σημαντική χωρική επιβάρυνση.

Οι εξαρτημένες από τα δεδομένα μέθοδοι από την άλλη μαθαίνουν τα hash codes και εν συνεχεία της hash συναρτήσεις χρησιμοποιώντας τα δεδομένα που έχουμε στη διάθεση μας. Τα δεδομένα εκπαίδευσης όπως θα τα αποκαλούμε συνήθως αξιοποιούνται έμμεσα, εννοώντας πως αξιοποιούμε τις σχέσεις μεταξύ των δεδομένων στον αρχικό χώρο χαρακτηριστικών και προσπαθούμε να μάθουμε εκείνα τα hash codes που διατηρούν τις αρχικές ιδιότητες των δεδομένων. Οι συγκεκριμένες hash μέθοδοι με τη σειρά τους χωρίζονται περαιτέρω σε 2 κατηγορίες. Στις μη-επιβλεπόμενες (unsupervised) και στην επιβλεπόμενες (supervised).

Οι μη-επιβλεπόμενες μέθοδοι αντιστοιχούν σε περιπτώσεις όπου δεν έχουμε πληροφορία σχετικά με το αν τα αντικείμενα που έχουμε στην διάθεση μας ανήκουν σε κάποια συγκεκριμένη κλάση. Σε αυτήν την περίπτωση η εκμάθηση των hash codes και των hash συναρτήσεων γίνεται χρησιμοποιώντας μόνο τις σχέσεις ομοιότητας ή απόστασης μεταξύ των δεδομένων. Χαρακτηριστικά παραδείγματα μη-επιβλεπόμενων

μεθόδων hashing αποτελούν οι Spectral Hashing, Iterative Quantization και Inductive Hashing on Manifolds.

Όταν κάθε αποθηκευμένο αντικείμενο συνοδεύεται και από την αντίστοιχη ετικέτα κλάσης, τότε οι επιβλεπόμενες τεχνικές μπορούν να αξιοποιήσουν και αυτές τις κλάσεις στην εκμάθηση των hash codes. Ακριβώς επειδή ενσωματώνουν περισσότερη πληροφορία κατά την αναζήτηση των hash codes αυτές οι μέθοδοι έχουν παρατηρηθεί να πετυχαίνουν καλύτερη απόδοση σε σχέση με τις μη-επιβλεπόμενες μεθόδους. Μερικά παραδείγματα τέτοιων μεθόδων αποτελούν οι Semi-Supervised Hashing, Minimal Loss Hashing και Supervised Hashing with Latent Factor.

Παρά το γεγονός πως, όπως είπαμε, οι περισσότερες επιβλεπόμενες hashing μέθοδοι παρέχουν πολλά υποσχόμενα αποτελέσματα, παρουσιάζουν επίσης και αρκετά προβλήματα. Αρχικά ο περιορισμός που έχουμε επιβάλει από την αρχή, ότι δηλαδή τα hash codes πρέπει να είναι δυαδικά, απαιτεί την επίλυση ενός διακριτού προβλήματος βελτιστοποίησης το οποίο στη γενική περίπτωση είναι αρκετά δύσκολο. Η λύση που επιστρατεύουν οι περισσότερες μέθοδοι για να το αντιμετωπίσουν είναι να μετατρέψουν το διακριτό πρόβλημα σε συνεχές και στη συνέχεια να επιλύσουν αυτό το συνεχές πρόβλημα βελτιστοποίησης. Επειδή όμως τα hash codes που θα προκύψουν θα αποτελούνται από πραγματικούς αριθμούς, συντελείται και ένα βήμα κβάντισης της λύσης όπου ανάλογα με το αν το κάθε ψηφίο του hash code είναι θετικό ή αρνητικό μετατρέπεται σε 1 ή -1 αντίστοιχα. Προφανώς η λύσεις που προκύπτουν με αυτόν τον τρόπο δεν αναμένεται να είναι βέλτιστες εξ' αιτίας του σφάλματος που εισάγει η μετατροπή του προβλήματος σε συνεχές.

Ένα άλλο σημαντικό πρόβλημα των επιβλεπόμενων μεθόδων κατακερματισμού προκύπτει από την χρήση των ετικετών που ενσωματώνουμε στην διαδικασία μάθησης των hash codes. Απαιτείται δηλαδή η χρήση του $n \times n$ μητρώου ομοιότητας S μεταξύ των δειγμάτων στον αρχικό χώρο χαρακτηριστικών, όπου n το πλήθος των αντικειμένων. Όταν έχουμε να κάνουμε με μεγάλο πλήθος δεδομένων, και ειδικά όταν μιλάμε για Big Data, το χωρικό κόστος $O(n^2)$ είναι απαγορευτικό.

Αυτή η χωρική πολυπλοκότητα, μεταξύ άλλων, κάνουν τις supervised hashing μεθόδους ιδιαίτερα χρονοβόρες και μη-κλιμακώσιμες. Μια πρόχειρη λύση που χρησιμοποιείται από τις περισσότερες τέτοιες μεθόδους όταν ο χρόνος είναι σημαντικός είναι η δειγματοληψία. Πιο συγκεκριμένα, αντί να χρησιμοποιούνται όλα τα δείγματα που έχουμε στη διάθεση μας σαν το σύνολο εκπαίδευσης, δειγματοληπτείται μόνο ένα μικρό υποσύνολο αυτών, ώστε να επιτυγχάνεται η γρήγορη εκπαίδευση της εκάστοτε μεθόδου. Κι εδώ γίνεται εύκολα αντιληπτό πως το γεγονός ότι δεν εκμεταλλευόμαστε το σύνολο των δεδομένων που έχουμε στη διάθεση μας έχει σαν αποτέλεσμα απώλεια πληροφορίας και φτωχή απόδοση.

Η μέθοδος Scalable Supervised Discrete Hashing (SSDH) [1] που προτείνεται σε αυτήν την εργασία σαν σκοπό έχει την αποδοτική αντιμετώπιση όλων των αναφερθέντων προβλημάτων διατηρώντας παράλληλα καλή απόδοση. Πρώτα από όλα η μέθοδος λύνει το διακριτό πρόβλημα βελτιστοποίησης αυτό καθ' αυτό, χωρίς καμία 'χαλάρωση', ενώ ταυτόχρονα αξιοποιείται το σύνολο των δεδομένων που έχουμε στη διάθεση μας, δεν απαιτεί δηλαδή κάποια δειγματοληψία.

Η μέθοδος SSDH αξιοποιεί και το μητρώο ομοιότητας S αλλά και τις ετικέτες κλάσεων (μητρώο Y) που έχουμε στη διάθεση μας για τα δεδομένα. Ακόμη, επειδή όπως θα δούμε, αναπαριστά το μητρώο των hash codes B σαν ένα μετασχηματισμό (προβολή) από τον χώρο των ετικετών, καταφέρνει να αποφεύγει την απαίτηση του

μεγάλου μητρώου \mathbf{S} σε κάθε βήμα της βελτιστοποίησης, βελτιώνοντας έτσι κατά πολύ το απαιτούμενο κόστος σε χώρο. Αυτή η προσέγγιση του μητρώου \mathbf{B} με ένα πραγματικό μητρώο έχει το πλεονέκτημα πως μπορεί να οδηγήσει σε καλύτερα αποτελέσματα, καθώς αντικαθιστούμε ένα δυαδικό μητρώο με ένα πραγματικό, που αναπόφευκτα θα φέρει περισσότερη πληροφορία. Όσον αφορά την βελτιστοποίηση, η μέθοδος SSDH χρησιμοποιεί έναν εναλλασσόμενο αλγόριθμο διακριτής βελτιστοποίησης που επιτρέπει την αποδοτική κλιμάκωση της μεθόδου όταν έχουμε να κάνουμε με μεγάλα σύνολα δεδομένων.

Όλες οι επιβλεπόμενες μέθοδοι χωρίζονται σε 2 κατηγορίες ανάλογα με το πώς φτάνουν στην εκμάθηση των hash codes και των hash συναρτήσεων. Οι περισσότερες μέθοδοι ανήκουν στην κατηγορία των one-step μεθόδων κατακερματισμού, οι οποίες μαθαίνουν τα hash codes και τις hash συναρτήσεις ταυτόχρονα. Αντίθετα οι two-step μέθοδοι κατακερματισμού διαιρούν την εκμάθηση των hash codes και hash συναρτήσεων σε 2 ξεχωριστά βήματα. Το πρώτο βήμα αντιστοιχεί στην παραγωγή των hash codes με την επίβλεψη κατάλληλων συναρτήσεων σφάλματος (loss function). Οι συναρτήσεις σφάλματος επιλέγονται με τέτοιο τρόπο ώστε να διατηρούν τις ιδιότητες (ομοιότητες) των δεδομένων από τον αρχικό χώρο στο χώρο των hash codes. Στο δεύτερο βήμα δοθέντων πλέον των hash codes οι two-step μέθοδοι κατακερματισμού επιχειρούν να μάθουν εκείνες τις hash συναρτήσεις που θα μετασχηματίζουν τα δεδομένα από τον αρχικό χώρο χαρακτηριστικών στα επιθυμητά hash codes.

Το πιο σημαντικό μεταξύ των 2 βημάτων είναι το πρώτο. Αυτό γιατί η απόδοση των two-step μεθόδων κατακερματισμού εξαρτάται σε μεγάλο βαθμό από την ποιότητα των hash codes που παράγονται στο πρώτο βήμα. Επομένως δίνεται συνήθως μεγαλύτερη βαρύτητα στο σχεδιασμό καλών συναρτήσεων σφάλματος. Όσον αφορά το δεύτερο βήμα μπορεί να αποδειχτεί πως η εκμάθηση μιας hash συνάρτησης που θα αντιστοιχίζει τα δεδομένα από τον αρχικό χώρο χαρακτηριστικών σε ένα bit της δυαδικής αναπαράστασης μοντελοποιείται σαν πρόβλημα δυαδικής ταξινόμησης. Αν λοιπόν το μήκος των hash codes είναι r , η εκμάθηση της συνάρτησης κατακερματισμού που μας πηγαίνει σε αυτόν το χώρο αντιστοιχεί στην επίλυση r προβλημάτων δυαδικής ταξινόμησης. Εδώ μπορούμε να χρησιμοποιήσουμε όποιο κατηγοριοποιητή θέλουμε, από γραμμικούς κατηγοριοποιητές μέχρι Μηχανές Διανυσμάτων Υποστήριξης (SVM) και Συνελκτικά Νευρωνικά δίκτυα. Είναι προφανές πως όσο πιο καλός είναι ο κατηγοριοποιητής που θα επιλέξουμε τόσο καλύτερα θα είναι τα αποτελέσματα, ωστόσο μπορεί να απαιτηθεί περισσότερος χρόνος εκπαίδευσης. Η μέθοδος SSDH που εξετάζουμε ανήκει στην κατηγορία των two-step μεθόδων κατακερματισμού και αυτό που την κάνει να ξεχωρίζει από όλες τις προηγούμενες μεθόδους είναι ο σχεδιασμός της συνάρτησης σφάλματος και ο τρόπος που αυτή βελτιστοποιείται.

Εν συνεχεία θα παρουσιάσουμε τον συμβολισμό που χρησιμοποιείται και στην εργασία ώστε να εξηγηθεί πως ακριβώς δουλεύει η μέθοδος SSDH παρακάτω. Πρώτα από όλα κάθε αντικείμενο δεδομένων συμβολίζεται σαν $\mathbf{x}_i \in R^d$ και η αντίστοιχη ετικέτα με το διάνυσμα $\mathbf{y}_i \in \{0,1\}^c$ ($i=1,2,\dots,n$). Υποθέτουμε πως έχουμε n αντικείμενα, c κλάσεις και ο αρχικός χώρος χαρακτηριστικών είναι d διαστάσεων. Όσον αφορά τα διανύσματα των ετικετών, μπορούμε να τα δούμε σαν one-hot διανύσματα, όπου το μόνο bit ίσο με 1 είναι το bit που αντιστοιχεί στην κλάση στην οποία ανήκει το αντικείμενο και όλα τα άλλα bit είναι 0. Συγκεντρώνοντας όλα τα αντικείμενα σαν γραμμές κατασκευάζεται το μητρώο χαρακτηριστικών $\mathbf{X} \in R^{n \times d}$. Όμοια

ομαδοποιώντας τα διανύσματα ετικετών σαν γραμμές κατασκευάζεται το μητρώο ετικετών $\mathbf{Y} \in \{0,1\}^{n \times c}$, το οποίο όπως αναφέραμε θα είναι δυαδικό. Πιο συγκεκριμένα αν \mathbf{Y}_{ik} το k -οστό στοιχείο της i -οστής γραμμής \mathbf{y}_i^T , τότε $\mathbf{Y}_{ik} = 1$ αν \mathbf{x}_i ανήκει στην κλάση k και $\mathbf{Y}_{ik} = 0$ διαφορετικά.

Το μητρώο σημασιολογικής ομοιότητας μεταξύ κάθε ζεύγους δειγμάτων στον αρχικό χώρο χαρακτηριστικών συμβολίζεται με $\mathbf{S} \in \{-1,1\}^{n \times n}$. Η τιμή 1 δηλώνει ομοιότητα μεταξύ 2 δειγμάτων και η τιμή -1 ότι δυο αντικείμενα είναι ανόμοια. Έχουμε αναφέρει πως ο στόχος της μεθόδου SSDH είναι η εκμάθηση δυαδικών hash codes. Στη συγκεκριμένη περίπτωση τα δυαδικά hash codes αποτελούνται από τις τιμές 1 και -1. Στόχος της μεθόδου SSDH επομένως είναι να μάθει ένα hash codes $b_i \in \{-1,1\}^r$ για κάθε αντικείμενο, όπου r το μήκος της κωδικολέξης. Συγκεντρώνοντας αυτά τα hash codes, όπως και για τα μητρώα χαρακτηριστικών και ετικετών, κατασκευάζεται το μητρώο των hash codes $\mathbf{B} \in \{-1,1\}^{n \times r}$, όπου κάθε γραμμή αντιστοιχεί στο hash code του αντίστοιχου αντικειμένου.

Τέλος η συνάρτηση κατακερματισμού που μετασχηματίζει τα αρχικά δεδομένα στα hash codes συμβολίζεται με F . Με βάση τα όσα ορίσαμε παραπάνω η συνάρτηση F είναι αυτή που μετατρέπει το μητρώο \mathbf{X} στο μητρώο \mathbf{B} , δηλαδή $\mathbf{B} = F(\mathbf{X}) = \text{sgn}(\mathbf{X}\mathbf{W})$. Από εδώ και στο εξής με sgn θα συμβολίζουμε τη στοιχείο προς στοιχείο συνάρτηση προσήμου, η οποία ορίζεται ως $\text{sgn}(x) = 1$ εάν $x \geq 0$ και -1 αλλιώς. Επίσης το μητρώο \mathbf{W} είναι το μητρώο προβολής από τον αρχικό χώρο χαρακτηριστικών στο χώρο των hash codes και θα δούμε πιο αναλυτικά στη συνέχεια πως προκύπτει η παραπάνω σχέση. Στην εργασία δεν αναφέρεται να εκτελείται κάποιο βήμα προ-επεξεργασίας των δεδομένων, μόνο ότι γίνεται η υπόθεση πως τα δεδομένα είναι κεντραρισμένα γύρω από το 0, δηλαδή $\sum_{i=1}^n \mathbf{x}_i = 0$.

Επειδή όπως αναφέραμε η μέθοδος SSDH αντιστοιχεί σε two-step μέθοδο κατακερματισμού θα αναλύσουμε τα 2 βήματα ξεχωριστά ξεκινώντας με το πιο σημαντικό, δηλαδή την σχεδίαση της συνάρτησης σφάλματος για την εκμάθηση των hash codes. Συνήθως η συνάρτηση σφάλματος που χρησιμοποιείται από την πλειοψηφία των επιβλεπόμενων μεθόδων είναι η εξής

$$\min_{\mathbf{B}} \|\mathbf{rS} - \mathbf{B}\mathbf{B}^T\|_F^2, \text{ s. t. } \mathbf{B} \in \{-1,1\}^{n \times r} \quad (1)$$

Όπως βλέπουμε χρησιμοποιείται το μητρώο \mathbf{S} για να αξιοποιηθεί η σημασιολογική πληροφορία που έχουμε στη διάθεση μας. Για τις επιβλεπόμενες μεθόδους κατακερματισμού τα παραγόμενα hash codes πρέπει να διατηρούν τη σημασιολογική ομοιότητα μεταξύ των δειγμάτων. Για να καταλάβουμε τώρα γιατί οι ομοιότητες μεταξύ των δεδομένων προσεγγίζονται με το γινόμενο $\mathbf{B}\mathbf{B}^T$ αρκεί να παρατηρήσουμε το εξής. Το εσωτερικό γινόμενο μεταξύ 2 δυαδικών hash codes αντιστοιχεί στο αντίστροφο της Hamming απόσταση τους. Το αντίστροφο της απόστασης όμως δηλώνει ομοιότητα. Με απλά λόγια επομένως η Εξ. (1) χρησιμοποιεί το εσωτερικό γινόμενο για να προσεγγίσει την σημασιολογική ομοιότητα μετρώντας το τετραγωνικό σφάλμα. Το βασικό μειονέκτημα της Εξ. (1) προκύπτει από την ανάγκη χρήσης του μεγάλου $n \times n$ μητρώου \mathbf{S} γεγονός που επιφέρει μεγάλο χωρικό και χρονικό κόστος.

Η νέα συνάρτηση σφάλματος που πρότειναν λοιπόν οι συγγραφείς με σκοπό να αποφύγουν τα παραπάνω προβλήματα είναι η

$$\min_{\mathbf{B}, \mathbf{G}} \|\mathbf{rS} - \mathbf{B}(\mathbf{Y}\mathbf{G})^T\|_F^2 + \mu \|\mathbf{B} - \mathbf{Y}\mathbf{G}\|_F^2, \text{ s. t. } \mathbf{B} \in \{-1,1\}^{n \times r} \quad (2)$$

Όπου $\mathbf{G} \in R^{c \times r}$ είναι ένα μητρώο προβολής από το \mathbf{Y} στο \mathbf{B} και μ είναι μια παράμετρος ισορροπίας. Βλέπουμε πως πλέον συμπεριλαμβάνεται στην εκμάθηση των hash codes και πληροφορία σχετικά με την κλάση στην οποία ανήκει κάθε αντικείμενο (μητρώο \mathbf{Y}). Από αυτό και μόνο το γεγονός περιμένουμε να βελτιωθεί η ακρίβεια των παραγόμενων hash codes. Επιπλέον προσεγγίζουν το μητρώο \mathbf{B} προβάλλοντας τις ετικέτες κλάσης \mathbf{Y} στο χώρο των hash codes. Το μητρώο \mathbf{G} , που εκτελεί αυτή την προβολή, θα έχει πραγματικά στοιχεία και έχει αποδειχτεί σε άλλη εργασία [2] πως η χρήση ενός πραγματικού μητρώου στη θέση του δυαδικού \mathbf{B} μπορεί να βελτιώσει την ποιότητα της προσέγγισης των ομοιοτήτων \mathbf{S} . Άρα αντικαθιστώντας το μητρώο \mathbf{B} με το μητρώο \mathbf{YG} αναμένουμε να παραχθούν καλύτερα hash codes, με δεδομένο [3] πως το μητρώο \mathbf{YG} προσεγγίζει αποτελεσματικά το μητρώο \mathbf{B} .

Αυτή η αντικατάσταση του \mathbf{B} όμως λύνει και το βασικό πρόβλημα όλων των επιβλεπόμενων μεθόδων, την ανάγκη $O(n^2)$ χωρικής πολυπλοκότητας. Θα δούμε πως η χρήση της συνάρτησης σφάλματος της Εξ. (2) απαιτεί τον υπολογισμό μόνο μια φορά (offline) του μητρώου \mathbf{SY} , με αποτέλεσμα την μείωση του χωρικού κόστους σε $O(n)$.

Επειδή όπως φαίνεται από την Εξ. (2) η συνάρτηση σφάλματος εξαρτάται από 2 μεταβλητές, τα μητρώα \mathbf{G} και \mathbf{B} , οι συγγραφείς πρότειναν έναν αλγόριθμο βελτιστοποίησης που εναλλάσσεται μεταξύ 2 βημάτων. Πρώτα κρατάνε σταθερό το μητρώο \mathbf{B} και ενημερώνουν το \mathbf{G} , και έπειτα ενημερώνουν το \mathbf{G} υποθέτοντας σταθερό το \mathbf{B} .

Το βήμα που αφορά την ενημέρωση του \mathbf{G} είναι γενικά εύκολο. Αν στην Εξ. (2) το \mathbf{B} είναι σταθερό τότε παίρνοντας την παράγωγο της συνάρτησης σφάλματος ως προς το \mathbf{G} και θέτοντας τη ίση με το μηδέν προκύπτει λύση κλειστής μορφής

$$\mathbf{G} = (\mathbf{YY}^T)^{-1}(r(\mathbf{SY})^T \mathbf{B} + \mu \mathbf{Y}^T \mathbf{B})(\mathbf{B}^T \mathbf{B} + \mu \mathbf{I}_{r \times r})^{-1} \quad (3)$$

Προς το παρόν όμως δεν αποφεύγεται το πρόβλημα χρήσης του $n \times n$ μητρώου \mathbf{S} . Αυτό που παρατήρησαν εδώ οι συγγραφείς είναι πως μπορούν να υπολογίσουν μία φορά το γινόμενο \mathbf{SY} και να το χρησιμοποιούν όποτε χρειάζεται κατά την διάρκεια της εκπαίδευσης. Αντικαθιστώντας λοιπόν $\mathbf{A} = \mathbf{SY}$ στην Εξ. (3) κατέληξαν στην σχέση

$$\mathbf{G} = (\mathbf{YY}^T)^{-1}(r(\mathbf{A})^T \mathbf{B} + \mu \mathbf{Y}^T \mathbf{B})(\mathbf{B}^T \mathbf{B} + \mu \mathbf{I}_{r \times r})^{-1} \quad (4)$$

Άξιο παρατήρησης είναι πως το μητρώο \mathbf{A} είναι διαστάσεων $n \times c$ όπου c το πλήθος των κλάσεων, το οποίο είναι συνήθως πολύ μικρότερο του n . Είναι αυτή η αντικατάσταση του \mathbf{S} με το μητρώο \mathbf{A} που οδηγεί πλέον σε χωρική πολυπλοκότητα $O(n)$, όπως αναφέραμε και πιο πάνω.

Αυτά όσον αφορά την ενημέρωση του \mathbf{G} . Για το επόμενο βήμα όπου κρατάμε σταθερό το \mathbf{G} για να ενημερώσουμε το \mathbf{B} υπάρχουν 2 υποπεριπτώσεις. Να έχουμε δεδομένα όπου κάθε δείγμα ανήκει σε μια και μόνο κλάση (single label) ή δεδομένα όπου κάθε αντικείμενο μπορεί να ανήκει σε περισσότερες από μια κλάσεις (multi label). Η 2^η περίπτωση δυσκολεύει αρκετά τα πράγματα.

Στην περίπτωση των single label δεδομένων οι ερευνητές πρώτα αντικατέστησαν την νόρμα Frobenius στην Εξ. (2) με την L_1 νόρμα, και έτσι το πρώτο μέρος της εξίσωσης γίνεται

$$\min_{\mathbf{B}} \|r\mathbf{S} - \mathbf{B}(\mathbf{YG})^T\|_1, \text{ s. t. } \mathbf{B} \in \{-1, 1\}^{n \times r} \quad (5)$$

Η λύση της Εξ. (5) είναι $\mathbf{B} = \text{sgn}(\mathbf{SYG})$. Επίσης όταν έχουμε single-label δεδομένα αποδεικνύεται πως $\text{sgn}(\mathbf{SYG}) = \text{sgn}(\mathbf{YG})$. Η λύση στο δεύτερο μέλος της Εξ. (2) είναι επίσης $\text{sgn}(\mathbf{YG})$, οπότε τελικά για single label δεδομένα η συνολική λύση της Εξ. (2) θα είναι

$$\mathbf{B} = \text{sgn}(\mathbf{YG}) \quad (6)$$

Στην περίπτωση των multi label δεδομένων τώρα $\text{sgn}(\mathbf{SYG}) \neq \text{sgn}(\mathbf{YG})$ και η $\mathbf{B} = \text{sgn}(\mathbf{YG})$ δεν αποτελεί πλέον λύση στην Εξ. (2). Εδώ οι συγγραφείς χρησιμοποιούν την μέθοδο βελτιστοποίησης Discrete Cyclic Coordinate descent (DCC) ώστε να καταλήξουν σε μια λύση κλειστής μορφής για το \mathbf{B} . Για το σκοπό αυτό, αναπτύσσοντας τη και εκτελώντας τις πράξεις, ξαναγράφουν την Εξ. (2) στην εξής μορφή

$$\begin{aligned} \min_{\mathbf{B}} & \|\mathbf{YGB}^T\|_F^2 - 2\text{Tr}(\mathbf{B}^T \mathbf{SYG} + \mu \mathbf{B}^T \mathbf{YG}) \\ & = \|\mathbf{YGB}^T\|_F^2 - 2\text{Tr}(\mathbf{B}^T (\mathbf{AG} + \mu \mathbf{YG})) \\ & = \|\mathbf{CB}^T\|_F^2 - 2\text{Tr}(\mathbf{B}^T \mathbf{Q}) \text{ s.t. } \mathbf{B} \in \{-1, 1\}^{n \times r} \quad (7) \end{aligned}$$

όπου $\mathbf{A} = \mathbf{SY}$, $\mathbf{C} = \mathbf{YG}$ και $\mathbf{Q} = \mathbf{AG} + \mu \mathbf{C}$. Σημειώνεται πως στα παραπάνω έχουν παραλείψει τους όρους $\|r\mathbf{S}\|_F^2, \mu\|\mathbf{B}\|_F^2$ και $\mu\|\mathbf{YG}\|_F^2$, οι οποίοι παραμένουν σταθεροί. Για τη λύση της Εξ. (7) απαιτείται η εκμάθηση των hash codes ένα bit τη φορά. Πιο αναλυτικά κάθε φορά θα μαθαίνεται επαναληπτικά μια στήλη του μητρώου \mathbf{B} , θεωρώντας όλες τις υπόλοιπες στήλες σταθερές. Για το σκοπό αυτό συμβολίζουν με \mathbf{b} την l-οστή στήλη του \mathbf{B} , όπου η l-οστή στήλη αντιστοιχεί στο l-οστό bit όλων των hash codes. Μπορούμε δηλαδή να δούμε, συμφωνά και με την προηγούμενη παρατήρηση, πως σε κάθε βήμα θα μαθαίνεται ένα συγκεκριμένο bit για όλα τα hash codes. Για όλα τα παρακάτω με \mathbf{B}' συμβολίζεται το μητρώο \mathbf{B} όταν αφαιρεθεί η στήλη \mathbf{b} . Με παρόμοιο σκεπτικό με \mathbf{Q}' θα συμβολίζεται το μητρώο \mathbf{Q} όταν από αυτό αφαιρεθεί η l-οστή του γραμμή \mathbf{q} , και με \mathbf{C}' το μητρώο \mathbf{C} , όταν από αυτό αφαιρεθεί η l-οστή του γραμμή \mathbf{c} . Στη συνέχεια αναπτύσσονται οι 2 όροι της Εξ. (7) ξεχωριστά. Από τον πρώτο όρο προκύπτει

$$\begin{aligned} \|\mathbf{CB}^T\|_F^2 & = \text{Tr}(\mathbf{BC}^T \mathbf{CB}^T) \\ & = \|\mathbf{bc}\|_F^2 + 2\mathbf{cC}'\mathbf{B}'^T \mathbf{b} + \text{const} \\ & = 2\mathbf{c}^T \mathbf{C}' \mathbf{B}'^T \mathbf{b} + \text{const} \end{aligned}$$

Ο όρος $\|\mathbf{bc}\|_F^2$ δεν επηρεάζει το αποτέλεσμα καθώς $\|\mathbf{bc}\|_F^2 = \text{Tr}(\mathbf{c}^T \mathbf{b}^T \mathbf{bc}) = \mathbf{nc}^T \mathbf{c} = \text{const}$. Για να δούμε γιατί αυτό το αποτέλεσμα είναι σταθερό πρέπει να σκεφτούμε πως προκύπτουν τα διάνυσμα \mathbf{c} . Το διάνυσμα \mathbf{c} θα είναι η l-οστή γραμμή του μητρώου \mathbf{C} , όπου το μητρώο $\mathbf{C} = \mathbf{YG}$. Το \mathbf{Y} σαν το μητρώο των ετικετών είναι σταθερό και το μητρώο \mathbf{G} στο παρόν βήμα είναι σταθερό. Άρα το διάνυσμα \mathbf{c} θα είναι και αυτό σταθερό στο παρόν βήμα.

Από τον 2° όρο της Εξ. (7) έχουμε

$$2\text{Tr}(\mathbf{B}^T \mathbf{Q}) = \mathbf{q}^T \mathbf{b} + \text{const}$$

Συνδυάζονται τώρα αυτούς τους 2 όρους, και βγάζοντας κοινό παράγοντα το διάνυσμα \mathbf{b} , οι συγγραφείς καταλήγουν στο παρακάτω πρόβλημα βελτιστοποίησης

$$\min_{\mathbf{c}} (\mathbf{c}^T \mathbf{C}' \mathbf{B}'^T + \mathbf{q}^T) \mathbf{b} \text{ s.t. } \mathbf{b} \in \{-1, 1\}^n$$

Η λύση στο παραπάνω πρόβλημα προκύπτει ίση με

$$\mathbf{b} = \text{sgn}(\mathbf{q} - \mathbf{B}' \mathbf{C}'^T \mathbf{c}) \quad (11)$$

Αυτό που είναι σημαντικό να παρατηρήσουμε εδώ είναι πως κάθε bit των hash codes (στήλη \mathbf{b}) μαθαίνεται με βάση όλα τα υπόλοιπα bits (μητρώο \mathbf{B}'). Όπως θα δούμε παρακάτω στα πειράματα που εκτελέστηκαν παρατηρείται πως αυτό το γεγονός, ότι δηλαδή κάθε bit μαθαίνεται με βάση όλα τα υπόλοιπα, έχει κάποιες επιδράσεις στο χρόνο εκπαίδευσης του αλγορίθμου.

Συνοπτικά τα βήματα για την εκμάθηση των hash codes και την διακριτή βελτιστοποίηση της συνάρτησης σφάλματος στον αλγόριθμο SSDH έχουν ως εξής. Παρέχουμε σαν είσοδο στον αλγόριθμο το μητρώο ετικετών \mathbf{Y} , το επιθυμητό μήκος των hash codes r , το πλήθος των επαναλήψεων που θα εκτελεστεί η βελτιστοποίηση και την παράμετρο μ . Στην συνέχεια υπολογίζεται μία φορά το γινόμενο $\mathbf{A} = \mathbf{S}\mathbf{Y}$. Στην έξοδο αναμένουμε να λάβουμε το μητρώο \mathbf{B} με τα hash codes για κάθε αντικείμενο καθώς και το μητρώο προβολής \mathbf{G} , όπως φαίνεται από την Εξ. (2). Τα μητρώο \mathbf{G} και \mathbf{B} αρχικοποιούνται με τυχαίες τιμές και εν συνεχεία εκτελούνται εναλλάξ οι ενημερώσεις τους. Κρατάμε σταθερό το \mathbf{B} και ενημερώνουμε το \mathbf{G} με χρήση της Εξ. (4). Έπειτα κρατάμε σταθερό το \mathbf{G} και ενημερώνουμε το \mathbf{B} με χρήση της εξίσωσης (6) αν έχουμε single label δεδομένα ή με χρήση της Εξ. (11) αν έχουμε multi label δεδομένα.

Όλα τα παραπάνω αφορούσαν το πρώτο βήμα της μεθόδου SSDH, δηλαδή την σχεδίαση της συνάρτησης σφάλματος για την παραγωγή των hash codes. Αφού έχουμε στη διάθεση μας τα hash codes το επόμενο βήμα του αλγορίθμου αφορά στην εκμάθηση της συνάρτησης κατακερματισμού που θα μετατρέπει τα δεδομένα από τον αρχικό χώρο χαρακτηριστικών στις δυαδικές τους αναπαραστάσεις στο χώρο των hash codes. Αναφέραμε πως αυτό το πρόβλημα μπορεί να μοντελοποιηθεί σαν ένα πρόβλημα δυαδικής ταξινόμησης και για κάθε bit μπορεί να χρησιμοποιηθεί οποιοσδήποτε αλγόριθμος κατηγοριοποίησης. Ο αλγόριθμος SSDH χρησιμοποιεί γραμμική παλινδρόμηση (linear regression). Σε αυτή τη περίπτωση η hash συνάρτηση προκύπτει λύνοντας

$$L_e = \|\mathbf{B} - \mathbf{X}\mathbf{W}\|_F^2 + \lambda_e \|\mathbf{W}\|_F^2 \quad (12)$$

Εδώ στόχος είναι να μάθουμε το μητρώο προβολής \mathbf{W} που μας πηγαίνει από τον αρχικό χώρο χαρακτηριστικών στο χώρο των hash codes. Ο όρος $\lambda_e \|\mathbf{W}\|_F^2$ χρησιμοποιείται για ομαλοποίηση (regularization), ώστε οι τιμές του μητρώου \mathbf{W} να μην πάρουν μεγάλες τιμές. Παίρνοντας την παράγωγο της παραπάνω σχέσης ως προς \mathbf{W} και θέτοντας τη ίση με το μηδέν προκύπτει η λύση

$$\mathbf{W} = (\mathbf{X}^T \mathbf{X} + \lambda_e \mathbf{I})^{-1} \mathbf{X}^T \mathbf{B} \quad (13)$$

Πλέον εφόσον έχουμε στη διάθεση μας τη hash συνάρτηση, ο αλγόριθμος SSDH μπορεί να παράγει hash codes για νέα query αντικείμενα. Υποθέτοντας ότι έχουμε τοποθετήσει τα νέα query αντικείμενα στις γραμμές ενός μητρώου \mathbf{X}_{query} , τα αντίστοιχα hash codes \mathbf{B}_{query} λαμβάνονται ως εξής

$$\mathbf{B}_{query} = F(\mathbf{X}_{query}) = \text{sgn}(\mathbf{X}_{query} \mathbf{W}) \quad (14)$$

Έπειτα οι συγγραφείς, για να επιβεβαιώσουν την αποτελεσματικότητα της μεθόδου τους, εκτέλεσαν ένα σύνολο πειραμάτων σε πραγματικά σύνολα δεδομένων. Τα σύνολα δεδομένων που επελέγησαν ήταν τα εξής. Το MNIST που αποτελείται από 70.000 εικόνες διαστάσεων 28×28 και οι οποίες αναπαριστούν τα δεκαδικά ψηφία. Κάθε εικόνα αναπαρίσταται από ένα διάνυσμα χαρακτηριστικών 784×1 . Το MIRFlickr το οποίο αποτελείται από 25.000 εικόνες από τον ιστότοπο Flickr. Αυτό είναι ένα multi label σύνολο δεδομένων καθώς σε κάθε εικόνα ανατίθεται τουλάχιστον 1 από 24 ετικέτες. Εδώ κάθε εικόνα αναπαρίσταται από ένα διάνυσμα 150 διαστάσεων που προκύπτει από το ιστόγραμμα κάθε εικόνας.

Το σύνολο CIFAR-10 το οποίο αποτελείται από 60.000 εικόνες διαστάσεων 32×32 και κάθε εικόνα ανήκει σε μια εκ των 10 κλάσεων. Κάθε εικόνα αναπαρίσταται με ένα 512×1 GIST διάνυσμα χαρακτηριστικών. Το σύνολο CIFAR-100 το οποίο αποτελείται από 100 κλάσεις και 60.000 εικόνες και πάλι διαστάσεων 32×32 . Και εδώ κάθε εικόνα αναπαρίσταται από 1 512×1 GIST διάνυσμα χαρακτηριστικών. Σημειώνουμε πως τα σύνολα δεδομένων CIFAR-10 και CIFAR-100 αντλούν από το ίδιο σύνολο 80.000.000 εικόνων. Το σύνολο NUS-WIDE, το οποίο αποτελείται από 269.648 εικόνες και 21 κλάσεις. Αυτό το σύνολο δεδομένων είναι επίσης multi label καθώς εικόνα ανήκει τουλάχιστον σε 1 εκ των 21 κλάσεων. Εδώ κάθε εικόνα αναπαρίσταται από ένα 500 διαστάσεων διάνυσμα bag of words με λέξεις που περιγράφουν την εκάστοτε εικόνα. Τέλος το σύνολο ImageNet το οποίο αποτελείται από 1.200.000.000 εικόνες, όπου κάθε μια ανήκει σε μια εκ των 1000 κλάσεων. Κάθε εικόνα αναπαρίσταται με ένα 1000 διαστάσεων bag of words διάνυσμα.

Κάθε σύνολο δεδομένων χωρίζεται, όπως συνηθίζεται, σε ένα σύνολο εκπαίδευσης και ένα σύνολο ελέγχου (εδώ query set). Για το σύνολο NUS-WIDE επιλέγονται τυχαία 100 εικόνες από κάθε κλάση για το query set, ενώ για το ImageNet το query set κατασκευάζεται επιλέγοντας τυχαία 5% των δειγμάτων του. Επίσης σημειώνεται πως ενώ για τα single label σύνολα δεδομένων 2 αντικείμενα θεωρούνται ίδια όταν ανήκουν στην ίδια κλάση για τα multi label γίνεται η εξής υπόθεση. Αν 2 δείγματα έχουν έστω και μια κοινή ετικέτα, δηλαδή έναν άσο σε κοινή θέση στα διανύσματα ετικετών τους, τότε θεωρούνται σημασιολογικά όμοια.

Στη συνέχεια θα δούμε συνοπτικά τις 6 μεθόδους με τις οποίες συγκρίθηκε η τεχνική SSDH. Πρώτα από όλα με την μέθοδο LSH, που αναφέραμε και πιο πάνω, και η οποία κατακερματίζει τα αντικείμενα της βάσης με τέτοιο τρόπο ώστε να διασφαλίζεται πως η πιθανότητα σύγκρουσης είναι πολύ μεγαλύτερη για αντικείμενα που είναι εγγύτερα από ότι για αντικείμενα που δεν έχουν κάποια σχέση. Ακόμη με τη μέθοδο Iterative Quantization (ITQ) η οποία αντιστοιχεί σε ένα απλό σχήμα εναλλασσόμενης βελτιστοποίησης για την εύρεση εκείνης της περιστροφής που θα αντιστοιχίσει, το κεντραρισμένο γύρω από το 0, σύνολο δεδομένων στις ακμές ενός κεντραρισμένου γύρω από το 0 δυαδικού υπερκύβου. Οι μέθοδοι LSH και ITQ είναι οι μόνες μη επιβλεπόμενες μέθοδοι με τις οποίες θα συγκριθεί ο αλγόριθμος SSDH.

Όσον αφορά τις επιβλεπόμενες μεθόδους, πρώτα έχουμε το Kernel Based Supervised Hashing (KSH), το οποίο αξιοποιεί την ισοδυναμία μεταξύ βελτιστοποίησης του εσωτερικού γινομένου των hash codes και των Hamming αποστάσεων. Επίσης τη μέθοδο Supervised Discrete Hashing (SDH), η οποία παράγει τα hash codes λύνοντας ένα πρόβλημα γραμμικής κατηγοριοποίησης. Τη μέθοδο Natural Supervised Hashing (NSH), η οποία χρησιμοποιεί τα διανύσματα ετικετών σαν δυαδικά hash codes και επιχειρεί να μάθει hash codes με τέτοιο τρόπο ώστε αυτά να

έχουν όμοια δομή με τα διανύσματα ετικετών. Επιπλέον τη μέθοδο Fast Supervised Discrete Hashing (FSDH), η οποία εκτελεί ένα είδος παλινδρόμησης των ετικετών των δειγμάτων εκπαίδευσης στα αντίστοιχα hash codes, με στόχο την επιτάχυνση της διαδικασίας. Τέλος τη μέθοδο Column Sampling based Discrete Supervised (COSDISH) η οποία εξάγει τα hash codes με χρήση μόνο σημασιολογικής πληροφορίας.

Για όλα τα πειράματα ο αλγόριθμος SSDH εκπαιδεύτηκε για 5 επαναλήψεις. Για τον αλγόριθμο KSH ότι θα αναφέρουμε στη συνέχεια αφορά την εκπαίδευση του με μόλις 2000 δείγματα για όλα τα σύνολα δεδομένων, καθώς έχει τεράστια χρονική πολυπλοκότητα. Οι συγγραφείς σύγκριναν τις παραπάνω τεχνικές χρησιμοποιώντας σαν βασική μετρική τη Mean Average Precision (MAP). Το Average Precision για ένα query ορίζεται ως

$$AP = \frac{1}{R} \sum_{r=1}^n Precision(r) \delta(r),$$

όπου R είναι το πλήθος των πραγματικών κοντινότερων γειτόνων του query, $\delta(r)$ είναι 1 αν το r-οστό ανακτηθέν αντικείμενο είναι πραγματικός κοντινότερος γείτονας και 0 αλλιώς, ενώ $precision(r)$ είναι η ακρίβεια των κορυφαίων r ανακτηθέντων αντικειμένων. Η μετρική MAP όταν έχουμε M το πλήθος queries αντιστοιχεί στο μέσο όρο των αντίστοιχων AP κάθε query, δηλαδή θα είναι

$$MAP = \frac{1}{M} \sum_{i=1}^M AP_i.$$

Επίσης οι ερευνητές εξέτασαν τις precision-recall καμπύλες, τις οποίες έλαβαν μεταβάλλοντας την Hamming ακτίνα των ανακτηθέντων αντικειμένων και υπολογίζοντας την ακρίβεια, την ανάκληση και το πλήθος των ανακτηθέντων αντικειμένων. Η μεταβολή της Hamming ακτίνας αντιστοιχεί σε μεταβολή του αποδεκτού σφάλματος μέχρι το οποίο κάποιο αντικείμενο θεωρείται κοντινότερος γείτονας σε ένα αντικείμενο query. Τέλος, όσον αφορά τις μετρικές, εξέτασαν και τις topN precision καμπύλες, οι οποίες δείχνουν πως μεταβάλλεται η ακρίβεια ανάλογα με το πλήθος των κορυφαίων N αντικειμένων που επιστρέφονται κάθε φορά. Επιπλέον να αναφέρουμε πως όλες οι μέθοδοι εξετάστηκαν για μήκος hash codes από 8 έως 96 bits.

Τα αποτελέσματα σχετικά με την μετρική MAP έδειξαν πως ο αλγόριθμος SSDH έχει τη καλύτερη απόδοση σε σχέση με όλες τις άλλες τεχνικές και για όλα τα επιλεγμένα μήκη κωδικολέξεων. Όσον αφορά τη χρονική πολυπλοκότητα οι μη-επιβλεπόμενες τεχνικές LSH και ITQ απαιτούν τον λιγότερο χρόνο εκπαίδευσης. Όλες οι επιβλεπόμενες μέθοδοι σύγκρισης ωστόσο έχουν καλύτερα αποτελέσματα όσον αφορά την ακρίβεια. Επίσης σημειώνεται πως στα single label σύνολα δεδομένων (MNIST, CIFAR-10, 100) ο αλγόριθμος SSDH εκπαιδεύεται ιδιαίτερα αποδοτικά. Πιο συγκεκριμένα χρειάζεται λιγότερο χρόνο σε σχέση με τις άλλες επιβλεπόμενες μεθόδους. Ακόμη ο χρόνος εκπαίδευσης του SSDH φαίνεται να είναι εύρωστος στα μήκη των hash codes, σε αντίθεση με τις τεχνικές KSH, SDH και COSDISH για παράδειγμα, οι οποίες απαιτούν περισσότερο χρόνο όσο αυξάνεται το μήκος των hash codes.

Όσον αφορά το multi label σύνολο δεδομένων (MIRFlickr) τώρα, ο αλγόριθμος SSDH εκπαιδεύεται και εδώ γρήγορα. Επειδή όμως σε αυτήν την περίπτωση τα bits της δυαδικής αναπαράστασης μαθαίνονται ένα την φορά (bit by bit) χρειάζεται περισσότερο χρόνο για να μάθει τα hash codes από ότι για τα single label σύνολα

δεδομένων. Συγκριτικά με άλλες μεθόδους όμως που μαθαίνουν και αυτές τα bit των hash codes ένα τη φορά, όπως KSH,SDH και COSDISH, ο αλγόριθμος SSDH είναι ο πιο γρήγορος. Σε ένα γενικότερο πλαίσιο οι συγγραφείς παρατηρούν πως όλες οι προς σύγκριση μέθοδοι έχουν χειρότερη απόδοση στα πιο δύσκολα σύνολα δεδομένων, όπου το πλήθος των δειγμάτων και ετικετών αυξάνεται αρκετά.

Ως προς τις precision recall καμπύλες η μέθοδος SSDH δείχνει να τα πηγαίνει καλύτερα από όλες τις άλλες τεχνικές. Οι topN precision καμπύλες επιβεβαιώνουν και αυτές την ανωτερότητα της SSDH μεθόδου, όπως μαρτυρά η σημαντική αύξηση της απόδοσης όταν χρησιμοποιείται η προτεινόμενη τεχνική. Άξιο παρατήρησης είναι το γεγονός πως η μέθοδος SSDH έχει πολύ καλύτερα αποτελέσματα όταν το N, το πλήθος των αντικειμένων που επιστρέφονται δηλαδή, είναι μικρό. Αυτό σημαίνει πως η τεχνική SSDH επιστρέφει πολύ πιο όμοια με το query αντικείμενα στην αρχή, μια ιδιότητα εξαιρετικά θεμιτή και σημαντική για οποιοδήποτε μοντέλο ανάκτησης.

Η σύγκριση των παραπάνω μεθόδων στα 2 μεγαλύτερα σύνολα δεδομένων (NUS-WIDE και ImageNet) έγινε ξεχωριστά ώστε να εκτιμηθεί καλύτερα η κλιμακωσιμότητα της μεθόδου. Για το NUS-WIDE, η μέθοδος SSDH είχε τα καλύτερα MAP αποτελέσματα σε σχέση με όλες τις άλλες μεθόδους. Επίσης η εκπαίδευση του αλγορίθμου SSDH φαίνεται να είναι γρήγορη και να κλιμακώνει καλά. Η μόνη μέθοδος με που προσεγγίζει την απόδοση του SSDH είναι η COSDISH, αλλά την ίδια στιγμή η μέθοδος COSDISH απαιτεί πολύ περισσότερο χρόνο εκπαίδευσης, γεγονός που μαρτυρά πως ο αλγόριθμος SSDH κλιμακώνει πολύ πιο εύκολα. Τέλος όσον αφορά το NUS-WIDE, οι μέθοδοι NSH και FSDH, έχουν καλύτερους χρόνους εκπαίδευσης από τον αλγόριθμο SSDH όσο το μέγεθος των hash codes αυξάνεται. Αυτό γιατί οι συγκεκριμένες μέθοδοι μαθαίνουν όλα τα bits των hash codes ταυτόχρονα και όχι ένα την φορά όπως ο SSDH. Παρόλα αυτά όμως ο αλγόριθμος SSDH έχει καλύτερη απόδοση, όσον αφορά την μετρική MAP, και από τις 2 προαναφερθείσες τεχνικές με μόνη διαφορά το ελάχιστο αυξημένο χρονικό κόστος. Με βάση όλες τις παραπάνω παρατηρήσεις σχετικά με τα αποτελέσματα του αλγορίθμου SSDH στο σύνολο δεδομένων NUS-WIDE, οι συγγραφείς κατέληξαν στο συμπέρασμα πως η μέθοδος SSDH αποτελεί την πιο πρακτική τεχνική επιβλεπόμενου κατακερματισμού, ιδιαίτερα για δεδομένα μεγάλου όγκου.

Συγκρίνοντας όλες τις μεθόδους στο σύνολο ImageNet για ακόμη μια φορά η μέθοδος SSDH επιτυγχάνει πολύ καλύτερη ακρίβεια από όλες τις άλλες τεχνικές. Πρέπει να τονιστεί ωστόσο πως η απόδοση όλων των τεχνικών στο σύνολο ImageNet είναι πολύ χειρότερη (κατά ένα παράγοντα 100) από ότι στο σύνολο NUS-WIDE για παράδειγμα. Αυτό, επιχειρηματολογούν οι συγγραφείς, οφείλεται στο γεγονός πως το ImageNet είναι γενικά πολύ πιο πολύπλοκο σύνολο δεδομένων, καθώς περιέχει περισσότερα από 1.000.000 δείγματα και 1000 κλάσεις. Συγκριτικά με τη μέθοδο SSDH, οι μέθοδοι NSH και FSDH απαιτούν λιγότερο χρόνο εκπαίδευσης, εις βάρος ωστόσο της εξαιρετικά χαμηλής, έως και απαγορευτικής, τους απόδοσης. Η τεχνική COSDISH είναι η μόνη με απόδοση που προσεγγίζει αυτή του SSDH, αλλά ο χρόνος εκπαίδευσης της αυξάνεται εκθετικά όσο αυξάνεται το μήκος των hash codes. Τα παραπάνω αποτελέσματα στο ImageNet αποτελούν, σύμφωνα με τους συγγραφείς, τρανταχτή απόδειξη πως η τεχνική SSDH δεν αποτελεί μόνο την πιο πρακτική αλλά και την πιο κλιμακώσιμη μέθοδο επιβλεπόμενου hashing.

Συμπερασματικά όλα τα παραπάνω πειραματικά αποτελέσματα επιδεικνύουν την ανωτερότητα της μεθόδου SSDH, ως προς την αποτελεσματικότητα της, την απόδοση

της, το χρόνο εκπαίδευσης και τη κλιμακωσιμότητα της, σε σχέση με μερικές από τις πιο σύγχρονες και αποδοτικές επιβλεπόμενες και μη-επιβλεπόμενες μεθόδους κατακερματισμού. Φαίνεται λοιπόν πως αυτή η σχετικά απλή ιδέα που είχαν οι συγγραφείς, να αντικαταστήσουν δηλαδή το δυαδικό μητρώο \mathbf{B} με το γινόμενο $\mathbf{Y}\mathbf{G}$, ώστε εν συνεχεία να μπορούν να υπολογίσουν το γινόμενο $\mathbf{S}\mathbf{Y}$ μια φορά για να αποφύγουν τη βελτιστοποίηση με χρήση του \mathbf{S} σε κάθε βήμα, είχε αποτέλεσμα. Ωστόσο σημαντικό ρόλο στην καλή απόδοση της μεθόδου έχει και η ενσωμάτωση του μητρώου ετικετών \mathbf{Y} στη διαδικασία εκμάθησης των hash codes. Εξ' αιτίας του αποδοτικού εναλλασσόμενου αλγορίθμου βελτιστοποίησης που προτείνεται για την διακριτή εκμάθηση των hash codes και της προσεκτικής σχεδίασης της συνάρτησης σφάλματος οι ερευνητές κατάφεραν να αξιοποιήσουν περισσότερη σημασιολογική πληροφορία με αποτέλεσμα η μέθοδος τους να πετυχαίνει υψηλότερη ακρίβεια σε αποδεκτό χρονικό κόστος. Βλέπουμε ότι η μέθοδος SSDH μπορεί να χειριστεί αποδοτικά ακόμη και τα εξαιρετικά μεγάλα σύνολα δεδομένων (NUS-WIDE, ImageNet).

3. *Fast Scalable Supervised Hashing*

Παρά την επιτυχία της μεθόδου SSDH και όλα της τα πλεονεκτήματα δεν παύει να έχει και κάποια μειονεκτήματα. Με αφορμή λοιπόν κάποιες αδυναμίες που εντόπισαν στη μέθοδο SSDH οι συγγραφείς επανήλθαν με μια καινούρια εργασία [4], στην οποία προτείνουν μια νέα μέθοδο, η οποία αντιμετωπίζει τις αδυναμίες της μεθόδου SSDH. Οι βασικές αδυναμίες της μεθόδου SSDH είναι οι εξής. Πρώτον, όπως αναφέραμε και πιο πάνω η μέθοδος SSDH κατάφερε να μειώσει το χωρικό κόστος $O(n^2)$ που είχε η χρήση του μητρώου ομοιότητας \mathbf{S} στη διαδικασία βελτιστοποίησης, ώστε πλέον το χωρικό κόστος να εξαρτάται γραμμικά από το μέγεθος της εισόδου, δηλαδή να γίνει $O(n)$. Ακόμη και αυτή η γραμμική εξάρτηση από το μέγεθος της εισόδου ωστόσο, όταν μιλάμε για Big Data, μπορεί να είναι απαγορευτική. Επίσης η μέθοδος SSDH υιοθετεί μια στρατηγική μάθησης των hash codes ένα bit τη φορά, γεγονός που αυξάνει τη χρονική της πολυπλοκότητα ειδικά για μεγάλα σύνολα δεδομένων. Επιπλέον η μέθοδος SSDH αξιοποιεί μόνο σημασιολογική πληροφορία, τις ετικέτες δηλαδή, για να παράξει τα hash codes γεγονός που δεν την καθιστά ιδιαίτερα εύρωστη στο θόρυβο.

Τη μέθοδο που πρότειναν σε αυτήν την εργασία οι συγγραφείς την ονόμασαν Fast Scalable Supervised Hashing (FSSH) και όπως θα δούμε είναι πολύ καλύτερη από την SSDH, αφού δίνει λύση σε όλα τα μειονεκτήματα της τελευταίας που αναφέραμε πιο πάνω. Πρώτον ενσωματώνοντας το μεγάλο μητρώο ομοιότητας \mathbf{S} σε έναν ενδιάμεσο όρο του οποίου το μέγεθος είναι ανεξάρτητο του πλήθους των δεδομένων n , η FSSH γίνεται πιο κατάλληλη μέθοδος για δεδομένα μεγάλου όγκου. Δεύτερον αντί να μαθαίνει τα hash codes ένα bit τη φορά, όλα τα bits μαθαίνονται ταυτόχρονα κάνοντας την εκπαίδευση του αλγορίθμου FSSH εύρωστη ακόμη και σε μεγαλύτερα μεγέθη hash codes. Τέλος εκτός από το μητρώο ομοιότητας και τις ετικέτες αξιοποιεί και τα αρχικά χαρακτηριστικά των δεδομένων.

Για να καταλάβουμε πως αξιοποιούνται τα χαρακτηριστικά των αρχικών δεδομένων πρέπει πρώτα να δούμε την έννοια του Kernelization. Το Kernelization αποτελεί μια πολύ γνωστή και πολύ ισχυρή τεχνική στο χώρο του επιβλεπόμενου κατακερματισμού. Γίνεται ως εξής. Για κάθε αντικείμενο από τον αρχικό χώρο δεδομένων $\mathbf{x} \in R^d$ δειγματοληπτούνται τυχαία m σημεία, $\mathbf{o}_1, \mathbf{o}_2, \dots, \mathbf{o}_m$ και το \mathbf{x}

αντιστοιχίζεται σε έναν m-διάστατο χώρο χαρακτηριστικών. Η αντιστοίχιση σε αυτό τον νέο μη-γραμμικό χώρο χαρακτηριστικών γίνεται με τη χρήση μιας συνάρτησης πυρήνα (kernel). Στην παρούσα εργασία οι συγγραφείς υιοθέτησαν την συνάρτηση RBF kernel. Σε αυτή την περίπτωση κάθε αντικείμενο \mathbf{x} αντιστοιχίζεται στο νέο χώρο χαρακτηριστικών με χρήση της παρακάτω συνάρτησης.

$$\phi(\mathbf{x}) = \left[\exp\left(\frac{\|\mathbf{x} - \mathbf{o}_1\|^2}{\sigma}\right), \dots, \exp\left(\frac{\|\mathbf{x} - \mathbf{o}_m\|^2}{\sigma}\right) \right]$$

Η παράμετρος σ εδώ καλείται ‘πλάτος’ του πυρήνα. Οι συγγραφείς έθεσαν $m=1000$ και η παράμετρος σ εκτιμήθηκε με βάση τη μέση Ευκλίδεια απόσταση μεταξύ των δειγμάτων εκπαίδευσης. Ο λόγος που κάνουμε αυτή την μετατροπή των δεδομένων από τον αρχικό χώρο σε έναν άλλο είναι εξ’ αιτίας της πολυπλοκότητας που μπορεί να παρουσιάζουν τα αρχικά δεδομένα. Αυτή η μη-γραμμική αντιστοίχιση ευελπιστούμε να κάνει τα πολύπλοκα δεδομένα στον αρχικό χώρο, να γίνουν πιο εύκολα διαχωρίσιμα στο νέο μη-γραμμικό χώρο χαρακτηριστικών και τις σχέσεις μεταξύ τους πιο εύκολα κατανοητές και αξιοποιήσιμες.

Οι δυο κύριοι στόχοι των ερευνητών σε αυτήν την εργασία ήταν, πρώτον η αξιοποίηση και των n διαθέσιμων δεδομένων στην εκπαίδευση χωρίς καμία δειγματοληψία. Και δεύτερον ο σχεδιασμός στρατηγικών για διακριτή βελτιστοποίηση που θα επέτρεπε την εκμάθηση όλων των bit των hash codes ταυτόχρονα και όχι ένα bit τη φορά. Ο σχεδιασμός της συνάρτησης σφάλματος για την εκμάθηση των hash codes έγινε με βάση αυτούς τους δυο πυλώνες όπως θα δούμε παρακάτω.

Οι hash συναρτήσεις που αντιστοιχίζουν τα αρχικά δεδομένα στο χώρο των hash codes και περιγράφονται από την Εξ.(14) τροποποιούνται λίγο και γίνονται

$$\mathbf{F}(\mathbf{X}) = \text{sgn}(\phi(\mathbf{X})\mathbf{W}) = \mathbf{B} \quad (15)$$

καθώς τώρα χρησιμοποιείται η αναπαράσταση των δεδομένων στο νέο μη-γραμμικό χώρο χαρακτηριστικών $\phi(\mathbf{X})$. Πρώτο μέλημα στο σχεδιασμό της συνάρτησης σφάλματος ήταν η διασφάλιση της ποιότητας των χαρακτηριστικών που παράγουν οι kernel που χρησιμοποιούνται, το οποίο επιτυγχάνεται με την εξής συνάρτηση σφάλματος

$$\min_{\mathbf{W}} \|\mathbf{B} - \phi(\mathbf{X})\mathbf{W}\|_F^2, s. t. \mathbf{B} \in \{-1,1\}^{n \times r} \quad (16)$$

Και εδώ οι συγγραφείς θέλουν να αντικαταστήσουν το δυαδικό μητρώο \mathbf{B} με ένα μητρώο πραγματικών τιμών, το οποίο θα φέρει περισσότερη πληροφορία και θα επιτρέψει την πιο ακριβή προσέγγιση του μητρώου ομοιότητας. Για το σκοπό αυτό επαναχρησιμοποιούνται οι πραγματικών τιμών αντιστοιχίσεις των kernel χαρακτηριστικών $\phi(\mathbf{X})\mathbf{W}$, ώστε να προσεγγιστεί το μητρώο ομοιότητας, ορίζοντας το εξής πρόβλημα βελτιστοποίησης

$$\min_{\mathbf{B}, \mathbf{W}} \|\mathbf{S} - (\phi(\mathbf{X})\mathbf{W})\mathbf{B}^T\|_F^2, s. t. \mathbf{B} \in \{-1,1\}^{n \times r} \quad (17)$$

Συνδυάζοντας τις δυο παραπάνω εξισώσεις προκύπτει η συνάρτηση σφάλματος μέχρι αυτό το σημείο.

$$\min_{\mathbf{B}, \mathbf{W}} \|\mathbf{S} - (\phi(\mathbf{X})\mathbf{W})\mathbf{B}^T\|_F^2 + \mu \|\mathbf{B} - \phi(\mathbf{X})\mathbf{W}\|_F^2, s. t. \mathbf{B} \in \{-1,1\}^{n \times r} \quad (18)$$

Και ενώ όπως παρατηρούν οι συγγραφείς το μητρώο ομοιότητας \mathbf{S} μπορεί να κατασκευαστεί από το μητρώο των ετικετών, παρόλα αυτά η χρήση του δεν είναι τετριμμένη. Αυτό γιατί, όπως θα δούμε, περιέχει επιπλέον πληροφορία που δεν μπορεί να ‘αντικατασταθεί’ από το μητρώο ετικετών και μόνο. Να σημειώσουμε πως στα πλαίσια της παρούσας εργασίας το μητρώο ετικετών συμβολίζεται πλέον με \mathbf{L} . Οι ετικέτες μεταξύ 2 αντικειμένων μπορεί να είναι περισσότερο όμοιες, να περιέχουν δηλαδή περισσότερους κοινούς άσους στην ίδια θέση, από ότι οι ετικέτες 2 άλλων αντικειμένων. Αυτή η ποσοτικοποίηση του πότε 2 αντικείμενα είναι περισσότερο όμοια από 2 άλλα δεν μπορεί να εκφραστεί μόνο με το δυαδικό μητρώο ομοιότητας \mathbf{S} το οποίο μας λέει απλά αν είναι ή δεν είναι 2 αντικείμενα όμοια. Για να αξιοποιήσουν λοιπόν και αυτήν την επιπλέον πληροφορία οι συγγραφείς επιχείρησαν να ενσωματώσουν και το μητρώο ετικετών \mathbf{L} στην μάθηση των hash codes, υποθέτοντας πως οι ετικέτες μπορούν να αντιστοιχιστούν στο χώρο των hash codes. Το αντίστοιχο πρόβλημα βελτιστοποίησης έχει ως εξής

$$\min_{\mathbf{B}, \mathbf{G}} \|\mathbf{B} - \mathbf{L}\mathbf{G}\|_F^2, s. t. \mathbf{B} \in \{-1, 1\}^{n \times r} \quad (19)$$

Όπου το μητρώο $\mathbf{G} \in R^{c \times r}$ είναι και εδώ το μητρώο που εκτελεί την προβολή των ετικετών στο χώρο των hash codes. Συνδυάζοντας την Εξ. (18) με την Εξ.(19) προκύπτει η τελική συνάρτηση σφάλματος της μεθόδου FSSH η οποία έχει ως εξής.

$$\min_{\mathbf{B}, \mathbf{G}, \mathbf{W}} \|\mathbf{S} - \phi(\mathbf{X})\mathbf{W}(\mathbf{L}\mathbf{G})^T\|_F^2 + \mu \|\mathbf{B} - \mathbf{L}\mathbf{G}\|_F^2 + \theta \|\mathbf{B} - \phi(\mathbf{X})\mathbf{W}\|_F^2 \quad (20)$$

$$s. t. \mathbf{B} \in \{-1, 1\}^{n \times r}$$

Υπάρχουν αρκετά πράγματα άξια παρατήρησης στη παραπάνω συνάρτηση. Ίσως το σημαντικότερο και πιο εμφανές είναι πως αξιοποιείται και το μητρώο ομοιότητας \mathbf{S} και το μητρώο ετικετών \mathbf{L} στην μάθηση των hash codes, γεγονός που κάνει τα παραγόμενα hash codes του μητρώου \mathbf{B} πολύ πιο πιθανό να διατηρούν τη σημασιολογική ομοιότητα. Ωστόσο όπως θα δούμε ευθύς αμέσως παρά το γεγονός ότι το μητρώο \mathbf{S} περιλαμβάνεται στη συνάρτηση σφάλματος δεν θα χρειαστεί να υπολογιστεί άμεσα. Πράγματι εισάγοντας έναν ενδιάμεσο όρο που προ-υπολογίζεται με βάση το \mathbf{S} μόνο μια φορά και κάποιους άλλους όρους με πολύ μικρότερο χωρικό κόστος αποφεύγεται το μεγάλο κόστος χρήσης του μητρώου \mathbf{S} .

Βλέπουμε πως η συνάρτηση σφάλματος της Εξ. (20) απαιτεί βελτιστοποίηση ως προς 3 όρους, τα μητρώα \mathbf{W} , \mathbf{G} , και \mathbf{B} . Ο αλγόριθμος βελτιστοποίησης που χρησιμοποιείται εναλλάσσεται μεταξύ 3 βημάτων. Πρώτον κρατάμε σταθερά τα μητρώα \mathbf{G} και \mathbf{B} και βελτιστοποιούμε ως προς \mathbf{W} , δεύτερον κρατώντας σταθερά τα μητρώα \mathbf{W} και \mathbf{B} ενημερώνουμε τα στοιχεία του μητρώου \mathbf{G} και τρίτον βελτιστοποιούμε ως προς \mathbf{B} υποθέτοντας σταθερά τα \mathbf{W} και \mathbf{G} .

Για το πρώτο βήμα που αφορά τη βελτιστοποίηση ως προς \mathbf{W} , όταν τα \mathbf{B} και \mathbf{G} είναι σταθερά η Εξ. (20) απλοποιείται στη μορφή

$$\min_{\mathbf{W}} \|\mathbf{S} - \phi(\mathbf{X})\mathbf{W}(\mathbf{L}\mathbf{G})^T\|_F^2 + \theta \|\mathbf{B} - \phi(\mathbf{X})\mathbf{W}\|_F^2$$

Και θέτοντας τη παράγωγο ως προς \mathbf{W} της παραπάνω εξίσωσης ίση με 0 η λύση προκύπτει

$$\mathbf{W} = (\phi(\mathbf{X})^T \phi(\mathbf{X}))^{-1} (\phi(\mathbf{X})^T \mathbf{S} \mathbf{L} \mathbf{G} + \theta \phi(\mathbf{X})^T \mathbf{B}) (\mathbf{G}^T \mathbf{L}^T \mathbf{L} \mathbf{G} + \theta \mathbf{I}_{r \times r})^{-1} = \mathbf{C}^{-1} (\mathbf{A} \mathbf{G} + \theta \phi(\mathbf{X})^T \mathbf{B}) (\mathbf{G}^T \mathbf{D} \mathbf{G} + \theta \mathbf{I}_{r \times r})^{-1} \quad (21)$$

Όπου $\mathbf{A} = \phi(\mathbf{X})^T \mathbf{S} \mathbf{L}$, $\mathbf{C} = \phi(\mathbf{X})^T \phi(\mathbf{X})$ και $\mathbf{D} = \mathbf{L}^T \mathbf{L}$. Τα μητρώα \mathbf{C}^{-1} και \mathbf{D} μπορούν να υπολογιστούν μια φορά πριν τη βελτιστοποίηση. Επίσης είναι ακριβώς αυτός ο ενδιάμεσος όρος $\mathbf{A} \in R^{m \times c}$ που επιτρέπει να αποφύγουμε την απευθείας βελτιστοποίηση με το μεγάλο μητρώο \mathbf{S} . Προφανώς το μέγεθος του \mathbf{A} είναι σταθερό και ανεξάρτητο του πλήθους των δειγμάτων n , που σημαίνει πως το \mathbf{A} μπορεί να υπολογιστεί εξαιρετικά αποδοτικά πριν την εκπαίδευση. Πλέον το χωρικό κόστος $O(n^2)$ για το \mathbf{S} μειώνεται δραματικά σε $O(mc)$. Αυτό περιμένουμε να έχει και μεγάλο όφελος στο χρόνο και στην απόδοση της εκπαίδευσης.

Για το δεύτερο βήμα της βελτιστοποίησης που αφορά την ενημέρωση του μητρώου \mathbf{G} , η Εξ. (20) μπορεί να ξαναγραφεί

$$\min_{\mathbf{G}} \|\mathbf{S} - \phi(\mathbf{X}) \mathbf{W} (\mathbf{L} \mathbf{G})^T\|_F^2 + \mu \|\mathbf{B} - \mathbf{L} \mathbf{G}\|_F^2$$

Θέτοντας τη παράγωγο ως προς \mathbf{G} τώρα ίση με 0 προκύπτει η λύση

$$\mathbf{G} = (\mathbf{L}^T \mathbf{L})^{-1} (\mu \mathbf{L}^T \mathbf{B} + \mathbf{L}^T \mathbf{S}^T \phi(\mathbf{X}) \mathbf{W}) (\mathbf{W}^T \phi(\mathbf{X})^T \phi(\mathbf{X}) \mathbf{W} + \mu \mathbf{I}_{r \times r})^{-1} = \mathbf{D}^{-1} (\mu \mathbf{L}^T \mathbf{B} + \mathbf{A}^T \mathbf{W}) (\mathbf{W}^T \mathbf{C} \mathbf{W} + \mu \mathbf{I}_{r \times r})^{-1} \quad (22)$$

Όπου $\mathbf{A} = \phi(\mathbf{X})^T \mathbf{S} \mathbf{L}$, $\mathbf{C} = \phi(\mathbf{X})^T \phi(\mathbf{X})$ $\mathbf{D} = \mathbf{L}^T \mathbf{L}$. Όπως και στο προηγούμενο βήμα έτσι και εδώ τα μητρώα \mathbf{C} και \mathbf{D}^{-1} μπορεί να υπολογιστούν μια φορά πριν την εκπαίδευση, οπότε περιμένουμε και η παραπάνω λύση να είναι αποδοτική.

Για το τρίτο και τελευταίο βήμα της βελτιστοποίησης που αφορά τη βελτιστοποίηση ως προς \mathbf{B} , όταν τεθούν σταθερά τα μητρώα \mathbf{G} και \mathbf{W} η Εξ. (20) γίνεται

$$\min_{\mathbf{B}} \mu \|\mathbf{B} - \mathbf{L} \mathbf{G}\|_F^2 + \theta \|\mathbf{B} - \phi(\mathbf{X}) \mathbf{W}\|_F^2$$

$$s. t. \mathbf{B} \in \{-1, 1\}^{n \times r}$$

η οποία μπορεί να ξαναγραφεί ως

$$\begin{aligned} \min_{\mathbf{B}} \mu \text{Tr}((\mathbf{B} - \mathbf{L} \mathbf{G})^T (\mathbf{B} - \mathbf{L} \mathbf{G})) + \theta \text{Tr}((\mathbf{B} - \phi(\mathbf{X}) \mathbf{W})^T (\mathbf{B} - \phi(\mathbf{X}) \mathbf{W})) \\ = (\mu + \theta) \|\mathbf{B}\|_F^2 - 2\mu \text{Tr}(\mathbf{B}^T \mathbf{L} \mathbf{G}) + \mu \|\mathbf{L} \mathbf{G}\|_F^2 - \theta \text{Tr}(\mathbf{B}^T \phi(\mathbf{X}) \mathbf{W}) \\ + \theta \|\phi(\mathbf{X}) \mathbf{W}\|_F^2 \\ s. t. \mathbf{B} \in \{-1, 1\}^{n \times r} \end{aligned}$$

Επειδή $\|\mathbf{B}\|_F^2$, $\|\mathbf{L} \mathbf{G}\|_F^2$ και $\|\phi(\mathbf{X}) \mathbf{W}\|_F^2$ είναι σταθεροί όροι η παραπάνω εξίσωση είναι ισοδύναμη με το ακόλουθο πρόβλημα

$$\min_{\mathbf{B}} -\text{Tr}(\mathbf{B}^T (\mu \mathbf{L} \mathbf{G} + \theta \phi(\mathbf{X}) \mathbf{W})), s. t. \mathbf{B} \in \{-1, 1\}^{n \times r}$$

Η λύση κλειστής μορφής της παραπάνω εξίσωσης είναι

$$\mathbf{B} = \text{sgn}(\mu \mathbf{L} \mathbf{G} + \theta \phi(\mathbf{X}) \mathbf{W}) \quad (23)$$

Το εξαιρετικά σημαντικό εδώ είναι να παρατηρήσουμε πως στην Εξ. (23) απαιτείται μόνο ένα βήμα για την εκμάθηση όλων των hash codes ταυτόχρονα.

Επομένως σε σύγκριση με τη μέθοδο SSDH όπου τα hash codes μαθαίνονται ένα bit τη φορά, η μέθοδος FSSH μπορεί να μάθει τα δυαδικά hash codes πολύ πιο γρήγορα.

Συνοψίζοντας, ο αλγόριθμος βελτιστοποίησης της μεθόδου FSSH έχει ως εξής. Δοθέντων του μητρώου ετικετών \mathbf{L} , των παραμέτρων μ και θ , του μεγέθους των hash codes, και του πλήθους των επαναλήψεων υπολογίζουμε αρχικά μια φορά πριν την εκπαίδευση το μητρώο $\mathbf{A} = \phi(\mathbf{X})^T \mathbf{S} \mathbf{L}$. Εν συνεχεία αρχικοποιούνται τυχαία τα μητρώα \mathbf{W} , \mathbf{G} και \mathbf{B} και επαναλαμβάνουμε εναλλάξ τα τρία βήματα της βελτιστοποίησης. Κρατάμε σταθερά τα \mathbf{G} και \mathbf{B} και ενημερώνουμε το \mathbf{W} με την Εξ. (21). Κρατάμε σταθερά τα \mathbf{W} και \mathbf{B} και ενημερώνουμε το \mathbf{G} με την Εξ. (22). Κρατάμε σταθερά τα \mathbf{W} και \mathbf{G} και ενημερώνουμε το \mathbf{B} με την Εξ. (23). Τελικά σαν έξοδο θα λάβουμε τα μητρώα \mathbf{W} και \mathbf{B} .

Αξίζει επίσης να παρατηρήσουμε πως οι λύσεις κλειστής μορφής που προκύπτουν και στα τρία στάδια της διαδικασίας βελτιστοποίησης παρέχουν 2 σημαντικά πλεονεκτήματα. Πρώτον ότι ο αλγόριθμος είναι σε θέση να μαθαίνει ακριβή hash codes. Δεύτερον πως ο αλγόριθμος θα συγκλίνει γρήγορα. Εξάλλου όλα τα bits του \mathbf{B} μπορούν να υπολογιστούν αποδοτικά σε ένα μόνο βήμα, γεγονός που κάνει τον αλγόριθμο εξαιρετικά εύρωστο ειδικά όταν απαιτείται η εκμάθηση μεγαλύτερου μήκους hash codes.

Μέχρι στιγμής δεν έχουμε αναφέρει αν η μέθοδος FSSH ανήκει στην κατηγορία των one-step ή στη κατηγορία των two-step μεθόδων κατακερματισμού. Αυτό γιατί όπως θα δούμε μπορούν να υλοποιηθούν 2 παραλλαγές της μεθόδου ώστε να εκτελείται η εκμάθηση των hash codes και hash συναρτήσεων σε ένα βήμα ή σε δυο ξεχωριστά βήματα. Συμβολίζουμε, όπως και πριν, με \mathbf{X}_{query} και \mathbf{B}_{query} τα αρχικά χαρακτηριστικά και τα δυαδικά hash codes των εισερχόμενων queries αντίστοιχα.

Στην εργασία η παραλλαγή όπου τα δυαδικά hash codes και οι συναρτήσεις hash μαθαίνονται σε ένα βήμα αναφέρεται σαν FSSH_os. Εδώ χρησιμοποιείται το μητρώο προβολής \mathbf{W} που έχει προκύψει από την βελτιστοποίηση της Εξ. (20). Με απλά λόγια ταυτόχρονη εκμάθηση των hash codes και hash συναρτήσεων σημαίνει πως επιστρέφονται την ίδια στιγμή τα μητρώα \mathbf{W} και \mathbf{B} . Όταν λοιπόν έρχονται νέα queries, τα δυαδικά hash codes προκύπτουν από τη σχέση

$$\mathbf{B}_{query} = \text{sgn}(\phi(\mathbf{X}_{query})\mathbf{W})$$

Η παραλλαγή 2-βημάτων της μεθόδου αναφέρεται σαν FSSH_ts και εδώ δεν αξιοποιείται το μητρώο \mathbf{W} . Αντίθετα ακολουθείται η στρατηγική που ακολουθείται σε όλες τις two-step μεθόδους κατακερματισμού. Πιο αναλυτικά εκπαιδεύονται r δυαδικοί κατηγοριοποιητές με βάση το μητρώο χαρακτηριστικών $\phi(\mathbf{X})$ και το μητρώο των δυαδικών hash codes \mathbf{B} , όπου σε κάθε bit αντιστοιχίζεται ένας κατηγοριοποιητής. Και ενώ μπορεί να χρησιμοποιηθεί οποιοσδήποτε κατηγοριοποιητής, για το FSSH_ts επιλέχθηκε η γραμμική παλινδρόμηση, κυρίως λόγω της αποδοτικότητας της. Το μητρώο προβολής \mathbf{P} το οποίο θα εκτελεί την προβολή από τον αρχικό χώρο χαρακτηριστικών $\phi(\mathbf{X})$ στο χώρο των hash codes λαμβάνεται λύνοντας ορίζοντας το ακόλουθο πρόβλημα

$$\|\mathbf{B} - \phi(\mathbf{X})\mathbf{P}\|_F^2 + \lambda_e \|\mathbf{P}\|_F^2$$

Όπου ο δεύτερος όρος $\|\mathbf{P}\|_F^2$ χρησιμοποιείται απλά για να μην λάβει μεγάλες τιμές το μητρώο \mathbf{P} και η λ_e αποτελεί παράμετρο ισορροπίας μεταξύ των 2 στόχων.

Παίρνοντας την παράγωγο της παραπάνω συνάρτησης ως προς \mathbf{P} και θέτοντας τη ίση με 0 προκύπτει η λύση

$$\mathbf{P} = (\phi(\mathbf{X})^T \phi(\mathbf{X}) + \lambda_e \mathbf{I})^{-1} \phi(\mathbf{X})^T$$

Οπότε για τα νέα queries τώρα τα δυαδικά hash codes παράγονται ως εξής

$$\mathbf{B}_{query} = \text{sgn}(\phi(\mathbf{X}_{query})\mathbf{P})$$

Και οι δυο παραλλαγές της μεθόδου FSSH συγκρίθηκαν χρησιμοποιώντας τις ίδιες ακριβώς μετρικές με τα ίδια σύνολα δεδομένων που συγκρίθηκε και η μέθοδος SSDH. Όσον αφορά τη μετρική MAP τα αποτελέσματα ήταν ανώτερα σε όλα τα σύνολα δεδομένων και για όλα τα μήκη hash codes. Αντίστοιχα και οι precision-recall και topN-precision καμπύλες μαρτυρούν την πολύ ανώτερη απόδοση και ευρωστία της μεθόδου FSSH. Εμείς ωστόσο θα σταθούμε περισσότερο στη σύγκριση μεταξύ των 2 μεθόδων SSDH και FSSH.

Το κίνητρο πίσω και από τις 2 μεθόδους ήταν το ίδιο. Ωστόσο η μέθοδος FSSH παρουσιάζει αρκετά πλεονεκτήματα έναντι της SSDH. Πρώτον η FSSH μπορεί να βελτιστοποιήσει για όλα τα bits σε ένα μόνο βήμα, αντίθετα από την SSDH που μαθαίνει ένα συγκεκριμένο bit για όλα τα αντικείμενα σε κάθε βήμα. Δεύτερον η μέθοδος SSDH μειώνει το χωρικό κόστος από $n \times n$ σε $n \times c$, το οποίο παραμένει μεγάλο και σχετικό με το πλήθος των δεδομένων, ενώ η μέθοδος FSSH καταφέρνει να ενσωματώσει το $n \times n$ μητρώο ομοιότητα \mathbf{S} σε έναν ενδιάμεσο όρο μεγέθους $m \times c$. Και οι 2 όροι m και c είναι σταθεροί, και στη συντριπτική πλειοψηφία των περιπτώσεων πολύ μικρότεροι του πλήθους των δεδομένων n . Τρίτον η μέθοδος SSDH παράγει τα hash codes χωρίς χρησιμοποιεί τα χαρακτηριστικά των δεδομένων αυτά κάθε αυτά παρά μόνο τις σχέσεις μεταξύ τους, ενώ η μέθοδος FSSH από την άλλη αξιοποιεί και τις σχέσεις μεταξύ των δεδομένων, και τη σημασιολογική πληροφορία από τις ετικέτες και τα χαρακτηριστικά των δεδομένων.

Στο πειραματικό κομμάτι τώρα συγκρίθηκαν οι μέθοδος SSDH με τη 2 βημάτων παραλλαγή FSSH_ts ώστε η σύγκριση να είναι πιο ‘δίκαιη’ αφού και η μέθοδος SSDH είναι μέθοδος 2 βημάτων. Οι δυο τεχνικές συγκρίθηκαν ως προς την απόδοση τους όσον αφορά τη μετρική MAP στα σύνολα δεδομένων CIFAR-10, MNIST και NUS-WIDE και για μήκη hash codes 16, 32, 64 και 96 bits και όσον αφορά το χρόνο εκπαίδευσης τους στο σύνολο δεδομένων NUS-WIDE, καθώς θεωρείται αρκούντως μεγάλο. Με εξαίρεση το σύνολο δεδομένων CIFAR-10 και για μήκη 16 και 32 bits, η μέθοδος FSSH επιφέρει τα καλύτερα MAP αποτελέσματα σε όλες τις περιπτώσεις. Αλλά και σε αυτές τις 2 υπό-περιπτώσεις όπου η τεχνική SSDH τα πηγαίνει καλύτερα, η απόδοση της μεθόδου FSSH είναι εξαιρετικά κοντά, ενώ ταυτόχρονα απαιτεί πολύ λιγότερο χώρο.

Όσον αφορά το χρόνο εκπαίδευσης η μέθοδος FSSH μπορεί να εκπαιδευτεί πολύ πιο γρήγορα από την SSDH στο NUS-WIDE. Ενώ ο αλγόριθμος SSDH καταναλώνει όλο και περισσότερο χρόνο όσο αυξάνεται το μήκος των hash codes, ο χρόνος εκπαίδευσης του αλγορίθμου FSSH παραμένει εξαιρετικά εύρωστος στο μήκος των hash codes. Ενδεικτικά αναφέρουμε πως για μήκος hash code 16 bit στο NUS-WIDE ο αλγόριθμος SSDH απαιτεί 14.5 δευτερόλεπτα και ο αλγόριθμος FSSH 13.7, για μήκος hash code 96 bit ο αλγόριθμος SSDH απαιτεί 130 δευτερόλεπτα ενώ ο

αλγόριθμος FSSH μόλις 15.5. Το τελικό συμπέρασμα των ερευνητών σχετικά με τις δυο τεχνικές είναι πως η μέθοδος FSSH αποτελεί την πιο πρακτική επιλογή ειδικά σε περιπτώσεις όπου απαιτούνται μεγαλύτερα μήκη hash codes σε σύνολα δεδομένων μεγάλου όγκου.

Επιπλέον με αφορμή την ευρεία χρήση και τα πολλά υποσχόμενα αποτελέσματα που παρουσιάζουν τα νευρωνικά δίκτυα σε διάφορους τομείς, οι ερευνητές αναρωτήθηκαν αν θα μπορούσαν ίσως να εκμεταλλευτούν την καλή τους απόδοση, ορίζοντα μια νέα παραλλαγή της μεθόδου τους, την FSSH_deep. Αυτή η παραλλαγή αντιστοιχεί σε στρατηγική hashing 2 βημάτων αφού η χρήση των νευρωνικών δικτύων αφορά το 2^ο βήμα εύρεσης των hash συναρτήσεων. Σημειώνεται πως όλες οι παραλλαγές της μεθόδου FSSH, συμπεριλαμβανομένης και της FSSH_deep δεν αποτελούν κλειστά end-to-end μοντέλα. Αυτό γιατί οι ερευνητές έδωσαν περισσότερη έμφαση στην επιλογή του κριτηρίου για την παραγωγή των hash codes, δηλαδή στο σχεδιασμό της συνάρτησης σφάλματος και του αλγορίθμου βελτιστοποίησης. Ωστόσο πιστεύουν πως η μέθοδος τους FSSH μπορεί να μετασχηματιστεί σχετικά εύκολα σε μια end-to-end μέθοδο.

Θα αναλύσουμε τώρα λίγο πιο διεξοδικά πως ακριβώς λειτουργεί η παραλλαγή FSSH_deep. Πρώτα από όλα έχουμε αναφέρει πως στις μεθόδους κατακερματισμού 2 βημάτων οποιοδήποτε προγνωστικό μοντέλο μπορεί να χρησιμοποιηθεί σαν hash συνάρτηση στο 2^ο βήμα, συμπεριλαμβανομένων και των νευρωνικών δικτύων. Στη παρούσα εργασία χρησιμοποιήθηκε ένα Συνελικτικό νευρωνικό δίκτυο, και πιο συγκεκριμένα το CNN-F [5]. Το δίκτυο εκπαιδεύτηκε λύνοντας ένα πολλών κατηγοριών πρόβλημα κατηγοριοποίησης και βελτιστοποιώντας τη συνάρτηση σφάλματος

$$L = -\frac{1}{n} \sum_{i=1}^n [b_i \log p_i + (1 - b_i \log(1 - p_i))]$$

Όπου b_i είναι το hash code του i -οστού δείγματος εκπαίδευσης έχοντας αντικαταστήσει τα -1 με το 0. Επίσης $p_i = \sigma(\mathbf{z}_i)$ όπου σ η σιγμοειδής συνάρτηση ενεργοποίησης και \mathbf{z}_i η έξοδος του δικτύου. Οι παράμετροι θ , τα βάρη δηλαδή του νευρωνικού δικτύου, μαθαίνονται με χρήση του αλγορίθμου Πίσω Διάδοσης του σφάλματος (Back Propagation) και του αλγορίθμου βελτιστοποίησης Stochastic Gradient Descent (SGD) για την μείωση της συνάρτησης σφάλματος. Πιο συγκεκριμένα η παράγωγος της συνάρτησης σφάλματος ως προς την έξοδο του δικτύου \mathbf{z}_i είναι $\frac{\partial L}{\partial \mathbf{z}_i} = \frac{1}{n}(\mathbf{p}_i - \mathbf{b}_i)$. Οι όροι $\frac{\partial L}{\partial \theta}$ και $\frac{\partial L}{\partial \mathbf{z}_i}$ υπολογίζονται με χρήση του κανόνα της αλυσίδας για την παράγωγο, ώστε να ενημερωθούν τα βάρη θ του δικτύου. Αφού ολοκληρωθεί η εκπαίδευση του νευρωνικού, τα hash codes για τα νέα query που έρχονται παράγονται με χρήση της σχέσης

$$\mathbf{B}_{query} = \text{sgn}(\mathbf{Z}_{query})$$

Όπου \mathbf{Z}_{query} είναι η έξοδος του νευρωνικού δικτύου αν στην είσοδο τροφοδοτήσουμε τα νέα αντικείμενα των queries. Επιπλέον, εκτός του πρώτου βήματος της τεχνικής FSSH_deep, τα χαρακτηριστικά \mathbf{X} εξάγονται πλέον από ένα προ-εκπαιδευμένο CNN-F δίκτυο. Επειδή στα πλαίσια της εργασίας το μόνο νευρωνικό δίκτυο που δοκιμάστηκε ήταν το CNN-F, οι συγγραφείς σημειώνουν πως ίσως θα ήταν

θεμιτό να εξεταστεί η χρήση και διαφορετικών μοντέλων νευρωνικών δικτύων στη θέση του CNN-F.

Οι ερευνητές σύγκριναν πειραματικά τη μέθοδο FSSH_deep με άλλες παρόμοιες μεθόδους κατακερματισμού 2 βημάτων που χρησιμοποιούν νευρωνικά δίκτυα σαν hash συναρτήσεις. Η σύγκριση έγινε στο σύνολο δεδομένων CIFAR-10 μεταξύ των μεθόδων DSRH, DSCH, DRSH, DPSH, VDSH, DTSH και DSDH. Η απόδοση της FSSH_deep όσο αφορά τη μετρική MAP για μήκη hash codes 24, 32 και 48 bit ήταν αρκετά καλή. Ωστόσο κάποιες από τις παραπάνω μεθόδους έχουν καλύτερη απόδοση. Αυτό οφείλεται στο γεγονός πως η εξαγωγή των χαρακτηριστικών στο πρώτο βήμα της μεθόδου είναι ανεξάρτητη από τη διαδικασία μάθησης των hash codes. Άρα είναι πιθανό τα χαρακτηριστικά που εξάγει το συνελκτικό νευρωνικό δίκτυο να μην είναι συμβατά με τη διαδικασία του κατακερματισμού. Αντίθετα όλες οι υπόλοιπες μέθοδοι εκτελούν ταυτόχρονα την μάθηση των χαρακτηριστικών και την μάθηση των hash codes. Αποτελούν με άλλα λόγια end-to-end μοντέλα. Και παρόλο που η μέθοδος FSSH_deep δεν είναι end-to-end, καταφέρνει να επιτυγχάνει ανταγωνιστική απόδοση, γεγονός που μαρτυρά την αποτελεσματικότητά της. Τέλος οι συγγραφείς θεωρούν πως εάν καταφέρουν να κάνουν την μέθοδο τους FSSH_deep end-to-end τότε η απόδοση θα βελτιωθεί περαιτέρω.

4. Εναλλακτικές Μέθοδοι Επιβλεπόμενου Κατακερματισμού

4.1. Supervised Short Length Hashing

Η βασική μέθοδος SSDH που είδαμε εμείς σχεδιάστηκε με τέτοιο τρόπο ώστε να είναι αποδοτική σε μεγάλα σύνολα δεδομένων και δοκιμάστηκε για μήκη hash codes από 16 έως 96 bits. Δεν αναφερθήκαμε όμως καθόλου στο πως το μέγεθος των hash codes μπορεί να επηρεάσει τα αποτελέσματα της μεθόδου. Όταν απαιτείται η χρήση πολύ μικρών hash codes τότε η απόδοση των περισσότερων μεθόδων, συμπεριλαμβανομένης και της SSDH, πάύει να είναι βέλτιστη. Λύση σε αυτό το πρόβλημα έρχεται να δώσει μια άλλη επιβλεπόμενη τεχνική που καλείται Supervised short-length hashing (SSLH) [6]. Το σημείο κλειδί της μεθόδου SSLH είναι η κοινή ανακατασκευή μεταξύ των μικρών hash codes και των αρχικών χαρακτηριστικών, ώστε να μειωθεί το σφάλμα σημασιολογικής πληροφορίας. Για την επίτευξη καλύτερης απόδοσης η μέθοδος SSLH συνδυάζει τη γραμμική παλινδρόμηση, με τεχνικές παραγοντοποίησης μητρώου και διακριτής βελτιστοποίησης.

Επίσης με στόχο να αυξηθεί η ευρωστία και η ακρίβεια των hash αναπαραστάσεων, προστίθεται ένας εύρωστος εκτιμητής ώστε να αξιοποιηθεί πλήρως ότι πληροφορία υπάρχει σχετικά με τις ετικέτες των δειγμάτων. Εκτενή πειράματα που εκτελέστηκαν από τους ερευνητές συγκρίνοντας τη μέθοδο τους, μεταξύ άλλων και με τη μέθοδο SSDH, και για μεγέθη hash code 4, 6, 8, 10, 32 και 48 bits σε σύνολα εικόνων, έδειξαν την ανωτερότητα της μεθόδου τους ως προς την απόδοση σε σχέση με όλες τις άλλες μεθόδους. Το αξιοπερίεργο είναι ότι σε κάποιες περιπτώσεις η μέθοδος SSLH πετυχαίνει μεγαλύτερη ακρίβεια ακόμα και για μεγαλύτερα μεγέθη hash code όπως 32 ή 48 bits. Οι συγγραφείς επιχειρηματολογούν πως η μέθοδος SSLH αποτελεί την μοναδική γραμμική μέθοδο κατακερματισμού που εστιάζει το ίδιο σε μικρά και μεγάλα μήκη hash codes διατηρώντας υψηλή ακρίβεια.

4.2. Supervised Discrete Hashing with Mutual Linear Regression

Μια άλλη γραμμική επιβλεπόμενη μέθοδος κατακερματισμού περιγράφεται στην εργασία [7]. Γενικά η σχέση μεταξύ ετικετών και hash codes χρησιμοποιείται από όλες τις επιβλεπόμενες μεθόδους. Ωστόσο οι περισσότερες μέθοδοι χρησιμοποιούν πάντα δυο διαφορετικές προβολές για να αναπαραστήσουν την από κοινού παλινδρόμηση μεταξύ των hash codes και των ετικετών κλάσεων. Αντίθετα σε αυτήν την εργασία [7] η μέθοδος Discrete Hashing with Mutual Linear Regression (SDHMLR) που προτείνεται χρησιμοποιεί μόνο μια σταθερή προβολή για να περιγράψει την γραμμική συσχέτιση μεταξύ των hash codes και των αντίστοιχων ετικετών κλάσης.

Πιο αναλυτικά εκτελείται παλινδρόμηση των hash codes στο αντίστοιχο μητρώο ετικετών με μια γραμμική προβολή και στη συνέχεια εκτελείται παλινδρόμηση του μητρώου ετικετών στα αντίστοιχα hash codes χρησιμοποιώντας την ίδια γραμμική προβολή. Σημειώνουμε πως αυτές οι δυο ενέργειες εκτελούνται ταυτόχρονα. Η γραμμική προβολή που θα μάθει ο αλγόριθμος μπορεί να περιγράψει μια σταθερή και μοναδική συσχέτιση μεταξύ των hash codes και των ετικετών. Το γεγονός ότι χρησιμοποιείται μόνο μια προβολή κάνει τη μέθοδο πιο σταθερή και πιο ακριβή. Επίσης οι ερευνητές πρότειναν και μια στρατηγική boosting για την περαιτέρω βελτίωση της απόδοσης, με μια μικρή χωρική και χρονική επιβάρυνση. Η διαφορά με τη χρήση της τεχνικής boosting είναι πως είναι δυνατόν να προκύψουν πιο αποδοτικές παράμετροι.

Συγκρίνοντας πειραματικά τη μέθοδο τους με διάφορες άλλες μεθόδους, μεταξύ των οποίων και η SSDH, σε 3 σύνολα δεδομένων εικόνων, ανάμεσα στα οποία ήταν το CIFAR-10 και το NUS-WIDE, καταλήγουν στο συμπέρασμα πως η τεχνική τους SDHMLR πετυχαίνει τα καλύτερα αποτελέσματα ως προς τη μετρική MAP.

4.3. Scalable Supervised Asymmetric Hashing with Semantic and latent Factor Embedding

Μια λίγο διαφορετική κατηγορία επιβλεπόμενων μεθόδων αποτελούν οι λεγόμενες ασύμμετρες μέθοδοι κατακερματισμού. Για παράδειγμα στην εργασία [8] προτείνεται η μέθοδος Scalable Supervised Asymmetric Hashing (SSAH), η οποία προσεγγίζει το μητρώο ομοιότητας με βάση το μέγιστο ασύμμετρο εσωτερικό γινόμενο 2 διαφορετικών μη-δυαδικών μητρώων. Ειδικότερα για να διερευνηθεί περιεκτικά η σημασιολογική πληροφορία των δεδομένων, οι ετικέτες και τα εκλεπτυσμένα latent χαρακτηριστικά θεωρούνται πως κατασκευάζουν τις hash συναρτήσεις και ενισχύουν τη διάκριση των παραγόμενων hash codes. Πιο συγκεκριμένα, η μέθοδος SSAH μαθαίνει 2 ξεχωριστές συναρτήσεις κατακερματισμού συνδυάζοντας την ελαχιστοποίηση του σφάλματος παλινδρόμησης των ετικετών και του σφάλματος κωδικοποίησης στα παραγόμενα latent χαρακτηριστικά.

Στη μέθοδο SSAH λοιπόν παράγονται δυο μητρώα χαρακτηριστικών. Ένα για τη σημασιολογική πληροφορία, τις ετικέτες, και ένα για με τα latent χαρακτηριστικά. Το πρώτο χρησιμοποιείται για την ανακάλυψη συσχετίσεων μεταξύ των ετικετών, των δυαδικών hash codes και των ομοιοτήτων μεταξύ των δειγμάτων. Το δεύτερο χρησιμοποιείται για να δώσει περισσότερη έμφαση στις κρυφές πληροφορίες που περιέχουν τα διαφορετικά στοιχεία των αρχικών δεδομένων. Επομένως συνοπτικά η

μέθοδος SSAH προτείνει ένα νέο πλαίσιο στο οποίο πετυχαίνουμε ταυτόχρονα τη διατήρηση των ομοιοτήτων μεταξύ των δειγμάτων, χρησιμοποιώντας τις ασύμμετρες προσεγγίσεις διπλής ροής και την εκμάθηση υψηλής ποιότητας hash codes που προέρχονται από δυο ξεχωριστές συναρτήσεις κατακερματισμού, με τέτοιο τρόπο ώστε τα δυο παραπάνω βήματα του αλγορίθμου να μπορούν να βελτιστοποιηθούν διαδραστικά και συντονισμένα. Τέλος ο αλγόριθμος βελτιστοποίησης που προτείνεται για την λύση του διακριτού προβλήματος, επιτρέπει την αποδοτική κλιμάκωση της μεθόδου σε μεγάλα σύνολα δεδομένων. Η αποτελεσματικότητα και η αποδοτικότητα της μεθόδου επιβεβαιώθηκε πειραματικά συγκρίνοντας της με κάποιες από τις καλύτερες ασύμμετρες μεθόδους σε διάφορα σύνολα δεδομένων.

4.4. SCRATCH: A Scalable Discrete Matrix Factorization Hashing for Cross-Modal Retrieval

Δεν έχουμε αναφερθεί καθόλου μέχρι τώρα σε περιπτώσεις όπου μπορεί να έχουμε ετερογενή δεδομένα τα οποία παράγονται από διαφορετικές πηγές. Σε αυτές τις περιπτώσεις έχουμε τη λεγόμενη cross-modal ανάκτηση. Μια cross-modal επιβλεπόμενη μέθοδος παρουσιάζεται στην εργασία [9]. Η συγκεκριμένη μέθοδος Scalable discrete mATrix factorization Hashing (SCRATCH) παρά το γεγονός ότι λύνει ένα λίγο διαφορετικό πρόβλημα, παρουσιάζει αρκετά κοινά με τη μέθοδο SSDH. Πρώτο αποτελεί διακριτή μέθοδο, δηλαδή δεν εφαρμόζεται κάποια χαλάρωση ώστε να μαθαίνονται hash codes πραγματικών τιμών, και δεύτερον αξιοποιεί το ίδιο αποδοτικά στη διαδικασία βελτιστοποίησης της συνάρτησης σφάλματος ολόκληρο το μητρώο ομοιότητας.

Τα αρχικά χαρακτηριστικά προβάλλονται σε ένα χώρο με τη διαδικασία του Kernelization που αναφέραμε και πιο πάνω. Η μέθοδος αξιοποιεί την συλλογική παραγοντοποίηση του μητρώου (Collective Matrix Factorization ή CMF) αυτών των χαρακτηριστικών και το μητρώο ετικετών με στόχο την πλήρη αξιοποίηση της διαθέσιμης πληροφορίας. Επίσης άλλη μια ομοιότητα της μεθόδου SCRATCH με τη μέθοδο SSDH είναι η ενσωμάτωση του μητρώου των ετικετών στη συνάρτηση σφάλματος. Στη μέθοδο SCRATCH η εκμάθηση των hash codes και των συναρτήσεων κατακερματισμού γίνεται ταυτόχρονα. Τέλος η χρονική πολυπλοκότητα της μεθόδου είναι γραμμική ως προς το μέγεθος της εισόδου, γεγονός που την κάνει κλιμακώσιμη σε μεγάλα σύνολα δεδομένων. Η πειραματική σύγκριση της μεθόδου SCRATCH με άλλες παρόμοιες cross-modal μεθόδους μαρτυρά την εξαιρετική της απόδοση.

4.5. Deep Collaborative Multi-View Hashing for Large Scale Image Search

Μια εντελώς διαφορετική οικογένεια μεθόδων είναι αυτών του multi-view hashing. Με τον όρο multi-view εννοείται η αξιοποίηση πολλαπλών χαρακτηριστικών, από διαφορετικές οπτικές, των αρχικών δεδομένων για την εκμάθηση των hash codes. Οι περισσότερες Multi-view μέθοδοι ωστόσο είτε βασίζονται σε σχετικά μικρά μοντέλα που αδυνατούν να συλλάβουν τις συσχετίσεις μεταξύ των διαφορετικών οπτικών ή σε

μη επιβλεπόμενα μοντέλα βαθιών νευρωνικών δικτύων. Στην εργασία [10] προτείνεται η μέθοδος Deep Collaborative Multi-view Hashing (DCMVH) η οποία σαν στόχο έχει να εκμεταλλευτεί τα χαρακτηριστικά των διαφορετικών οπτικών και να 'μάθει' συνεργατικά τα πολλών οπτικών hash codes με τη χρήση μιας βαθιάς αρχιτεκτονικής νευρωνικού δικτύου.

Η αρχιτεκτονική αυτή αποτελείται από πολλά διαφορετικά δίκτυα, ένα για κάθε οπτική, και ένα δίκτυο που συνδυάζει όλα αυτά για την εκμάθηση των hash codes. Πιο συγκεκριμένα στη μέθοδο DCMVH διαφορετικά επίπεδα του νευρωνικού δικτύου συσχετίζονται με συγκεκριμένα δείγματα ή τις ομοιότητες μεταξύ δειγμάτων αντίστοιχα. Με αυτόν τον τρόπο ενισχύεται η ικανότητα των επιπέδων αναπαράστασης να διαχωρίζουν μεταξύ των δειγμάτων, ενώ ταυτόχρονα η συμπληρωματικότητα των χαρακτηριστικών που έχουν προκύψει από διαφορετικές οπτικές μπορεί να αξιοποιηθεί αποδοτικά.

Αναφορές

- [1] X. Luo, Y. Wu και X.-S. Xu, «Scalable Supervised Discrete Hashing for Large-Scale Search,» The Web conference, 2018.
- [2] A. Gordo, F. Perronnin, Y. Gong και S. Lazebnik , «Asymmetric Distances for Binary Embeddings,» *IEEE Transactions on Pattern Analysis and Machine Intelligence*, τόμ. 36, αρ. 1, pp. 33-47, 2013.
- [3] J. Gui, T. Liu, Z. Sun, D. Tao και T. Tan, «Fast Supervised Discrete Hashing,» *IEEE Transactions on Pattern Analysis and Machine Intelligence*, τόμ. 40, αρ. 2, pp. 490-496, 2018.
- [4] X. Luo, L. Nie, X. He, Y. Wu, Z.-D. Chen και X.-S. Xu, «Fast Scalable Supervised Hashing,» 2018.
- [5] K. Chatfield, K. Simonyan, A. Vedaldi και A. Zisserman, «Return of the Devil in the Details: Delving Deep into Convolutional Nets,» 2014.
- [6] X. Liu, X. Nie, Q. Zhou, X. Xi, L. Zhu και Y. Yin, «Supervised Short-Length Hashing,» 2019.
- [7] X. Liu, X. Nie, Q. Zhou και Y. Yin, «Supervised Discrete Hashing With Mutual Linear Regression,» In Proceedings of the 27th ACM International Conference on Multimedia (MM '19), New York, 2019.
- [8] Z. Zhang, Z. Lai, Z. Huang, W. K. Wong, G.-S. Xie, L. Liu και L. Shao, «Scalable Supervised Asymmetric Hashing With Semantic and Latent Factor Embedding,» *IEEE Transactions on Image Processing*, τόμ. 28, αρ. 10, pp. 4803-4818, 2019.
- [9] C.-X. Li, Z.-D. Chen, P.-F. Zhang, X. Luo, L. Nie , W. Zhang και X.-S. Xu, «SCRATCH: A Scalable Discrete Matrix Factorization Hashing for Cross-Modal Retrieval,» ACM Multimedia, 2018.

- [10] L. Zhu, Z. Cheng, J. Li και H. Zhang, «Deep Collaborative Multi-View Hashing for Large-Scale Image Search,» *IEEE Transactions on Image Processing*, τόμ. 29, pp. 4643-4655, 2020.