

**Πανεπιστήμιο Πειραιώς
Τμήμα Πληροφορικής
Έτος: 2023 - 2024**



**Μάθημα:
«Ανάλυση Εικόνας»
Εργασία: Υπολογιστική Εργασία Μαθήματος
Εξάμηνο: 7ο**

Ομάδα εργασίας:
Αιμιλιανός Κουρπάς-Δανάς Π20100,
Αναστάσιος Μελαχροινούδης Π20124

Ημερομηνία παράδοσης : 12.02.2024

Σημείωση

Τα προγράμματα γράφτηκαν στο προγραμματιστικό περιβάλλον VScode-jupyter notebook.

Στον κώδικα υπάρχουν περιεκτικά σχόλια ουσίας για την καλύτερη επεξήγησή του.

Σας έχουμε στείλει:

1)Το PDF

2)Τον κωδικα python

3)Το jupyter notebook

4)Ένα html αρχείο που έγινε convert απο το jupyter notebook αυτοματα για να δειτε μια εκτελεση του αλγορίθμου με αποτελεσματα και σχολια.

ΕΙΣΑΓΩΓΗ

Σύμφωνα με τις οδηγίες της υπολογιστικής εργασίας, η εργασία αφορά την ανάπτυξη ενός αλγορίθμου γραφοθεωρίας για την ανάκτηση εικόνων με βάση το περιεχόμενό τους. Ο αναφερόμενος αλγόριθμος βασίζεται στο άρθρο με τίτλο "Multimedia Retrieval through Unsupervised Hypergraph-based Manifold Ranking".

Αναλυτική περιγραφή της υπολογιστικής διαδικασίας

Ο προτεινόμενος αλγόριθμος, που αναλύεται στο προηγούμενο άρθρο, ονομάζεται "Log-Based Hypergraph of Ranking Reference" και αποτελείται από πέντε βασικά βήματα που εφαρμόστηκαν στο πλαίσιο της συγκεκριμένης εργασίας.

Για την εκτέλεση του αλγορίθμου, προηγείται η εξαγωγή του συνόλου των χαρακτηριστικών αντικειμενικού περιεχομένου για κάθε πολυμεσικό αντικείμενο. Το άρθρο παρουσιάζει μια συνάρτηση εξαγωγής χαρακτηριστικών, ενώ στην υλοποίησή μας χρησιμοποιήσαμε ένα προ-εκπαιδευμένο μοντέλο του torchvision. Αντικαταστήσαμε το επίπεδο εξόδου με ένα επίπεδο χωρίς λειτουργία, επιτρέποντας μας να εξάγουμε τα χαρακτηριστικά απευθείας από το τελευταίο κρυφό επίπεδο.

Προτού εξετάσουμε τα βήματα του αλγορίθμου, απαιτείται η καθορισμός ορισμένων εννοιών. Αρχικά, ο αλγόριθμος εργάζεται με ένα σύνολο $C = \{o_1, o_2, \dots, o_n\}$, όπου το o_n αντιπροσωπεύει ένα πολυμεσικό αντικείμενο στο σύνολο δεδομένων, και το $|C| = n$, όπου n αναφέρεται στον αριθμό των πολυμεσικών αντικειμένων. Ο στόχος του αλγορίθμου είναι, για ένα συγκεκριμένο αντικείμενο o_q , να ανακαλύψει τα k πιο κοντινά αντικείμενα. Αυτό πραγματοποιείται μέσω μιας ταξινομημένης λίστας $T_q = \{t_q(1), t_q(2), \dots, t_q(n)\}$, όπου $t_q(i)$, $i \in [N]$, υποδηλώνει τη θέση του αντικειμένου o_i στην ταξινομημένη λίστα του o_q . Για την ταξινόμηση χρησιμοποιείται η συνάρτηση ομοιότητας ρ . Επομένως, αν σε μια λίστα T_q το o_i προηγείται του o_j , τότε $t_q(i) < t_q(j)$, καθιστώντας το $\rho(q, i) \geq \rho(q, j)$. Στην υλοποίησή μας, η αρχική συνάρτηση ομοιότητας ήταν το αντίστροφο της Ευκλείδειας απόστασης, όπως είχε εκτιμηθεί και στο μάθημα. Τέλος, ορίζονται τα γειτονικά σύνολα $N(q, k)$, που περιλαμβάνουν τα k πιο κοντινά αντικείμενα σε σχέση με το συγκεκριμένο αντικείμενο o_q . Σημειώνεται ότι, λόγω του υψηλού κόστους δημιουργίας των λιστών για μεγάλα σύνολα πολυμεσικών αντικειμένων, η εργασία προτείνει τη διατήρηση μόνο των L καλύτερων αντικειμένων σε ένα νέο υποσύνολο $CL \subset C$. Για λόγους ευκολίας, αντικαθιστούμε τον όρο "πολυμεσικό αντικείμενο" με τον όρο "εικόνα", δεδομένου ότι η εργασία και το μάθημα επικεντρώνονται στις εικόνες.

Βήματα αλγορίθμου:

Κανονικοποίηση Σειράς Κατάταξης:

Εφαρμόζεται μια διαδικασία κανονικοποίησης σε κάθε λίστα T_i , όπου υπολογίζεται ένα νέο μετρικό ομοιότητας $\rho(i,j) = 2L - (t_i(j) + t_j(i))$, με σκοπό να βελτιώσει τη συμμετρία των σχέσεων γειτονιάς $N(q,k)$. Στη συνέχεια, πραγματοποιείται η ταξινόμηση κάθε λίστας.

Δηλαδή, το νέο μετρικό ομοιότητας υπολογίζεται αφαιρώντας τις θέσεις των δύο αντικειμένων j και i από το διπλάσιο του μέγεθους της λίστας L . Αυτή η διαδικασία στοχεύει στη βελτίωση της ισορροπίας και της συμμετρίας στις σχέσεις γειτονιάς, προκειμένου να επιτευχθεί πιο αξιόπιστη και συνεκτική κατάταξη των γειτονικών αντικειμένων. Στη συνέχεια, η κάθε λίστα ταξινομείται για να ληφθεί υπόψη το ανανεωμένο μετρικό κατάταξης και να επιτευχθεί η επιθυμητή συμμετρία στις γειτονικές σχέσεις.

Κατασκευή Υπεργραφήματος:

Το υπεργράφημα αναπαριστά μια εκτεταμένη μορφή ενός γραφήματος, όπου οι ακμές αντιστοιχούν σε μη-κενά υποσύνολα των κορυφών. Αυτές οι ακμές επιτρέπουν τη σύνδεση οποιωνδήποτε κορυφών, προσφέροντας περισσότερες πληροφορίες για τις σχέσεις μεταξύ τους. Για ένα δεδομένο γράφημα $G = (V, E, w)$, όπου V είναι το σύνολο των κορυφών και E το σύνολο των υπερακμών με $U \in E = V$, κάθε υπερακμή ei καθορίζεται ως ένα υποσύνολο κορυφών, αντιστοιχώντας κάθε εικόνα oi με βάση το k -γειτονικό σύνολο και τους αντίστοιχους γείτονες.

Σε αντίθεση με την κλασική προσέγγιση δυαδικής συμμετοχής, χρησιμοποιούμε μια πιθανοτική προσέγγιση. Για κάθε εικόνα oi και $oj \in N(x,k)$, το μέτρο συμμετοχής $r(ei,uj)$ υπολογίζεται ως:

$$r(e_i, u_j) = \sum_{o_x \in N(i,k) \wedge o_j \in N(x,k)} w_p(i, x) \times w_p(x, j)$$

Η συνάρτηση $w_p(i,x)$ αναθέτει βάρος στην εικόνα ox βάσει της θέσης της στην ταξινομημένη λίστα t_i . Ειδικότερα, το βάρος υπολογίζεται ως:

$$w_p(i, x) = 1 - \log_k \tau_i(x)$$

Αυτή η συνάρτηση δίνει υψηλό βάρος (ίσο με 1) στην πρώτη θέση της λίστας (που είναι η ίδια η εικόνα στόχος) και μειώνει ταχύως τα βάρη για τις επόμενες ταξινομημένες θέσεις. Στόχος είναι η ανάθεση υψηλών τιμών βαρών στις πρώτες θέσεις, όπου η αποτελεσματικότητα των ταξινομημένων λιστών είναι ανώτερη.

Με βάση τις τιμές της $r(ei,uj)$, καθορίζουμε τον πίνακα εγγύησης (incidence matrix) του υπεργραφήματος:

$$H(e_i, u_j) = r(e_i, u_j)$$

Τέλος, για κάθε υπερακμή ei , υπολογίζουμε ένα βάρος $w(ei)$, που δηλώνει την αυτοπεποίθηση των σχέσεων μεταξύ των κορυφών της υπερακμής:

$$w(e_i) = \sum_{j \in N_h(i,k)} H(i, j)$$

Υπολογισμός Ομοιότητας Υπερακμών:

Στον αρχικό υπολογισμό του πίνακα ομοιότητας S , πρέπει να ληφθούν υπ' όψιν δύο βασικές υποθέσεις. Η πρώτη υπόθεση βασίζεται στην ιδέα ότι παρόμοιες εικόνες έχουν παρόμοιες ταξινομημένες λίστες, επομένως και παρόμοιες υπερακμές. Λαμβάνοντας υπόψη ότι οι πληροφορίες για τη σχετικότητα των υπερακμών είναι ήδη κωδικοποιημένες στον πίνακα εγγύησης (incidence matrix), μπορούμε να ορίσουμε ένα μέτρο ομοιότητας μεταξύ δύο υπερακμών e_i και e_j ως εξής: $S_h = H \cdot H^T$.

Η δεύτερη υπόθεση υποδηλώνει ότι παρόμοιες εικόνες αναμένεται να συσχετίζονται με τις ίδιες υπερακμές, και αυτό μας οδηγεί στον πίνακα $S_u = H^T \cdot H$.

Εφόσον και οι δύο παραπάνω πίνακες περιέχουν σχετική πληροφορία, ο πίνακας ομοιότητας S μπορεί να υπολογιστεί μέσω του Hadamard product (στοιχειοθετικού πολλαπλασιασμού): $S = S_h \circ S_u$. Αυτή η διαδικασία επιτρέπει τη συνένωση των πληροφοριών από τις δύο υποθέσεις για τη δημιουργία ενός ολοκληρωμένου πίνακα ομοιότητας υπερακμών.

Υπολογισμός Καρτεσιανού Γινομένου μεταξύ Υπερακμών:

Για την απευθείας εξαγωγή σχέσεων ζευγαριών από τα στοιχεία των υπερακμών, πρέπει να υπολογιστεί το καρτεσιανό γινόμενο τους. Καθώς κάθε υπερακμή συνδέει ένα σύνολο κορυφών, το καρτεσιανό γινόμενο ορίζεται ως

$$\setminus(e_q^2 = e_q \times e_i = \{(u_x, u_y) : u_x \in e_q \wedge u_y \in e_i\}.$$

Επομένως, για κάθε ζεύγος κορυφών που ανήκει στο καρτεσιανό γινόμενο των υπερακμών, ορίζεται μια σχέση σχετικότητας κατά ζεύγη. Στη συνέχεια, υπολογίζεται η συνάρτηση $\setminus(\rho)$ εκ νέου, λαμβάνοντας υπόψη το βάρος $w(e_q)$ που αντιπροσωπεύει την αυτοπεποίθηση της υπερακμής e_q . Ο βαθμός συμμετοχής των κορυφών u_i, u_j δίνεται από τη σχέση $\rho(e_q, u_i, u_j) = w(e_q) \times H(e_q, u_i) \times H(e_q, u_j)$.

Το μέτρο ομοιότητας του καρτεσιανού γινομένου υπολογίζεται με τη χρήση του πίνακα C , λαμβάνοντας υπόψη τις σχέσεις σε όλες τις υπερακμές. Κάθε θέση στον πίνακα C υπολογίζεται ως

$$C(i, j) = \sum_{e_q \in E^{(u_i, u_j)} \wedge (u_i, u_j) \in e_q^2} \rho(u_i, u_j).$$

Αξιολόγηση Ομοιότητας μέσω Κατασκευασμένου Υπεργράφου:

Το μέτρο ομοιότητας που προέκυψε από τις υπερακμές και το καρτεσιανό γινόμενο παρέχει σημαντική πληροφορία για την πολυπλοκότητα των δεδομένων. Η αξιοποίηση αυτής της πληροφορίας γίνεται μέσω ενός πίνακα συγγένειας W , ο οποίος συνδυάζει τους πίνακες C και S :

$$W = C \circ S$$

Ο βαθμός συγγένειας που προκύπτει από το W μας επιτρέπει να εφαρμόσουμε μια διαδικασία κατάταξης, δημιουργώντας έτσι μια νέα ταξινομημένη λίστα T .

Η διαδικασία αυτή περιγράφεται ως εξής:

1. Έστω $T(0)$ η αρχική λίστα που υπολογίστηκε από τα αντικειμενικά χαρακτηριστικά.
2. Η λίστα $T(t+1)$ υπολογίζεται με βάση το $W(t)$.
3. Μετά από έναν ορισμένο αριθμό επαναλήψεων, ο οποίος καθορίζεται στο πρόγραμμα, υπολογίζεται η τελική ταξινομημένη λίστα T για κάθε εικόνα q που ανήκει στο σύνολο C .

Προγραμματιστική Υλοποίηση

Ο αλγόριθμος υλοποιήθηκε σε python , στο jupyter notebook ο οποίος έχει τον κωδικά σε κελιά και με περισσότερη τεκμηρίωση βημα-βημα.

Για την υλοποίηση χρειαστηκε:

- οι βιβλιοθήκες pytorch (+torchvision), matplotlib, numpy ,random , gdown,timm
- Ως σύνολο δεδομένων χρησιμοποιούμε το σύνολο δεδομένων Caltech101 (συγκεκριμένα ένα υποσύνολο του για να μειωθεί ο χρόνος εκτέλεσης). Το dataset αυτό περιέχεται στα datasets του torchvision και εγκαθίσταται μέσω της βιβλιοθήκης. Δυστυχώς δεν δουλευε απο το προγραμμα να το καλέσουμε και να κατεβει αυτοματα αλλα το καναμε manually download και το βάλαμε στο αρχείο του project μας.
- Για την εξαγωγή χαρακτηριστικών χρησιμοποιήσαμε ένα από τα διαθέσιμα προεκπαιδευμένα μοντέλα της βιβλιοθήκης torchvision.
- Για την προβολη των εικωνων ενα helper.py απο ενα github repository.

Στην υλοποίησή μας, αποφεύγουμε να ορίζουμε συγκεκριμένες εικόνες ως εικόνες ερώτησης (query). Αντ' αυτού, λειτουργούμε με ένα σύνολο N εικόνων. Υπολογίζουμε τις ομοιότητες μεταξύ κάθε δυνατού ζεύγους εικόνων, δημιουργώντας έναν πίνακα διαστάσεων $N \times N$. Κάθε γραμμή του πίνακα αναπαριστά τα σκορ της εκάστοτε εικόνας έναντι όλων των υπολοίπων. Με αυτόν τον τρόπο, έχουμε την ευελιξία να επιλέξουμε οποιαδήποτε εικόνα από τις N ως εικόνα ερώτησης και να ανακτήσουμε τις εικόνες με τα υψηλότερα σκορ από την αντίστοιχη γραμμή του πίνακα.

Εξαγωγή χαρακτηριστικών

Για την εξαγωγή χαρακτηριστικών από τις εικόνες, χρησιμοποιήσαμε ένα προεκπαιδευμένο νευρωνικό δίκτυο τύπου Vision Transformer (vit_base_patch16_224), το οποίο έχει εκπαιδευτεί στο σύνολο δεδομένων ImageNet για την κατηγοριοποίηση. Κατεβάζοντας τα τελικά βάρη και κατασκευάζοντας το δίκτυο, απενεργοποιούμε τον υπολογισμό των gradients για να διατηρήσουμε τα βάρη αμετάβλητα.

Για την εξαγωγή των χαρακτηριστικών, απενεργοποιούμε το τελευταίο επίπεδο που αντιστοιχεί στην κλάση της εικόνας στο ImageNet. Επιλέγουμε να αντικαταστήσουμε αυτό το επίπεδο με ένα άδριο Sequential layer που δεν επιδρά στην έξοδο, διατηρώντας την έξοδο του τελευταίου κρυφού επιπέδου ως τα χαρακτηριστικά που χρησιμοποιούμε ως είσοδο στον αλγόριθμο. Στη συνέχεια, τυπώνουμε το μοντέλο για εξατομικευμένη κατανόηση της δομής του.

Παραδείγματα Εκτέλεσης

Τα ids παρακάτω δεν αντιστοιχούν στις ίδιες εικόνες σε κάθε εκτέλεση.

Θα δειξουμε 7 παραδείγματα εκτέλεσης.

Όσον αφορά το precision-recall εξηγείται στην επομενη ενοτητα.

ΠΑΡΑΔΕΙΓΜΑ 1

Query Image:



Τα score που πήραμε με τα ids είναι:

[(2.207931078188474, 1), (0.21343984370492775, 1356), (0.015259401746890366, 2716)]

(2.207931078188474, 1)	(0.21343984370492775, 1356)	(0.015259401746890366, 2716)
		

Precision[0]=1.0

Recall[0]=0.08695652173913043

ΠΑΡΑΔΕΙΓΜΑ 2

Query Image:



Τα score που πήραμε με τα ids είναι:

(2.402190537018941, 2), (0.21343984370492775, 1161),
(0.02288910262033555, 922), (0.015259401746890366, 3258)

(2.402190537018941, 2)	(0.21343984370492775, 1161)
	
0.02288910262033555, 922)	(0.015259401746890366, 3258)
	

Precision[1]=1.0

Recall[1]=0.12




ΠΑΡΑΔΕΙΓΜΑ 3

Query Image:



Τα score που πήραμε με τα ids είναι:

[(3.4498923096694907, 3), (0.21343984370492775, 857),
(0.21343984370492775, 3279), (0.015259401746890366, 3344)]

(3.4498923096694907, 3)	(0.21343984370492775, 857)
	
(0.21343984370492775, 3279)	(0.015259401746890366, 3344)
	

Precision[2]=0.666666666666

Recall[2]=0.0392156862745098

ΠΑΡΑΔΕΙΓΜΑ 4

Query Image:



Τα score που πήραμε με τα ids είναι:

[(3.4498923096694907, 4), (0.6047548240938942, 396),
(0.21343984370492775, 1360)]

(3.4498923096694907, 4)	(0.6047548240938942, 396)	(0.21343984370492775, 1360)
		

Precision[3]=1.0

Recall[3]=0.046511627906976744

ΠΑΡΑΔΕΙΓΜΑ 5

Query Image:



Τα score που πήραμε με τα ids είναι:

[(2.207931078188474, 7), (0.21343984370492775, 2174),
(0.015259401746890366, 694)]

(2.207931078188474, 7)	(0.21343984370492775, 2174)	(0.015259401746890366, 694)
		

Precision[4]=1.0

Recall[4]=0.08333333333333333

ΠΑΡΑΔΕΙΓΜΑ 6

Query Image:



Τα score που πήραμε με τα ids είναι:

[(2.402190537018941, 9), (0.23182300398972688, 1248),
(0.015259401746890366, 824), (0.015259401746890366, 2541)]

(2.402190537018941, 9)	(0.23182300398972688, 1248)
	
(0.015259401746890366, 824)	(0.015259401746890366, 2541)
	

Precision[5]=1.0

Recall[5]=0.01079136690647482

ΠΑΡΑΔΕΙΓΜΑ 7

Query Image:



Τα score που πήραμε με τα ids είναι:

[(5.25718679814929, 12), (0.46418641623249673, 2346),
(0.257732613193808, 2117), (0.21343984370492775, 2199),
(0.033642562031689485, 1371), (0.033642562031689485, 1648),
(0.005387443063889855, 1412)]

(5.25718679814929, 12)	(0.46418641623249673, 2346)	(0.257732613193808, 2117)
		
(0.21343984370492775, 2199)	(0.033642562031689485, 1371)	(0.033642562031689485, 1648)
		

(0.005387443063889855, 1412)



Precision[6]=1.0

Recall[6]=0.4

6) Μέτρηση ακρίβειας

Για την αξιολόγηση της ακρίβειας του αλγορίθμου, επιλέξαμε να χρησιμοποιήσουμε δύο δημοφιλείς μετρικές για tasks Information Retrieval: το Precision και το Recall. Χρησιμοποιούμε τα πραγματικά labels που παρέχονται από το σύνολο δεδομένων Caltech101 για τον υπολογισμό των μετρικών.

Για το Precision, υπολογίζουμε πόσες από τις επιστρεφόμενες εικόνες έχουν το ίδιο label με την εικόνα ερώτησης. Η μετρική ορίζεται ως:

$$\text{Precision} = \frac{\text{Επιστρεφόμενες εικόνες με σωστό label}}{\text{Συνολικές επιστρεφόμενες εικόνες}}$$

Για το Recall, υπολογίζουμε το ποσοστό των εικόνων με σωστό label που επιστράφηκαν σε σχέση με όλες τις εικόνες με το συγκεκριμένο label. Η μετρική ορίζεται ως:

$$\text{Recall} = \frac{\text{Επιστρεφόμενες εικόνες με σωστό label}}{\text{Συνολικές εικόνες με σωστό label}}$$

Σε περιπτώσεις όπου χρησιμοποιούμε μικρά υποσύνολα ως υπερακμές, τα recall μπορεί να είναι χαμηλότερα, καθώς το σύνολο των διαθέσιμων εικόνων για κάθε label είναι περιορισμένο. Επίσης, η ευαισθησία του μοντέλου σε λεπτομέρειες μπορεί να οδηγήσει σε διακυμάνσεις στα αποτελέσματα, καθώς ανιχνεύει διαφορές μεταξύ εικόνων με κοινά labels που οφείλονται σε μικρές διαφορές, παραβλέποντας τα βασικά χαρακτηριστικά.

7) Πηγές

- Multimedia Retrieval through Unsupervised Hypergraph-based Manifold Ranking, Daniel Carlos Guimaraes Pedronette et. al. IEEE Transactions on Image Processing December 2019
- Pytorch datasets <https://pytorch.org/vision/stable/datasets.html>
- Finetuning vision models
https://pytorch.org/tutorials/beginner/finetuning_torchvision_models_tutorial.html#inputs
- Torch vision models and pre trained weights
<https://pytorch.org/vision/stable/models.html>
- Caltech101 dataset
https://drive.google.com/uc?id=137RyRjvTBkBilfeYBNZBtViDhQ6_Ewsp
- HyperNetX library <https://pnnl.github.io/HyperNetX/build/index.html>
- Helper function to plot images from a torch dataset
<https://raw.githubusercontent.com/udacity/deep-learning-v2-pytorch/master/intro-to-pytorch/helper.py>
- Χειρόγραφες σημειώσεις μαθήματος