Μηχανική Μάθηση

2ο σετ ασκήσεων

Καραγιαννίδης Χρήστος 4375

Νεφέλη-Ελένη Κατσιλέρου 4385

Αρχικά με την χρήση της load_images() φορτώσαμε τα δεδομένα από τον φάκελο train data. Συγκεκριμένα διαλέγουμε 10 random αριθμούς που θα αντιστοιχούν στους φακέλους από το dataset (από το 0 έως το 4000). Παίρνουμε το path του κάθε φακέλου με τη χρήση αυτών των random αριθμών και βάζουμε σε μια λίστα όλες τις φωτογραφίες που έχει μέσα. Έπειτα φιλτράρουμε το πλήθος των φωτογραφιών κάθε φακέλου ώστε να έχει μέχρι 50 φωτογραφίες (όπως ζητείται στην εκφώνηση). Κάποιοι φάκελοι έχουν λιγότερες από 50 φωτογραφίες οπότε από αυτούς τις κρατάμε όλες. Επιστρέφουμε έναν array τύπου allFiles[number_of_folders][number_of_files_in_each_folder].

1) Για την μείωση των δεδομένων με την μέθοδο pca δημιουργήσαμε την συνάρτηση pcaFunc. Η pcaFunc καλεί την load_images() και αποθηκεύει το array με τις φωτογραφίες σε μια μεταβλητή. Για κάθε αρχείο σε κάθε φάκελο αυτού του πίνακα κάνουμε την μείωση των διαστάσεων με την χρήση της βιβλιοθήκης sklearn και συγκεκριμένα της συνάρτησης PCA και αποθηκεύουμε την reduced Image σε έναν 2d array που λέγεται reducedImages και που εν τέλει τον επιστρέφουμε στο τέλος της συνάρτησης PCA.

2) Για την υλοποίηση του αλγορίθμου Kmeans φτιάξαμε τη συνάρτηση Kmeans η οποία καλεί την pcaFunc για το μέγεθος M που θέλει και κρατάει σε έναν πίνακα τις reduced φωτογραφίες. Για κάθε φάκελο (ακόμα είναι ομαδοποιημένες οι φωτογραφίες) διαλέγει μια random φωτογραφία με

την χρήση της pickRandomFrom η οποία διαλέγει μια random φωτογραφία, βρίσκει τα red average, green average και blue average και επιστρέφει αυτόν την 3αδα. Βάζει σε έναν αρχικά κενό πίνακα με όνομα groups την 3αδα με τα average χρώματα που επιστράφηκαν και επίσης τα βάζει και σε έναν πίνακα με όνομα medians ο οποίος θα περιέχει 10 3αδες με τα μέσα του κάθε group. Για κάθε μια φωτογραφία σε κάθε φάκελο του οργανωμένου 2d array “reducedImages” , παίρνει την κάθε φωτογραφία και την βάζει σε έναν 1d array με όνομα allPictures μαζι με την τιμή του πραγματικού group στο οποίο ανήκε καθώς επίσης και μια τιμή στην οποία θα μπει το prediction του group αργότερα (γι’ αυτό και την έχουμε -1). Για κάθε μια φωτογραφία στον καινούριο μονοδιάστατο πίνακα allPictures καλούμε την συνάρτηση group η οποία είναι υπεύθυνη για να κάνει classification της φωτογραφίας σε κάποιο από τα υπάρχοντα groups. Συγκεκριμένα για την φωτογραφία που της δόθηκε ως όρισμα καλεί την συνάρτηση findDistanceToAllMedians για να βρει την απόσταση από όλα τα μέσα (όλων των group δηλαδή). Η findDistanceToAllMedians βρίσκει τις average τιμές των red,green,blue και υπολογίζει είτε την ευκλείδεια είτε την συνημιτονοειδή απόσταση (αναλόγως με το τι της δόθηκε ως όρισμα) και επιστρέφει το position δηλαδή το group με το οποίο η φωτογραφία που της δόθηκε ως όρισμα έχει την μικρότερη απόσταση. Η group παίρνει την 3αδα με τα average red,green,blue και την κάνει append στην θέση position του πίνακα groups (άρα στο group που προβλέψαμε ότι ανήκει) . Στο τέλος της η group επιστρέφει τον πίνακα groups και το position. Βάζουμε στην θέση 2 της κάθε φωτογραφίας(που κρατούσε την πρόβλεψη group και ήταν αρχικοποιημένη στο -1) στον πίνακα allPictures την μεταβλητή position που είναι η πρόβλεψη του group που κάναμε για την εκάστοτε φωτογραφία. Μετά καλούμε την findNewMedian() η οποία παίρνει ως όρισμα ένα group και βρίσκει το καινούριο μέσο του (μετά δηλαδή από την προσθήκη του καινούριου στοιχείου στο group).

Για τον υπολογισμό του purity παίρνουμε από την κάθε φωτογραφία την τιμή του πραγματικού της group και αυξάνουμε την τιμή ενός ξεχωριστού πίνακα στην θέση του group που προβλέψαμε ότι ανήκει η φωτογραφία κατά 1 . Έτσι μετράμε για κάθε φωτογραφία που κανονικά ανήκει στο

group A που προβλέψαμε ότι ανήκει. Ο πίνακας είναι της μορφής clusterA = [18, 12, 1, 3, 7, 0, 2, 3, 4, 0] που σημαίνει ότι οι φωτογραφίες που κανονικά ανήκουν στον A προβλέψαμε ότι 18 από αυτές ανήκουν όντως στον A , 12 ανήκουν στον B ,…, 0 ανήκουν J. Για να βρούμε την τιμή του purity παίρνουμε το άθροισμα των max του κάθε cluster και το διαιρούμε με το πλήθος των φωτογραφιών που κατηγοριοποιήσαμε στο σύνολο (~500).

Για τον υπολογισμό της f-measure διατρέχουμε όλους τους πίνακες clusters που δημιουργήσαμε παραπάνω , και βρίσκουμε ποια είναι η πλειοψηφούσα πραγματική κατηγορία . Για κάθε φωτογραφία στο σετ με όλες τις φωτογραφίες αν η τιμή του πραγματικού group ταυτίζεται με το max group που βρήκαμε και η πρόβλεψη που κάναμε είναι ίδια με το max group τότε αυξάνουμε τα TP κατά 1. Αν η πρόβλεψη ήταν max αλλά η πραγματική τιμή δεν ήταν max αυξάνουμε τα FP κατά 1. Αν η πραγματική τιμή ήταν max αλλά δεν προβλέψαμε max τότε αυξάνουμε τα FN κατά 1. Αν δεν ήταν κανένα από αυτά αυξάνουμε τα TN κατά 1. Υπολογίζουμε τα precision και recall όπως μας δίνονται οι τύποι χρησιμοποιώντας τα TP,FP,FN και καλούμε την συνάρτηση fmeasure() που θα υπολογίσει το Fmeasure για a=1 του εκάστοτε cluster σύμφωνα με τον τύπο που δόθηκε. Την τιμή που θα επιστρέψει η fmeasure() θα την κρατήσουμε σε έναν μονοδιάστατο πίνακα allFmeasures τον οποίο στο τέλος θα χρησιμοποιήσουμε για να αθροίσουμε όλα τα περιεχόμενα του και να βρούμε το Total Fmeasure που ζητείται.

Method	dimension of data (M)	Purity	F-measure
K-means (Euclidean distance)	64*	0.273	1.731807327
	50	0.288	2.743836131
	25	0.299	2.228081395
K-means (Cosine distance)	64*	0.244	1.600398823
	50	0.258	2.456225863
	25	0.254	2.427486287

* Βάλαμε 64 γιατί αυτό είναι το μέγιστο dimension μιας φωτογραφίας(64x64x3) οπότε όταν βάζαμε 100 έβγαζε error