Τμήμα Μηχανικών Η/Υ & Πληροφορικής

Πανεπιστήμιο Ιωαννίνων

Λένα Πλιάκου, Χριστόδουλος Ασημινίδης

Μεταπτυχιακό Μάθημα: «Μηχανική Μάθηση»

Θέμα εργασίας: Μείωση διάστασης και εφαρμογή στην αναγνώριση προσώπων

Επιβλέποντες Καθηγητές: Κωνσταντίνος Μπλέκας, Άρης Λύκας

Ιούνιος 2019

**Άσκηση (α): Ταξινόμηση Προσώπων**

Αρχικά κατεβάζουμε το αρχείο από τη βάση δεδομένων Face Database. Κάνουμε εξαγωγή και αμέσως μετά φορτώνουμε το αρχείο στο script της Python 2.7 όπου και έχει υλοποιηθεί όλη η εργασία. Διαιρούμε τα αρχεία σε 200 εικόνες για εκπαίδευση και 200 εικόνες για έλεγχο και ονομάζουμε τους φακέλους testing και training αντίστοιχα. Εισάγουμε στο πάνω τμήμα του script το μονοπάτι που ανήκουν τα αρχεία στον υπολογιστή μας. Αφού γίνει αυτό υλοποιούμε τη συνάρτηση read\_pgm η οποία επιστρέφει την εικόνα PGM σε μορφή πίνακα numpy. Το αμέσως επόμενο βήμα που υλοποιούμε είναι η κατασκευή τριών συναρτήσεων, path\_all\_data, path\_training και path\_testing ώστε να βάλουμε και συνεπώς να επσιτρέψουμε σε μορφή λίστας τις εικόνες που επιθυμούμε κάθε φορά. Συνεχίζοντας κατασκευάζουμε τη συνάρτηση pca\_face με δυο ορίσματα, μία λίστα με τις εικόνες και μία τιμή η οποία είναι η τιμή που παίρνει η έτοιμη συνάρτηση της python PCA για τον προσδιορισμό των components. Σε αυτή την συνάρτηση επιστρέφουμε τα ιδιοδιανύσματα με τη βοήθεια της βιβλιοθήκης numpy. Συνέχεια έχει η υλοποίηση μιας επανάληψης for-loop μέσα στο script με τιμές 10, 25, 50, 75, 100 όπου είναι οι τιμές για τον αριθμό μείωσης διάστασης της πραγματικής διάστασης των δεδομένων. Έτσι, μέσα στην επανάληψη καλούμε την συνάρτηση pca\_face και μέσα σε αυτή τη συνάρτηση βάζουμε ως πρώτο όρισμα την συνάρτηση path\_training και ως δεύτερο τις τμές 10, 25, 50, 75, 100. Έτσι, παίρνουμε τα ιδιοδιανύσματα του συνόλου εκπαίδευσης για όλες τις τιμές που περιγράφηκαν παραπάνω σε μορφή πίνακα. Εφόσον γίνει αυτό, παίρνουμε κάθε μία τιμή των ιδιοδιανυσμάτων που αποθηκεύσαμε σε έναν πίνακα. Δημιουργούμε ακόμα μία μορφή επανάληψης for-loop μέσα στο script της python όπου παίρνουμε μια-μια τις τιμές των ιδιοδιανυσμάτων του πίνακα και εφαρμόζουμε pca καλώντας την συνάρτηση pca\_face με ορίσματα το σύνολο ελέγχου και τις τιμές των ιδιοδιανυσμάτων του συνόλου εκπαίδευσης που περιγράψαμε παραπάνω.

Εν συνεχεία, χρίζουμε τους classifiers με την ευκλείδεια απόσταση και την απόσταση του συνημίτονου όπου αποθηκεύουμε τον ταξινομητή στο φάκελο που είμαστε, μετά τον καλούμε θέτοντας του κοντινότερους γείτονες που θέλουμε να πάρουμε στην δική μας περίπτωση πήραμε τους 2 πιο κοντινούς γείτονες. Αμέσως μετά, βλέπουμε ποιοι κάνουμε matching και μετά πρόβλεψη χρησιμοποιώντας τον ταξινομητή που αποθηκεύσαμε και τέλος, εμφανίζουμε μία ετικέτα με την πρόβλεψη που κάναμε. Οι υλοποιήσεις για τους ταξινομητές στις δύο διαφορετικές αποστάσεις έγιναν με τις συναρτήσεις knn\_classifier\_euclidean\_distance() και knn\_classifier\_cosine\_distance().

**Άσκηση (β): Ομαδοποίηση Προσώπων**

Αφού εφαρμόσουμε PCA σε όλες τις εικόνες με τον ίδιο τρόπο όπως στο ερώτημα (1) και με διάφορες τιμές της μειωμένης διάστασης, 100%, 75%, 50%, 25%, 10% το επόμενο βήμα είναι να κάνουμε ομαδοποίηση σε 10 ομάδες με τη μέθοδο k-means χρησιμοποιώντας την βιβλιοθήκη scikit-learn κάνοντας import στο script που έχουμε χτίσει. Έτσι, παίρνουμε τα εξής αποτελέσματα

Το Άθροισμα των τετραγωνικών αποστάσεων των δειγμάτων που δώσαμε στα πιο κοντινά κέντρα των ομάδων είναι: 253872.14142044846

Οι ετικέτες του κάθε σημείου δίνεται από τον πίνακα: [7, 4, 0, 4, 0, 0, 0, 4, 4, 0, 3, 7, 3, 3, 3, 7, 3, 3, 3, 3, 2, 2,

2, 2, 2, 2, 2, 2, 2, 2, 0, 0, 8, 8, 8, 8, 0, 8, 0, 0, 8, 0, 8, 8,

8, 0, 0, 0, 0, 0, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 2, 8, 8, 2, 2, 8,

2, 8, 2, 8, 2, 7, 7, 7, 4, 4, 2, 7, 7, 4, 4, 7, 7, 7, 7, 7, 7, 7,

7, 7, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 4, 4, 4, 4, 4, 4, 6, 4, 4, 2,

2, 4, 2, 2, 2, 2, 2, 4, 2, 2, 5, 5, 5, 5, 3, 7, 5, 3, 3, 3, 5, 3,

5, 5, 3, 3, 3, 3, 5, 5, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 3, 3, 3,

3, 3, 3, 3, 3, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 8, 0, 8, 0, 8, 8,

8, 0, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 4, 2, 4, 4, 4, 4, 2, 2,

2, 2, 8, 6, 6, 8, 6, 6, 8, 6, 8, 4, 5, 5, 5, 5, 5, 5, 3, 5, 5, 5,

8, 0, 8, 8, 8, 8, 7, 3, 0, 7, 5, 5, 5, 5, 5, 5, 3, 5, 5, 3, 5, 6,

1, 6, 6, 6, 5, 5, 5, 5, 4, 4, 4, 4, 4, 4, 4, 2, 4, 4, 5, 5, 5, 5,

5, 5, 5, 5, 5, 5, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 8, 0, 8, 8, 8, 8,

0, 8, 0, 8, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 8, 8, 6, 8, 6, 6, 6, 8,

6, 8, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 1, 1, 1, 1, 1, 1, 1, 5, 5, 5,

3, 7, 3, 8, 7, 7, 7, 7, 8, 3, 8, 8, 0, 0, 8, 0, 8, 0, 8, 0, 8, 0,

8, 8, 8, 0, 0, 8, 0, 0, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 7, 7, 2, 7,

7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 3, 3, 3, 3, 3, 3,

3, 3, 3, 3]

Οι συντεταγμένες των κέντρων των ομάδων δίνονται από τον πίνακα πινάκων:

[[-55.77771776, 7.61565749, -15.90525503],

[ 82.85007772, 19.24170854, -10.67340029],

[ 16.44496062, 11.64775595, 41.14730751],

[ 1.54694918, -43.20392178, -3.86418205],

[-22.75507386, 45.85909856, 26.6643095 ],

[ 49.55235068, -35.08766865, -14.14549696],

[ 31.66461452, 61.24766025, -5.94953146],

[-20.58564363, -17.40159364, 33.74364662],

[-18.18547206, 16.77872557, -34.82673789],

[-65.13098952, -64.14475486, -24.98642192]]

Τέλος, ο αριθμός των επαναλήψεων που έγινε είναι: 7.

The model scenarios of the two exercises performed to a system that is Dual core 2.0GHz with 4GB RAM and 500GB storage size and runs under Windows 10 Operating System. Python 2.7 and 3.7 versions have been used. Libraries and modules such as numpy, pandas, scikit-learn, keras, tensorflow have been imported into the scripts for this project to be carried out successfully.