**Σχολή Ηλεκτρολογών Μηχανικών και Μηχανικών Υπολογιστών**

**ΤΗΛ311-Στατιστική Μοντελοποίηση και Αναγνώριση Προτύπων**

# *3η Σειρά Ασκήσεων*

**Μαραγκάκη Μαρία**
Α.Μ.:2015030153

**Πολυτεχνείο Κρήτης**
Ημερομηνία παράδοσης: 29/05/2020

# Contents

# 1 Κατηγοριοποίηση κειμένου με k-NN

Στη συγκεκριμένη άσκηση υπολοποιήθηκε ο αλγόριθμος k-NN με σκοπό την κατηγοριοποίηση των κειμένων που δόθηκαν.

1. Σε πρώτη φάση για την εύρεση των σημαντικότερων λέξεων στα κείμενα υπολογίστηκε η εντροπία κάθε λέξης. Δημιουργήθηκε ένας πίνακας που περιέχει τις εντροπίες των λέξεων και από αυτόν επιλέχθηκαν οι 300 λέξεις που έχουν την μικρότερη εντροπία, βρίσκονται δηλαδή λιγές φορές στο κάθε κείμενο, πράγμα που καθιστά ευκολότερη την ομαδοποίηση τους. Οι συχνότητες εμφάνισης των λέξεων βρίσκονται σε έναν term-document πίνακα M διαστάσεων nDxnT, όπου nD ο αριθμός των γραμμών και κατα αντιστοιχία των κειμένων και nT ο αριθμός των στηλών και κατά αντιστοιχία των λέξεων. Για τον υπολογισμό της εντροπίας είναι αναγκάιος ο υπολογισμός της κανονικοποιημένης συχνότητας της λέξης που υπολογίζεται με τον παρακάτω τύπο: $p_{ij} = \frac{f_{ij}}{\sum_{i=1}^{nD} f_{ij}}$. Στη συνέχεια υπολογίστηκαν οι εντροπίες των λέξεων με τον τύπο του Shannon $H_{ij} = -\sum_{i-1}^{n} p_{ij} log(p_{ij})$. Έπειτα δημιουργήθηκε ένα νέο λεξικό που περιέχει τις 300 σημαντικότερες λέξεις με την χαμηλότερη εντροπία και τις αντίστοιχες τιμές $f_{ij}$ που υπολογίστηκαν με την συνάρτηση *tfidf*.

2. Στη συνέχεια υλοποήθηκε ο αλγόριθμος k-NN ο οποίος ταξινομεί τα δεδομένα με βάση τον αριθμό τον κοντινότερων γειτόνων και τον τύπο της μετρικής απόστασης. Οι τύποι για την μετρική απόσταση που χρησιμοποιήθηκαν ήταν:

   - Ευκλείδεια απόσταση: $d(\mathbf{x},\mathbf{y}) = \sum_{i=1}^{p}(x_i - y_i)^2$
   - Cosine similarity: $cos(\theta) = \frac{\sum_{i=1}^{p}(x_i y_i)}{\sqrt{\sum_{i=1}^{p}(x_i)^2}\sqrt{\sum_{i=1}^{p}(y_i)^2}}$

   Αφού επιλεγεί η επιθυμητή μετρική απόσταση επιστρέφεται ένας διάνυσμα αποστάσεων όπου για την ευκλείδεια μεγαλύτερες τιμές στις αποστάσεις σημαίνει μεγαλύτερη απόσταση των δειγμάτων. Ενώ για το cosine similarity μεγαλύτερες τιμές στις αποστάσεις υποδεικνύουν μεγαλύτερη ομοιότητα στα κείμενα. Συνεπώς στην πρώτη περίπτωση έγινε ταξινόμηση του πίνακα κατά αύξουσα σειρά και επιλέχθηκαν από τα train labels τα πρώτα k με βάση τις θέσεις των ταξινομημένων αποστάσεων, όπου k ο αριθμός των κοντινότερων γειτόνων, ενώ στη δεύτερη περίπτωση πραγματοποιήθηκε ταξινόμηση κατά φθήνουσα σειρά και επιλέχθηκαν τα train labels τα πρώτα k. Κατόπιν, ως label του test δείγματος, επιλέχθηκε η κλάση η οποία εμφανίζεται περισσότερες φορές στους k κοντινότερους γείτονες.

3. Τέλος, εφαρμόστηκε K-Fold Cross-validation με 5 folds, όπου δημιουργήται μια επαναληπτική διαδικασία στην οποία κάθε φορά ένα fold είναι το test fold και τα υπόλοιπα 4 είναι

τα train με σκοπό την εκπαίδευση του μοντέλου. Τα αποτελέσματα παρουσιάζονται παρακάτω.

(a) Για την ευκλείδεια απόσταση τα αποτελέσματα είναι τα εξής:

| Number of K nearest neighbors: 1 | | | | |
|---|---|---|---|---|
| Fold 1 | Fold 2 | Fold 3 | Fold 4 | Fold 5 |
| 31.4% | 27.07% | 27.07% | 29.22% | 26.33% |
| Total Accuracy: 28.22% | | | | |

| Number of K nearest neighbors: 3 | | | | |
|---|---|---|---|---|
| Fold 1 | Fold 2 | Fold 3 | Fold 4 | Fold 5 |
| 22.91% | 23.1% | 23.82% | 24.18% | 24.54% |
| Total Accuracy: 23.71% | | | | |

| Number of K nearest neighbors: 5 | | | | |
|---|---|---|---|---|
| Fold 1 | Fold 2 | Fold 3 | Fold 4 | Fold 5 |
| 23.46% | 23.82% | 23.1% | 21.87% | 22.74% |
| Total Accuracy: 23% | | | | |

| Number of K nearest neighbors: 10 | | | | |
|---|---|---|---|---|
| Fold 1 | Fold 2 | Fold 3 | Fold 4 | Fold 5 |
| 22.06% | 22.74% | 22.06% | 22.38% | 22.14% |
| Total Accuracy: 22.27% | | | | |

(b) Για το Cosine Similarity τα αποτελέσματα είναι τα εξής:

| Number of K nearest neighbors: 1 | | | | |
|---|---|---|---|---|
| Fold 1 | Fold 2 | Fold 3 | Fold 4 | Fold 5 |
| 22.14% | 22.38% | 21.75% | 22.38% | 22.38% |
| Total Accuracy: 22.2% | | | | |

| Number of K nearest neighbors: 3 | | | | |
|---|---|---|---|---|
| Fold 1 | Fold 2 | Fold 3 | Fold 4 | Fold 5 |
| 22.14% | 22.38% | 22.38% | 22.06% | 22.06% |
| Total Accuracy: 22.2% | | | | |

| Number of K nearest neighbors: 5 | | | | |
|---|---|---|---|---|
| Fold 1 | Fold 2 | Fold 3 | Fold 4 | Fold 5 |
| 22.38% | 22.06% | 22.38% | 21.83% | 22.38% |
| Total Accuracy: 22.2% | | | | |
| Number of K nearest neighbors: 10 | | | | |
| Fold 1 | Fold 2 | Fold 3 | Fold 4 | Fold 5 |
| 22.38% | 22.38% | 22.06% | 22.14% | 22.06% |
| Total Accuracy: 22.2% | | | | |

Παρατηρείται ότι οι αποδόσεις και στις δυο περιπτώσεις δεν είναι οι αναμενόμενες καθώς ο αλγόριθμος έχει αρκετά μεγάλες αποδόσεις όσο αυξάνεται ο αριθμός των γειτόνων. Η απόκλιση αυτή οφείλεται είτε σε λανθασμένη υλοποίηση του αλγορίθμου είτε σε λανθασμένη είσοδο δεδομένων.

## 2 K-means clustering

Στη συγκεκριμένη άσκηση υλοποιήθηκε ο αλγόριθμος k-means με σκοπό αρχικά την ομαδοποίηση δεδομένων σε K-κλάσεις και στη συνέχεια την εφαρμογή του για την συμπίεση μιας εικόνας. Ο αλγόριθμος K-means λειτουργεί ως εξής, αρχικά θέτει τυχαίες τιμές στα κέντρα των κλάσεων και στην συνέχεια βελτιώνει την αρχική του επιλογή τοποθετώντας δείγματα στην κλάση με την μικρότερη απόσταση μεταξύ του δείγματος και του κέντρου της κλάσης και ξαναυπολογίζοντας τα κέντρα των κλάσεων.

1. Σε πρώτη φάση, υλοποιήθηκε συνάρτηση *findClosestCentroinds.m* όπου υπολογίζει την κοντινότερη απόσταση $c^i := j$ η οποία ελαχιστοποιέι την απόσταση $||x^i - \mu^j||^2$ απο τα αρχικά κέντρα των κλάσεων, όπου $c^i$ είναι ο δείκτης του κοντινότερου κέντρου στο $x^i$ και $\mu_j$ είναι το διάνυσμα τιμών του κέντρου j.

2. Εν συνεχεία, συπληρώθηκε η συνάρτηση *computeCentroids.m* η οποία υπλογίζει τα κέντρα των κλάσεων ως εξής:

$$\mu_k = \frac{1}{|C_k|}\Sigma_{i\in|C_k|}x^i$$

όπου $|C_k|$ είναι το σύνολο των παραδειγμάτων που έχουν αντιστοιχηθεί στην κλάση k. Οπότε βρέθηκε εκτελώντας την πρώτη συνάρτηση ότι αρχικά τα κοντινότερα κεντροειδή για τα αρχικά δείγματα είναι τα 1,3,2 όπως ήταν αναμενόμενο. Έπειτα εκτελώντας

την δέυτερη συνάρτηση βρέθηκε ότι τα κεντροειδή που υπολογίστηκαν ήταν τα εξής: $[2.428301, 3.157924]$, $[5.8135032.633656]$, $[7.1193873.616684]$, όπως ήταν αναμενόμενο.

3. Έπειτα, χρησιμοποιήθηκαν οι παραπάνω συναρτήσεις με σκοπό την εύρεση των κέντρων στο δοσμένο dataset τα αποτελέσματα παρουσιάζονται παρακάτω.

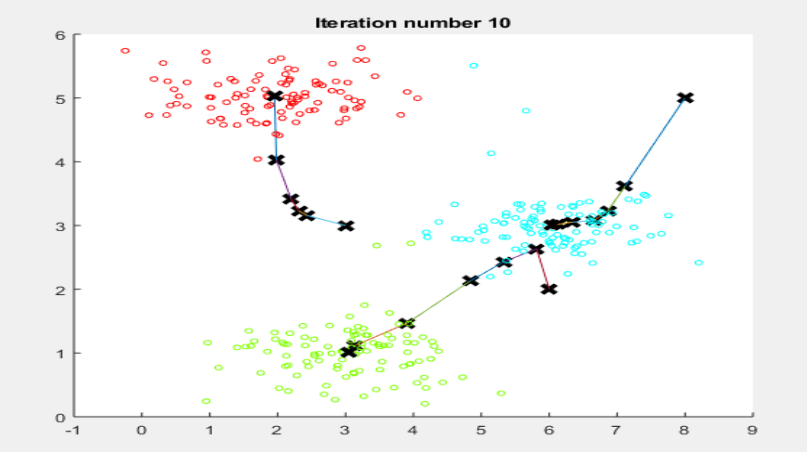


Figure 1: Εύρεση κεντροειδών έπειτα απο 10 επαναλήψεις του αλγορίθμου

Παρατηρείται ότι έπειτα απο 10 διαδοχικές επαναλήψεις, αλλάζει σταδιακά το κέντρο κάθε κλάσης στη 7η επανάληψη έχει βρεθέι το κέντρο κάθε κλάσης.

4. Τέλος εφαρμόστηκε ο αλγόριθμος με σκοπό την συμπίεση μια εικόνας. Το αποτέλεσμα παρουσιάζεται παρακάτω.

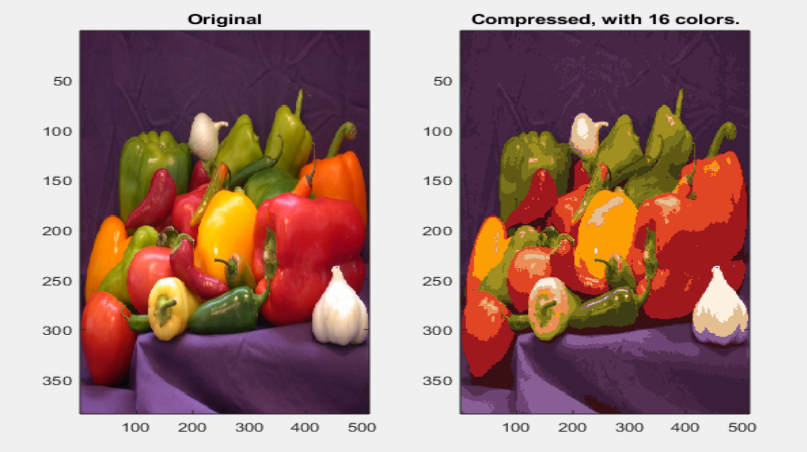


Figure 2: Συμπίεση εικόνας με την εφαρμογή του K-means

Η διαδικασία ήταν η εξής, αρχικά εφαρμόστηκε ο αλγόριθμος σε όλα τα pixel της εικόνας και στη συνέχεια έγινε αντιστοίχηση του κάθε Pixel στο κοντινότερο απο τα 16 κέντρα κλάσεων με αποτέλεσμα η εικόνα να συμπιεστεί με μόνο 16 χρώματα.

# 3 GMMs και εκτίμηση με τον αλγόριθμο Expectation Maximization

Στη συγκεκριμένη άσκηση υλοποιήθηκε ο αλγόριθμος Expectation Maximization για την εκπαίδευση των παραμέτρων ενός μοντέλου Gaussian Mixture(GMM). Συγκεκριμένα ο αλγόριθμος εφαρμόστηκε στα λουλούδια Fisher Iris με σκοπό την εύρεση της κλάσης κάθε λουλουδιού. Τα βήματα που ακολουθήθηκαν ήταν τα εξής:

- Αρχικά, αρχικοποιήθηκαν τα $\mu_\kappa$, $\Sigma_\kappa$,$\pi_\kappa$ με τη βοήθεια του αλγορίθμου kmeans.
- Στη συνέχεια, υπολογίστηκαν οι πιθανότητες likelihood για κάθε δείγμα και κάθε κλάση και κατόπιν σύμφωνα με τον παρκάτω τύπου υπολογίστηκε το $\gamma(z_{nk})$.

$$\gamma(z_{nk}) = (\pi_\kappa * N(x_n|\mu_\kappa, \Sigma_\kappa))$$

- Κατόπιν, υπολογίστηκαν οι τιμές $\mu_\kappa^{new}$, $\Sigma_\kappa^{new}$,$\pi_\kappa^{new}$ σύμφωνα με του παρακάτω τύπους:

$$\mu_\kappa^{new} = \frac{1}{N_k}\Sigma_{n=1}^{N}\gamma(z_{nk}) * x_n$$

$$\Sigma_\kappa^{new} = \frac{1}{N_k}\Sigma_{n=1}^{N}\gamma(z_{nk}) * (x_n - \mu_\kappa^{new}) * (x_n - \mu_\kappa^{new})^T$$

$$\pi_\kappa^{new} = \frac{N_k}{N}$$

  Οπότε για κάθε κλάση επιστρέφοντα οι τιμές $P_m$,M,S

- Σύμφωνα με τις τιμές που επιστρέφονται, την Pm, τον μέσο και τον πίνακα συνδιασποράς κάθε κλάσης, υπολογίζεται η likelihood πιθανότητα κάθε κλάσης για κάποιο στοιχείο και το στοιχείο ταξινομείται στην κλάση με τη μεγαλύτερη likelihood. Παρατηρείται ότι ο αλγόριθμος δεν έχει κάποια σταθερή απόδοση και αυτο οφείλεται στην αρχικοποίηση των κλάσεων απο τον K-means. Συνεπώς κάθε φορά που εκτελείται ο κώδικας η απόδοση είναι διαφορετική με μέγιστη απόδοση να είναι αυτή της τάξης του 96,6% δηλαδή να ταξινομούνται μόλις 5 δείγματα σε λάθος κλάση.

# 4 Παραπομπές


- Διαλέξεις μαθήματος και διαφάνειες φροντιστηρίου.
- https://www.mathworks.com/matlabcentral/fileexchange/67018-k-nearest-neighbors-knn-algorithm
- https://www.mathworks.com/matlabcentral/fileexchange/47355-k-means-algorithm-with-the-application-to-image-compression?focused=3831055tab=function