Τεχνική Αναφορά *για την εργασία στις*

Αποθήκες & Εξόρυξη Δεδομένων

Συσταδοποίηση Δεδομένων και Εντοπισμός Ανωμαλιών

Νέρων Μιχαήλ Παναγιωτόπουλος [3990], Χρήστος Μπαλακτσής [3865]

ΑΡΙΣΤΟΤΕΛΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΟΝΙΚΗΣ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Εργαλεία

Ο κώδικας που απαντά στα ερωτήματα 1-5 αναπτύχθηκε σε **Python** (v3.10) με την υποστήριξη της βιβλιοθήκης **PySpark** για την εκτέλεση των αλγορίθμων συσταδοποίησης σε περιβάλλον **Apache Spark**.

Προαιρετικά, αξιοποιήθηκαν και οι βιβλιοθήκες **matplotlib** και **pandas** για τη διαχείριση των δεδομένων γραφικά και δομικά, αντίστοιχα. Για plotting των αποτελεσμάτων μπορεί να γίνει pip install της πρώτης και uncomment των αντίστοιχων εντολών της main.py.

Εκτέλεση σε περιβάλλον Linux

```
$ python3 main.py <dataset>.csv
```

όπου <dataset> το όνομα του αρχείου δεδομένων.

Για την εγκατάσταση των βιβλιοθηκών:

```
$ pip install -r requirements.txt
```

Προεπεξεργασία Δεδομένων

Για τον **μετασχηματισμό** των δεδομένων στην **κλίμακα 0-1**, εφαρμόστηκε σε κάθε στήλη X ο τύπος $X'_i = \frac{X_i - \min(X)}{\max(X) - \min(X)}$.

```python
def normalize(df: DF) -> DF:
    min0 = float(df.agg(F.min("_c0")).collect()[0][0])
    max0 = float(df.agg(F.max("_c0")).collect()[0][0])
    min1 = float(df.agg(F.min("_c1")).collect()[0][0])
    max1 = float(df.agg(F.max("_c1")).collect()[0][0])

    min_max_values = {"_c0":{},"_c1":{}}
    min_max_values["_c0"]["min"] = min0
    min_max_values["_c0"]["max"] = max0
    min_max_values["_c1"]["min"] = min1
    min_max_values["_c1"]["max"] = max1

    df_normalized = df.withColumn("_c0", (df["_c0"] - min0) / (max0 - min0))\
        .withColumn("_c1", (df["_c1"] - min1) / (max1 - min1))

    return df_normalized, min_max_values
```

Στις γραμμές 7-11 αποθηκεύουμε τα ακρότατα κάθε στήλης, προκειμένου να αποκανονικοποιήσουμε τα outliers σε επόμενο ερώτημα.

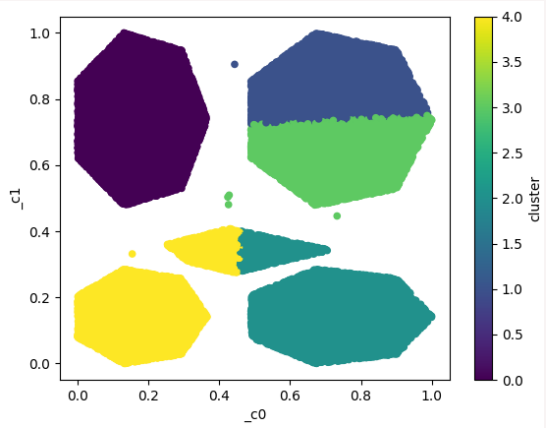
Για τον **καθαρισμό** των δεδομένων πριν την απόθεσή τους ως είσοδο στον αλγόριθμο K-Means, χρησιμοποιήθηκε η μέθοδος **dropna** της βιβλιοθήκης **pyspark.sql.dataframe**, η οποία **καθάρισε** όλες τις εγγραφές (γραμμές του csv) που ήταν ελλιπείς ή ασύμφωνες με τα πρότυπα του csv για dataframe.

```python
def cleanup(df: DF) -> DF:
    cleandf = df.dropna(how='any')
    return cleandf
```

Συσταδοποίηση με K-Means

Απαιτούμενο της εργασίας ήταν η χρήση της μεθόδου **KMeans** της βιβλιοθήκης **pyspark.ml.clustering** για τη συσταδοποίηση των δεδομένων, δοθέντος a priori του πλήθους των συστάδων. Συνεπώς, υλοποιήθηκε η συνάρτηση **kmeansCluster** που καλεί την εν λόγω μέθοδο με καταλλήλως μορφοποιημένα dataframes και **τα επιστρέφει labeled με βάση τη συστάδα που αποδόθηκαν.**

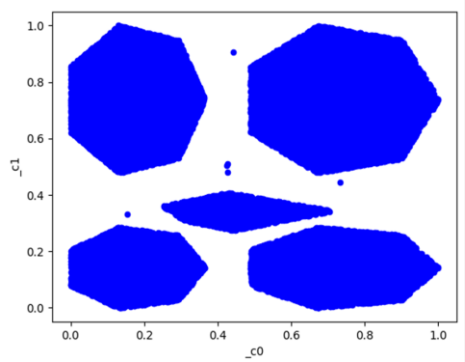
```python
def kmeansCluster(df: DF, n: int):
    vec_assembler = VectorAssembler(inputCols = ["_c0", "_c1"],
                                    outputCol='features', handleInvalid="skip")
    final_data = vec_assembler.transform(df)
    kmeans = KMeans(k=n, featuresCol="features", predictionCol="cluster")
    model = kmeans.fit(final_data)
    clustered = model.transform(final_data)
    return clustered.select("_c0", "_c1", "cluster")
```



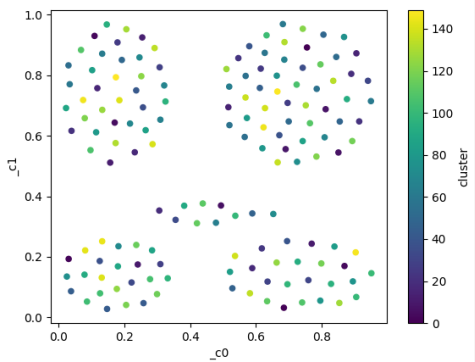
Ωστόσο, αμέσως έγινε αντιληπτό ότι, τόσο εξαιτίας του **θορύβου** (outliers) όσο και της **κατανομής** και φύσης των δεδομένων και των συστάδων, **ο K-Means αποτυγχάνει να συσταδοποιήσει κατάλληλα για k=5**, καθώς ενέχει μεγάλη ευαισθησία σε αυτούς τους παράγοντες σε συνδυασμό με την επιλογή του *k*.

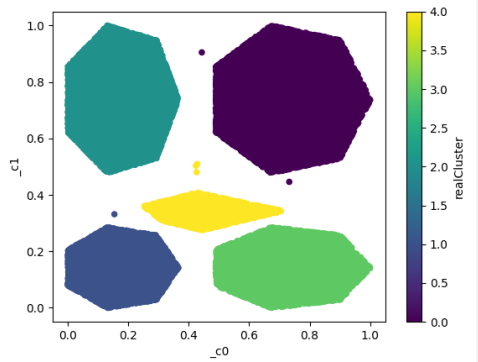
Συσταδοποίηση με K-Means & Single-Link

Μετά από πειραματισμό με διάφορους αλγορίθμους (ιεραρχικούς, τμηματοποίησης) και με στόχο τη διατήρηση της χαμηλής χωρικής και χρονικής πολυπλοκότητας, καταλήξαμε στην παρακάτω προσέγγιση.



150 κέντρα





KMeans (Apache Spark) για ***k=150*** στο αρχικό dataset

Single-Link (self-defined) για την **ένωση των κοντινότερων κέντρων** μέχρι να μείνουν **5** (σ.σ. 5 συστάδες).

$O(n^2)$ αλλά **n=150**, οπότε τελικά είναι **μικρό** το κόστος (22500 προσπελάσεις)

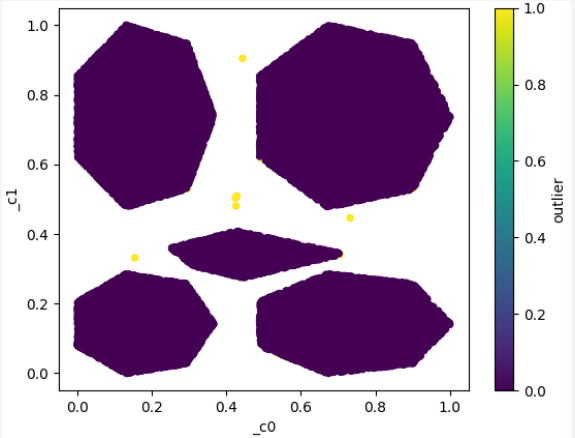
Συνένωση των σημείων
(πραγματοποιείται ταυτόχρονα με τη συνένωση κέντρων, δημιουργώντας μια νέα συστάδα ως ένωση δύο άλλων)

Ανίχνευση Ακροτάτων & Ανωμαλιών

Για τον εντοπισμό των **outliers**, υλοποιήθηκε η μέθοδος **`findOutliers(df: DF, n_stdev: int)`**, αξιοποιώντας το 1ο βήμα της συσταδοποίησης (k-means, *k=150*).

Κάνοντας την παραδοχή ότι **εντός κάθε συστάδας** τα σημεία της είναι προσεγγιστικά **ομοιόμορφα κατανεμημένα**, η προσέγγισή μας είναι η ακόλουθη:

1. Χρησιμοποιούμε τη συσταδοποίηση, προκειμένου **να υπολογίσουμε την απόσταση μεταξύ κάθε σημείου και του κέντρου της συστάδας του**.
2. **Υπολογίζουμε τη μέση απόσταση σημείου-κέντρου για κάθε συστάδα και την τυπική απόκλιση**.
3. Χωρίζουμε τα σημεία σε **δύο ομάδες**, **ακρότατα και μη-ακρότατα**, συναρτήσει της **απόκλισής τους από την αντίστοιχη μέση απόσταση της συστάδας τους** (βήμα 2), χρησιμοποιώντας την **τυπική απόκλιση ως άνω όριο** για να βρούμε τα ακρότατα (outliers) σε χρόνο $O(n)$.
4. Συγκεντρώνουμε τα σημεία της ομάδας των ακροτάτων και τα **αποκανονικοποιούμε** για να αντιστοιχηθούν με τα πραγματικά δεδομένα εισόδου.



Πρόταση συσταδοποίησης από το ChatGPT

Ζητήθηκαν από το ChatGPT οδηγίες για τον εντοπισμό συστάδων σε ένα σύνολο δισδιάστατων δεδομένων, δεδομένου εκ των προτέρων του πλήθους των συστάδων και λαμβάνοντας υπόψη την ανάγκη χαμηλής πολυπλοκότητας και τη δυνατότητα παραλληλισμού της διαδικασίας. Η συζήτηση μπορεί να βρεθεί εδώ και η εκτενής κριτική αξιολόγηση εδώ.

- Προτείνει τη χρήση του αλγορίθμου **k-means** και περιγράφει αφαιρετικά τη λογική του *(τυχαία επιλογή 5 σημείων ως κέντρα προσπέλαση κάθε σημείου και ένταξή του στην κοντινότερη ομάδα, αναδρομικό επαναϋπολογισμό των κέντρων των συστάδων κλπ.)*. Πρόκειται για **μια επιστημονικά αναμενόμενη προσέγγιση**, μιας και τόσο για δεδομένο αριθμό συστάδων όσο και για τη δυνατότητα παραλληλισμού ενδείκνυται γενικά η χρήση του. (Jin & Han, 2011)
- Πρόκειται για λελογισμένου/**χαμηλού κόστους πρόταση** και πράγματι εύκολα **παραλληλοποιήσιμη**. Μάλιστα, προτείνει -μεταξύ άλλων- τη χρήση του **Spark** για την κατανομή και παραλληλοποίηση των δεδομένων και διαδικασιών. (Han, 2023)
- Σχολιάζει το **πρόβλημα** τερματισμού/απόδοσης του αλγορίθμου λόγω αδυναμίας **σύγκλισης**, σε συμφωνία με την περιγραφή του φαινομένου στη βιβλιογραφία. **Σωστά προτείνει ως τρόπο περιορισμού του προβλήματος σύγκλισης** τον ορισμό κατάλληλου κατωφλίου στη μέτρηση της μεταβολής της κατανομής, ενώ **σχετικά με την επιλογή του κατάλληλου *k* απέτυχε να αντιπροτείνει τεχνικές** που βελτιώνουν το αποτέλεσμα του k-means, όταν το *k* είναι γνωστό, καταδεικνύοντας ότι πρέπει να οριστεί με τη δεδομένη του τιμή (λ.χ. *k*=5) σε κάθε τέτοια περίπτωση. Για τις περιπτώσεις που το *k* δεν είναι γνωστό, **ορθά προτείνει τον πειραματισμό με διάφορες τιμές** σε συνδυασμό με τεχνικές (πχ. elbow) καθορισμού προϋποθέσεων τερματισμού της διαδικασίας.
- **Προτείνει ορθά τη χρήση και αξιοποίηση εμπειρικών τεχνικών και μετρικών**, όπως τον προσδιορισμό συντελεστή σιλουέτας κ.α. για την εκτίμηση του μέτρου επιτυχίας της συσταδοποίησης.

Πρόταση συσταδοποίησης από το ChatGPT

- Για τον **χειρισμό των outliers** προτείνει την εξέταση τεχνικών **προεπεξεργασίας** των δεδομένων για την αφαίρεσή τους. **Σωστά** αναφέρει ότι η παρουσία τους διαστρεβλώνει τα αποτελέσματα του k-means. Προτείνει μια σειρά από τεχνικές που μπορούν να εφαρμοστούν (όπως Z-score) για την αφαίρεση των οριακών τιμών, αλλά και αλγορίθμους, όπως ο **DBSCAN αντί του k-means**, τονίζοντας την αξιοποίηση της διατήρησης τους σε ξεχωριστή ομάδα, **αγνοώντας** όμως την αύξηση της πολυπλοκότητας.

- Παρά τον λεκτικό προσδιορισμό των χαρακτηριστικών των δεδομένων (**διακριτές συστάδες** και **ύπαρξη ανωμαλιών στο χωρικό ενδιάμεσο τους**, αγνοεί τη φύση των δεδομένων και **αποτυγχάνει να προτείνει ένα συνδυασμό τεχνικών** που να καλύπτει τόσο την -παραλληλοποιήσιμη- συσταδοποίηση όσο και την ανίχνευση των οριακών τιμών. Εκπίπτει μάλιστα σε **αντίφαση**, καθώς επισημαίνει τη βαρύτητα του domain knowledge και των data features, αλλά πρακτικά δεν τα λαμβάνει υπόψη.

- Χρησιμοποιώντας τη μηχανή του ChatGPT 4 (μέσω του Bing Chat) για την παράθεση screenshot της κατανομής των δεδομένων στον χώρο, αντιπρότεινε ως τιμή του *k* το 6, **θεωρώντας το σύνολο των outliers μια ακόμη συστάδα**, τεχνική που **αντιστρατεύεται της βιβλιογραφίας**, μιας και ο k-means τυπικά δε θα ενώσει όλες τις οριακές τιμές σε ξεχωριστή συστάδα, αλλά μάλλον θα ανακαλύψει νέα κατανομή. (Barai & Dey, 2017)

Η προτεινόμενη λύση του ChatGPT ακολουθεί στα περισσότερα σημεία της την επιστημονική πρακτική, αγνοώντας όμως σημαντικά την ιδιαιτερότητα του εκάστοτε προβλήματος και υποπίπτοντας σε μερικά λογικά σφάλματα. Για γνωστά χαρακτηριστικά δεδομένων δεν είναι σε θέση να παράξει καινοτόμα μεθοδολογία για τον εντοπισμό των συστάδων. Σημειώνει σωστά ότι πρέπει να δίνεται έμφαση στη *γνώση πεδίου* και στην *μορφή των δεδομένων*, αλλά πρακτικά αδυνατεί να εκμεταλλευτεί τις επιπλέον πληροφορίες. Σχετικά με την υλοποίηση του k-means απαντάει λανθασμένα σχετικά με τον αριθμό του *k* όταν παρατηρεί τις συστάδες απεικονιστικά, ενώ η θεώρηση του συνόλου των outliers ως ξεχωριστή συστάδα προδίδει ότι αγνοείται το πώς σχηματίζονται σταδιακά τις συστάδες.

Ευχαριστούμε!

Ερωτήσεις;

Βιβλιογραφικές αναφορές

Barai, A., & Dey, L. (2017). Outlier Detection and Removal Algorithm in K-Means and Hierarchical Clustering. World Journal of Computer Application and Technology, 24.

Han, M. (2023). Research on optimization of K-means Algorithm Based on Spark. 2023 IEEE 6th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC) (pp. 1829-1836). Chongqing, China: IEEE.

Jin, X., & Han, J. (2011). K-Means Clustering. In Encyclopaedia of Machine Learning (pp. 563-564). Boston, MA: Springer.

Shalmoli Gupta, R. K. (2017). Local Search Methods for k-Means with Outliers. VLDB Endowment.

mpalaktsc@csd.auth.gr

neronmkp@csd.auth.gr