ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

Προηγμένα Θέματα Βάσεων Δεδομένων

Εξαμηνιαία εργασία – Θέμα 3ο

Χειμερινό εξάμηνο 2020

Μασούρης Αθανάσιος

03115189 9ο εξάμηνο

Μεθοδολογία

Το αρχείο "yellow_tripdata_1m.csv" περιέχει δεδομένα που αφορούν διαδρομές taxi στην Νέα Υόρκη. Περιέχονται πληροφορίες για το id της διαδρομής, την ημερομηνία και ώρα έναρξης και λήξης της διαδρομής, το γεωγραφικό μήκος και πλάτος των σημείων επιβίβασης και αποβίβασης και τέλος το κόστος της διαδρομής.

Παράδειγμα:

Id, Ημ/Ώρα επιβίβασης, Ημ/Ώρα αποβίβασης, Γεωγρ. μήκος επιβίβασης, Γεωγρ. πλάτος επιβίβασης Γεωγρ. μήκος αποβίβασης, Γεωγρ. πλάτος αποβίβασης, Κόστος διαδρομής

369367789289,2015-03-27 18:29:39,2015-03-27 19:08:28,-73.975051879882813,40.760562896728516,-73.847900390625,40.7326850 89111328,34.8

Στο θέμα 3ο ζητείται να βρούμε τις κεντρικές συντεταγμένες των top 5 περιοχών επιβίβασης πελατών, με τη χρήση του αλγορίθμου k-means (5 clusters) του οποίου ζητείται υλοποίηση σε spark.

Δίνεται επίσης ο ψευδοκώδικας για τον αλγόριθμο k-means:

```
points = read_data()
k = 5
MAX_ITERATIONS = 3

# Initialize centroids
centroids = get_k_first_points(points, k)
iterations = 1
while not (iterations > MAX_ITERATIONS) {
    iterations += 1

    # For each point in the dataset, chose the closest centroid.
    # Make that centroid's index the point's label.
    points_and_labels = get_labels(points, centroids)

    # Each new centroid is the mean of the points that have the
    # same centroid's label.
    new_centroids = get_new_centroids(points_and_labels)
    centroids = new_centroids
}
print(centroids)
```

Με βάση τον αλγόριθμο αυτό παρατηρούμε ότι πρέπει αρχικά να φορτώσουμε τα δεδομένα μας. Για να το κάνουμε δημιουργούμε πρώτα SparkContext και στη συνέχεια συνδεόμαστε στην πύλη 9000 του master από τον οποίο παίρνουμε τα δεδομένα "yellow_tripdata_1m.csv" που έχουμε φορτώσει στο hdfs.

Αρχικά, παρατηρούμε ότι θέλουμε να βρούμε τα κεντρικά σημεία των top 5 περιοχών για τα σημεία επιβίβασης. Συνεπώς από τα δεδομένα που περιέχονται στο αρχείο, χρήσιμα είναι μόνο τα πεδία που αφορούν τις συντεταγμένες του σημείου επιβίβασης.

Επειδή όπως είδαμε παραπάνω τα δεδομένα είναι σε μία γραμμή χωρισμένα με ",", εφαρμόζουμε μια συνάρτηση map σε κάθε γραμμή δεδομένων, την οποία χωρίζουμε στα "," και κρατάμε σε μία tuple μόνο τα πεδία για τις συντεταγμένες των σημείων επιβίβασης.

*Προαιρετικό: Παρατηρήθηκε ότι υπήρχαν δεδομένα στο αρχείο για τα οποία δεν είχαν καταγραφεί τα σημεία επιβίβασης, δηλαδή είχαν μηδενικές τιμές στις συντεταγμένες σημείου επιβίβασης. Αν θέλουμε μπορούμε να αφαιρέσουμε αυτά τα δεδομένα με τη χρήση μιας συνάρτησης filter στην οποία κρατάμε μόνο δεδομένα με τιμές συντεταγμένων εντός κάποιον καθορισμένων προσεγγιστικών διαστημάτων για τα γεωγραφικά μήκη και πλάτη της Αμερικής.

Αφού πλέον έχουμε τα δεδομένα, μπορούμε να πάμε στην αρχικοποίηση των centroids. Επιλέχθηκαν τα πρώτα 5 δεδομένα, ως αρχικές τιμές για τα centroids, τα οποία στον κώδικά μας αποθηκεύονται σε μία δομή python dictionary με keys το id του cluster (0,1,2,3,4,5) και value την tuple με τις συντεταγμένες. Για την αρχικοποίηση υλοποιήσαμε τη συνάρτηση "initialize_centroids(data, k)".

Έπειτα ξεκινάμε την επαναληπτική διαδικασία του αλγορίθμου k-means (μέχρι να φτάσουμε τις ορισμένες MAX_ITERATIONS). Στο σώμα αυτής της επαναληπτικής διαδικασίας πρέπει να πραγματοποιήσουμε τα εξής:

- Κατάταξη των σημείων σε clusters
- Ενημέρωση κέντρων

Για το πρώτο, χρησιμοποιούμε μια συνάρτηση map στην οποία αντιστοιχούμε κάθε γραμμή από τα αρχικά δεδομένα μας σε κάποιο cluster μέσω της συνάρτησης "cluster(x, centroids)". Στη συνάρτηση αυτή αρχικοποιούμε την απόσταση στο άπειρο και στη συνέχεια υπολογίζουμε τη μικρότερη απόσταση* του σημείου x από όλα τα centroids που έχουμε. Επιστρέφουμε το id (key στο παραπάνω dictionary) του κοντινότερου centroid.

*απόσταση: Η απόσταση υπολογίζεται με βάση τη φόρμουλα haversine για την οποία υλοποιούμε μια συνάρτηση υπολογισμού.

Έπειτα έχοντας ενα RDD με τα δεδομένα μας αντιστοιχισμένα σε κάποιο cluster, πάμε στο δεύτερο στάδιο. Για την ενημέρωση των κέντρων υλοποιήσαμε τη συνάρτηση "update_centroids" που παίρνει ως είσοδο το παραπάνω RDD. Στη συνάρτηση αυτή, εφαρμόζουμε αρχικά μια map προκειμένου κάθε γραμμή δεδομένων να έχει ως key το cluster στο οποίο έχει καταταχθεί και ως value την τούπλα με τις συντεταγμένες του σημείου επιβίβασης και έναν άσσο (1) ο οποίος θα χρησιμοποιηθεί για να μετρήσουμε το πλήθος των σημείων σε κάθε cluster. Έπειτα με τη χρήση μίας συνάρτησης reduce, αθροίζουμε ουσιαστικά για κάθε cluster τόσο τις συντεταγμένες, όσο και τους άσσους, των σημείων που ανήκουν σε αυτό. Τέλος, με μία ακόμα συνάρτηση map, για κάθε μία από τις πέντε γραμμές δεδομένων που έχουν προκύψει από την παραπάνω συνάρτηση reduce, βρίσκουμε το μέσο σημείο διαιρώντας με το άθροισμα

των άσσων (πλήθος σημείων στο cluster) τα αθροίσματα των συντεταγμένων. Επιστρέφουμε τελικά ενα dictionary με τις νέες τιμές των κέντρων.

Μετά τον επαναληπτικό βρόχο, τυπώνουμε το dictionary που περιέχει τις τελικές τιμές των centroids για κάθε cluster.