Προχωρημένα Θέματα Βάσεων Δεδομένων

Πέτρος Βαβαρούτσος
03115125

Θέμα 3ο: Machine Learning - Ομαδοποίηση δεδομένων με εκτέλεση τουk-means αλγόριθμου

Με τον αλγόριθμο kmeans θα γίνει ομαδοποίηση των σημείων επιβίβασης των πελατών σε k=5 περιοχές (clusters) και θα βρεθεί το κέντρο των σημείων της κάθε περιοχής. Αφού έγινε μία μικρή μελέτη του dataset παρατηρήθηκε ότι υπήρχαν missing values στις συντεταγμένες x,y. Πιο συγκεκριμένα υπήρχαν συντεταγμένες x,y οι οποίες είχαν τιμή ίση με 0 και αντιστοιχούσαν κάπου βαθιά στην θάλασσα. Υποθέτουμε, λοιπόν, πως οι συντεταγμένες αυτές ήταν λάθος αφού μελετάμε τις διαδρομές των ταξί στην Νέα Υόρκη και, γι' αυτό το λόγο, παραλείφθηκαν. Στην συνέχεια, παρατίθεται ψευδοκώδικας για τα παρακάτω.

```
map(key, value):
    String data = value.toString()
    String line = data.split(‘,’)
    float x = float(line[3])
    float y = float(line[4])
    if (x!=0 and y!=0)
        emit(x, y)
```

Μετά το φιλτράρισμα του dataset για missing values εκτελούμε επαναληπτικά τον αλγόριθμο (για αριθμό επαναλήψεων ίσο με 3). Η επαναληπτική εκτέλεση αυτή αναλύεται παρακάτω:

Αρχικά , για κάθε σημείο x, y υπολογίζεται η haversine απόστασή του από όλα τα κέντρα. Στην συνέχεια, βρίσκουμε το πιο κοντινό σε αυτό κέντρο και επιστρέφεται το index του.

```
map(key, value):
    index = 0
    float minDistance = haversine(value[0], value[1],
centroid[0][0], centroid[0][1])
    for i in range(k-1):
        dist =  haversine(value[0], value[1],
centroid[i+1][0], centroid[i+1][1])
        if (dist < minDistance):
            minDistance = dist
            index = i+1
    emit(index, (value, 1))
```

Μετά την εκτέλεση του παραπάνω map το RDD που δημιουργείται είναι της μορφής

(index of closest centroid,(points,1))

Το 1 στο (points, 1) το χρειαζόμαστε για να υπολογίσουμε το πλήθος των σημείων που έχουν το ίδιο index. Στην συνέχεια επιθυμούμε να επαναυπολογίσουμε τα κέντρα. Ως εκ τούτου αρχικά θα υπολογίσουμε το άθροισμα των σημείων και το πλήθος τους ανάλογα με το index.

```
reduce(key, values):
    float sum_x = 0
    float sum_y = 0
    int num = 0
    for x in values:
        sum_x += x[0][0]
        sum_y += x[0][1]
        num += x[1]
    emit(key, ((sum_x, sum_y), num))
```

Τέλος, για τον υπολογισμό των κέντρων διαιρούμε το άθροισμα των σημείων με το πλήθος τους.

```
map(key, value):
    int num = value[1]
    emit(key, (value[0][0]/num, value[0][1]/num))
```

Μετά την εκτέλεση του παραπάνω κώδικα επαναληπτικά 3 φορές τυπώνουμε τα κέντρα και τα εξάγουμε σε ένα αρχείο στο hdfs.

Τέλος, παρατίθενται τα αποτελέσματα από την εκτέλεση του κώδικα:

```
Centroid Coordinates
1 (-74.01403589451454, 40.71059112138094)
2 (-73.94046141190235, 40.763626885130115)
3 (-73.9957046634548, 40.72206538337711)
4 (-74.00370483926541, 40.73870530458936)
5 (-73.98400182991335, 40.752753031855725)
```