Big Data - 2η ομαδική εργασία

Ομάδα
Σωτηρίου Σωτήριος, dai19052
Κακαλές Δημήτριος, dai19062

Προβλήματα

- Προσπαθώντας αμέσως να τρέξουμε τον κώδικα καταλάβαμε από τα σφάλματα που προέκυψαν πως απαιτείται η εγκατάσταση περαιτέρω λογισμικού. Απαιτούνταν η βιβλιοθήκη της Python NumPy. Εδώ αντιμετωπίσαμε και την κύρια δυσκολία μας, καθώς ενώ κάναμε εγκατάσταση της βιβλιοθήκης , υπήρχε πρόβλημα με την συμβατότητα μεταξύ της NumPy και κάποιων εκδόσεων της Python. Έτσι μετά από αρκετή αναζήτηση καταφέραμε να βρούμε μέσω του Github λύση στο πρόβλημα μας, κάνοντας εγκατάσταση της κατάλληλης έκδοσης Python.
- Επόμενη δυσκολία ήταν ο τρόπος υπολογισμού του συντελεστή περιγράμματος (Silhouette score), αφού δεν υπήρχε κάποια άμεση υλοποίηση του στην βιβλιοθήκη MLLib. Ωστόσο κάνοντας συνδυασμό των 2 βιβλιοθηκών ML και MLLib καταφέραμε να βρούμε την λύση που ψάχναμε.

Τα υπόλοιπα προβλήματα που αντιμετωπίσαμε δεν είναι άξια αναφοράς. Είχαν να κάνουν με επεξεργασία του κώδικα και διόρθωση σφαλμάτων που προέκυπταν μέχρι να φτάσουμε στο τελικό αποτέλεσμα.

Επεξήγηση Κώδικα

Σύμφωνα και με την εκφώνηση της εργασίας, ξεκινήσαμε με τον έτοιμο κώδικα της βιβλιοθήκης MLLib που υλοποιεί τον αλγόριθμο K-Means. Πάνω σε αυτόν κάναμε τροποποιήσεις για να καταλήξουμε στο τελικό αποτέλεσμα.
Πρώτη τροποποίηση ήταν η εισαγωγή σ'αυτόν μιας επιπλέον παραμέτρου η οποία ορίζει τον τρόπου αρχικοποίησης του αλγορίθμου (initializationMode) και μπορεί να δεχτεί τις 2 τιμές (random,k-means||).
Επόμενη αλλαγή είναι ο τρόπος με τον οποίο διαβάζει ένα αρχείο ο αλγόριθμος. Εδώ έγινε εισαγωγή του κατάλληλου path για διάβασμα αρχείων μέσω του hdfs και μετατροπή των δεδομένων αυτών στην κατάλληλη μορφή για χρήση.
Παρακάτω γίνονται κάποιοι έλεγχοι σχετικά με τις παραμέτρους του κώδικα και το κατά πόσο είναι έγκυρες οι τιμές τους.
Αμέσως μετά χωρίζουμε το πρόγραμμα σε 2 περιπτώσεις. Η πρώτη περίπτωση συναντάται όταν περάσουμε ως όρισμα την τιμή 0 για την παράμετρο k. Αυτό που συμβαίνει τότε είναι ο αλγόριθμος να μπαίνει σε λειτουργία δοκιμής διάφορων τιμών του k (πλήθους συστάδων) ανάμεσα σε ένα πλήθος τιμών (k=3,6,9,...,27). Κατά την λειτουργία αυτή του αλγορίθμου υπολογίζεται για κάθε k κι εμφανίζεται στην οθόνη:

- Το πλήθος των συστάδων
- Τα τελικά κέντρα των συστάδων

- Ο συντελεστής περιγράμματος
- Το μέγεθος των συστάδων
- Το συνολικό άθροισμα των τετραγώνων των αποστάσεων

Έτσι λοιπόν περνώντας ως όρισμα για k = 0 μπορούμε να επιλέξουμε μια καλή τιμή του k για τα δεδομένα μας.

Σε περίπτωση τώρα που το k δεν είναι 0 προχωράμε στην άλλη λειτουργία η οποία κάνει συσταδοποίηση των δεδομένων για το συγκεκριμένο k με initializationMode=mode. Αυτή η λειτουργία εμφανίζει :

- Τα τελικά κέντρα των συστάδων
- Το συνολικό άθροισμα των τετραγώνων των αποστάσεων (SSE).

Τέλος να αναφέρουμε πως αυτή είναι η λειτουργία στην οποία βασίζονται κι οι μετρήσεις μας.

Μετρήσεις

2 SLAVES

Mode = K-means|| , K = 6

	Avg	Max	Min
SSE	58094212.26118421733	58094212.26118422	58094212.261184216
Time	1.9	1.9	1.9

Mode = Random , K = 6

	Avg	Max	Min
SSE	1634313757.302516933	1967528545.8892922	968475177.4350181
Time	4.5	6.8	3.2

1 SLAVE

Mode = K-means|| , K = 6

	Avg	Max	Min
SSE	58094212.26118421733	58094212.26118422	58094212.261184216
Time	1.9	1.9	1.9

Mode = Random , K = 6

	Avg	Max	Min
SSE	361554495.57863847733	968475062.213547	58094212.261184216
Time	2.86	5.7	1.4

Παρατηρούμε πως ορίζοντας το initializationMode = k-means|| ο χρόνος μειώνεται κατά πολύ σε σχέση με την περίπτωση του initializationMode = random. Κάτι που είναι αναμενόμενο αφού το συγκεκριμένο mode αφορά βελτίωση του ήδη υπάρχοντα αλγορίθμου.
Επίσης εκτός από ταχύτητα παρατηρούμε πως το mode = k-means|| είναι καλύτερο κι ως προς τα σφάλματα συγκριτικά με το mode = random.
Τα παραπάνω συμπεράσματα ισχύουν τόσο κατά την χρήση 1 όσο και κατά την χρήση 2 slaves.
Καταλαβαίνουμε λοιπόν πως για τα συγκεκριμένα δεδομένα η επιλογή του initializationMode = k-means|| κάνει τον αλγόριθμο αποδοτικότερο.