Πρακτική Εργασία στο Μάθημα

Εξόρυξη από μεγάλα δεδομένα

Ποιητής Μαρίνος – 17

Ζαΐκης Δημήτριος – 8

Παράδειγμα εκτέλεσης:

```
>> spark-submit --class “Main” cure-algorithm_2.11-0.2.jar <arg(0)> <arg(1)>
<arg(2)> <arg(3)> <arg(4)> <arg(5)> <arg(6)> <arg(7)> <arg(8)> <arg(9)>
```

Παράμετροι (προαιρετικοί)

arg(0) - Τοποθεσία φακέλου δεδομένων: URL/Path του φακέλου που περιέχει τα αρχεία data*.txt. Αν δεν οριστεί, θεωρεί πως ο φάκελος βρίσκεται στο ίδιο επίπεδο με το εκτελέσιμο .jar

arg(1) - Τοποθεσία αρχείου python: URL/Path του αρχείου main.py με την υλοποίηση του ιεραρχικού αλγόριθμου. Αν δεν οριστεί, θεωρεί πως το αρχείο βρίσκεται στο φάκελο \src\main\python\main.py σε σχέση με το εκτελέσιμο .jar

arg(2) - Πλήθος clusters: Το τελικό πλήθος από clusters που θέλουμε ο αλγόριθμος να προβλέψει

arg(3) - Μέγεθος δείγματος: Πειραματικά ένα μέγεθος που λειτουργούσε καλά και από άποψη σιλουέτας και από άποψη χρόνου είναι το 0.001

arg(4) - Πλήθος ενδιάμεσων clusters: Το πλήθος από τα clusters που δημιουργούνται από τον kmeans (στην περίπτωση του baseline αλγορίθμου) και του SHAS ή του ιεραρχικού σε python στην περίπτωση του CURE.

> Προσοχή: Αν οριστεί η παράμετρος merge να είναι false τότε τα clusters που υπολογίζονται σε αυτό το βήμα είναι τα τελικά και όχι τα ενδιάμεσα.

arg(5) - Πλήθος αντιπροσώπων: Το πλήθος των σημείων που αντιπροσωπεύουν ένα cluster

arg(6) - Παράγοντας συρρίκνωσης: Κατά πόσο θα συρρικνωθούν οι αντιπρόσωποι προς το κέντρο του cluster (πειραματικά από προηγούμενες δουλειές, φαίνεται ότι μια καλή τιμή είναι το 0.2)

arg(7) - From_python: αν είναι αληθής τότε θα τρέξει τον ιεραρχικό που έχει υλοποιηθεί σε python, αλλιώς θα τρέξει τον SHAS που αποτελεί παράλληλη υλοποίηση ιεραρχικού σε scala.

> Από πειράματα φάνηκε ότι για μέγεθος δείγματος 1% ο ιεραρχικός σε python έχει πρόβλημα με τη μνήμη, ενώ ο SHAS τρέχει χωρίς πρόβλημα.
>
> Ωστόσο για μικρότερα μεγέθη, ο ιεραρχικός της python είναι πιο γρήγορος.

arg(8) - WithRepresentatives: αν είναι αληθής υλοποιεί την ανάθεση των σημείων σε clusters βρίσκοντας τον κοντινότερο αντιπρόσωπο (η κανονική συμπεριφορά του CURE είναι αυτή). Αντίθετα, αν είναι ψευδής η ανάθεση γίνεται με βάση το κοντινότερο κέντρο του cluster που υπολογίζεται ως το κέντρο των συρρικνωμένων αντιπροσώπων. Πειραματικά φαίνεται ότι

> η σιλουέτα δεν εξαρτάται από τη μέθοδο

ο υπολογισμός με βάση τα κέντρα είναι σαφώς πιο γρήγορος συγκριτικά με τους αντιπροσώπους

arg(9) - Merge: αν είναι αληθής τότε ο CURE θα κάνει συνένωση των clusters σε ζευγάρια μέχρι να πέσει από το πλήθος των ενδιάμεσων clusters στο πλήθος των τελικών clusters. Αν είναι ψευδής, τότε γίνεται μόνο υπολογισμός αντιπροσώπων και συρρίκνωση χωρίς συνένωση. Πειραματικά φαίνεται:

Η συνένωση δεν προκαλεί διαφοροποιήσεις στη σιλουέτα

Η συνένωση είναι πολύ αργή σε σχέση με την απλή περίπτωση, ενώ δεν είναι πλήρως παραλληλοποιήσιμη καθώς απαιτεί επαναληπτικό υπολογισμό των νέων clusters και εκ νέου ενέργεια πάνω σε αυτά. Δεν βρέθηκε σαφής διατύπωση του τρόπου που γίνεται η συρρίκνωση για περαιτέρω βελτίωση.

Ακολουθία εκτέλεσης

A.

1. Φόρτωμα όλων των δεδομένων και εκτέλεση του kmeans για να υπολογισθούν τα ενδιάμεσα clusters.
2. Χρήση των ενδιάμεσων clusters είτε από τον ιεραρχικό σε python ή από τον SHAS για τον υπολογισμό των τελικών clusters.
3. Εύρεση σιλουέτας για αυτήν τη διαδικασία.

B.

1. Δειγματοληψία των δεδομένων και εκτέλεση είτε του ιεραρχικού σε python ή του SHAS για την εύρεση των ενδιάμεσων clusters.
2. Χρήση των ενδιάμεσων clusters από το επόμενο κομμάτι του CURE για την εύρεση των τελικών clusters.
3. Εύρεση σιλουέτας για αυτήν τη διαδικασία.