Παράλληλος Προγραμματισμός σε Συστήματα Μηχανικής Μάθησης

Τμήμα Ηλεκτρολόγων Μηχανικών και Τεχνολογίας Υπολογιστών

Εργαστήριο 1:
Ο αλγόριθμος των κ-μέσων "K-MEANS"

Δασούλας Ιωάννης – 1053711 – 5ο Έτος

Εισαγωγή

Η αναφορά αφορά την υλοποίηση του αλγορίθμου K-means. Για κάθε εργασία έχει δημιουργηθεί ένα ξεχωριστό αρχείο c (kmeans_a.c για την Εργασία 1, kmeans_b.c για την Εργασία 2 και ούτω καθ' εξής). Αναλυτικά εξηγήσεις και σχόλια για τα βήματα που ακολουθούνται υπάρχουν σε κάθε αρχείο. Επίσης, ενώ τα προγράμματα είχαν υλοποιηθεί, οι μετρήσεις για την αναφορά έγιναν 2 μέρες πριν την ημερομηνία παράδοσης, όπου και το μηχάνημα παρουσίαζε μικρές καθυστερήσεις στους χρόνους.

Εργασία 1

Σε κάθε επανάληψη των βημάτων του αλγορίθμου επαναπροσδιορίζονται οι κλάσεις των σημείων κι έπειτα τα κέντρα στα οποία ανήκουν. Οπότε, για N δεδομένα διάστασης Nv τα οποία πρέπει να ομαδοποιηθούν σε Nc, κέντα οι 3 βασικοί πίνακες που χρειάζονται είναι ένας πίνακας στον οποίο περιέχονται όλα τα δεδομένα διαστάσεων N*Nv, ένας πίνακας στον οποίο περιέχονται όλα τα κέντρα διαστάσεων Nc*Nv και ένας πίνακας που περιέχει την κλάση του αντίστοιχου σημείου διαστάσεων N.

Ο αλγόριθμος αποτελείται από 4 βήματα, οπότε μπορεί να εκφραστεί με 4-5 βασικές συναρτήσεις. Μία συνάρτηση δημιουργίας των δεδομένων (CreateData()), μία συνάρτηση επιλογής των πρώτων τυχαίων κέντρων (CreateCenters()), μία συνάρτηση μέτρησης των αποστάσεων των δεδομένων με τα κέντρα και αντιστοίχισης του κάθε δεδομένου στο πλησιέστερο κέντρο (Classification()) και μία συνάρτηση επαναϋπολογισμού των νέων κέντρων βάσει της ομαδοποίησης που έγινε, παίρνοντας των μέσο όρο των δεδομένων που ανήκουν στην ομάδα. Φυσικά, για την πραγματική λειτουργία του προγράμματος χρειάζεται και μία συνάρτηση ελέγχου της σύγκλισης του αλγορίθμου βάσει ενός κατωφλιού με το οποίο συγκρίνεται η μεταβολή της

αθροιστικής απόστασης των διανυσμάτων από τα αντίστοιχα κέντρα (Terminate()), αλλά και μία κύρια συνάρτηση για να τρέχει το πρόγραμμα (main()).

Εργασία 2

Επόμενο βήμα είναι η κατασκευή των συναρτήσεων. Σε σχέση με την πρώτη έκδοση, προστέθηκε η συνάρτηση CheckIfTaken() η οποία ελέγχει αν τα πρώτα κέντρα που επιλέγονται είναι ξεχωριστά μεταξύ τους και η Euclidian Distance που υπολογίζει και επιστρέφει την ευκλείδεια απόσταση ενός δεδομένου με ένα κέντρο κάθε φορά που καλείται (χρησιμοποιείται στο classification). Επίσης, προστέθηκε η PrintVec() , η οποία δεν έχει κάποιο ρόλο στην ροή του αλγόριθμου αλλά χρησιμοποιείται για την αναπαράσταση και τον έλεγχο των πινάκων. Αναλυτικά σχόλια για την λειτουργία υπάρχουν στο αρχείο.

Εργασία 3

Αφού διαπιστώθηκε η καλή λειτουργία του προγράμματος στην Εργασία 2, στην Εργασία 3 έγιναν κάποιες αλλαγές για να βελτιωθεί η απόδοσή του. Αρχικά, στην συνάρτηση main() προστέθηκε ο μετρητής επαναλήψεων iterations.

Η επόμενη αλλαγή έγινε στην συνάρτηση επιλογής των πρώτων κέντρων (CreateCenters()). Αντί να βρίσκονται τυχαία indexes, τα οποία θα έπρεπε να ελέγχονται κάθε φορά αν είναι ίδια ή όχι, τώρα χρησιμοποιήθηκε η συνάρτηση memcpy() η οποία αντιγράφει τα πρώτα Nc δεδομένα από τον πίνακα δεδομένων στον πίνακα κέντρων, διαδικασία που γίνεται πολύ γρηγορότερα και είναι έγκυρη μιας και η άσκηση γίνεται πάνω σε τυχαία δεδομένα.

Μία άλλη αλλαγή είναι η χρήση macro εντολής για τον πολλαπλασιασμό στην συνάρτηση υπολογισμού της ευκλείδειας απόστασης.

Η μεγαλύτερη αλλαγή, όμως, έγινε στην συνάρτηση υπολογισμού των νέων κέντρων (EstimateCenters()). Αντί για δύο πίνακες που χρησιμοποιούνταν για προσωρινή αποθήκευση δεδομένων, πλέον χρησιμοποιείται μόνο μία μεταβλητή (η class_members) για μέτρηση των δεδομένων που έχει η κάθε κλάση. Αρχικά, μηδενίζεται ο πίνακας κέντρων με τη χρήση της συνάρτησης memset() που έχει καλύτερη απόδοση από τον μηδενισμό όλων των στοιχείων ένα ένα με δομή επανάληψης. Έπειτα, με 2 δομές επανάληψης και ένα if ελέγχονται οι κλάσεις με τη σειρά για τον αριθμό των δεδομένων που έχουν. Στο τέλος με τη μέθοδο loop jamming, υπολογίζονται στον ίδιο ευρύτερο βρόγχο τα νέα κέντρα, διαιρώντας με την class_members που έχει υπολογισθεί μόλις.

Εργασία 4

Για την εργασία 4, χρησιμοποιήθηκε ο κώδικας της εργασίας 3 και τέθηκαν οι αρχικές τιμές που ζητούνται. Επίσης, ορίστηκε το πρόγραμμα να σταματήσει στις 16 επαναλήψεις για να μετρηθεί ο χρόνος του.

- Για τις 16 επαναλήψεις ο χρόνος ήταν συνήθως κοντά στα 10.5 λεπτά. Την ημέρα μέτρησης το αποτέλεσμα ήταν το εξής:

```
Iterations:16
Algorithm terminated succesfully!

real    11m36,021s
user    11m35,671s
sys     0m0,132s
```

Εικόνα 1: Χρόνος για εργασία 4

- Τα αποτελέσματα του profiler ήταν τα εξής:

```
Each sample counts as 0.01 seconds.
  %   cumulative   self              self     total
 time   seconds   seconds    calls   s/call   s/call  name
 99.29    686.78   686.78 161600000     0.00     0.00  EuclidianDistance
  0.97    693.49     6.71       16     0.42     0.42  EstimateCenters
  0.21    694.93     1.44       16     0.09    43.01  Classification
  0.12    695.78     0.84        1     0.84     0.84  CreateData
  0.00    695.78     0.00       16     0.00     0.00  Terminate
  0.00    695.78     0.00        1     0.00     0.00  CreateCenters
```

Εικόνα 2: Χρόνοι συναρτήσεων για Εργασία 4

Παρατηρείται ότι σχεδόν όλος ο χρόνος αφορά την συνάρτηση υπολογισμού των αποστάσεων. Αυτό σημαίνει ότι είχε αποτέλεσμα η βελτίωση της συνάρτησης υπολογισμού των κέντρων η οποία πριν τις αλλαγές αφορούσε το 14% των χρόνων ενώ τώρα αφορά μόνο το 1% περίπου.

Εργασία 5

Στα πλαίσια της εργασίας 5, επιχειρήθηκαν πολλές αλλαγές για καλύτερη απόδοση της συνάρτησης υπολογισμού της απόστασης πριν γίνει η βελτιστοποίηση -O2. Παρόλα αυτά, οι αλλαγές που δοκιμάστηκαν (pointers, loop unrolling, χρήση <math.h>) δεν είχαν ουσιαστικό αποτέλεσμα. Με την βελτιστοποίηση -O2 ο χρόνος εκτέλεσης ήταν κοντά στα 3 λεπτά.

- Την ημέρα μέτρησης ο χρόνος για τις 16 επαναλήψεις ήταν:

```
Iterations:16
Algorithm terminated succesfully!

real    3m27,049s
user    3m26,742s
sys     0m0,205s
```

Εικόνα 3: Χρόνοι για Εργασία 5

- Τα αποτελέσματα του profiler ήταν:

```
Each sample counts as 0.01 seconds.
  %   cumulative   self              self     total
 time   seconds   seconds    calls  Ts/call  Ts/call  name
 99.49    204.98   204.98                             Classification
  0.86    206.75     1.77                             EstimateCenters
  0.42    207.62     0.87                             CreateData
```

Εικόνα 4: Χρόνοι συναρτήσεων για Εργασία 5

Τα αποτελέσματα είναι αναμενόμενα. Εδώ, προφανώς στην συνάρτηση classification() περιέχει την συνάρτηση EuclidianDistance() στην οποία γίνονται οι υπολογισμοί των αποστάσεων, για αυτό και έχει τόσο μεγάλο ποσοστό.

Εργασία 6

Για την Εργασία 6, προστέθηκαν οι δοσμένες εντολές για βελτιστοποίηση του gcc. Μία αλλαγή που έγινε είναι η αφαίρεση της PrintVector() συνάρτησης που πλέον δεν χρειάζεται, διότι ο compiler προσπαθούσε να την βελτιστοποιήσει.

Με την εντολή για ενημέρωση των βελτιστοποιήσεων προέκυψαν οι παρακάτω βελτιστοποιήσεις των loops:

```
kmeans_e.c:127:3: optimized: loop vectorized using 16 byte vectors
kmeans_e.c:120:5: optimized: loop vectorized using 16 byte vectors
kmeans_e.c:103:2: optimized: loop vectorized using 16 byte vectors
kmeans_e.c:103:2: optimized: loop vectorized using 16 byte vectors
kmeans_e.c:103:2: optimized: loop vectorized using 16 byte vectors
kmeans_e.c:127:3: optimized: loop vectorized using 16 byte vectors
kmeans_e.c:120:5: optimized: loop vectorized using 16 byte vectors
```

Εικόνα 5: Βελτιστοποιήσεις βρόγχων

Οι βελτιστοποιήσεις είναι για τον βρόγχο της συνάρτησης υπολογισμού των αποστάσεων (γραμμή 103) και για τους βρόγχους υπολογισμού των νέων κέντρων (γραμμές 120 και 127).

- Την ημέρα μέτρησης ο χρόνος για τις 16 επαναλήψεις ήταν:

```
Iterations:16
Algorithm terminated succesfully!

real    0m52,634s
user    0m52,468s
sys     0m0,148s
```

Εικόνα 6: Χρόνοι για την Εργασία 6

- Τα αποτελέσματα του profiler ήταν:

```
Each sample counts as 0.01 seconds.
  %   cumulative   self              self     total
 time   seconds   seconds    calls  Ts/call  Ts/call  name
 98.88     49.95    49.95                             Classification
  1.61     50.77     0.81                             CreateData
```

Εικόνα 7: Χρόνοι συναρτήσεων για την Εργασία 6

Στους χρόνους φαίνεται ότι έχει μειωθεί λίγο το ποσοστό χρόνου της συνάρτησης Ευκλείδειας απόστασης, ενώ έχει πρακτικά μηδενιστεί ο χρόνος του επαναϋπολογισμού των κέντρων, που στις πρώτες εκδόσεις απασχολούσε για μεγάλο χρόνο τον compiler.