

ΛΟΓΙΣΜΙΚΟ & ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΣΥΣΤΗΜΑΤΩΝ

ΥΨΗΛΗΣ ΕΠΙΔΟΣΗΣ

ΑΚΑΔΗΜΑΪΚΟ ΕΤΟΣ 2017/2018

Εισαγωγή

Η ομαδοποίηση δεδομένων (clustering) είναι μια εφαρμογή που τα τελευταία χρόνια χρησιμοποιείται όλο και πιο συχνά για την εξαγωγή χρήσιμης πληροφορίας από μεγάλο πλήθος δεδομένων. Στόχος της είναι ο διαχωρισμός ενός συνόλου δεδομένων σε ένα αριθμό ομάδων (συστάδων), ώστε τα δεδομένα μιας ομάδας να είναι περισσότερο όμοια μεταξύ τους παρά με τα στοιχεία άλλων ομάδων. Αν και κάποιοι αλγόριθμοι συσταδοποίησης είναι ιδιαίτερα γνωστοί λόγω της απλότητας στην κατανόηση και την υλοποίηση τους (k-means, k-medoids, ...), υπάρχουν και άλλοι αλγόριθμοι που δίνουν εξαιρετικά αποτελέσματα αλλά έχουν μεγαλύτερο υπολογιστικό φόρτο. Ένας από αυτούς είναι και ο Principal Direction Divisive Partitioning (PDDP). Μπορείτε να βρείτε περισσότερες πληροφορίες για τον αλγόριθμο στην ιστοσελίδα <http://www-users.cs.umn.edu/~boley/publications/papers/PDDP.pdf>. Όπως θα αναλυθεί με μεγαλύτερη λεπτομέρεια στην συνέχεια, ο αλγόριθμος επεξεργάζεται πρώτα τα δεδομένα εισόδου και υπολογίζει το «κύριο συστατικό» (principal component), ένα ιδιοδιάνυσμα δηλαδή βάσει του οποίου στην συνέχεια διαχωρίζει τα στοιχεία εισόδου σε δύο υποσύνολα. Η διαδικασία είναι αναδρομική και ο ίδιος αλγόριθμος εφαρμόζεται στην συνέχεια σε καθένα από τα υποσύνολα που δημιουργήθηκαν στο προηγούμενο βήμα, έως ότου δημιουργηθεί το επιθυμητό πλήθος συστάδων.

Στα πλαίσια της εργασίας δεν θα υλοποιήσετε τον πλήρη αλγόριθμο PDDP, αλλά μόνο τον κύριο υπολογισμό ενός βήματος του αλγορίθμου, το οποίο θα αναλυθεί στην συνέχεια. Ωστόσο, στην διαδικασία θα πρέπει να ληφθούν υπόψη από την μεριά σας περιορισμοί οι οποίοι προκύπτουν από τον ολοκληρωμένο αλγόριθμο.

Το βασικό βήμα του αλγόριθμου PDDP

Ο αλγόριθμος PDDP δέχεται ως είσοδο ένα μητρώο M , διαστάσεων $n \times m$, του οποίου κάθε στήλη $d = (\delta_1 \delta_2 \dots \delta_n)$ αντιστοιχεί σε ένα στοιχείο του συνόλου δεδομένων που επιθυμούμε να κατατάξουμε σε μια συστάδα. Κάθε στοιχείο δεδομένων είναι πολυδιάστατο και αποτελείται από n διαστάσεις, αποτελείται δηλαδή από ένα σύνολο τιμών, καθεμία εκ των οποίων αντιστοιχεί στην τιμή του στοιχείου σε μια διάσταση. Το σύνολο των στοιχείων που επιθυμούμε να κατατάξουμε σε συστάδες είναι m και τυπικά $m \gg n$. Κατά συνέπεια, το μητρώο M είναι «κοντό» (έχει μικρό ύψος n) και «πλατύ» (έχει μεγάλο πλάτος m). Στα πλαίσια της εργασίας θα θεωρήσουμε πως όλα τα στοιχεία του μητρώου είναι αριθμοί κινητής υποδιαστολής διπλής ακρίβειας (double).

Το πρώτο βήμα στην επεξεργασία του μητρώου M είναι να βρεθεί ο μέσος όρος όλων των στοιχείων δεδομένων, πρέπει δηλαδή να πραγματοποιηθεί ο υπολογισμός:

$$w = \frac{1}{m} \cdot (d_1 + d_2 + \dots + d_m)$$

Αν συμβολίσουμε με $e = (111 \dots 1)^T$ ένα διάνυσμα κατάλληλου πάντα μήκους το οποίο αποτελείται μόνο από μονάδες, τότε ο παραπάνω υπολογισμός μπορεί επίσης να εκφραστεί ως:

$$w = \frac{1}{m} \cdot (d_1 + d_2 + \dots + d_m) = \frac{1}{m} \cdot M \cdot e$$

Το επόμενο βήμα είναι να υπολογιστεί το μητρώο:

$$C = (M - w \cdot e^T)^T \cdot (M - w \cdot e^T) = A^T \cdot A, \text{ όπου } A = (M - w \cdot e^T)$$

Με λίγα λόγια, πρέπει να αφαιρεθεί από κάθε στήλη του μητρώου M ο μέσος όρος των στοιχείων εισόδου (μητρώο A) και το αποτέλεσμα να πολλαπλασιαστεί με τον ανάστροφο. Παρατηρήστε πως το τελικό μητρώο C έχει διαστάσεις $m \times m$. Καθώς το m μπορεί να είναι της τάξης των εκατομμυρίων, το μητρώο C δεν υπολογίζεται ρητά, καθώς θα απαιτούσε τεράστιο χώρο στην μνήμη για την αποθήκευση του, ο οποίος τυπικά δεν είναι διαθέσιμος. Επιπλέον, ούτε το μητρώο A μπορεί να υπολογιστεί ρητά, καθώς επίσης απαιτεί αρκετό χώρο στην μνήμη. Θα μπορούσε κάποιος να σκεφτεί πως μπορεί να αντικαταστήσει τις τιμές του μητρώου M με τις νέες τιμές που προκύπτουν μετά την αφαίρεση του μέσου όρου (να αποθηκεύσει δηλαδή το μητρώο A στην θέση του μητρώου M). Θυμίζουμε ωστόσο πως ο παραπάνω υπολογισμός είναι ένα μόνο βήμα του αλγορίθμου PDDP. Μετά τον διαμοιρασμό των στοιχείων εισόδου σε δύο συστάδες, η διαδικασία πρέπει να επαναληφθεί στα πλαίσια των δεδομένων κάθε συστάδας. Επομένως, πρέπει να έχουμε τα **αρχικά** στοιχεία του μητρώου M για να προχωρήσουμε στην διαδικασία αυτή. Κατά συνέπεια, δεν είναι δυνατόν να υπολογιστούν ρητά τα μητρώα A και C και θα δούμε αμέσως πως αντιμετωπίζονται τα θέματα αυτά.

Επόμενο βήμα στον αλγόριθμο είναι να υπολογιστεί το «κύριο συστατικό» (principal component) του μητρώου C , δηλαδή το ιδιοδιάνυσμα που αντιστοιχεί στην μεγαλύτερη ιδιοτιμή του μητρώου. Υπάρχουν αρκετοί αλγόριθμοι για να πραγματοποιηθεί ο υπολογισμός. Στα πλαίσια της εργασίας θα χρησιμοποιήσετε την μέθοδο “power iteration”, η οποία είναι μια εξαιρετικά απλή (αλλά όχι τόσο αποδοτική) επαναληπτική μέθοδος. Θεωρήστε ένα τυχαίο διάνυσμα x_0 μήκους m . Για ευκολία μπορείτε να χρησιμοποιήσετε το διάνυσμα $x_0 = (111 \dots 1)$. Η μέθοδος υπολογίζει σε κάθε βήμα ένα νέο διάνυσμα x_{k+1} από το προηγούμενο διάνυσμα x_k :

$$x_{k+1} = \frac{C \cdot x_k}{\|C \cdot x_k\|} = \frac{(M - w \cdot e^T)^T \cdot (M - w \cdot e^T) \cdot x_k}{\|(M - w \cdot e^T)^T \cdot (M - w \cdot e^T) \cdot x_k\|}$$

Ο αλγόριθμος τερματίζει (έχει συγκλίνει) όταν το νέο διάνυσμα έχει μικρή διαφορά από το προηγούμενο διάνυσμα. Το τελευταίο διάνυσμα που υπολόγισε ο αλγόριθμος είναι το ζητούμενο ιδιοδιάνυσμα. Παρατηρήστε πως το γεγονός ότι πολλαπλασιάζουμε το μητρώο C με ένα διάνυσμα λύνει αυτομάτως το πρόβλημα που αφορούσε τον μη ρητό υπολογισμό των μητρώων C και A . Πρέπει απλώς να προσπελαύνουμε τα κατάλληλα στοιχεία του μητρώου M και των διανύσματος w, x_k, x_{k+1} σε κάθε επανάληψη της μεθόδου power iteration.

Συμπερασματικά, ο αλγόριθμος που θα πρέπει να υλοποιήσετε είναι ο παρακάτω:

Διάβασμα μητρώου M από αρχείο

Υπολογισμός μέσου όρου στοιχείων (διάνυσμα w)

$x_0 = (111 \dots 1)^T$

do

$$x_{k+1} = \frac{(M - w \cdot e^T)^T \cdot (M - w \cdot e^T) \cdot x_k}{\|(M - w \cdot e^T)^T \cdot (M - w \cdot e^T) \cdot x_k\|}$$

while $\|x_{k+1} - x_k\| > \varepsilon$

Αν και δεν ζητείται στα πλαίσια της εργασίας, αναφέρουμε πως ο διαχωρισμός των στοιχείων σε συστάδες γίνεται με βάση το πρόσημο του ιδιοδιανύσματος σε κάθε θέση. Για όσα στοιχεία του μητρώου M το αντίστοιχο στοιχείο στο ιδιοδιάνυσμα είναι θετικό τοποθετούνται σε μια συστάδα, ενώ για όσα είναι αρνητικό τοποθετούνται σε δεύτερη συστάδα.

Ζητούμενα της άσκησης

Στα πλαίσια της άσκησης σας ζητείται να υλοποιήσετε και να πάρετε μετρήσεις απόδοσης για 2 διαφορετικές υλοποιήσεις του παραπάνω αλγόριθμου. **Οι υλοποιήσεις θα πρέπει να είναι γενικές, υπό την έννοια ότι θα πρέπει να υποστηρίζεται οποιοδήποτε μέγεθος μητρώου (και όχι, π.χ., μόνο μητρώα με μεγέθη που είναι δυνάμεις του 2).** Επιπλέον, για λόγους εξοικονόμησης μνήμης, θα πρέπει να υπάρχει αποθηκευμένο στην μνήμη μόνο το μητρώο M . Τα στοιχεία των μητρώων και διανυσμάτων θα είναι τύπου “double”. Πιο συγκεκριμένα:

- 1) Υλοποιήστε σε CUDA τον πιο απλό τρόπο πραγματοποίησης της πράξης στην κάρτα γραφικών: Δεσμεύστε καθολική μνήμη (global memory) για το μητρώο, μεταφέρετε τα δεδομένα και κάνετε τις πράξεις. **Κατά την χρονομέτρηση μην λαμβάνετε υπ’ όψη σας τους χρόνους μεταφοράς δεδομένων από και προς την κάρτα γραφικών! Λάβετε υπ’ όψη μόνο τον χρόνο υπολογισμού.**
- 2) Προσπαθήστε να βελτιστοποιήσετε την απόδοση της υλοποίησης σας αξιοποιώντας την ιεραρχία μνήμης της CUDA. Χρησιμοποιείτε καταχωρητές, κοινή μνήμη (shared memory), streams, εσωτερική αναδιοργάνωση των δεδομένων στην κάρτα γραφικών και ότι άλλο θεωρείτε ότι μπορεί να σας δώσει καλύτερη απόδοση! **Θα πρέπει οπωσδήποτε να λάβετε υπόψη σας το γεγονός πως προσπελαύνετε στοιχεία του ίδιου μητρώου, τόσο κατά στήλες (A^T) αλλά ταυτόχρονα και κατά γραμμές (A). Το γεγονός αυτό θα πρέπει να ληφθεί υπόψη τόσο κατά την εκμετάλλευση της κοινής μνήμης, όσο και για την αξιοποίηση της προσπέλασης συνεχόμενων στοιχείων στην καθολική μνήμη (coalesced accesses in main memory).** Οποιοδήποτε υλικό βρείτε (βιβλία, GPU Gems, δημοσιεύσεις σε επιστημονικά περιοδικά ή συνέδρια, κλπ) μπορείτε φυσικά να το χρησιμοποιήσετε.

Είναι προφανές πως το σημαντικότερο κομμάτι της άσκησης είναι το τελευταίο στάδιο. Αυτό που ζητείται είναι να δείξετε ότι έχετε κατανοήσει την αρχιτεκτονική των καρτών γραφικών και πως μπορείτε να απεικονίσετε αποδοτικά έναν αλγόριθμο στην αρχιτεκτονική αυτή, λαμβάνοντας υπ’ όψη σας όλες τις παραμέτρους του αλγόριθμου και της αρχιτεκτονικής. Μας ενδιαφέρει η απόδοση της λύσης σας!

Διαδικαστικά

Η εργασία θα πρέπει να γίνει σε ομάδες των 2 ή 3 ατόμων. Η διαχείριση των ομάδων θα γίνει μέσω της ηλεκτρονικής πλατφόρμας “Open eClass” του Πανεπιστημίου Πατρών (<http://eclass.upatras.gr>). Για τον σκοπό αυτό θα πρέπει όλοι οι φοιτητές που επιθυμούν να παραδώσουν εργασία να εγγραφούν πρώτα στην παραπάνω πλατφόρμα. Στην συνέχεια, ένα άτομο από κάθε ομάδα θα αναλάβει να δηλώσει την ομάδα του μέχρι την **Τρίτη, 07/11/2017 και ώρα 23:59:59**. Το άτομο αυτό θα είναι επίσης υπεύθυνο για όλη την επικοινωνία της ομάδας μαζί μας, καθ’ όλη την διάρκεια του εξαμήνου και μέχρι την παράδοση της άσκησης. Η ομάδα θα δηλωθεί μέσω e-mail στην διεύθυνση christofid@ceid.upatras.gr. Για την ευκολότερη ταξινόμηση από την μεριά μας και την δυνατότητα αυτόματης προώθησης, το e-mail θα πρέπει να έχει τον εξής τίτλο (subject):

[HPC17-18] Δήλωση ομάδας

Το περιεχόμενο του e-mail θα πρέπει να είναι ο Α.Μ. και το ονοματεπώνυμο του φοιτητή που κάνει την δήλωση της ομάδας. Στην συνέχεια θα αναλάβουμε να φτιάξουμε μια ομάδα στο “Open eClass” και θα σας ενημερώσουμε για τον αριθμό της ομάδας σας.

Για όσες ομάδες έχουν ήδη ζητήσει μέσω e-mail πρόσβαση στο σύστημα GPU του εργαστηρίου, μαζί με το λογαριασμό τους στο σύστημα δημιουργήθηκε και η αντίστοιχη ομάδα στο eClass οπότε **δεν** χρειάζεται να ξαναστείλουν e-mail.

Σε περίπτωση που χρειαστεί επιπλέον επικοινωνία μαζί μας μέσω e-mail, αυτή θα πρέπει να γίνει είτε με τον κ. Δημήτριο Χριστοφιδέλλη (christofid@ceid.upatras.gr) είτε με τον κ. Ιωάννη Βενέτη (venetis@ceid.upatras.gr). Για την ευκολότερη ταξινόμηση από την μεριά μας και την δυνατότητα αυτόματης προώθησης, ο τίτλος (subject) κάθε e-mail θα πρέπει να ξεκινάει με [HPC17-18].

Παραδοτέα

Τα παραδοτέα για την εργασία σας είναι μια γραπτή αναφορά και ο κώδικας της άσκησης που θα αναπτύξετε. Η προθεσμία παράδοσης της εργασίας ορίζεται η Κυριακή 14/01/2018 και ώρα 23:59:59. Η εργασία θα πρέπει να παραδωθεί αποκλειστικά μέσω της ηλεκτρονικής πλατφόρμας “Open eClass” (εργασίες που θα αποσταλούν μέσω e-mail δεν θα βαθμολογηθούν). Μετά την είσοδο σας στο σύστημα θα πρέπει να μεταβείτε στο μάθημα “Λογισμικό & Προγραμματισμός Συστημάτων Υψηλής Επίδοσης” και στο μενού αριστερά να μεταβείτε στο “Εργασίες”. Κάθε ομάδα θα παραδώσει μια φορά μόνο την εργασία (όχι κάθε φοιτητής ξεχωριστά). Βεβαιωθείτε πως στο εξώφυλλο της γραπτής αναφοράς αναφέρονται τα ονόματα και οι ΑΜ όλων των συμμετεχόντων της ομάδας.

Στην αναφορά επικεντρωθείτε στην επεξήγηση της παραλληλοποίησης που κάνατε, πως αξιοποιήσατε τις δυνατότητες της CUDA, πως απεικονίσατε τον αλγόριθμο στην αρχιτεκτονική σας, στις μετρήσεις σας και στα διαγράμματα που θα προσθέσετε.

Παράρτημα Α

Για να εγκαταστήσετε την CUDA στο σύστημα σας μεταβείτε στην ιστοσελίδα <https://developer.nvidia.com/cuda-zone> και επιλέξτε τον σύνδεσμο “Downloads”. Επιλέξτε το πακέτο που θα κατεβάσετε ανάλογα με το σύστημα σας.

Σε περιβάλλον Windows, αν και δεν είναι απαραίτητο, η ύπαρξη του Visual Studio βοηθάει ιδιαίτερα στην ανάπτυξη εφαρμογών. Κατά την εγκατάσταση της CUDA εγκαθίσταται μια επέκταση για το Visual Studio ειδικά για την ανάπτυξη εφαρμογών CUDA. Αν δεν υπάρχει το Visual Studio η ανάπτυξη προγραμμάτων μπορεί να γίνει σε οποιονδήποτε κειμενογράφο (editor) και να χρησιμοποιείται απευθείας ο μεταγλωττιστής της CUDA (nvcc) από την “Γραμμή Εντολών” (“Command Prompt”). Αντίστοιχα, σε περιβάλλον Linux ο μεταγλωττιστής καλείται από το κέλυφος (shell).

Αν η ομάδα σας δεν διαθέτει σύστημα με κάρτα γραφικών της NVidia (ή αυτή δεν υποστηρίζεται από την CUDA) επικοινωνήστε μαζί μας για να σας δώσουμε πρόσβαση σε δικό μας σύστημα. Θα μπορείτε να συνδέεστε με απομακρυσμένη πρόσβαση σε αυτό (ssh).

Σημαντική παρατήρηση: Το σύστημα στο οποίο θα σας δωθεί πρόσβαση έχει δύο κάρτες γραφικών που υποστηρίζουν CUDA. Μόνο η κάρτα με Device ID 1 έχει δυνατότητα επεξεργασίας αριθμών κινητής υποδιαστολής διπλής ακρίβειας (double). Δείτε την συνάρτηση cudaSetDevice() για να θέσετε ποια κάρτα γραφικών θα χρησιμοποιηθεί για την εκτέλεση των υπολογιστικών πυρήνων της εφαρμογής σας.