Βελτιστοποίηση Απόδοσης του αλγορίθμου SG-tSNE-Π με την χρήση Μονάδων Επεξεργασίας Γραφικών

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Ιακωβίδης Ιωάννης

ΑΡΙΣΤΟΤΕΛΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΟΝΙΚΗΣ
Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Η/Υ

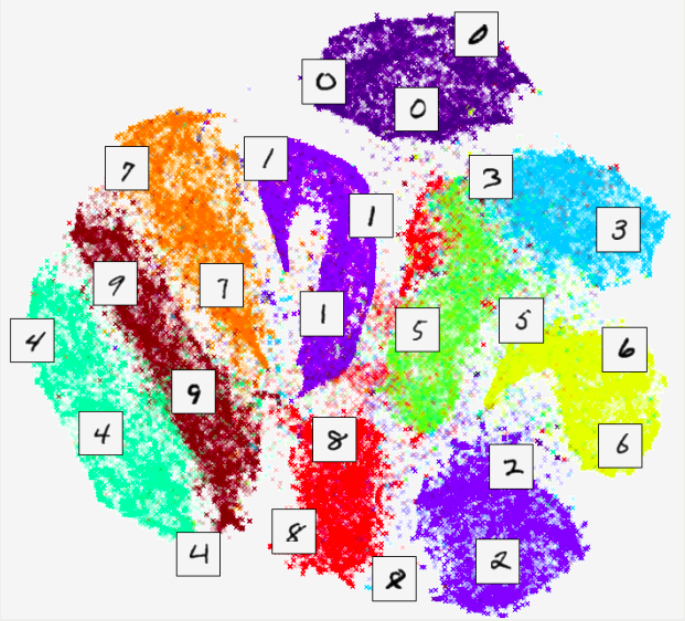
Επιβλέπων:
κ. Νικόλαος Πιτσιάνης,
κ. Δημήτριος Φλώρος

ΘΕΣΣΑΛΟΝΙΚΗ, ΙΟΥΛΙΟΣ 2021

Περιεχόμενα

t-distributed Stochastic Neighborhood Embedding (t-SNE)

Ο t-distributed Stochastic Neighborhood Embedding (t-SNE) (1) είναι ένας ευρέως διαδεδομένος αλγόριθμος μείωσης της διαστατικότητας, ο οποίος χρησιμοποιείται κυρίως για την οπτικοποίηση πολυδιάστατων δεδομένων.



- Έστω $x_i \in \mathbb{R}^d, i = 1, \ldots, n$ στοιχεία δεδομένων στον πολυδιάστατο χώρο.
- ο t-SNE δίνει έξοδο $y_i \in \mathbb{R}^s, i = 1, \ldots, n$ στοιχεία δεδομένων σε έναν ολιγοδιάστατο χώρο, όπου το s είναι συνήθως 2 ή 3 για οπτική απεικόνιση.
- Ο στόχος του t-SNE είναι να διατηρήσει την τοπική δομή των δεδομένων.
- Αυτό γίνεται με την ελαχιστοποίηση απόκλισης βαθμών ομοιότητας που ορίζονται στον πολυδιάστατο και ολιγοδιάστατο χώρο.

Αλγόριθμος t-SNE

- Από τις αποστάσεις d_{ij} των $(x_i)_{i=1}^n$ ορίζουμε

$$p_{i|j} = \frac{e^{-d_{ij}^2/2\sigma_i^2}}{\sum_{l\neq i} e^{-d_{il}^2/2\sigma_i^2}} \quad \text{and} \quad p_{ij} = \frac{p_{i|j} + p_{j|i}}{2n}$$

  ως τους βαθμούς ομοιότητας ανάμεσα στα στοιχεία του πολυδιάστατου χώρου.

- Και ως τους τους βαθμούς ομοιότητας των στοιχείων $(y_i)_{i=1}^n$ έχουμε

$$q_{ij} = \frac{(1 + \|y_i - y_j\|^2)^{-1}}{\sum_{k\neq l}(1 + \|y_k - y_l\|^2)^{-1}}$$

- Η απεικόνιση t-SNE παράγεται με την ελαχιστοποίηση της, $C(Y)$, Kullback Leibler (KL) απόκλισης

$$C(Y) = KL(P \mid Q) = \sum_{i\neq j} p_{ij} \log \frac{p_{ij}}{q_{ij}}.$$

Παράγωγος της Συνάρτησης Κόστους

Παράγωγος t-SNE

- Η ελαχιστοποίηση αυτή μπορεί να γίνει με την χρήση της μεθόδου βαθμωτής κατάβασης (gradient descent).
- Η παράγωγος της συνάρτησης κόστους είναι

$$\frac{\partial C(Y)}{\partial y_i} = 4 \sum_{j \neq i} p_{ij} q_{ij} Z(y_i - y_j) - 4 \sum_{j \neq i} q_{ij}^2 Z(y_i - y_j)$$

- Ισχύει ότι η τυχαία αρχικοποίηση των y_i σε μία μικρή περιοχή του μηδενός οδηγεί σε εκθετική σύγκληση και παράγει απεικονίσεις (ενσωματώσεις) που έχουν ελκυστικές θεωρητικές ιδιότητες.

Παράγωγος της Συνάρτησης Κόστους

Ερμηνεία της Παραγώγου

- Η παράγωγος μπορεί να ερμηνευθεί ως ένα άθροισμα δύο όρων $\frac{1}{4}\frac{\partial C(Y)}{\partial y_i} = F_{attr,i} - F_{rep,i}$ του Ελκτικού και του Απωθητικού.
- Ελκτικός όρος $F_{attr,i} = \sum_{j \neq i} p_{ij} q_{ij} Z(y_i - y_j)$
- Απωθητικός όρος $F_{rep,i} = \sum_{j \neq i} q_{ij}^2 Z(y_i - y_j)$
- Με αυτόν τον τρόπο μπορούμε να ερμηνεύουμε την μέθοδο ως ένα πρόβλημα προσομοίωσης N-σωμάτων.
- Μπορούμε να παρατηρήσουμε ότι ο αφελείς υπολογισμός της παραγώγου απαιτεί $O(n^2)$ πράξεις. Απαγορευτικό για μεγάλα σύνολα δεδομένων.

Ελκτικός Όρος

- Από την έκφραση $F_{attr,i} = \sum_{j \neq i} p_{ij} q_{ij} Z(y_i - y_j)$ αν ορίσουμε $P = \{p_{ij}\}$και $Q = \{q_{ij}\}$ $n \times n$ ο υπολογισμός του του ελκτικού όρου παρομοιάζει τον υπολογισμό ενός γινομένου πίνακα διάνυσμα με πίνακα $P \odot Q$.
- Από την έκφραση των p_{ij} έχουμε ότι τα βάρη μειώνονται εκθετικά με το τετράγωνο της απόστασης.
- Για αυτό το λόγο μετά από μία συγκεκριμένη απόσταση οι βαθμοί ομοιότητας των υπολοίπων σημείων είναι αμελητέοι.
- Έτσι χρησιμοποιώντας για κάθε σημείο μόνο τα βάρη των k κοντινότερων γειτόνων του έχουμε μία αραιή προσέγγιση του P.
- Οπότε υπολογισμός του Ελκτικού όρου παρομοιάζει τον υπολογισμό ενός γινομένου αραιού πίνακα με διάνυσμα $O(n \cdot k)$.

Απωθητικός Όρος

- Για τον απωθητικό όρο δεν μπορεί να γίνει τέτοιου είδους προσέγγιση καθώς τα q_{ij} αλλάζουν σε κάθε επανάληψη και η t-κατανομή έχει μεγαλύτερη ουρά της κανονικής.
- Σε αυτή την περίπτωση γίνεται μία Low-rank προσέγγιση του πίνακα Q μέσω ενός πλέγματος παρεμβολής στον ολιγοδιάστατο χώρο για τη προσέγγιση των όρου.
- Λόγο της δομής του υπολογισμού μπορεί να γίνει χρήση του Fast Fourier Transform (FFT) για επιπλέον επιτάχυνση. Ο χρόνος για τον υπολογισμό με τη χρήση p σημείων παρεμβολής ανά διάσταση είναι $O(p^d \cdot log(p) + n \cdot p)$.
- Η ακρίβεια της μεθόδου εξαρτάται από το μέγεθος του πλέγματος. Αν θέλουμε να έχουμε σφάλμα μικρότερο από μία σταθερή τιμή καθώς η περιοχή ενσωμάτωσης μεγαλώνει πρέπει να μεγαλώσουμε το πλέγμα μας.

tSNE-CUDA

- Η πλέον αποδοτικότερη υλοποίηση GPU σε CUDA της μέθοδο t-SNE σε δύο διαστάσεις (2).
- Αυτή η μέθοδος χρησιμοποιεί τις παραπάνω μεθόδους προσέγγισης για την εκτέλεση της gradient descent.
- Για τον υπολογισμό του ελκτικού όρου χρησιμοποιείται η δομή δεδομένον COO για την αποθήκευση του αραιού πίνακα.
- Ο υπολογισμός του απωθητικού όρου γίνεται με την χρήση του cuFFT. Λόγο της διάρκειας χρόνου σχεδίασης του πλάνου εκτέλεσης του cuFFT. Η t-SNE-CUDA διαθέτει έναν σταθερό αριθμό σημείων για την προσέγγιση του απωθητικού όρου σε κάθε επανάληψη.

SG-tSNE-Π

- Αυτή η υλοποίηση παρουσιάζει επιτάχυνση και ως προς τον ελκτικό και απωθητικό όρο από τις προϋπάρχουσες CPU υλοποιήσεις (3).
- Επίσης η μέθοδος μειώνει τις απαιτήσεις χώρου της μεθόδου κάνοντας αποδοτική την απεικόνιση σε μεγαλύτερο αριθμό διαστάσεων.
- Επίσης η μέθοδος αποτελεί γενικεύει τη μέθοδο t-SNE, στην απεικόνιση αραιών γράφων.

SG-tSNE-Π

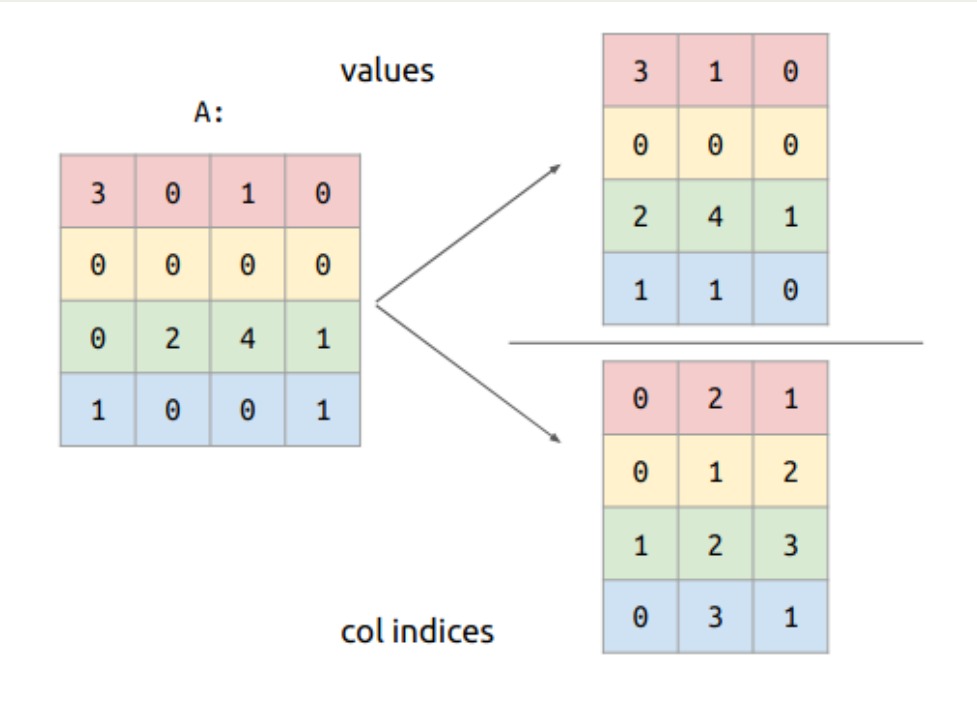
- Για τον υπολογισμό του ελκτικού όρου χρησιμοποιείται η δομή δεδομένων CSB (Compressed Sparse Blocks (4)). Μία δομή που αποθηκεύει υποπίνακες του πίνακα και οδηγεί σε καλή χρήση της κρυφής μνήμης και εξαιρετική απόδοση.
- Συνήθως η χρήση δομών αποθήκευσης αραιών πινάκων που αποθηκεύουν υποπίνακες του πίνακα οδηγούν σε μεγάλο αριθμό περιττού γεμίσματος μηδενικών.
- Για την μείωση αυτού του γεμίσματος γίνονται μεταθέσεις του πίνακα με μία τεχνική διαμερισμού γράφων που ονομάζεται Nested Disection (5).
- Το χρονικό κόστος των μεταθέσεων εξαργυρώνεται από τον αριθμό των εκτελέσεων της μεθόδου.
- Για τον απωθητικό υπολογισμό χρησιμοποιείται μία τεχνική παρεμβολής η οποία επιταχύνεται με την χρήση του FFTW.
- Για επιπλέον επιτάχυνση πριν τον υπολογισμό εκτελείται μετάθεση των σημείων ώστε να έχουμε τοπικότητα κατά την διάρκεια του υπολογισμού και έτσι καλύτερη χρήση της κρυφής μνήμης.
- Επίσης η μέθοδος αυτή επίσης οδηγεί σε καλύτερη ακρίβεια για το ίδιο μέγεθος πλέγματος και μειώνει σημαντικά τις απαιτήσεις χώρου. Με αυτόν τον τρόπο γίνεται εφικτή και αποδοτική η απεικόνιση σε 3 και 4 διαστάσεις.

SG-tSNE-CUDA

- Η υλοποίηση GPU σε CUDA που αναπτύχθηκε στα πλαίσια της διπλωματικής.
- Η μέθοδος μας βασίζεται στην υλοποίηση SG-tSNE-Π.
- Για τον υπολογισμό του ελκτικού όρου χρησιμοποιείται η υβριδική δομή δεδομένον HYB (ELL-COO) για την αποθήκευση του αραιού πίνακα.
- Η υλοποίηση του απωθητικού υπολογισμού βασίζεται στην υλοποίηση SG-tSNE-Π αλλά γίνεται σε CUDA με την χρήση της βιβλιοθήκης CUFFT.
- Η SG-tSNE-CUDA διαθέτει έναν σταθερό αριθμό σημείων για την προσέγγιση του απωθητικού όρου σε κάθε επανάληψη όμοια με την t-SNE-CUDA.

HYB

- Η δομή HYB αποθηκεύει τα στοιχεία του πίνακα σε δύο δομές μία δομή COO και μία ELL.
- Με την δομή ELL ένας $M \times N$ αραιός πίνακας με το πολύ K μη-μηδενικά ανά γραμμή στοιχεία αποθηκεύεται σε έναν πυκνό πίνακα $M \times K$ που αποθηκεύει τα δεδομένα του πίνακα και έναν $M \times K$ πυκνό πίνακα δεικτών στήλης για αυτά.



HYB

- Η υλοποίηση του γινομένου πίνακα διανύσματος με την χρήση της ELL δομής είναι πολύ αποδοτική. Καθώς η προσπέλαση των στοιχείων του πίνακα από τα νήματα CUDA είναι ευθυγραμμισμένη.
- Από πειραματικά αποτελέσματα ισχύει ότι μία πλήρη κατειλημμένη ELL δομή είναι περίπου τρεις φορές πιο γρήγορη από την αντίστοιχη COO εκτός όταν ο αριθμός των γραμμών είναι αρκετά μικρός (6).
- Η απόδοσή της ELL μειώνετε γρήγορα καθώς ο αριθμός των μη-μηδενικών του πίνακα κυμαίνεται. Αφού χρειάζεται όλο και περισσότερο γέμισμα με μηδενικά για να κατασκευαστεί.
- Για αυτό το λόγο δημιουργείται η δομή HYB που αποθηκεύει έναν προκαθορισμένο αριθμό στοιχείων από κάθε γραμμή σε ELL και έπειτα τα υπόλοιπα σε COO.
- Στην περίπτωσή μας αυτός ο προκαθορισμένος αριθμός καθορίζεται ως ο βέλτιστος λαμβάνοντας υπόψη την διαφορά ταχύτητας εκτέλεσης υπολογισμού του γινομένου πίνακα-διάνυσμα των δομών ELL και COO μέσω της βιβλιοθήκης Cusp (7).
- Καθώς αυτός επιλέγεται με γνώση της σχετικής ταχύτητας του ELL ως προς το COO. Δεν υπάρχει κίνδυνος και η τελική υλοποίηση είναι καθολικά αποδοτικότερη από την αντίστοιχή με χρήση COO (t-SNE-CUDA).

SG-tSNE-HYB

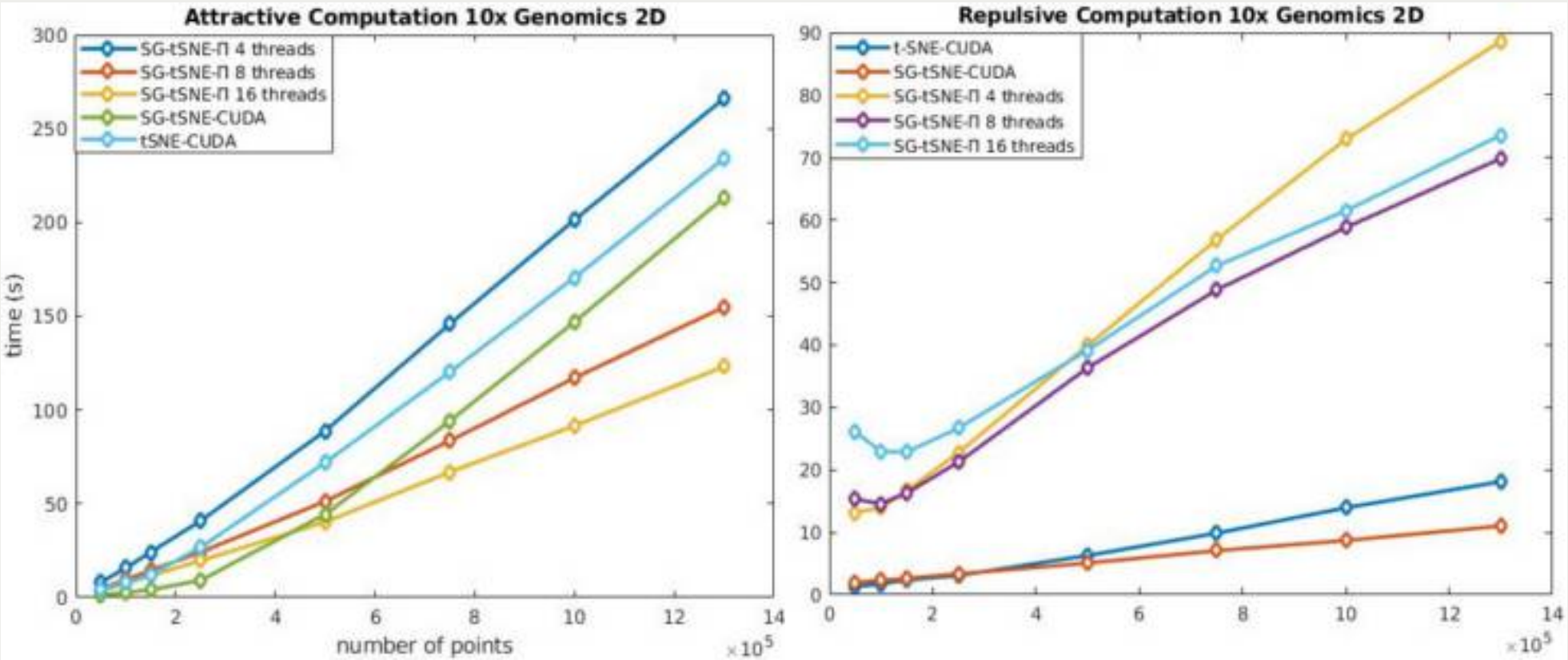
- Όπως θα δούμε στην επόμενη ενότητα ο απαιτούμενος χρόνος για τον υπολογισμό του ελκτικού όρου είναι πολύ μεγαλύτερος από αυτόν του απωθητικού και κυριαρχεί τον συνολικό.
- Επίσης όταν έχουμε στην διάθεση μας ένα αρκετά αποδοτικό πολυπύρηνο σύστημα, με την χρήση μεγάλου αριθμού νημάτων, ο χρόνος υπολογισμού του ελκτικού όρου της SG-tSNE-Π είναι μικρότερος απο αυτόν της SG-tSNE-CUDA.
- Για αυτούς τους λόγους κατασκευάσαμε μία υβριδική υλοποίηση η οποία υπολογίζει τον ελκτικό όρο στην CPU με την μέθοδο της SG-tSNE-Π ενώ τον απωθητικό στην GPU.
- Με αυτόν τον τρόπο μπορούμε να κρύψουμε το χρονικό χρόνο του απωθητικού καθώς είναι ανεξάρτητοι και εκτελούνται παράλληλα.

Πειράματα σε μεγάλα σύνολα δεδομένων

Στις ακόλουθες παραγράφους θα παρουσιάσουμε πειραματικά αποτελέσματα για τις απεικονίσεις του συνόλου δεδομένων κυττάρων ποντικιών της 10x Genomics. Συγκεκριμένα θα παράγουμε πειράματα για τυχαία υποσύνολα του μεγέθους 100k, 250k, 500k, 750k, 1.3m στοιχείων. Με αυτόν τον τρόπο θα δούμε πως ο χρόνος των διαφόρων υλοποιήσεων αλλάζει με τον αριθμό των σημείων.

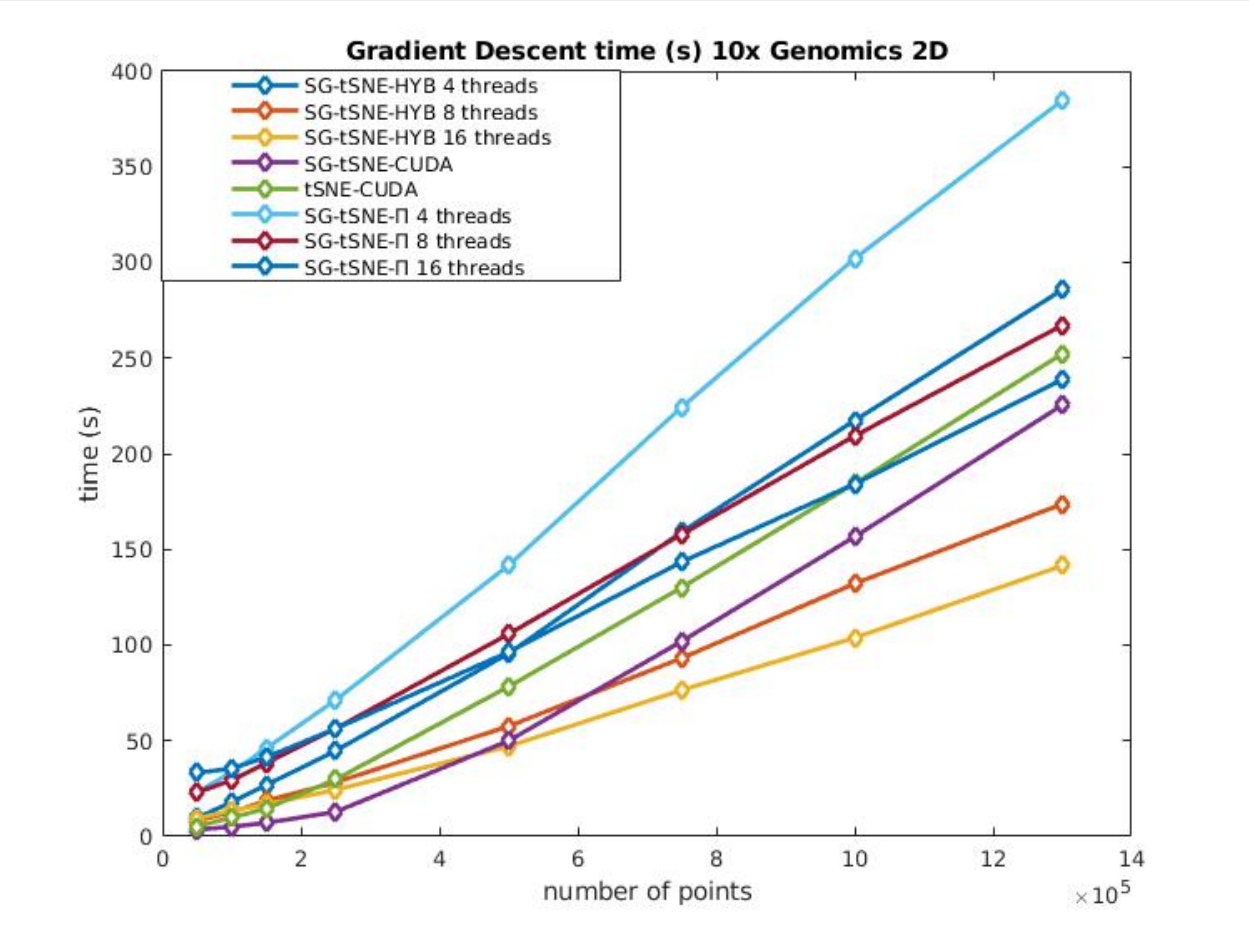
Πειράματα σε μεγάλα σύνολα δεδομένων

Όπως βλέπουμε για μεγάλο αριθμό νημάτων και σημείων ο ελκτικός χρόνος της SG-tSNE-Π είναι μικρότερος από αυτόν των GPU υλοποιήσεων ενώ ο απωθητικός χρόνος πολύ μεγαλύτερος.



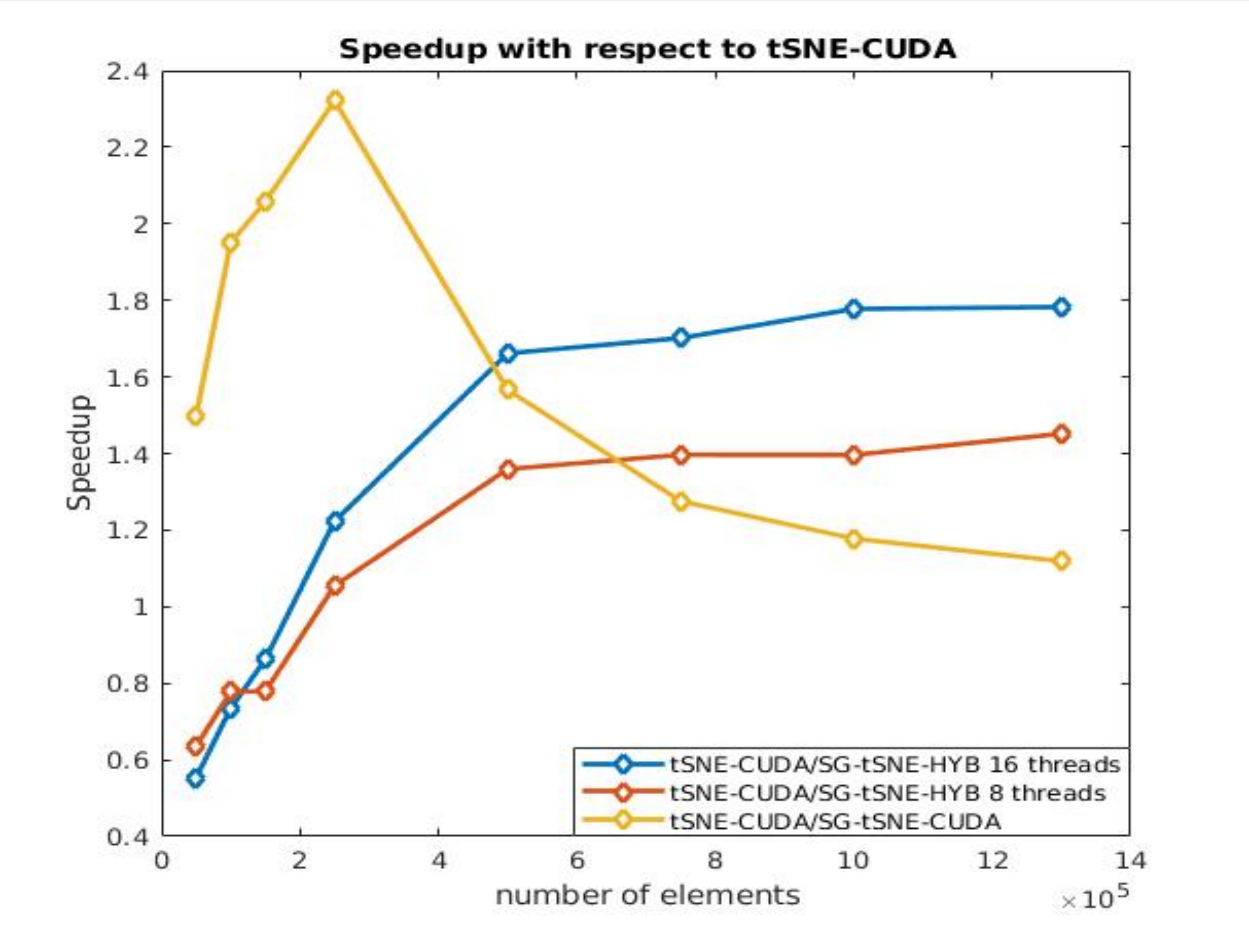
Πειράματα σε μεγάλα σύνολα δεδομένων

Τα παραπάνω έχουν ως αποτέλεσμα η υβριδική μας υλοποίηση να είναι η αποδοτικότερη.



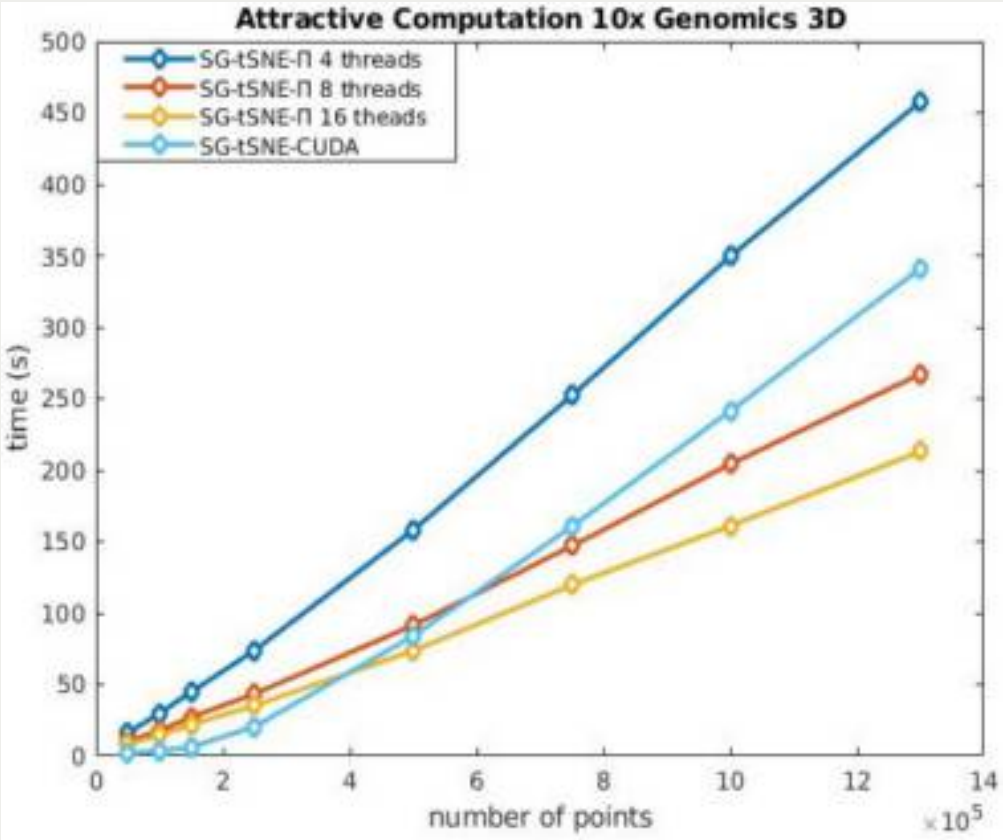
Πειράματα σε μεγάλα σύνολα δεδομένων

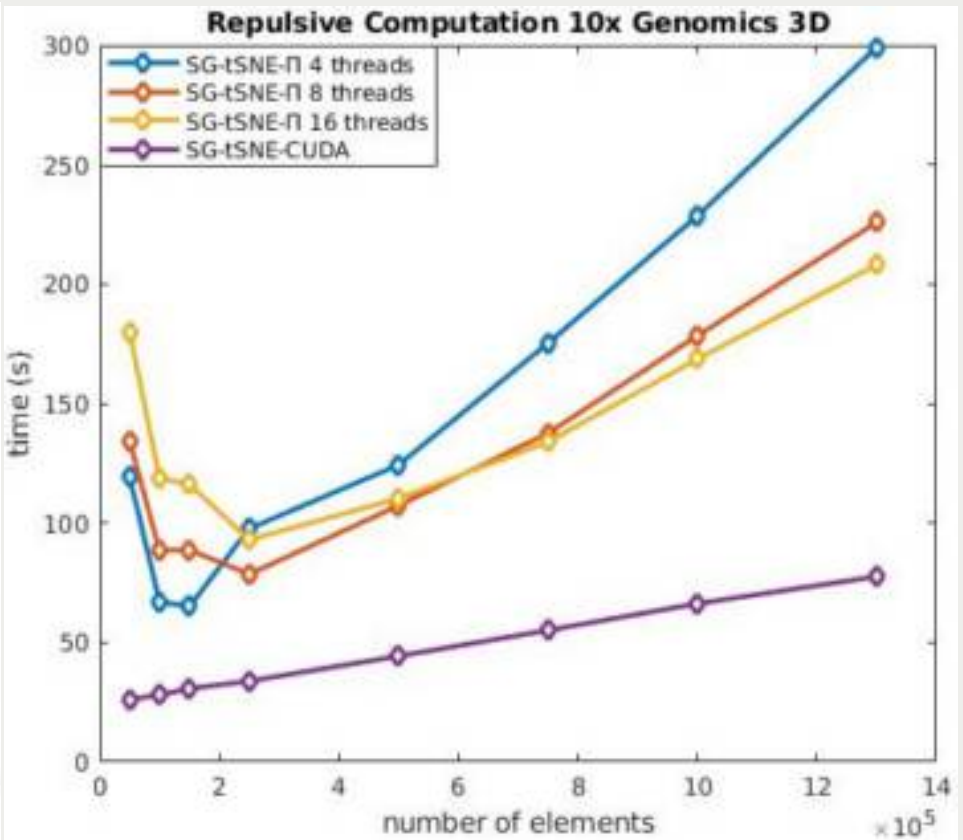
Παρακάτω βλέπουμε την επιτάχυνση των υλοποιήσεων μας ως προς την t-SNE-CUDA.



Πειράματα σε μεγάλα σύνολα δεδομένων

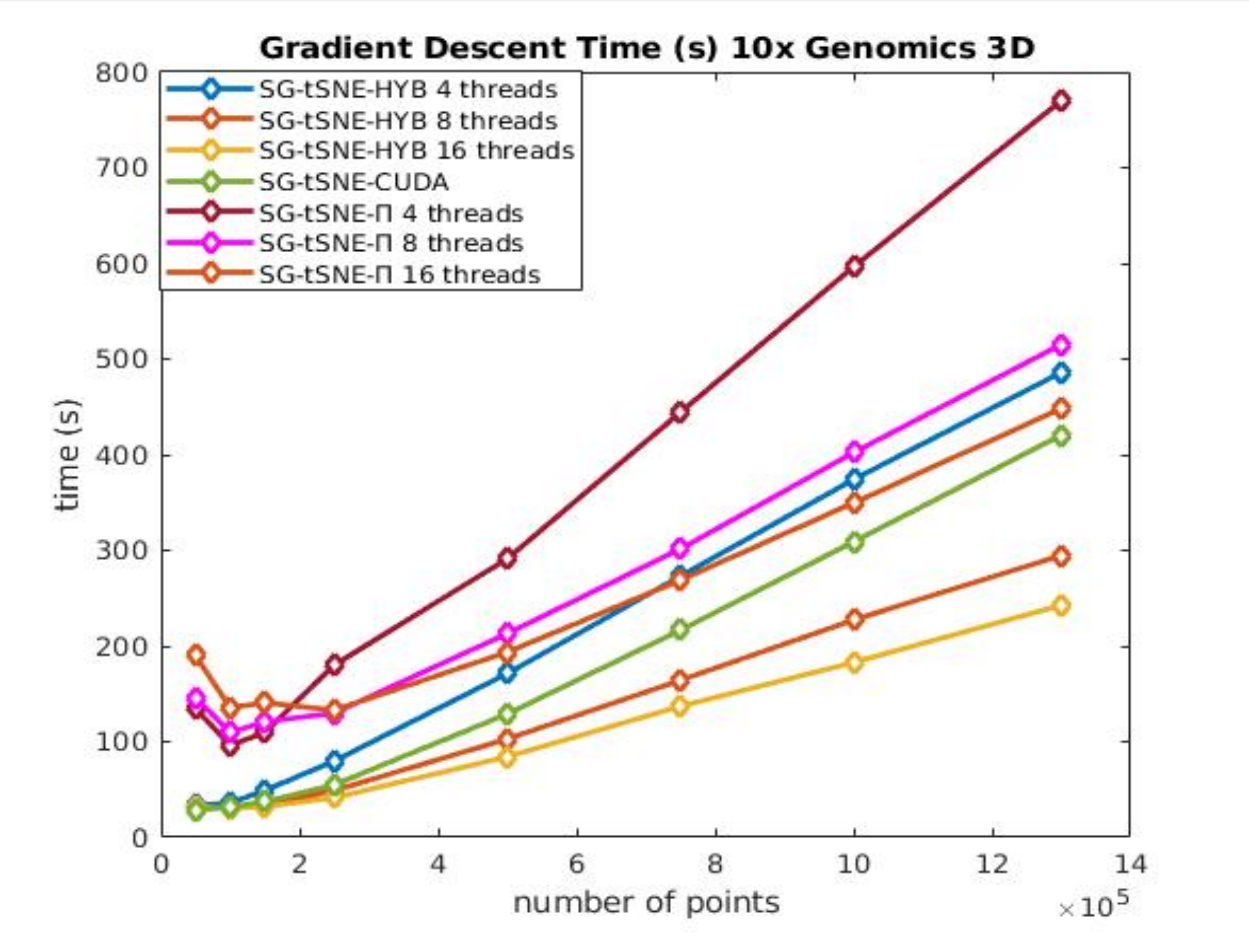
Οι παραπάνω παρατηρήσεις είναι ακόμα πιο εμφανείς για απεικόνιση σε τρεις διαστάσεις. Σε αυτήν την περίπτωση ο χρόνος του απωθητικού υπολογισμού είναι μεγαλύτερος οπότε η χρήση της SG-tSNE-HYB είναι ακόμα προτιμότερη.





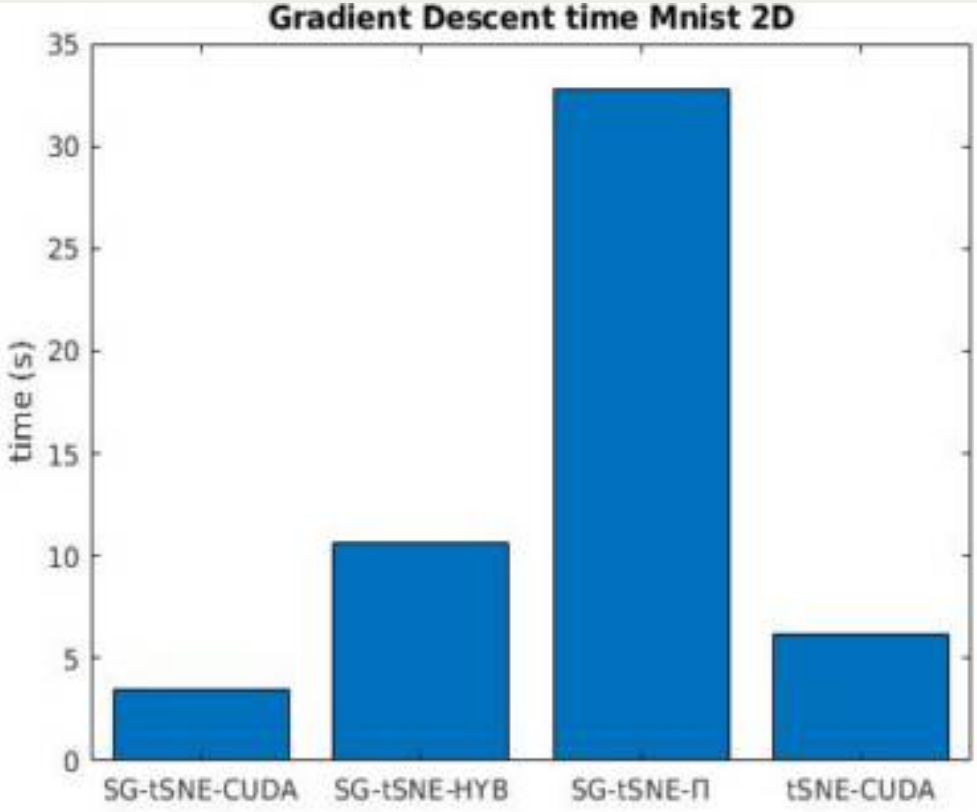
Πειράματα σε μεγάλα σύνολα δεδομένων

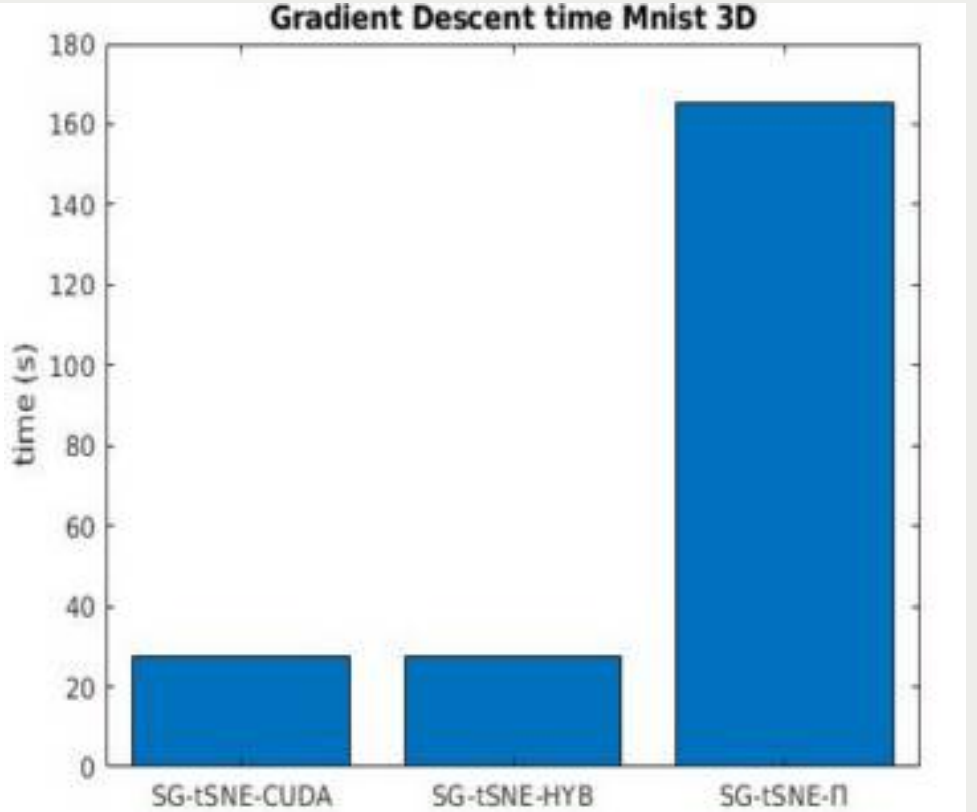
Σε αυτήν την περίπτωση ο χρόνος του απωθητικού υπολογισμού είναι μεγαλύτερος οπότε η χρήση της SG-tSNE-HYB είναι ακόμα προτιμότερη.



Πειράματα σε μικρότερα σύνολα δεδομένων

Παρακάτω βλέπουμε τους χρόνους για απεικόνιση του συνόλου Mnist από της υλοποιήσεις. Βλέπουμε ότι η SG-tSNE-CUDA είναι η ταχύτερη για απεικόνηση σε δύο διαστάσεις ενώ η SG-tSNE-HYB σε τρεις.





Συμπεράσματα

- Οι υλοποιήσεις που αναπτύξαμε κατά την διάρκεια της διπλωματικής επιταχύνουν τις υπάρχουσες. Με αποδοτική απεικόνιση και σε τρεις διαστάσεις.
- Η επιτάχυνση και συνολικός χρόνος του υπολογισμού εξαρτάται από τη δομή του αραιού πίνακα.
- Η υβριδική μας υλοποίηση που υπολογίζει τον ελκτικό χρόνο στην CPU και απωθητικό χρόνο στην GPU είναι η αποδοτικότερη. Για μεγάλο αριθμό δεδομένων και νημάτων.
- Χρήση της υβριδικής υλοποίησης συνίσταται όταν έχουμε στην διάθεση μας ένα πολύ αποδοτικό σύστημα και θέλουμε να απεικονίσουμε ένα μεγάλο σύνολο δεδομένων.
- Σε άλλες περιπτώσεις συνιστάται η χρήση της SG-tSNE-CUDA η οποία είναι αποδοτικότερη από την t-SNE-CUDA λόγω της διαφορετικής δομής υπολογισμού του ελκτικού όρου.

Μελλοντική Έρευνα

- Αφού το χρονικό κόστος του απωθητικού υπολογισμού μπορεί να κρυφτεί με την χρήση μίας υβριδικής CPU-GPU υλοποίησης.
- Ισχύει ότι το χρονικό κόστος της μεθόδου αποτελείται από το χρονικό κόστος του ελκτικού όρου.
- Χρήση καλύτερου αλγόριθμου/δομής για τον υπολογισμό του ελκτικού όρου.
- Βέβαια θεωρείται ότι ο υπολογισμός του SpMV με την χρήση GPU και την δομή HYB που χρησιμοποιούμε αποτελεί συχνά την αποδοτικότερη επιλογή, δηλαδή η απόδοση γενικεύει σε μεγάλο αριθμό μοτίβων αραιότητας (8).

Ευχαριστώ για την Παρακολούθηση.

Παρακαλώ Ερωτήσεις.

Βιβλιογραφία I

L. van der Maaten and G. Hinton, ``Visualizing data using t-SNE,'' *Journal of Machine Learning Research*, vol. 9, pp. 2579{2605, 2008. (Online). Available: http://www.jmlr.org/papers/v9/vandermaaten08a.html.

D. M. Chan, R. Rao, F. Huang, and J. F. Canny, *T-sne-cuda: Gpu-accelerated t-sne and its applications to modern data*, 2018. arXiv: 1807.11824 [cs.LG].

N. Pitsianis, A.-S. Iliopoulos, D. Floros, and X. Sun, ``Spaceland embedding of sparse stochastic graphs,'' in *2019 IEEE High Performance Extreme Computing Conference (HPEC)*, 2019, pp. 1{8. DOI: 10.1109/HPEC.2019.8916505.

A. Buluç, J. T. Fineman, M. Frigo, J. R. Gilbert, and C. E. Leiserson, ``Parallel sparse matrix-vector and matrix-transpose-vector multiplication using compressed sparse blocks,'' in *Proceedings of the Twenty-First Annual Symposium on Parallelism in Algorithms and Architectures*, ser. SPAA '09, Calgary, AB, Canada: Association for Computing Machinery, 2009, 233{244, ISBN: 9781605586069. DOI: 10.1145/1583991.1584053. (Online). Available: https://doi.org/10.1145/1583991.1584053.

Βιβλιογραφία II


G. Karypis and V. Kumar, ``Analysis of multilevel graph partitioning,'' in *Proceedings of the 1995 ACM/IEEE Conference on Supercomputing*, ser. Supercomputing '95, San Diego, California, USA: Association for Computing Machinery, 1995, 29{es, ISBN: 0897918169. DOI: 10.1145/224170.224229. (Online). Available: https://doi.org/10.1145/224170.224229.


N. Bell and M. Garland, ``Implementing sparse matrix-vector multiplication on throughput-oriented processors,'' in *Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis*, ser. SC '09, Portland, Oregon: Association for Computing Machinery, 2009, ISBN: 9781605587448. DOI: 10.1145/1654059.1654078. (Online). Available: https://doi.org/10.1145/1654059.1654078.


S. Dalton, N. Bell, L. Olson, and M. Garland, *Cusp: Generic parallel algorithms for sparse matrix and graph computations*, Version 0.5.0, 2014. (Online). Available: http://cusplibrary.github.io/.


Y. M. Tsai, T. Cojean, and H. Anzt, ``Sparse linear algebra on amd and nvidia gpus -- the race is on,'' in *High Performance Computing*, P. Sadayappan, B. L. Chamberlain, G. Juckeland, and H. Ltaief, Eds., Cham: Springer International Publishing, 2020, pp. 309{327, ISBN: 978-3-030-50743-5.