ΔΙΙΔΡΥΜΑΤΙΚΟ ΔΙΑΤΜΗΜΑΤΙΚΟ
ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
ΕΦΑΡΜΟΣΜΕΝΗ ΕΠΙΧΕΙΡΗΣΙΑΚΗ ΕΡΕΥΝΑ &
ΑΝΑΛΥΣΗ»

ΣΤΡΑΤΙΩΤΙΚΗ ΣΧΟΛΗ
ΕΥΕΛΠΙΔΩΝ
Τμήμα Στρατιωτικών
Επιστημών

ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ
Σχολή Μηχανικών
Παραγωγής & Διοίκησης

ΜΕΤΑΠΤΥΧΙΑΚΗ ΔΙΑΤΡΙΒΗ

ΑΛΓΟΡΙΘΜΟΙ ΓΙΑ ΠΡΟΒΛΗΜΑΤΑ ΜΑΖΙΚΩΝ ΔΕΔΟΜΕΝΩΝ: ΔΕΙΓΜΑΤΟΛΗΨΙΑ STREAMING SKETCHING

Διατριβή που υπεβλήθη για την μερική ικανοποίηση των απαιτήσεων για την απόκτηση Μεταπτυχιακού Διπλώματος

του **Νικητόπουλου Αλέξανδρου (ΑΜ: 2018018030)**

ΜΑΙΟΣ 2021

Η Μεταπτυχιακή Διατριβή του Νικητόπουλου Αλέξανδρου εγκρίνεται:

ΤΡΙΜΕΛΗΣ ΕΞΕΤΑΣΤΙΚΗ ΕΠΙΤΡΟΠΗ

Καθηγητής Δάρας Νικόλαος (Επιβλέπων) ,.

Καθηγητής Ματσατσίνης Νικόλαος,Nikolaos Matsatsinis Digitally signed by Nikolaos Matsatsinis Date: 2021.05.25 08:26:50 +03'00'.....

Καθηγητής Παπαδάκης Νικόλαος,Νικόλαος Παπαδάκης.....

ΣΕΛΙΔΑ ΣΚΟΠΙΜΑ ΚΕΝΗ

Έτος 2021

Περίληψη

Τα τελευταία χρόνια, η πρόοδος στην τεχνολογία υλικού διευκόλυνε τη δυνατότητα συνεχούς συλλογής δεδομένων. Οι απλές συναλλαγές της καθημερινής ζωής, όπως η χρήση πιστωτικής κάρτας, τηλεφώνου ή η περιήγηση στον Ιστό, οδηγούν σε αυτοματοποιημένη αποθήκευση δεδομένων. Ομοίως, οι εξελίξεις στην τεχνολογία των πληροφοριών έχουν οδηγήσει στη δημιουργία μεγάλων ροών δεδομένων σε δίκτυα IP. Σε πολλές περιπτώσεις, αυτοί οι μεγάλοι όγκοι δεδομένων μπορούν να εξαχθούν για ενδιαφέρουσες πληροφορίες σε μια μεγάλη ποικιλία εφαρμογών. Ο τεράστιος όγκος των υποκείμενων δεδομένων οδηγεί σε μια σειρά υπολογιστικών προκλήσεων καθώς και προκλήσεων που αφορούν την εξαγωγή των δεδομένων αυτών. Η δειγματοληψία μαζικών δεδομένων ασχολείται με προβλήματα μαζικών δεδομένων (massive data) όπου τα δεδομένα εισόδου (ένας γράφος, μια μήτρα ή κάποιο άλλο αντικείμενο) είναι πολύ μεγάλα για να αποθηκευτούν στη μνήμη τυχαίας προσπέλασης. Ένα μοντέλο για τέτοια προβλήματα είναι το μοντέλο ροής (streaming model), όπου τα δεδομένα είναι ορατά μόνο μία φορά. Στο μοντέλο ροής, η φυσική τεχνική αντιμετώπισης των μαζικών δεδομένων είναι η δειγματοληψία. Στην παρούσα εργασία μελετώνται βιβλιογραφικά μια σειρά αλγορίθμων που σχετίζονται με προβλήματα διαχείρισης μαζικών δεδομένων σε ροές, όπως η δειγματοληψία, η εξαγωγή, η ταξινόμηση και η επεξεργασία, εστιάζοντας στη δειγματοληψία Streaming Sketching

Λέξεις Κλειδιά: Μαζικά Δεδομένα, Ροές Δεδομένων, Αλγόριθμοι, Μηχανική Μάθηση, Δειγματοληψία Streaming Sketching

Abstract

In recent years, advances in hardware technology have facilitated continuous data collection. Simple everyday transactions, such as using a credit card, telephone or web browsing, lead to automated data storage. Similarly, advances in information technology have led to the creation of large data streams in IP networks. In many cases, these large volumes of data can be exported for interesting information in a wide variety of applications. The huge volume of underlying data leads to a number of computational challenges as well as challenges related to the extraction of this data. Mass data sampling deals with massive data problems where the input data (a graph, a matrix or some other object) is too large to be stored in random access memory. One model for such problems is the streaming model, where the data is only visible once. In the flow model, the natural technique for dealing with bulk data is sampling. In the present work, a number of algorithms related to problems of mass data management in flows, such as sampling, export, classification and processing, are studied in the literature, focusing on Streaming Sketching.

Keywords: Mass Data, Data Flows, Algorithms, Machine Learning, Streaming Sketching sampling

Περιεχόμενα

1. Εισαγωγή

Τα τελευταία χρόνια, η πρόοδος στην τεχνολογία υλικού διευκόλυνε τη δυνατότητα συνεχούς συλλογής δεδομένων. Οι απλές συναλλαγές της καθημερινής ζωής, όπως η χρήση πιστωτικής κάρτας, τηλεφώνου ή η περιήγηση στον Ιστό, οδηγούν σε αυτοματοποιημένη αποθήκευση δεδομένων. Ομοίως, οι εξελίξεις στην τεχνολογία των πληροφοριών έχουν οδηγήσει στη δημιουργία μεγάλων ροών δεδομένων σε δίκτυα IP. Σε πολλές περιπτώσεις, αυτοί οι μεγάλοι όγκοι δεδομένων μπορούν να εξαχθούν για ενδιαφέρουσες πληροφορίες σε μια μεγάλη ποικιλία εφαρμογών. Ο τεράστιος όγκος των υποκείμενων δεδομένων οδηγεί σε μια σειρά υπολογιστικών προκλήσεων καθώς και προκλήσεων που αφορούν την εξαγωγή των δεδομένων αυτών:

• Με την αύξηση του όγκου των δεδομένων, δεν καθίσταται πλέον δυνατή η αποτελεσματική επεξεργασία τους χρησιμοποιώντας πολλαπλά περάσματα. Αντίθετα, μπορεί κανείς να επεξεργαστεί ένα στοιχείο δεδομένων το πολύ μία φορά, κάτι που οδηγεί σε περιορισμούς στην εφαρμογή των υποκείμενων αλγορίθμων. Επομένως, οι αλγόριθμοι εξαγωγής ροής πρέπει συνήθως να σχεδιαστούν έτσι ώστε να λειτουργούν με μονό πέρασμα δεδομένων.

• Στις περισσότερες περιπτώσεις, υπάρχει μια εγγενής χρονική συνιστώσα στη διαδικασία εξαγωγής ροής επειδή τα δεδομένα ενδέχεται να εξελίσσονται με την πάροδο του χρόνου. Αυτή η συμπεριφορά των ροών δεδομένων αναφέρεται ως χρονική τοποθεσία. Επομένως, μια απλή προσαρμογή των αλγορίθμων εξαγωγής μονού περάσματος μπορεί να μην αποτελεί αποτελεσματική λύση. Οι αλγόριθμοι εξαγωγής ροής πρέπει να σχεδιαστούν προσεκτικά με σαφή εστίαση στην εξέλιξη των υποκείμενων δεδομένων.

Ένα άλλο σημαντικό χαρακτηριστικό των ροών δεδομένων είναι ότι συχνά εξάγονται με κατανεμημένο τρόπο. Επιπλέον, οι μεμονωμένοι επεξεργαστές μπορεί να έχουν περιορισμένη επεξεργασία και μνήμη. Παραδείγματα τέτοιων περιπτώσεων περιλαμβάνουν δίκτυα αισθητήρων, στα οποία μπορεί να είναι επιθυμητή η εκτέλεση ροής δεδομένων εντός δικτύου με περιορισμένη επεξεργασία και μνήμη (Cormode et al., 2005; Kollios et al., 2005).

Εν συνεχεία ακολουθεί επισκόπηση των διαφορετικών αλγορίθμων εξαγωγής ροής καθώς και των προκλήσεων που σχετίζονται με κάθε πρόβλημα.

2. Μηχανική Μάθηση και Εξόρυξη Δεδομένων

Ο πιο κοινά αποδεκτός ορισμός της «εξόρυξης δεδομένων» είναι η ανακάλυψη «μοντέλων» για δεδομένα. Ένα «μοντέλο», ωστόσο, μπορεί να είναι ένα από πολλά πράγματα. Παρακάτω αναφέρονται οι πιο σημαντικές οδηγίες ως προς την μοντελοποίηση. Οι στατιστικολόγοι ήταν οι πρώτοι που χρησιμοποίησαν τον όρο «εξόρυξη δεδομένων». Αρχικά, η «εξόρυξη δεδομένων» ή η «εκσκαφή δεδομένων» ήταν ένας υποτιμητικός όρος που αναφερόταν σε προσπάθειες εξαγωγής πληροφοριών που δεν υποστηρίζονταν από τα δεδομένα. Στη συνέχεια παρουσιάζεται το είδος των σφαλμάτων που μπορεί κανείς να κάνει προσπαθώντας να εξαγάγει ό,τι δεν υπάρχει πραγματικά στα δεδομένα. Σήμερα, η «εξόρυξη δεδομένων» έχει θετική σημασία. Τώρα, οι στατιστικολόγοι βλέπουν την εξόρυξη δεδομένων ως την κατασκευή ενός *στατιστικού μοντέλου*, δηλαδή, μια υποκείμενη κατανομή από την οποία αντλούνται τα ορατά δεδομένα (Murray & Scime, 2015).

Υπάρχουν ορισμένοι που θεωρούν την εξόρυξη δεδομένων συνώνυμη με τη μηχανική μάθηση. Δεν υπάρχει αμφιβολία ότι κάποια εξόρυξη δεδομένων χρησιμοποιεί κατάλληλα αλγόριθμους από τη μηχανική μάθηση. Οι επαγγελματίες της μηχανικής μάθησης χρησιμοποιούν τα δεδομένα ως εκπαιδευτικό σύνολο, για να εκπαιδεύσουν έναν αλγόριθμο (έναν από τους πολλούς τύπους) που χρησιμοποιούνται από τους επαγγελματίες της μηχανικής μάθησης, όπως τα δίκτυα Bayes, μηχανές διανυσμάτων υποστήριξης, δέντρα αποφάσεων, κρυφά μοντέλα Markov και πολλά άλλα (Wu & Ding, 2013).

Υπάρχουν καταστάσεις όπου η χρήση δεδομένων με αυτόν τον τρόπο έχει νόημα. Η τυπική περίπτωση όπου η μηχανική μάθηση είναι μια καλή προσέγγιση είναι όταν έχει κανείς λίγη ιδέα για το τι ψάχνει στα δεδομένα. Για παράδειγμα, είναι μάλλον ασαφές τι συμβαίνει με τις ταινίες που κάνουν ορισμένους θεατές να τους αρέσουν ή να μην τους αρέσουν. Έτσι, απαντώντας στην «πρόκληση Netflix» για την επινόηση ενός αλγορίθμου που προβλέπει την αξιολόγηση των ταινιών από τους χρήστες, με βάση ένα δείγμα των απαντήσεών τους, οι αλγόριθμοι μηχανικής μάθησης έχουν αποδειχθεί αρκετά επιτυχημένοι (Koedinger et al., 2015).

2.1. Ερωτήματα Ροής (Stream Queries)

Υπάρχουν δύο τρόποι με τους οποίους πραγματοποιούνται ερωτήματα σχετικά με τις ροές. Σε μια ειδική τοποθεσία εντός του επεξεργαστή αποθηκεύονται *σταθερά ερωτήματα* (standing queries). Αυτά τα ερωτήματα, κατά μία έννοια, εκτελούνται μόνιμα και παράγουν αποτελέσματα σε κατάλληλους χρόνους (Weiss & Davison, 2010).

ΠΑΡΑΔΕΙΓΜΑ 1. Η ροή που παράγεται από τον αισθητήρα θερμοκρασίας επιφάνειας-ωκεανού μπορεί να έχει ένα σταθερό ερώτημα για την έξοδο μιας ειδοποίησης όποτε η θερμοκρασία υπερβαίνει τους 25 βαθμούς Κελσίου. Αυτό το ερώτημα απαντάται εύκολα, καθώς εξαρτάται μόνο από το πιο πρόσφατο στοιχείο ροής.

Εναλλακτικά, μπορεί να υπάρχει ένα σταθερό ερώτημα που, κάθε φορά που φθάνει μια νέα ανάγνωση, παράγει τον μέσο όρο των 24 πιο πρόσφατων μετρήσεων. Αυτό το ερώτημα μπορεί επίσης να απαντηθεί εύκολα, εάν γίνει αποθήκευση των 24 πιο πρόσφατων στοιχείων ροής. Όταν φτάσει ένα νέο στοιχείο ροής, μπορεί κανείς να διαγράψει από το χώρο εργασίας το 25ο πιο πρόσφατο στοιχείο, αφού δεν θα χρειαστεί ξανά (εκτός εάν υπάρχει κάποιο άλλο ερώτημα που το απαιτεί) (Hand & Adams, 2014).

Ένα άλλο ερώτημα που μπορεί να πραγματοποιηθεί είναι η μέγιστη θερμοκρασία που έχει καταγραφεί ποτέ από αυτόν τον αισθητήρα. Η απάντηση σε αυτό το ερώτημα θα γίνει διατηρώντας μια απλή περίληψη: Το μέγιστο όλων των στοιχείων ροής που έχουν ειδωθεί. Δεν είναι απαραίτητο να καταγραφεί ολόκληρη η ροή. Όταν φτάσει ένα νέο στοιχείο ροής, συγκρίνεται με το αποθηκευμένο μέγιστο και ορίζεται το μέγιστο σε όποιο είναι μεγαλύτερο. Στη συνέχεια μπορεί να δοθεί απάντηση στο ερώτημα παράγοντας την τρέχουσα τιμή του μέγιστου. Ομοίως, εάν επιθυμεί κανείς τη μέση θερμοκρασία για όλη την ώρα, πρέπει να καταγραφθούν μόνο δύο τιμές: Ο αριθμός των μετρήσεων που έχουν σταλεί ποτέ στη ροή και το άθροισμα αυτών των μετρήσεων. Οι τιμές αυτές προσαρμόζονται εύκολα κάθε φορά που έρχεται μια νέα ανάγνωση και μπορεί να παραχθεί το πηλίκο τους ως απάντηση στο ερώτημα (Wu & Ding, 2013).

Η άλλη μορφή ερωτήματος είναι η *ad-hoc*, μια ερώτηση που τέθηκε μία φορά για την τρέχουσα κατάσταση μιας ροής ή ροών. Εάν δεν αποθηκεύονται όλες οι ροές στο

σύνολό τους, όπως συνήθως δεν γίνεται, τότε δεν μπορεί κανείς να περιμένει να απαντήσει σε αυθαίρετα ερωτήματα σχετικά με τις ροές. Εάν υπάρχει κάποια ιδέα τι είδους ερωτήματα θα ζητηθούν μέσω της διεπαφής ερωτήσεων ad-hoc, τότε μπορεί κανείς να προετοιμαστεί για αυτά αποθηκεύοντας κατάλληλα μέρη ή περιλήψεις ροών (Koedinger et al., 2015).

Εάν θέλει κανείς τη δυνατότητα να υποβάλει μια μεγάλη ποικιλία ερωτημάτων ad-hoc, μια κοινή προσέγγιση είναι να αποθηκεύσει ένα *συρόμενο παράθυρο* κάθε ροής στο χώρο εργασίας. Ένα συρόμενο παράθυρο μπορεί να είναι τα πιο πρόσφατα *n* στοιχεία μιας ροής, για μερικά *n*, ή μπορεί να είναι όλα τα στοιχεία που έφτασαν τις τελευταίες *t* μονάδες χρόνου, π.χ. μία ημέρα. Εάν θεωρηθεί κάθε στοιχείο ροής ως πλειάδα, μπορεί κανείς να αντιμετωπίσει το παράθυρο ως σχέση και να το ζητήσει με οποιοδήποτε ερώτημα SQL. Φυσικά, το σύστημα διαχείρισης ροής πρέπει να διατηρήσει το παράθυρο ενημερωμένο, διαγράφοντας τα παλαιότερα στοιχεία καθώς εισέρχονται νέα (Murray & Scime, 2015).

2.2.Δειγματοληψία Δεδομένων σε Ροή

Ως το πρώτο παράδειγμα διαχείρισης δεδομένων ροής, θα εξεταστεί η εξαγωγή αξιόπιστων δειγμάτων από μια ροή. Όπως με πολλούς αλγόριθμους ροής, το «κόλπο» περιλαμβάνει τη χρήση κατακερματισμού με έναν κάπως ασυνήθιστο τρόπο.

Το παράδειγμα εκτέλεσης είναι χαρακτηριστικό του ακόλουθου γενικού προβλήματος. Η ροή αποτελείται από πλειάδες με *n* στοιχεία. Ένα υποσύνολο των στοιχείων είναι τα *κλειδιά* στοιχεία, στα οποία θα βασίζεται η επιλογή του δείγματος. Στο τρέχον παράδειγμα μας, υπάρχουν τρία στοιχεία - χρήστης, ερώτημα και χρόνος - εκ των οποίων μόνο ο *χρήστης* είναι στο κλειδί. Ωστόσο, θα μπορούσε επίσης κάποιος να πάρει ένα δείγμα ερωτημάτων κάνοντας το *ερώτημα* να είναι το κλειδί ή ακόμη και να πάρει ένα δείγμα του ζεύγους χρήστης - ερώτημα κάνοντας και τα δύο αυτά στοιχεία να αποτελούν το κλειδί (Hand & Adams, 2014).

Για να ληφθεί ένα δείγμα μεγέθους *a* / *b*, κατακερματίζεται η τιμή κλειδιού για κάθε πλειάδα σε *b* buckets και γίνεται αποδεκτή η πλειάδα για το δείγμα εάν η τιμή κατακερματισμού είναι μικρότερη από *a*. Εάν το κλειδί αποτελείται από περισσότερα

από ένα στοιχεία, η συνάρτηση κατακερματισμού πρέπει να συνδυάσει τις τιμές για αυτά τα στοιχεία για να δημιουργήσει μία μόνο τιμή κατακερματισμού. Το αποτέλεσμα θα είναι ένα δείγμα που αποτελείται από όλες τις πλειάδες με συγκεκριμένες τιμές κλειδιών. Οι επιλεγμένες τιμές κλειδιού θα είναι περίπου *a / b* όλων των τιμών - κλειδιά που εμφανίζονται στη ροή (Maia et al., 2020).

2.3.Ανάλυση Φίλτρου Bloom

Εάν μια τιμή κλειδιού είναι στο *S*, τότε το στοιχείο θα περάσει σίγουρα μέσω του φίλτρου Bloom. Ωστόσο, εάν η τιμή κλειδιού δεν είναι στο *S*, πάλι ενδέχεται να περάσει. Πρέπει να γίνει κατανοητό πώς υπολογίζεται η πιθανότητα ενός *ψευδούς θετικού*, ως συνάρτηση του *n*, του μήκους του πίνακα bits *m*, του αριθμού των μελών του *S* και *k*, του αριθμού των συναρτήσεων κατακερματισμού (Wu & Ding, 2013).

Το μοντέλο που χρησιμοποιείται είναι σαν να ρίχνει κανείς βελάκια στους στόχους. Ας υποτεθεί ότι υπάρχουν *x* στόχοι και *y* βελάκια. Κάθε βελάκι είναι εξίσου πιθανό να χτυπήσει οποιονδήποτε στόχο. Αφού ριφθούν τα βελάκια, πόσους στόχους μπορεί κανείς να περιμένει να χτυπήσει τουλάχιστον μία φορά; Η ανάλυση έχει ως εξής:

- Η πιθανότητα ότι ένα δεδομένο βελάκι δεν θα χτυπήσει έναν δεδομένο στόχο είναι (x – 1)/x.
- Η πιθανότητα ότι κανένα από τα *y* βελάκια δεν θα χτυπήσει έναν δεδομένο στόχο είναι $\left(\frac{x-1}{x}\right)^y$. Η έκφραση αυτή μπορεί να γραφτεί και ως $\left(1-\frac{1}{x}\right)^{x\left(\frac{y}{x}\right)}$.
- Χρησιμοποιώντας τότε την προσέγγιση $(1-\varepsilon)^{1/\varepsilon} = 1/e$ για κάποιο μικρό ε, συμπεραίνει κανείς ότι η πιθανότητα κανένα από τα *y* βελάκια δεν θα βρει έναν δεδομένο στόχο είναι $e^{-y/x}$.

2.4.Ο Αλγόριθμος Flajolet – Martin

Είναι δυνατόν να εκτιμηθεί ο αριθμός των διακριτών στοιχείων μέσω του κατακερματισμού των στοιχείων του καθολικού συνόλου σε μια συμβολοσειρά bit που είναι επαρκώς μεγάλη. Το μήκος της συμβολοσειράς bit πρέπει να είναι αρκετό ώστε να υπάρχουν πιο πιθανά αποτελέσματα της συνάρτησης κατακερματισμού από ότι υπάρχουν στοιχεία του καθολικού συνόλου. Για παράδειγμα, 64 bit αρκούν για να

κατακερματιστούν τα URL. Θα επιλεχθούν πολλές διαφορετικές συναρτήσεις κατακερματισμού για τον κατακερματισμό κάθε στοιχείου της ροής χρησιμοποιώντας αυτές τις συναρτήσεις. Η σημαντική ιδιότητα μιας συνάρτησης κατακερματισμού είναι ότι όταν εφαρμόζεται στο ίδιο στοιχείο, παράγει πάντα το ίδιο αποτέλεσμα. Σημειώστε ότι αυτή η ιδιότητα ήταν επίσης απαραίτητη για την τεχνική δειγματοληψίας της προηγούμενης ενότητας (Koedinger et al., 2015).

Η ιδέα πίσω από τον αλγόριθμο Flajolet – Martin είναι ότι όσο περισσότερα διαφορετικά στοιχεία βλέπει κανείς στη ροή, τόσο περισσότερες τιμές κατακερματισμού θα δει. Καθώς παρατηρούνται περισσότερες διαφορετικές τιμές κατακερματισμού, γίνεται πιο πιθανό μια από αυτές τις τιμές να είναι «ασυνήθιστη». Η ιδιαίτερη ασυνήθιστη ιδιότητα που θα εκμεταλλευτούμε είναι ότι η τιμή τελειώνει σε πολλά 0s, αν και υπάρχουν πολλές άλλες επιλογές (Desale et al., 2015).

Κάθε φορά που εφαρμόζεται μια συνάρτηση κατακερματισμού h σε ένα στοιχείο ροής a, η συμβολοσειρά bit $h(a)$ θα τελειώνει σε κάποιο αριθμό από 0s, πιθανώς κανένα. Θα αποκαλείται αυτός ο αριθμός το *μήκος ουράς* για a και h. Έστω το R να είναι το μέγιστο μήκος της ουράς οποιουδήποτε ορατού a μέχρι στιγμής στη ροή. Στη συνέχεια, θα χρησιμοποιηθεί η εκτίμηση 2^R για τον αριθμό των διακριτών στοιχείων που εμφανίζονται στη ροή (Wu & Ding, 2013).

Η εκτίμηση αυτή έχει διαισθητικό νόημα. Η πιθανότητα ότι ένα δοσμένο στοιχείο ροής a που έχει $h(a)$ τελειώνει σε τουλάχιστον r 0s είναι 2^{-r}. Έστω τώρα ότι υπάρχουν m διακριτά στοιχεία μέσα στην ροή. Τότε η πιθανότητα κανένα από αυτά να έχει μήκος ουράς τουλάχιστον r είναι $(1 - 2^{-r})^m$. Τέτοιου είδους εκφράσεις πρέπει να είναι γνώριμες σε αυτό το σημείο. Μπορεί να ξαναγραφτεί η παραπάνω εξίσωση ως: $((1 - 2^{-r})^{2r})^{m2^{-r}}$. Υποθέτοντας ότι το r είναι εύλογα μεγάλο, η εσωτερική έκφραση είναι της μορφής $(1 - \varepsilon)^{1/\varepsilon}$, η οποία είναι προσεγγιστικά ίσο με 1/e. Έτσι, η πιθανότητα του να μην βρεθεί ένα στοιχείο ροής με τουλάχιστον r μηδενικά στο τέλος της τιμής κατακερματισμού του είναι $e^{-m2^{-r}}$. Συμπερασματικά προκύπτει ότι (Hand & Adams, 2014):

- Εάν το m είναι πολύ μεγαλύτερο από 2^r, τότε η πιθανότητα να βρεθεί μια ουρά μήκους τουλάχιστον r πλησιάζει στο 1.
- Εάν το m είναι πολύ μικρότερο από 2^r, τότε η πιθανότητα εύρεσης μήκους ουράς τουλάχιστον r πλησιάζει το 0.

Συμπεραίνεται από αυτά τα δύο σημεία ότι η προτεινόμενη εκτίμηση του m, που είναι 2^R (υπενθυμίζεται ότι R είναι το μεγαλύτερο μήκος ουράς για οποιοδήποτε στοιχείο ροής) είναι απίθανο να είναι είτε πολύ υψηλή είτε πολύ χαμηλή (Koedinger et al., 2015).

2.5.Τρόποι Αντιμετώπισης Άπειρων Ροών

Τεχνικά, η εκτίμηση που χρησιμοποιήθηκε για τις δεύτερες και υψηλότερες ροπές υποθέτει ότι το n, το μήκος ροής, είναι σταθερό. Στην πράξη, το n μεγαλώνει με το χρόνο. Αυτό το γεγονός, από μόνο του, δεν προκαλεί προβλήματα, καθώς αποθηκεύονται μόνο οι τιμές των μεταβλητών και πολλαπλασιάζεται κάποια συνάρτηση αυτής της τιμής με το n όταν είναι ώρα να εκτιμηθεί η ροπή. Εάν μετρηθεί ο αριθμός των στοιχείων ροής που εμφανίζονται και η τιμή αυτή αποθηκευτεί, η οποία απαιτεί μόνο $\log n$ bits, τότε έχει κανείς n διαθέσιμα όποτε το χρειάζεται.

Ένα πιο σοβαρό πρόβλημα είναι ότι πρέπει να είναι κανείς προσεκτικός στο πώς επιλέγονται οι θέσεις για τις μεταβλητές. Εάν γίνει αυτή η επιλογή μία για πάντα, τότε καθώς η ροή μεγαλώνει, είμαστε προκατειλημμένοι υπέρ των πρώτων θέσεων και η εκτίμηση της ροπής θα είναι υπερβολικά μεγάλη. Από την άλλη πλευρά, αν περιμένουμε πάρα πολύ καιρό για να διαλέξουμε θέσεις, τότε νωρίς στη ροή δεν έχουμε πολλές μεταβλητές και έτσι θα προκύψει μια αναξιόπιστη εκτίμηση (Maia et al., 2020).

Η σωστή τεχνική είναι η διατήρηση όσων μεταβλητών μπορούν να αποθηκευτούν ανά πάσα στιγμή και η απόρριψη μερικών καθώς μεγαλώνει η ροή. Οι απορριφθείσες μεταβλητές αντικαθίστανται από νέες, με τρόπο που ανά πάσα στιγμή, η πιθανότητα επιλογής μιας θέσης για μια μεταβλητή είναι η ίδια με εκείνη της επιλογής οποιασδήποτε άλλης θέσης. Ας υποτεθεί ότι υπάρχει χώρος για την αποθήκευση των s μεταβλητών. Στη συνέχεια, οι πρώτες s θέσεις της ροής επιλέγονται η καθεμία ως η θέση μιας από τις s μεταβλητές (Maia et al., 2020).

Επαγωγικά, ας υποτεθεί ότι έχουν παρατηρηθεί n στοιχεία ροής και η πιθανότητα οποιασδήποτε συγκεκριμένης θέσης να είναι η θέση μιας μεταβλητής είναι ομοιόμορφη, δηλαδή s / n. Όταν το $(n + 1)$-*οστό* φτάσει, επιλέγεται η θέση αυτού με πιθανότητα $s/(n+1)$. Εάν δεν επιλεχθεί, οι s μεταβλητές διατηρούν τις ίδιες θέσεις τους. Ωστόσο, εάν επιλεχθεί η (n + 1)-*οστή* θέση, τότε απορρίψτε μία από τις τρέχουσες s μεταβλητές, με την ίδια πιθανότητα. Αντικαταστήστε αυτό που απορρίπτεται από μια νέα μεταβλητή του οποίου το στοιχείο είναι αυτό στη θέση $n + 1$ και του οποίου η τιμή είναι 1.

Σίγουρα, η πιθανότητα ότι η θέση $n + 1$ επιλέγεται για μια μεταβλητή είναι αυτή που πρέπει να είναι: $s / (n + 1)$. Ωστόσο, η πιθανότητα κάθε άλλης θέσης είναι επίσης $s / (n + 1)$, όπως μπορεί να αποδειχθεί με επαγωγή στο n. Με την επαγωγική υπόθεση, πριν από την άφιξη του $(n + 1)$-*οστού* στοιχείου ροής, αυτή η πιθανότητα ήταν s / n. Με πιθανότητα $1 − s / (n + 1)$ η (n + 1)-*οστή* θέση δεν θα επιλεχθεί και η πιθανότητα καθεμιάς από τις πρώτες n θέσεις παραμένει s / n. Ωστόσο, με την πιθανότητα $s / (n + 1)$, επιλέγεται η (n + 1)-*οστή* θέση και η πιθανότητα για καθεμία από τις πρώτες n θέσεις μειώνεται κατά τον παράγοντα $(s - 1) / s$.

2.6.Μοντέλα Ροής Δεδομένων

Η *ροή* εισόδου $a_1, a_2, \ldots$ φτάνει σειριακά, αντικείμενο προς αντικείμενο, και περιγράφει ένα υποκείμενο *σήμα* **A**, μια μονοδιάστατη συνάρτηση **A**: $[1 \cdots N] \rightarrow R$. Η είσοδος μπορεί να περιλαμβάνει πολλαπλές ροές ή πολυδιάστατα σήματα, αλλά δεν λαμβάνονται αυτές οι παραλλαγές προς το παρόν. Τα μοντέλα διαφέρουν ως προς τον τρόπο με τον οποίο τα a_i's περιγράφουν το **A**.

• *Μοντέλο Χρονοσειρών*. Κάθε a_i ισούται με **A**$[i]$ και εμφανίζεται με αυξανόμενη σειρά των i. Αυτό είναι ένα κατάλληλο μοντέλο για δεδομένα χρονοσειρών όταν, για παράδειγμα, παρατηρεί κάποιος τον όγκο της κίνησης σε έναν σύνδεσμο IP κάθε 5 λεπτά, ή τον όγκο συναλλαγών NASDAQ κάθε λεπτό, κ.λπ. Σε κάθε τέτοια «περίοδο», παρατηρείται η επόμενη νέα ενημέρωση.

- ***Μοντέλο Ταμειακής Μηχανής.*** Εδώ τα a_i's είναι προσαυξήσεις των **A**[j]. Σκεφτείτε τα $a_i = (j, I_i)$, $I_i \geq 0$, ότι σημαίνουν $\mathbf{A}_i[j] = A_{i-1}[j] + I_i$ όπου $\mathbf{A}_i$ είναι η

κατάσταση του σήματος αφού εμφανιστεί το στοιχείο i_{th} στη ροή. Όπως και σε μια ταμειακή μηχανή, τα πολλαπλά a_i's θα μπορούσαν να αυξήσουν ένα δεδομένο **A**[j] με την πάροδο του χρόνου. Αυτό είναι ίσως το πιο δημοφιλές μοντέλο ροής δεδομένων. Ταιριάζει σε εφαρμογές όπως παρακολούθηση διευθύνσεων IP που έχουν πρόσβαση σε έναν διακομιστή ιστού, διευθύνσεις IP πηγών που στέλνουν πακέτα μέσω συνδέσμου κ.λπ. επειδή οι ίδιες διευθύνσεις IP ενδέχεται να έχουν πρόσβαση στον διακομιστή Ιστού πολλές φορές ή να στέλνουν πολλά πακέτα στον σύνδεσμο με την πάροδο του χρόνου. Αυτό το μοντέλο έχει εμφανιστεί στο παρελθόν στη λογοτεχνία, αλλά επίσημα βαφτίστηκε (Gilbert et al., 2001) με αυτό το όνομα. Μια ειδική, απλούστερη περίπτωση αυτού του μοντέλου είναι όταν το a_i είναι μια αυθαίρετη *μεταλλαγή* (permutation) του **A**[j], δηλαδή, τα στοιχεία δεν επαναλαμβάνονται στη ροή, αλλά εμφανίζονται εκτός σειράς.

- ***Περιστροφικό Μοντέλο ή Μύλος.*** Εδώ τα a_i's είναι ενημερώσεις για τα **A**[j]. Σκεφτείτε τα $a_i = (j, I_i)$, να σημαίνουν $\mathbf{A}_i[j] = \mathbf{A}_{i-1}[j] + I_i$ όπου $\mathbf{A}_i$ είναι το σήμα αφού εμφανιστεί το στοιχείο i_{th} στη ροή και το I_i μπορεί να είναι θετικό ή αρνητικό. Αυτό είναι το πιο γενικό μοντέλο. Εμπνέεται ήπια από έναν πολυσύχναστο σταθμό του μετρό της Νέας Υόρκης, όπου η περιστροφική πύλη (τουρνικέ) παρακολουθεί τους ανθρώπους που φθάνουν και αναχωρούν συνεχώς. Ανά πάσα στιγμή, ένας μεγάλος αριθμός ανθρώπων βρίσκεται στο μετρό. Αυτό είναι το κατάλληλο μοντέλο για τη μελέτη πλήρως δυναμικών καταστάσεων όπου υπάρχουν εισαγωγές αλλά και διαγραφές, αλλά συχνά είναι δύσκολο να αποκτηθούν ισχυρά όρια σε αυτό το μοντέλο. Αυτό το μοντέλο έχει εμφανιστεί και στο παρελθόν με διαφορετικές μορφές, αλλά βαφτίζεται εδώ με το όνομά του.

Υπάρχει μια μικρή λεπτομέρεια, σε ορισμένες περιπτώσεις, τα $\mathbf{A}_i[j] \geq 0$ για όλα τα i, ανά πάσα στιγμή. Αυτό γενικά αναφέρεται ως το *αυστηρό* περιστροφικό μοντέλο.

Για παράδειγμα, σε μια βάση δεδομένων, μπορεί κανείς να διαγράψει μόνο μια εγγραφή που έχει εισαχθεί. Ως εκ τούτου, η κατανομή του αριθμού εγγραφών με δεδομένη τιμή χαρακτηριστικού δεν μπορεί να είναι αρνητική.

Από την άλλη πλευρά, υπάρχουν περιπτώσεις όπου οι ροές μπορεί να είναι *μη αυστηρές*, δηλαδή, $\mathbf{A}_i[j] < 0$ για ορισμένα i. Αυτό συμβαίνει για παράδειγμα ως μαθησιακό τεχνούργημα, ας πούμε $\mathbf{A}_1 - \mathbf{A}_2$ με τις ροές $\mathbf{A}_1$ και $\mathbf{A}_2$ να διανέμονται σε δύο διαφορετικούς τόπους αντίστοιχα ως ταμειακή μηχανή ή αυστηρές ροές περιστροφικού μοντέλου.

Θα πρέπει να αποφεύγεται η διάκριση μεταξύ των δύο περιστροφικών μοντέλων εκτός εάν είναι απαραίτητο. Συνήθως εργάζεται κανείς στο αυστηρό περιστροφικό μοντέλο, εκτός εάν το μη αυστηρό μοντέλο απαιτείται ρητά.

Τα μοντέλα σε φθίνουσα σειρά γενικότητας είναι τα εξής: Περιστροφικό μοντέλο, ταμειακή μηχανή, χρονοσειράς. (Μια πιο συμβατική περιγραφή των μοντέλων εμφανίζεται στο (Gilbert et al., 2001). Από θεωρητική άποψη, φυσικά κάποιος επιθυμεί να σχεδιάσει αλγόριθμους στο περιστροφικό μοντέλο, αλλά από πρακτική άποψη, ένα από τα άλλα μοντέλα, αν και ασθενέστερα, μπορεί να είναι πιο κατάλληλο για μια εφαρμογή. Επιπλέον, μπορεί να είναι (αποδεδειγμένα) δύσκολο να σχεδιαστούν αλγόριθμοι σε ένα γενικό μοντέλο, και κάποιος μπορεί να πρέπει να συμβιβαστεί με αλγόριθμους σε ένα ασθενέστερο μοντέλο.

Πρέπει να υπολογιστούν διάφορες συναρτήσεις στο σήμα **A** σε διαφορετικούς χρόνους κατά τη διάρκεια της ροής. Αυτές οι συναρτήσεις μπορούν να θεωρηθούν ως ερωτήματα για τη δομή δεδομένων που διασχίζονται με τις ενημερώσεις. Υπάρχουν διαφορετικά μέτρα απόδοσης.

- Χρόνος επεξεργασίας ανά στοιχείο a_i στη ροή. (*Proc. Time*)
- Χώρος που χρησιμοποιείται για την αποθήκευση της δομής δεδομένων στο $\mathbf{A}_t$ τη χρονική στιγμή t. (*Storage*)
- Χρόνος που απαιτείται για τον υπολογισμό συναρτήσεων στο **A**. (*Compute or query time*.)

Υπάρχει επίσης ο χώρος εργασίας που απαιτείται για τον υπολογισμό της συνάρτησης ή την εκτέλεση του ερωτήματος. Ακολουθεί μια αναδιατύπωση των λύσεων στα δύο παζλ στην αρχή όσον αφορά τα μοντέλα ροής δεδομένων και τα μέτρα απόδοσης. Το παζλ στην Ενότητα 1.1 βρίσκεται στο μοντέλο ταμειακών μηχανών (αν και τα

αντικείμενα δεν επαναλαμβάνονται). Η συνάρτηση που πρέπει να υπολογιστεί είναι {j | **A**[j] = 0}, δοθέντος ότι | {j | **A**[j] = 0} | ≤ k. Για την απλούστερη λύση που παρουσιάστηκε για k = 1, ο χρόνος επεξεργασίας ανά στοιχείο και αποθήκευση ήταν O(logn) και ο χρόνος υπολογισμού ήταν O (1). Η συνάρτηση υπολογίζεται στο τέλος της ροής, οπότε ο χρόνος υπολογισμού εδώ ισχύει μόνο μία φορά. Στην άσκηση της αλιείας στην Ενότητα 1.2, είναι και πάλι το μοντέλο ταμειακών μηχανών. Η συνάρτηση που πρέπει να υπολογιστεί είναι η γ[t] ανά πάσα στιγμή t. Ο χρόνος επεξεργασίας ανά στοιχείο ήταν O (*k*log (1/ε)), ο χώρος αποθήκευσης ήταν O (*k* log*u* log (1/ε)) και ο χρόνος υπολογισμού ήταν O (k) για την εκτίμηση της γ [t] έως 1 + ε κατά προσέγγιση υπό την προϋπόθεση ότι ήταν μεγάλη (τουλάχιστον 1 / k) με την επιθυμητή, σταθερή πιθανότητα επιτυχίας. Το ερώτημα για την εκτίμηση του γ[t] μπορεί να προκύψει ανά πάσα στιγμή t κατά τη ροή, οπότε ο υπολογισμός του χρόνου εφαρμόζεται κάθε φορά που εκτιμάται το γ [t]. Το παζλ στην Ενότητα 1.3 δεν ταιριάζει στα μοντέλα ροής δεδομένων παραπάνω, καθώς η λύση του χρησιμοποιεί πολλαπλά περάσματα ή τυχαία πρόσβαση επί του σήματος.

Οι αναγνώστες μπορούν να κατανοήσουν την τεχνική πρόκληση που θέτει αυτό το ζητούμενο σε αντίθεση με μια παραδοσιακή δυναμική δομή δεδομένων, όπως το ισορροπημένο δέντρο αναζήτησης που επεξεργάζεται κάθε ενημέρωση σε χρόνο O(logN) και υποστηρίζει το ερώτημα σε χρόνο O(logN), αλλά χρησιμοποιεί γραμμικό χώρο για την αποθήκευση των δεδομένων εισόδου. Οι αλγόριθμοι ροής δεδομένων μπορούν ομοίως να θεωρηθούν ότι διατηρούν μια δυναμική δομή δεδομένων, αλλά περιορίζονται στη χρήση *υπο-γραμμικού* χώρου αποθήκευσης και των επιπτώσεων που συνοδεύουν. Μερικές φορές, το ζητούμενο γίνεται πιο ελαστικό έτσι ώστε:

Ανά πάσα στιγμή t στη ροή δεδομένων, ο χρόνος επεξεργασίας και η αποθήκευση ανά στοιχείο πρέπει να είναι ταυτόχρονα o(N, t) (κατά προτίμηση, polylog (N, t)), αλλά ο χρόνος υπολογισμού μπορεί να είναι μεγαλύτερος.

Αυτό προτάθηκε στο (Henzinger et al., 1998), χρησιμοποιήθηκε σε λίγα έργα και ισχύει σε περιπτώσεις όπου ο υπολογισμός γίνεται λιγότερο συχνά από το ρυθμό ενημέρωσης. Ακόμα, το ότι το πεδίο *N* και η είσοδος *t* είναι τόσο μεγάλα που απαιτούν τη χρήση μόνο *polylog (N, t)* αποθήκευσης μπορεί στην πραγματικότητα να σημαίνει ότι ο

χρόνος υπολογισμού ακόμη και γραμμικός στο πεδίο ή την είσοδο μπορεί να είναι απαγορευτικός σε εφαρμογές για ένα συγκεκριμένο ερώτημα.

Ακολουθεί ένα ή δύο σχόλια για το ζητούμενο. Πρώτον, γιατί περιορίζουμε τον εαυτό μας σε ένα μικρό (υπο-γραμμικό) χώρο. Συνήθως, κάποιος λέει ότι αυτό συμβαίνει επειδή η ροή δεδομένων είναι τόσο μαζική που μπορεί να μην γίνεται να αποθηκευθούν όλα όσα εμφανίζονται. Αυτό το επιχείρημα είναι κομψό. Ακόμα κι αν η ροή δεδομένων είναι τεράστια, αν περιγράφει ένα συμπαγές σήμα (δηλαδή, το N είναι μικρό), μπορεί να διατεθεί γραμμικός χώρος στο N και να λυθούν προβλήματα στο συμβατικό υπολογιστικό πλαίσιο. Για παράδειγμα, έστω μια μαζική ροή ταυτοτήτων IDs ατόμων και των ηλικιών τους σε χρόνια, και το μόνο που πρέπει να υπολογιστεί είναι συναρτήσεις για την κατανομή της ηλικίας των ανθρώπων ενώ το σήμα είναι πάνω από N λιγότερο από 150, το οποίο είναι τετριμμένο ως προς τη διαχείριση. Αυτό που κάνει τις ροές δεδομένων μοναδικές είναι ότι υπάρχουν εφαρμογές όπου οι ροές δεδομένων περιγράφουν σήματα σε ένα πολύ μεγάλο σύμπαν. Για παράδειγμα, το N μπορεί να είναι

- αριθμός πηγών, ζευγών διευθύνσεων IP προορισμού (ο οποίος είναι επί του παρόντος 2^{64} και ενδέχεται να γίνει μεγαλύτερος εάν υιοθετηθούν μεγαλύτερες διευθύνσεις IP στο μέλλον),
- αριθμός χρονικών διαστημάτων όπου έγιναν ορισμένες παρατηρήσεις (που αυξάνεται γρήγορα με την πάροδο του χρόνου), ή
- Διευθύνσεις http στον Ιστό (κάτι που είναι δυνητικά άπειρο, δεδομένου ότι τα ερωτήματα Ιστού γράφονται μερικές φορές σε κεφαλίδες http).

Γενικότερα, και αυτό είναι πολύ πιο πειστικό, οι ροές δεδομένων είναι παρατηρήσεις πάνω σε πολλά χαρακτηριστικά και οποιοδήποτε υποσύνολο χαρακτηριστικών μπορεί να περιλαμβάνει το πεδίο του σήματος σε μια εφαρμογή και αυτό οδηγεί σε δυνητικά μεγάλους χώρους πεδίων, ακόμη και αν τα μεμονωμένα πεδία χαρακτηριστικών είναι μικρά. Όπως θα φανεί, υπάρχουν σοβαρές εφαρμογές όπου η διαθέσιμη μνήμη είναι πολύ περιορισμένη. Είναι ο συνδυασμός μεγάλου μεγέθους των σημάτων και της περιορισμένης διαθέσιμης μνήμης που οδηγεί στα μοντέλα ροής δεδομένων.

Δεύτερον, γιατί χρησιμοποιείται η συνάρτηση polylog? Αυτό συμβαίνει επειδή ο λογάριθμος του μεγέθους εισόδου είναι το κατώτερο όριο στον αριθμό των bit που απαιτούνται για την ευρετηρίαση και την αναπαράσταση του σήματος, και το poly δίνει ένα οικείο δωμάτιο για «παιχνίδι».

Τέλος, υπάρχει μια γνωστική αναλογία που εξηγεί ποιοτικά τα επιθυμητά δεδομένα και μπορεί να είναι ελκυστική. Τα ανθρώπινα όντα, αντιλαμβάνονται κάθε στιγμή της ζωής τους μέσω μιας σειράς αισθητηριακών παρατηρήσεων (οπτική, ακουστική, νευρική, κ.λπ.). Ωστόσο, κατά τη διάρκεια της ζωής τους, καταφέρνουν να αφαιρέσουν και να αποθηκεύσουν μόνο μέρος των παρατηρήσεων και να λειτουργήσουν επαρκώς ακόμη και αν δεν μπορούν να θυμηθούν κάθε λεπτομέρεια κάθε στιγμής της ζωής τους. Οι άνθρωποι είναι βιολογικές μηχανές επεξεργασίας ροών δεδομένων.

3. Αλγόριθμοι Δειγματοληψίας Μαζικών Δεδομένων

Η δειγματοληψία μαζικών δεδομένων ασχολείται με προβλήματα μαζικών δεδομένων (massive data) όπου τα δεδομένα εισόδου (ένας γράφος, μια μήτρα ή κάποιο άλλο αντικείμενο) είναι πολύ μεγάλα για να αποθηκευτούν στη μνήμη τυχαίας προσπέλασης. Ένα μοντέλο για τέτοια προβλήματα είναι το μοντέλο ροής (streaming model), όπου τα δεδομένα είναι ορατά μόνο μία φορά. Στο μοντέλο ροής, η φυσική τεχνική αντιμετώπισης των μαζικών δεδομένων είναι η δειγματοληψία. Η δειγματοληψία γίνεται «εν κινήσει» (Zhang, 2010).

Για να παρουσιαστεί η βασική έννοια της δειγματοληψίας εν κινήσει έστω το ακόλουθο σενάριο. Από μια ροή n θετικών πραγματικών αριθμών $\alpha_1, \alpha_2, \ldots, \alpha_n$, ζητείται να επιλεχθεί ένα στοιχείο δείγματος α_i έτσι ώστε η πιθανότητα επιλογής ενός στοιχείου να είναι ανάλογη με την τιμή του.

3.1.Ροπές Συχνότητας των Ροών Δεδομένων

Μια σημαντική κατηγορία προβλημάτων αφορά τις ροπές συχνότητας των ροών δεδομένων. Εδώ μια ροή δεδομένων $\alpha_1, \alpha_2, \ldots, \alpha_n$, μήκους n αποτελείται από σύμβολα α_i από ένα αλφάβητο m πιθανών συμβόλων τα οποία για ευκολία δηλώνονται ως $\{1, 2, \ldots, m\}$. Σε όλη αυτή την ενότητα, τα n, m και α_i θα έχουν αυτές τις έννοιες και το s (για το σύμβολο) θα υποδηλώνει ένα γενικό στοιχείο του $\{1, 2, \ldots, m\}$. Η συχνότητα f_s του συμβόλου s είναι ο αριθμός των εμφανίσεων του s στη ροή. Για ένα μη αρνητικό ακέραιο p, η p-οστή ροπή συχνότητας της ροής είναι (Zhang, 2010):

$$\sum_{s=1}^{m} f_s^p$$

Να σημειωθεί ότι η ροπή συχνότητας $p = 0$ αντιστοιχεί στον αριθμό των διακριτών συμβόλων που εμφανίζονται στη ροή. Η πρώτη ροπή συχνότητας είναι απλά n, το μήκος της συμβολοσειράς.

3.2.Αριθμός των Διακριτών Στοιχείων σε μια Ροή Δεδομένων

Έστω μια ακολουθία $\alpha_1, \alpha_2, \ldots, \alpha_n$ από n στοιχεία, κάθε α_i είναι ένας ακέραιος στο εύρος 1 έως m όπου τα n και m είναι πολύ μεγάλα. Ας υποτεθεί ότι πρέπει να

προσδιοριστεί ο αριθμός των διακριτών α_i στην ακολουθία. Κάθε α_i μπορεί να αντιπροσωπεύει έναν αριθμό πιστωτικής κάρτας που εξάγεται από μια ακολουθία συναλλαγών με πιστωτική κάρτα και πρέπει να προσδιοριστεί το πόσοι διαφορετικοί λογαριασμοί πιστωτικών καρτών υπάρχουν. Το μοντέλο είναι μια ροή δεδομένων όπου τα σύμβολα φαίνονται ένα – ένα κάθε φορά.

Έστω $\alpha_1, \alpha_2, \ldots, \alpha_n$ μια ακολουθία από στοιχεία όπου κάθε $\alpha_i \in \{1, 2, \ldots, m\}$. Ο αριθμός των διακριτών στοιχείων μπορεί να εκτιμηθεί με χωρική πολυπλοκότητα $O(\log m)$. Έστω $S \subseteq \{1, 2, \ldots, m\}$ το σύνολο των στοιχείων που εμφανίζονται σε μια ακολουθία. Ας υποτεθεί ότι τα στοιχεία του S επιλέχθηκαν ομοιόμορφα και τυχαία από το $\{1, 2, \ldots, m\}$. Έστω min το ελάχιστο στοιχείο του S. Γνωρίζοντας το ελάχιστο στοιχείο του S επιτρέπει την εκτίμηση του μεγέθους του S. Τα στοιχεία του S διαμερίζουν το σύνολο $\{1, 2, \ldots, m\}$ σε $|S| + 1$ υπο-σύνολα μεγέθους περίπου $\frac{m}{|S|+1}$ το καθένα. Συνεπώς, το ελάχιστο στοιχείο του S θα πρέπει να έχει τιμή κοντά στο $\frac{m}{|S|+1}$. Η επίλυση του $min = \frac{m}{|S|+1}$ αποδίδει $|S| = \frac{m}{min} - 1$. Εφόσον μπορεί να προσδιοριστεί το min, αυτό μπορεί να συντελέσει σε μια εκτίμηση του $|S|$.

Η παραπάνω ανάλυση απαιτούσε τα στοιχεία του S να συλλέγονται ομοιόμορφα και τυχαία από το $\{1, 2, \ldots, m\}$. Αυτό γενικά δεν συμβαίνει όταν υπάρχει μια ακολουθία $\alpha_1, \alpha_2, \ldots, \alpha_n$ ένα στοιχείο από τα στοιχεία $\{1, 2, \ldots, m\}$. Σαφώς εάν τα στοιχεία του S αποκτήθηκαν επιλέγοντας τα $|S|$ μικρότερα στοιχεία του $\{1, 2, \ldots, m\}$, η παραπάνω τεχνική θα έδινε εσφαλμένη απάντηση (Kane et al., 2013). Εάν τα στοιχεία δεν συλλέγονται ομοιόμορφα τυχαία, μπορεί να εκτιμηθεί ο αριθμός των διακριτών στοιχείων; Ο τρόπος επίλυσης αυτού του προβλήματος είναι να χρησιμοποιηθεί μια συνάρτηση κατακερματισμού h όπου

$$h : \{1, 2, \ldots, m\} \rightarrow \{0, 1, 2, \ldots, M - 1\}.$$

Για την μέτρηση του αριθμού των διακριτών στοιχείων στην είσοδο, αρκεί η μέτρηση του αριθμού των στοιχείων στο αντιστοιχισμένο σύνολο $\{h(\alpha_1), h(\alpha_2), \ldots\}$. Το συμπέρασμα είναι ότι το $\{h(\alpha_1), h(\alpha_2), \ldots\}$ συμπεριφέρεται σαν ένα τυχαίο υποσύνολο, έτσι το παραπάνω ευρετικό πόρισμα που χρησιμοποιεί το ελάχιστο για να

εκτιμήσει τον αριθμό των στοιχείων ισχύει. Εάν χρειαζόταν τα $\{h(\alpha_1), h(\alpha_2), \ldots\}$ να είναι εντελώς ανεξάρτητα, η πολυπλοκότητα χώρου που θα απαιτούταν για την αποθήκευση της συνάρτησης κατακερματισμού θα ήταν πολύ υψηλή. Ευτυχώς, χρειάζεται μόνο αμφίδρομη (2-way) ανεξαρτησία (Zhang, 2010).

3.3.Καταμέτρηση του Αριθμού Εμφανίσεων ενός Δεδομένου Στοιχείου

Για τον υπολογισμό του αριθμού των εμφανίσεων ενός στοιχείου σε μια ροή απαιτείται το πολύ χώρος log n όπου το n είναι το μήκος της ροής. Σαφώς, για κάθε μήκους ροή που συμβαίνει στην πράξη, μπορεί να έχει κανείς χώρο log n. Για το λόγο αυτό, το ακόλουθο υλικό δεν μπορεί ποτέ να χρησιμοποιηθεί στην πράξη, αλλά η τεχνική είναι ενδιαφέρουσα και μπορεί να δώσει πληροφορίες για τον τρόπο επίλυσης κάποιου άλλου προβλήματος (Poularakis et al., 2013).

Έστω μια συμβολοσειρά από 0 και 1 του μήκους n στην οποία θέλει κάποιος να μετρήσει τον αριθμό των εμφανίσεων του 1. Σαφώς, εάν είχε κανείς log n bits μνήμης, θα μπορούσε να παρακολουθήσει τον ακριβή αριθμό των 1. Ωστόσο, μπορεί να προσεγγίσει τον αριθμό μόνο με log log n bits.

Ας είναι m ο αριθμός των 1 που εμφανίζονται στην ακολουθία. Έστω μια τιμή k έτσι ώστε το 2^k να είναι περίπου ο αριθμός των εμφανίσεων m. Η αποθήκευση του k απαιτεί μόνο log log n bits μνήμης. Ο αλγόριθμος λειτουργεί ως εξής. Ξεκίνησε με $k = 0$. Για κάθε εμφάνιση του 1, πρόσθεσε ένα στο k με πιθανότητα $1/2^k$. Στο τέλος της συμβολοσειράς, η ποσότητα $2^k - 1$ είναι η εκτίμηση του m. Για την απόκτηση ενός κέρματος που πέφτει με «κεφαλή» με πιθανότητα $1/2^k$, ρίξτε ένα τίμιο κέρμα, ένα που πέφτει με «κεφαλή» με πιθανότητα 1/2, k φορές και αναφέρετε κεφαλή εάν το τίμιο κέρμα πέσει κάτω με «κεφαλή» σε όλες τις k ρίψεις (Teng, 2010).

Με δεδομένο το k, κατά μέσο όρο θα χρειαστούν 2^k άσσοι πριν το k επαυξηθεί. Συνεπώς, ο αναμενόμενος αριθμός από 1 για να παράξουν την τρέχουσα τιμή του k είναι $1 + 2 + 4 + \ldots + 2^{k-1} = 2^k - 1$.

3.3.1. Καταμέτρηση Συχνών Στοιχείων

Πρώτα έστω το πολύ απλό πρόβλημα της ψήφου των n ατόμων. Υπάρχουν m υποψήφιοι, $\{1, 2, \ldots, m\}$. Πρέπει να προσδιοριστεί εάν ένας υποψήφιος θα λάβει την

πλειοψηφία και αν ναι ποιος. Επισήμως, δίνεται μια ροή ακέραιων αριθμών $\alpha_1, \alpha_2, \ldots, \alpha_n$, καθένα α_i ανήκει στο $\{1, 2, \ldots, m\}$, και πρέπει να προσδιοριστεί αν υπάρχει κάποιο $s \in \{1, 2, \ldots, m\}$ που εμφανίζεται περισσότερες από $n / 2$ φορές και εάν ναι, ποιο είναι το s. Είναι εύκολο να παρατηρηθεί ότι για την επίλυση του προβλήματος ακριβώς κατά την ανάγνωση μόνο μιας ροής δεδομένων με ντετερμινιστικό αλγόριθμο, απαιτείται χώρος $\Omega(n)$ (Madduri & Bader, 2009).

3.3.2. Αλγόριθμος Πλειοψηφίας

Αποθήκευσε το α_1 και αρχικοποίησε έναν μετρητή σε ένα. Για κάθε επόμενο α_i, αν το α_i είναι το ίδιο με το τρέχον αποθηκευμένο αντικείμενο, αύξησε τον μετρητή κατά ένα. Εάν διαφέρει, μείωσε τον μετρητή κατά ένα υπό τον όρο ότι ο μετρητής είναι μη μηδενικός. Εάν ο μετρητής είναι μηδέν, αποθήκευσε το α_i και όρισε τον μετρητή σε ένα (Zhang, 2010).

Για να αναλυθεί ο αλγόριθμος, είναι βολικό να δει κανείς το βήμα της μείωσης του μετρητή ως «εξάλειψη» δύο αντικειμένων, το νέο και αυτό που προκάλεσε την τελευταία αύξηση στον μετρητή. Είναι εύκολο να κατανοηθεί ότι εάν υπάρχει ένα στοιχείο πλειοψηφίας s, πρέπει να αποθηκευτεί στο τέλος. Εάν όχι, κάθε εμφάνιση του s έχει εξαλειφθεί. αλλά κάθε τέτοια εξάλειψη προκαλεί επίσης την εξάλειψη ενός άλλου αντικειμένου και έτσι για να μην αποθηκευτεί το πλειοψηφικό στοιχείο στο τέλος, πρέπει να έχουν εξαλειφθεί περισσότερα από n στοιχεία, μια αντίφαση (Poularakis et al., 2013).

3.3.3. Αλγόριθμος Συχνότητας

Διατηρήστε μια λίστα αντικειμένων που μετρούνται. Αρχικά η λίστα είναι κενή. Για κάθε στοιχείο, εάν είναι ίδιο με κάποιο στοιχείο στη λίστα, αυξήστε τον μετρητή του κατά ένα. Εάν διαφέρει από όλα τα στοιχεία της λίστας, τότε εάν υπάρχουν λιγότερα από k στοιχεία στη λίστα, προσθέστε το στοιχείο στη λίστα και θέστε το μετρητή του σε ένα. Εάν υπάρχουν ήδη k στοιχεία στη λίστα μειώστε κάθε έναν από τους τρέχοντες μετρητές κατά ένα. Διαγράψτε ένα στοιχείο από τη λίστα εάν η μέτρηση του γίνει μηδέν (Teng, 2010).

4. Αλγόριθμοι Εξαγωγής Ροών Δεδομένων

Στη συνέχεια παρουσιάζονται τα βασικά προβλήματα εξαγωγής ροής καθώς και οι προκλήσεις που σχετίζονται με κάθε πρόβλημα.

Ομαδοποίηση ροής δεδομένων. Πρόκειται για ένα ευρέως μελετημένο πρόβλημα στη βιβλιογραφία εξαγωγής δεδομένων. Ωστόσο, η προσαρμογή των αυθαίρετων αλγορίθμων ομαδοποίησης στις ροές δεδομένων είναι δύσκολη λόγω των περιορισμών μονού περάσματος στο σύνολο δεδομένων. Ενδιαφέρον παρουσιάζει μία προσαρμογή του αλγορίθμου k-means (Guha et al., 2000) που χρησιμοποιεί μια προσέγγιση βασισμένη σε επιμερισμό του συνόλου δεδομένων. Αυτή η προσέγγιση χρησιμοποιεί προσαρμογή μιας τεχνικής k-means για τη δημιουργία συστάδων σε ολόκληρη τη ροή δεδομένων. Στο πλαίσιο των ροών δεδομένων, μπορεί να είναι πιο επιθυμητό να καθοριστούν οι συστάδες σε συγκεκριμένους ορίζοντες που ορίζονται από τον χρήστη παρά σε ολόκληρο το σύνολο δεδομένων. Αναλύθηκε η τεχνική μικρο-συστάδων (Aggarwal et al., 2003) η οποία καθορίζει τις συστάδες σε ολόκληρο το σύνολο δεδομένων, καθώς και μια ποικιλία εφαρμογών μικρο-συστάδων που μπορούν να πραγματοποιήσουν αποτελεσματική ανάλυση βάσει συνόψεων του συνόλου δεδομένων. Για παράδειγμα, η μικρο-συστάδα μπορεί να επεκταθεί στο πρόβλημα της ταξινόμησης σε ροές δεδομένων (Aggarwal et al., 2004) ενώ σε πολλές περιπτώσεις, μπορεί να χρησιμοποιηθεί για αυθαίρετες εφαρμογές εξαγωγής δεδομένων, όπως η προστασία προσωπικών δεδομένων ή η εκτίμηση ερωτημάτων.

Ταξινόμηση ροής δεδομένων. Το πρόβλημα της ταξινόμησης είναι ίσως ένα από τα πιο ευρέως μελετημένα στο πλαίσιο της εξαγωγής ροής δεδομένων. Το πρόβλημα της ταξινόμησης καθίσταται δυσκολότερο από την εξέλιξη της υποκείμενης ροής δεδομένων. με συνέπεια να απαιτείται ο σχεδιασμός αποτελεσματικών αλγόριθμων ικανών να λάβουν υπόψη τη χρονική θέση. Καλύπτεται μια μεγάλη ποικιλία αλγορίθμων ταξινόμησης για ροές δεδομένων, μερικοί τους οποίους έχουν σχεδιαστεί ως καθαρά μονόδρομες προσαρμογές των συμβατικών αλγορίθμων ταξινόμησης (Domingos & Hulten, 2000), ενώ άλλοι (όπως οι μέθοδοι στο (Aggarwal et al., 2004; Hulten et al., 2001) είναι πιο αποτελεσματικοί στη λογιστική εξέλιξη της υποκείμενης ροής δεδομένων και φυσικά αναλύονται πλεονεκτήματα του καθενός.

Συχνή εξόρυξη προτύπων. Το πρόβλημα της εξόρυξης συχνών προτύπων παρουσιάστηκε για πρώτη φορά στο (Agrawal et al., 1993), και αναλύθηκε εκτενώς για τη συμβατική περίπτωση συνόλων μόνιμων δεδομένων δίσκου. Στην περίπτωση ροών δεδομένων, μπορεί κανείς να θέλει να βρει τα συνηθισμένα σύνολα στοιχείων είτε από ένα ολισθαίνον παράθυρο είτε από ολόκληρη τη ροή δεδομένων (Giannella et al., 2002; Jin & Agrawal, 2005).

Αλλαγή ανίχνευσης σε ροές δεδομένων: Τα πρότυπα σε μια ροή δεδομένων μπορεί να εξελίσσονται με την πάροδο του χρόνου ενώ αρκετά συχνά είναι επιθυμητή η παρακολούθηση και η ανάλυση των αλλαγών αυτών με την πάροδο του χρόνου. Στην εργασία των (Dasu et al., 2005; Kifer et al., 2004), συζητήθηκαν αρκετές μέθοδοι για την ανίχνευση αλλαγών των ροών δεδομένων. Επιπλέον, η εξέλιξη της ροής δεδομένων μπορεί επίσης να επηρεάσει τη συμπεριφορά των υποκείμενων αλγορίθμων εξαγωγής δεδομένων, καθώς τα αποτελέσματα μπορούν να καταστούν ξεπερασμένα με την πάροδο του χρόνου.

Ανάλυση κυβικής ροής (Stream Cube) πολλαπλών διαστάσεων ροών: Μεγάλο μέρος των δεδομένων ροής βρίσκεται σε έναν πολυδιάστατο χώρο και σε σχετικά χαμηλό επίπεδο αφαίρεσης, ενώ οι περισσότεροι αναλυτές ενδιαφέρονται για δυναμικές αλλαγές σχετικά υψηλού επιπέδου σε κάποιο συνδυασμό διαστάσεων. Προκειμένου για δυναμικά και εξελισσόμενα χαρακτηριστικά υψηλού επιπέδου, καθίσταται σχεδόν απαραίτητη μία πολυεπίπεδη, πολυδιάστατη διαδικτυακή αναλυτική επεξεργασία (on-line analytical processing-OLAP) ροής δεδομένων. Αυτή η αναγκαιότητα απαιτεί τη διερεύνηση νέων αρχιτεκτονικών που μπορούν να διευκολύνουν την on-line αναλυτική επεξεργασία πολυδιάστατων δεδομένων ροής (Chen et al., 2002; Dong et al., 2001).

Ενδιαφέρον παρουσιάζει η αρχιτεκτονική κυβικής ροής που εκτελεί αποτελεσματικά την on-line μερική συγκέντρωση πολυδιάστατων δεδομένων ροής, συλλαμβάνει τα βασικά δυναμικά και εξελισσόμενα χαρακτηριστικά των ροών δεδομένων και διευκολύνει τη γρήγορη OLAP σε δεδομένα ροής. Η αρχιτεκτονική Stream Cube διευκολύνει την online αναλυτική επεξεργασία δεδομένων ροής και αποτελεί μια προκαταρκτική δομή για online εξαγωγή ροής.

Αποβολή φορτίου σε ροές δεδομένων: Οι ροές δεδομένων δημιουργούνται από διεργασίες που είναι ξένες προς την εφαρμογή επεξεργασίας ροής με συνέπεια να καθίσταται αδύνατος ο έλεγχος του ρυθμού εισερχόμενης ροής, με συνέπεια το σύστημα να πρέπει να διαθέτει τη δυνατότητα γρήγορης προσαρμογής σε διαφορετικούς ρυθμούς επεξεργασίας εισερχόμενων ροών. Ένας συγκεκριμένος τύπος προσαρμοστικότητας είναι η ικανότητα υποβάθμισης της απόδοσης χάρη στην "απόρριψη φορτίου" (πτώση μη επεξεργασμένων ποσοτήτων για τη μείωση του φορτίου του συστήματος) όταν οι απαιτήσεις που τίθενται στο σύστημα δεν μπορούν να ικανοποιηθούν πλήρως με δεδομένους διαθέσιμους πόρους. Παρουσιάζονται λοιπόν οι αλγόριθμοι εκείνοι οι οποίοι καθορίζουν το σε ποια σημεία (σε ένα σύνολο ερωτημάτων) πρέπει να εκτελείται η απόρριψη φορτίου και ποια ποσότητα φορτίου πρέπει να αποβάλλεται σε κάθε σημείο προκειμένου να ελαχιστοποιηθεί ο βαθμός ανακρίβειας που εισάγεται στις απαντήσεις ερωτημάτων.

Υπολογισμοί ολισθαίνοντος παραθύρου σε ροές δεδομένων. Πολλές από τις δομές της σύνοψης που συζητήθηκαν χρησιμοποιούν ολόκληρη τη ροή δεδομένων για να κατασκευάσουν την αντίστοιχη δομή σύνοψης. Το μοντέλο υπολογισμού ολισθαίνοντος παραθύρου βασίζεται στην παραδοχή ότι είναι πιο σημαντικό να χρησιμοποιούνται πρόσφατα δεδομένα στον υπολογισμό ροής δεδομένων (Datar et al., 2002). Επομένως, η επεξεργασία και η ανάλυση γίνεται μόνο σε ένα σταθερό ιστορικό της ροής δεδομένων.

Δημιουργία σύνοψης σε ροές δεδομένων. Ο μεγάλος όγκος ροών δεδομένων θέτει μοναδικούς περιορισμούς χώρου και χρόνου στη διαδικασία υπολογισμού. Πολλές επεξεργασίες ερωτημάτων, λειτουργίες βάσης δεδομένων και αλγόριθμοι εξαγωγής απαιτούν αποτελεσματική εκτέλεση που μπορεί να είναι δύσκολο να επιτευχθεί με μια γρήγορη ροή δεδομένων. Σε πολλές περιπτώσεις, μπορεί να είναι αποδεκτό να δημιουργηθούν κατά προσέγγιση λύσεις για τέτοια προβλήματα. Τα τελευταία χρόνια έχει αναπτυχθεί μια σειρά συνόψεων, οι οποίες μπορούν να χρησιμοποιηθούν σε συνδυασμό με μια ποικιλία τεχνικών εξαγωγής και επεξεργασίας ερωτημάτων (Garofalakis et al., 2002). Ορισμένες βασικές μέθοδοι σύνοψης περιλαμβάνουν εκείνες της δειγματοληψίας, των κυμάτων, των σκίτσων και των ιστογραμμάτων.

Επεξεργασία σύνδεσης σε ροές δεδομένων. Η σύνδεση σε ροές είναι μια θεμελιώδης λειτουργία για τη συσχέτιση πληροφοριών από διαφορετικές ροές κάτι ιδιαίτερα χρήσιμο σε πολλές εφαρμογές, όπως τα δίκτυα αισθητήρων στα οποία οι ροές που προέρχονται από διαφορετικές πηγές μπορεί να χρειάζεται να σχετίζονται μεταξύ τους. Στη ρύθμιση της ροής, οι ποσότητες εισόδου φθάνουν συνεχώς και οι ποσότητες των αποτελεσμάτων πρέπει επίσης να παράγονται συνεχώς. Η υπόθεση ότι τα δεδομένα εισαγωγής είναι ήδη αποθηκευμένα ή ευρετηριοποιημένα ή ότι ο ρυθμός εισαγωγής μπορεί να ελέγχεται από το σχέδιο ερωτημάτων δεν ευσταθεί.. Επίσης δεν είναι δυνατή η εφαρμογή αλγορίθμων τυπικής σύνδεσης που χρησιμοποιούν λειτουργίες αποκλεισμού, π.χ. η ταξινόμηση. Οι συμβατικές μέθοδοι για την εκτίμηση του κόστους και τη βελτιστοποίηση ερωτημάτων είναι επίσης ακατάλληλες, επειδή προϋποθέτουν πεπερασμένη εισαγωγή. Επιπλέον, η μακροχρόνια φύση των ερωτημάτων ροής απαιτεί πιο προσαρμοσμένες στρατηγικές επεξεργασίας που μπορούν να αντιδράσουν σε αλλαγές και διακυμάνσεις στα δεδομένα και στα χαρακτηριστικά ροής. Γενικά, προκειμένου να υπολογιστεί το πλήρες αποτέλεσμα μιας σύνδεσης ροής, πρέπει να διατηρηθούν όλες τις προηγούμενες είσοδοι ως μέρος της κατάστασης επεξεργασίας, επειδή μια νέα πλειάδα μπορεί να συνδεθεί με μια αυθαίρετα παλιά πλειάδα που έφτασε κάποια στιγμή στο παρελθόν. Αυτό το πρόβλημα επιδεινώνεται εξαιτίας των απεριόριστων ροών εισόδου, των περιορισμένων πόρων επεξεργασίας και των απαιτήσεων υψηλής απόδοσης, καθώς είναι αδύνατο μακροπρόθεσμα να διατηρηθεί όλο το ιστορικό σε γρήγορη μνήμη.

Ευρετηριοποίηση ροών δεδομένων. Το πρόβλημα της ευρετηριοποίησης ροών δεδομένων επιχειρεί να δημιουργήσει μια ευρετηριοποιημένη αναπαράσταση, έτσι ώστε να είναι δυνατή η αποτελεσματική απάντηση διαφορετικών ειδών ερωτημάτων, όπως ερωτήματα συγκέντρωσης ή ερωτήματα βάσει τάσεων. Αυτό είναι ιδιαίτερα σημαντικό στην περίπτωση ροής δεδομένων λόγω του τεράστιου όγκου των υποκείμενων δεδομένων.

Μείωση διαστάσεων και πρόβλεψη σε ροές δεδομένων. Λόγω της έμφυτης χρονικής φύσης των ροών δεδομένων, τα προβλήματα της μείωσης της διάστασης και της πρόβλεψης είναι ιδιαίτερα σημαντικά. Όταν υπάρχει μεγάλος αριθμός ταυτόχρονης

ροής δεδομένων, είναι δυνατός ο συσχετισμός μεταξύ διαφορετικών ροών δεδομένων προκειμένου να προκύψουν αποτελεσματικές προβλέψεις (Sakurai et al., 2005; Yi et al., 2000) σχετικά με τη μελλοντική συμπεριφορά της ροής δεδομένων. Επίσης αναλύθηκε μία μέθοδος μείωσης διαστάσεων και προβλέψεων για το πρόβλημα των ροών δεδομένων, η MUSCLES (Yi et al., 2000) και διερευνήθηκε η εφαρμογή της σε ροές δεδομένων. Σημαντικός επίσης είναι και ο αλγόριθμος SPIRIT, ο οποίος διερευνά τη σχέση μεταξύ μείωσης διαστάσεων και πρόβλεψης σε ροές δεδομένων. πιο συγκεκριμένα διερευνάται η χρήση ενός συμπαγούς αριθμού κρυφών μεταβλητών για την πλήρη περιγραφή της ροής δεδομένων. Αυτή η συμπαγής αναπαράσταση μπορεί επίσης να χρησιμοποιηθεί για αποτελεσματική πρόβλεψη των ροών δεδομένων.

Κατανεμημένη εξαγωγή ροών δεδομένων. Σε πολλές περιπτώσεις, οι ροές δημιουργούνται σε πολλούς κατανεμημένους υπολογιστικούς κόμβους. Η ανάλυση και παρακολούθηση δεδομένων σε τέτοια περιβάλλοντα απαιτεί τεχνολογία εξαγωγής δεδομένων βασισμένη σε βελτιστοποίηση πλήθους κριτηρίων, όπως το κόστος επικοινωνίας σε διαφορετικούς κόμβους, καθώς και οι υπολογιστικές, μνήμες ή απαιτήσεις αποθήκευσης σε κάθε κόμβο. Απαιτείται λοιπόν μία ολοκληρωμένη έρευνα για την προσαρμογή διαφορετικών συμβατικών αλγορίθμων εξόρυξης στην κατανεμημένη περίπτωση που θα αναχαιτίζει τα προβλήματα ομαδοποίησης, ταξινόμησης, εξακρίβωσης, συχνής εξαγωγής μοτίβων και συνόψισης.

Εξαγωγή ροής στα δίκτυα αισθητήρων. Με τις πρόσφατες εξελίξεις στην τεχνολογία υλικού, κατέστη δυνατή η παρακολούθηση μεγάλων ποσοτήτων δεδομένων με κατανεμημένο τρόπο με τη χρήση τεχνολογίας αισθητήρων. Οι μεγάλες ποσότητες δεδομένων που συλλέγονται από τους κόμβους του αισθητήρα καθιστούν το πρόβλημα της παρακολούθησης δύσκολο σε διάφορα τεχνολογικά επίπεδα. Οι κόμβοι αισθητήρων έχουν περιορισμένη τοπική αποθήκευση, υπολογιστική ισχύ και διάρκεια ζωής της μπαταρίας, ως αποτέλεσμα των οποίων είναι επιθυμητό να ελαχιστοποιείται η αποθήκευση, η επεξεργασία και η επικοινωνία από αυτούς τους κόμβους. Το πρόβλημα διευρύνεται περαιτέρω από το γεγονός ότι ένα δεδομένο δίκτυο μπορεί να διαθέτει εκατομμύρια κόμβους αισθητήρα και επομένως να μην είναι εφικτό να

εντοπίζονται όλα τα δεδομένα σε έναν δεδομένο συνολικό κόμβο για ανάλυση τόσο από άποψη αποθήκευσης όσο και από άποψη επικοινωνίας.

4.1.Συγκέντρωση Μαζικών Ροών Δεδομένων

Η πρόοδος στην τεχνολογία υλικού επιτρέπει την καταγραφή αυτόματων συναλλαγών καθώς άλλων στοιχείων πληροφοριών της καθημερινής ζωής με ταχύ ρυθμό. Τέτοιες διαδικασίες δημιουργούν τεράστια ποσά διαδικτυακών δεδομένων που αυξάνονται με απεριόριστο ρυθμό. Αυτά τα είδη διαδικτυακών δεδομένων αναφέρονται ως ροές δεδομένων. Τα ζητήματα σχετικά με τη διαχείριση και την ανάλυση των ροών δεδομένων έχουν διερευνηθεί εκτενώς τα τελευταία χρόνια λόγω των αναδυόμενων, επικείμενων και ευρέων εφαρμογών τους (Domingos & Hulten, 2001; O'Callaghan et al., 2002).

Πολλά σημαντικά προβλήματα όπως η ομαδοποίηση και η ταξινόμηση έχουν μελετηθεί ευρέως όσον αφορά την εξαγωγή δεδομένων. Ωστόσο, η πλειονότητα αυτών των μεθόδων ενδέχεται να μην λειτουργεί αποτελεσματικά σε ροές δεδομένων. Οι ροές δεδομένων δημιουργούν ιδιαίτερες προκλήσεις σε έναν αριθμό αλγορίθμων εξαγωγής δεδομένων, όχι μόνο λόγω του τεράστιου όγκου των διαδικτυακών ροών δεδομένων, αλλά και λόγω του γεγονότος ότι τα δεδομένα στις ροές ενδέχεται να εμφανίζουν χρονικές συσχετίσεις. Τέτοιοι χρονικοί συσχετισμοί μπορούν να βοηθήσουν στην αποκάλυψη σημαντικών χαρακτηριστικών εξέλιξης δεδομένων και μπορούν επίσης να χρησιμοποιηθούν για την ανάπτυξη αποδοτικών και αποτελεσματικών αλγορίθμων εξαγωγής. Επιπλέον, οι ροές δεδομένων απαιτούν διαδικτυακή εξαγωγή, κατά την οποία το επιθυμητό είναι η εξαγωγή δεδομένων με συνεχή τρόπο. Επιπλέον, το σύστημα πρέπει να έχει την ικανότητα να εκτελεί ανάλυση off line βάσει των απαιτήσεων των χρηστών. Το παραπάνω μοιάζει με ένα διαδικτυακό πλαίσιο αναλυτικής επεξεργασίας (online analytical processing-OLAP) που χρησιμοποιεί το παράδειγμα της εφάπαξ προεπεξεργασίας, υποβάλλοντας ερωτήσεις πολλές φορές.

Με βάση τις παραπάνω εκτιμήσεις, προτείνται ένα νέο πλαίσιο εξαγωγής ροών, το οποίο υιοθετεί ένα πλαίσιο παραθύρου με κλίση χρόνου, λαμβάνει τη μικρο-συστάδα ως διαδικασία προεπεξεργασίας και ενσωματώνει την προεπεξεργασία με τη σταδιακή,

δυναμική διαδικασία εξαγωγής. Η προεπεξεργασία μικρο-συστάδας συμπιέζει αποτελεσματικά τα δεδομένα, διατηρεί τη γενική χρονική τοποθεσία τους και διευκολύνει τόσο την εντός όσο και την εκτός σύνδεσης ανάλυση, καθώς και την ανάλυση των τρεχόντων δεδομένων και τις κανονικότητες εξέλιξης αυτών.

Στην εργασία του ο (Aggarwal 2007), επικεντρώθηκε κυρίως στην εφαρμογή αυτής της τεχνικής όσον αφορά δύο προβλήματα: (1) την ομαδοποίηση ροής και (2) την ταξινόμηση ροής. Σκοπός της προσέγγισης είναι η χρήση μιας διαδικτυακής συνοπτικής προσέγγισης που είναι αποτελεσματική και επιτρέπει την αποτελεσματική επεξεργασία των ροών δεδομένων.

4.2.Ομαδοποίηση Εξελισσόμενων Ροών Δεδομένων

Το πρόβλημα ομαδοποίησης ορίζεται ως εξής: ένα δεδομένο σύνολο σημείων δεδομένων, πρέπει να διαχωριστεί σε μία ή περισσότερες ομάδες παρόμοιων αντικειμένων. Η ομοιότητα των αντικειμένων μεταξύ τους ορίζεται τυπικά με τη χρήση ευρέως ερευνημένων στη κοινότητα βάσεων δεδομένων, εξαγωγής δεδομένων και στατιστικών (Guha et al., 1998; Jain, & Dubes, 1988) λόγω της χρήσης τους σε ένα ευρύ φάσμα εφαρμογών. Πρόσφατα, το πρόβλημα ομαδοποίησης μελετήθηκε επίσης στο πλαίσιο του περιβάλλοντος ροής δεδομένων (Guha et al., 2003; O'Callaghan et al., 2002).

Ένας προηγούμενος αλγόριθμος που ονομάζεται STREAM (O'Callaghan et al., 2002) υποθέτει ότι οι ομάδες πρέπει να υπολογίζονται σε ολόκληρη τη ροή δεδομένων. Ενώ μια τέτοια εργασία μπορεί να είναι χρήσιμη σε πολλές εφαρμογές, ωστόσο ένα πρόβλημα ομαδοποίησης είναι δυνατόν να προκύψει μόνο σε ένα τμήμα μιας ροής δεδομένων. Αυτό συμβαίνει επειδή η ροή πρέπει να θεωρηθεί ως μια άπειρη διαδικασία που αποτελείται από δεδομένα που εξελίσσονται συνεχώς με το χρόνο. Ως αποτέλεσμα, οι υποκείμενες συστάδες μπορεί κατ' επέκταση να αλλάξουν κι αυτές σημαντικά με την πάροδο του χρόνου. Η φύση των συστάδων μπορεί να ποικίλλει τόσο με τη στιγμή κατά την οποία υπολογίζονται όσο και με τον χρονικό ορίζοντα κατά τον οποίο μετρούνται. Για παράδειγμα, ένας αναλυτής δεδομένων μπορεί να επιθυμεί να εξετάσει συστάδες που σημειώθηκαν τον τελευταίο μήνα, τον περασμένο χρόνο ή την τελευταία

δεκαετία. Τέτοιες συστάδες μπορεί να είναι σημαντικά διαφορετικές. Επομένως, μία από τις εισόδους στον αλγόριθμο ομαδοποίησης είναι ένας χρονικός ορίζοντας πάνω στον οποίο βρίσκονται οι συστάδες. Ο διαδικτυακό αλγόριθμο που χρησιμοποιείται για τη συγκέντρωση ροών δεδομένων σήμερα είναι ο CluStream

4.2.1. Διαδικτυακή συντήρηση μικρο-συστοιχιών - Ο αλγόριθμος CluStream

Η φάση μικρο-συστάδων είναι το on line τμήμα συλλογής στατιστικών δεδομένων του αλγορίθμου. Αυτή η διαδικασία δεν εξαρτάται από κανένα χρήστη εισόδου, όπως ο χρονικός ορίζοντας ή η απαιτούμενη λεπτομέρεια της διαδικασίας ομαδοποίησης. Ο στόχος είναι να διατηρηθούν τα στατιστικά στοιχεία σε ένα αρκετά υψηλό επίπεδο (χρονικής και χωρικής) διακριτότητας, έτσι ώστε να μπορεί να χρησιμοποιηθεί αποτελεσματικά από τα στοιχεία εκτός σύνδεσης, όπως η ομαδοποίηση συγκεκριμένης μακρο-εντολής καθώς και η ανάλυση εξέλιξης. Η βασική ιδέα του αλγορίθμου συντήρησης μικρο-συστάδων αντλεί ιδέες από τους αλγόριθμους k-means και πλησιέστερου γείτονα. Ο αλγόριθμος λειτουργεί με επαναληπτικό τρόπο, διατηρώντας διαρκώς ένα τρέχον σύνολο μικρο-συστάδων. Υποτίθεται ότι αποθηκεύονται συνολικά q μικρο-συστάδες συστημένων ανά πάσα στιγμή από τον αλγόριθμο. Έστω τα μικροσυστήματα M_1 .. .M_q. Σε συνδυασμό με κάθε μικρο-συστάδα i, δημιουργείται ένα μοναδικό αναγνωριστικό. Εάν συγχωνευθούν δύο μικρο-συστάδες (όπως θα γίνει εμφανές από τις λεπτομέρειες του αλγορίθμου συντήρησής), δημιουργείται μια λίστα αναγνωριστικών για την αναγνώριση των συστατικών μικρο-συστάδων. Η τιμή του q καθορίζεται από την ποσότητα της κύριας μνήμης που είναι διαθέσιμη για την αποθήκευση των μικρο-συστάδων. Επομένως, οι τυπικές τιμές του q είναι σημαντικά μεγαλύτερες από τον φυσικό αριθμό των συστάδων στα δεδομένα, αλλά είναι επίσης σημαντικά μικρότερες από τον αριθμό των σημείων δεδομένων που φθάνουν σε μεγάλο χρονικό διάστημα για μια μαζική ροή δεδομένων. Αυτές οι μικρο-συστάδες αντιπροσωπεύουν το τρέχον στιγμιότυπο συστάδων και αλλάζουν κατά τη διάρκεια της ροής καθώς φθάνουν νέα σημεία. Η κατάστασή τους αποθηκεύεται μακριά στο δίσκο κάθε φορά που ο χρόνος ρολογιού διαιρείται από α^i για οποιονδήποτε ακέραιο i. Ταυτόχρονα, τυχόν μικρο-συστάδες της τάξης r που είχαν αποθηκευτεί κάποια στιγμή

στο παρελθόν πιο απομακρυσμένα από τις μονάδες α^{l+r} διαγράφονται από τον αλγόριθμο.

Το πρώτο βήμα είναι να δημιουργηθούν οι αρχικές q μικρο-συστάδες. Αυτό γίνεται χρησιμοποιώντας μια διαδικασία εκτός σύνδεσης στην αρχή της διαδικασίας υπολογισμού ροής δεδομένων. Στην αρχή της ροής δεδομένων, αποθηκεύονται τα πρώτα σημεία InitNumber στο δίσκο και χρησιμοποιείται ένας τυπικός αλγόριθμος συμπλέγματος k-means για τη δημιουργία των αρχικών μικρο-συστάδων q. Η τιμή του Init Number επιλέγεται να είναι τόσο μεγάλη όσο επιτρέπεται από την υπολογιστική πολυπλοκότητα ενός αλγόριθμου k-means που δημιουργεί συστάδες q.

Μόλις δημιουργηθούν αυτά τα αρχικά μικροσυστήματα, ξεκινά η διαδικτυακή διαδικασία ενημέρωσης των μικρο-συστάδων. Κάθε φορά που φθάνει ένα νέο σημείο $\overline{X_{\iota_k}}$ δεδομένων, οι μικροσυστάδες ενημερώνονται ώστε να αντικατοπτρίζουν τις αλλαγές. Κάθε σημείο δεδομένων πρέπει είτε να απορροφηθεί από μία μικρο-συστάδα, είτε πρέπει να τοποθετηθεί σε μία δική του συστάδα. Είναι προτιμότερη η απορρόφηση του σημείου δεδομένων σε μία υπάρχουσα μικροσυστάδα. Αρχικά υπολογίζεται η απόσταση κάθε σημείου δεδομένων από τα κεντροειδή των μικρο-συστάδων M_1. ..M_q. Έστω η τιμή απόστασης του σημείου δεδομένων $\overline{X_{\iota_k}}$, στο κεντροειδές της μικρο-συστάδας M_j από dist (M_j, $\overline{X_{\iota_k}}$). Δεδομένου ότι το κεντροειδές της μικρο-συστάδας διατίθεται στον φορέα χαρακτηριστικών συμπλέγματος, αυτή η τιμή μπορεί να υπολογιστεί σχετικά εύκολα.

4.2.2. Ομαδοποίηση ροής προβολής υψηλών διαστάσεων

Η μέθοδος μπορεί επίσης να επεκταθεί και στην περίπτωση προβλεπόμενης ομαδοποίησης ροής υψηλών διαστάσεων. Οι αλγόριθμοι αναφέρονται ως HPSTREAM. Η περίπτωση υψηλών διαστάσεων παρουσιάζει μια ειδική πρόκληση για τη συγκέντρωση αλγορίθμων ακόμη και στον παραδοσιακό τομέα των στατικών συνόλων δεδομένων, γεγονός που οφείλεται στην βραδύτητα των δεδομένων στην περίπτωση υψηλής διάστασης. Σε χώρο υψηλών διαστάσεων, όλα τα ζεύγη σημείων τείνουν να είναι σχεδόν ισοδύναμα μεταξύ τους. Συνεπώς ο ορισμός συστάδων με βάση την απόσταση με ουσιαστικό τρόπο είναι συχνά μη ρεαλιστικός. Μερικές πρόσφατες

εργασίες για δεδομένα υψηλών διαστάσεων χρησιμοποιούν τεχνικές για προβλεπόμενη ομαδοποίηση που μπορούν να καθορίσουν συστάδες για ένα συγκεκριμένο υποσύνολο διαστάσεων (Aggarwal et al., 1999; Agrawal et al., 1998). Σε αυτές τις μεθόδους, οι ορισμοί των συστάδων είναι τέτοιοι ώστε κάθε συστάδα να είναι συγκεκριμένη για μια συγκεκριμένη ομάδα διαστάσεων. Το γεγονός αυτό μετριάζει σε κάποιο βαθμό το πρόβλημα βραδύτητας στον χώρο υψηλής διαστάσεων. Ακόμα κι αν ένα σύμπλεγμα μπορεί να μην ορίζεται ουσιαστικά σε όλες τις διαστάσεις λόγω της βραδύτητας των δεδομένων, μπορεί να βρεθεί πάντα ένα υποσύνολο των διαστάσεων στις οποίες συγκεκριμένα υποσύνολα σημείων σχηματίζουν υψηλής ποιότητας συστάδες. Φυσικά, αυτά τα υποσύνολα διαστάσεων μπορεί να διαφέρουν ανάλογα με τις διάφορες συστάδες. Τέτοιες συστάδες αναφέρονται ως προβλεπόμενες (Aggarwal et al., 1999)

4.3. Ταξινόμηση ροών δεδομένων με προσέγγιση μικρο-συστάδων

Ένα σημαντικό πρόβλημα εξαγωγής δεδομένων που μελετήθηκε στο πλαίσιο των ροών δεδομένων είναι αυτό της ταξινόμησης ροής (Duda & Hart, 1973). Η κύρια ώθηση στην εξαγωγής ροής δεδομένων στο πλαίσιο της ταξινόμησης ήταν αυτή της εξαγωγής μονού περάσματος (Hulten et al., 2001). Σε γενικές γραμμές, η χρήση εξαγωγής μονού περάσματος δεν αναγνωρίζει τις αλλαγές που έχουν συμβεί στο μοντέλο από την αρχή της διαδικασίας δημιουργίας ροής (Aggarwal, 2003). Ενώ η εργασία των (Hulten et al., 2001) λειτουργεί με την αλλαγή των ροών δεδομένων στο χρόνο, το επίκεντρο έγγυται στην παροχή αποτελεσματικών μεθόδων για τη σταδιακή ενημέρωση του μοντέλου ταξινόμησης. Σημειώνεται ότι η ακρίβεια ενός τέτοιου μοντέλου δεν μπορεί να είναι μεγαλύτερη από το καλύτερο μοντέλο ολισθαίνοντος παραθύρου σε μια ροή δεδομένων. Όπως τελικώς δείχνουν τα εμπειρικά αποτελέσματα, η πραγματική συμπεριφορά της ροής δεδομένων καταγράφεται σε ένα χρονικό μοντέλο που είναι ευαίσθητο στο επίπεδο εξέλιξης της ροής δεδομένων.

Η διαδικασία ταξινόμησης μπορεί να απαιτεί ταυτόχρονη κατασκευή και δοκιμή μοντέλου σε περιβάλλον που εξελίσσεται διαρκώς με την πάροδο του χρόνου. Έστω ότι η διαδικασία δοκιμής εκτελείται ταυτόχρονα με τη διαδικασία εκπαίδευσης. Αυτό συμβαίνει συχνά σε πολλές πρακτικές εφαρμογές, στις οποίες επισημαίνεται μόνο ένα μέρος των δεδομένων, ενώ το υπόλοιπο όχι. Επομένως, τέτοια δεδομένα μπορούν να

διαχωριστούν στη ροή εκπαίδευσης (με ετικέτα) και στη ροή δοκιμών (χωρίς ετικέτα). Η κύρια διαφορά στην κατασκευή των μικρο-συστάδων είναι ότι οι μικρο-συστάδες συνδέονται με μια ετικέτα τάξης. Επομένως, ένα εισερχόμενο σημείο δεδομένων στη ροή εκπαίδευσης μπορεί να προστεθεί μόνο σε μία μικρο-συστάδα που ανήκει στην ίδια τάξη. Συνεπώς, οι μικρο-συστάδες κατασκευάζονται σχεδόν με τον ίδιο τρόπο όπως ο αλγόριθμος χωρίς επίβλεψη, με επιπλέον περιορισμό την ετικέτας κατηγορίας.

Από τη σκοπιά των δοκιμών, το σημαντικό σημείο που πρέπει να σημειωθεί είναι ότι το πιο αποτελεσματικό μοντέλο ταξινόμησης δεν παραμένει σταθερό με την πάροδο του χρόνου, αλλά ποικίλλει ανάλογα με την εξέλιξη της ροής δεδομένων. Εάν ένα στατικό μοντέλο ταξινόμησης χρησιμοποιήθηκε για μια εξελισσόμενη ροή δοκιμών, η ακρίβεια της υποκείμενης διαδικασίας ταξινόμησης είναι πιθανό να μειωθεί ξαφνικά όταν υπάρχει μια ξαφνική έκρηξη εγγραφών που ανήκουν σε μια συγκεκριμένη κατηγορία. Σε μια τέτοια περίπτωση, ένα μοντέλο ταξινόμησης που κατασκευάζεται χρησιμοποιώντας μικρότερο ιστορικό δεδομένων είναι πιθανό να παρέχει καλύτερη ακρίβεια. Σε άλλες περιπτώσεις, ένα μεγαλύτερο ιστορικό εκπαίδευσης παρέχει μεγαλύτερη αντοχή.

Στη διαδικασία ταξινόμησης μιας εξελισσόμενης ροής δεδομένων, είτε η βραχυπρόθεσμη είτε η μακροπρόθεσμη συμπεριφορά της ροής μπορεί να είναι σημαντική, συχνά όμως δεν μπορεί να είναι γνωστό εκ των προτέρων το ποια θα παίξει αυτό το ρόλο.

Το πώς λαμβάνεται η απόφαση να χρησιμοποιηθεί ένα παράθυρο ή ο ορίζοντας των εκπαιδευτικών δεδομένων ώστε να έχουμε την καλύτερη ακρίβεια ταξινόμησης αποτελεί ένα σημαντικό ερώτημα. Ενώ τεχνικές όπως τα δέντρα αποφάσεων είναι χρήσιμα για την εξαγωγή ροής δεδομένων με μονό περάσμα (Hulten et al., 2001), αυτές δεν μπορούν εύκολα να χρησιμοποιηθούν στο πλαίσιο ενός ταξινομητή κατ 'απαίτηση σε ένα εξελισσόμενο περιβάλλον, ένας τέτοιος ταξινομητής απαιτεί ταχεία μεταβολή στη διαδικασία επιλογής ορίζοντα λόγω της εξέλιξης της ροής δεδομένων. Επιπλέον, είναι πολύ ακριβό να παρακολουθείται ολόκληρο το ιστορικό των δεδομένων στην αρχική του λεπτομέρεια. Επομένως, η διαδικασία ταξινόμησης κατ 'απαίτηση εξακολουθεί να απαιτεί τον κατάλληλο μηχανισμό για αποτελεσματική

συλλογή στατιστικών δεδομένων προκειμένου να εκτελεστεί η διαδικασία ταξινόμησης.

5. Αλγόριθμοι για Προβλήματα Μαζικών Δεδομένων - Δειγματοληψία Streaming Sketching

5.1.Αλγοριθμικές Τεχνικές Επεξεργασίας Ροών δεδομένων

Γενικά, η δειγματοληψία υποδηλώνει μια διαδικασία βασισμένη στην επιλογή μικρότερου αριθμού στοιχείων από μια μεγαλύτερη ομάδα. Μία τέτοια προσέγγιση μπορεί να είναι χρήσιμη στο περιβάλλον ροής. Έστω το μοντέλο cash register. Η προσέγγιση δειγματοληψίας εφαρμόζεται ομαλά: Από τη μεγάλη ομάδα όλων των στοιχείων $a_1, a_2, \ldots, a_m$ στη ροή εισόδου, ο αλγόριθμος επιλέγει μια ομάδα μεγέθους μικρότερη, στις περισσότερες περιπτώσεις πολύ μικρότερη, από το m που θα διατηρηθεί στη μνήμη για να καταναλώσει ένα χωρικό υπογραμμικό σε m. Η ιδέα είναι ότι στο τέλος της ροής ή όποτε ερωτάται ο αλγόριθμος, χρησιμοποιούνται τα απομνημονευμένα στοιχεία, δηλαδή το δείγμα, για να αποκτηθούν πληροφορίες για ολόκληρη τη ροή. Φυσικά, η ακρίβεια αυτών των πληροφοριών εξαρτάται σε μεγάλο βαθμό από το πόσο καλά το δείγμα αντιπροσωπεύει ολόκληρη τη ροή. Η σχεδίαση ενός χαρακτηριστικού δείγματος είναι η πρόκληση για οποιαδήποτε προσέγγιση δειγματοληψίας.

Αν και υπάρχουν ορισμένες ντετερμινιστικές μέθοδοι δειγματοληψίας στον τομέα των αλγορίθμων ροής, π.χ. (Greenwald & Khanna, 2001; Shrivastava et al., 2004), το κυρίαρχο μέρος των προσεγγίσεων δειγματοληψίας σε αυτόν τον τομέα είναι τυχαιοποιημένο.

5.1.1. Δειγματοληψία δεξαμενών

Έστω ότι πρέπει να γίνει δειγματοληψία από τη ροή εισόδου $a_1, a_2, \ldots, a_m$ ενός ενιαίου στοιχείου s τυχαία, δηλαδή, με τέτοιο τρόπο ώστε η πιθανότητα δειγματοληψίας να είναι η ίδια για κάθε στοιχείο εισαγωγής. Ως εκ τούτου, απαιτείται το $P_r[a_l$ είναι το δείγμα $s]=1/m$ για $1 \leq l \leq m$.

Είναι σημαντικό να σημειωθεί ότι σχεδιάζεται ένα ομοιόμορφο δείγμα όσον αφορά όλα τα στοιχεία εισαγωγής και όχι σε σχέση με τα στοιχεία του universe U. Ως εκ τούτου, για τους σκοπούς της δειγματοληψίας, διακρίνεται η διαφορά μεταξύ δύο

στοιχείων εισαγωγής a_i και a_j εφόσον το $i \neq j$, ακόμη και αν οι a_i και a_j δηλώνουν το ίδιο στοιχείο του U.

Για παράδειγμα, από τη ροή εισόδου 2, 1, 2, 5, πρέπει να επιλεγεί ένα από τα τέσσερα στοιχεία εισαγωγής ομοιόμορφα τυχαία, δηλαδή, το καθένα με πιθανότητα 1/4 χωρίς να ενδιαφέρει ότι δύο από αυτά τα στοιχεία αντιπροσωπεύουν το ίδιο στοιχείο του U. Φυσικά, το στοιχείο 2 του U επιλέγεται ως δείγμα με πιθανότητα 2/4 λόγω των δύο αντίστοιχων στοιχείων εισαγωγής a_1 και a_3, ενώ τα στοιχεία 1 και 5 αποτελεί το καθένα δείγμα με πιθανότητα 1/4. Πράγματι, αυτό επιδιώκεται δεδομένου ότι το στοιχείο 2 εμφανίζεται δύο φορές συχνότερα από κάθε ένα από τα 1 και 5.

Η συχνότητα εμφάνισης ενός στοιχείου επηρεάζει αναλογικά την πιθανότητα να είναι δείγμα. Επομένως, ο σχεδιασμός και η εξέταση δειγμάτων καθιστά δυνατή την εξαγωγή πληροφοριών σχετικά με την κατανομή συχνότητας της ροής εισόδου..

Η δειγματοληψία ενός στοιχείου εισαγωγής ομοιόμορφα τυχαία από τη ροή (το μήκος m της ροής είναι γνωστό εκ των προτέρων) αποτελεί μια πολύ απλή εργασία: Πριν ακόμη διαβαστεί η ροή, ο αλγόριθμος επιλέγει έναν αριθμό $l \epsilon \{1, 2, \ldots, m\}$ ομοιόμορφα τυχαία. Τότε διαβάζει τη ροή μέχρι το στοιχείο a_l που επιλέχθηκε ως δείγμα. Για την κατανάλωση χώρου, ο αλγόριθμος πρέπει να δημιουργήσει l και να απομνημονεύσει l και a_l, ενώ απαιτεί να μετράει τον αριθμό των στοιχείων ροής έως l. Δεδομένου ότι μια μνήμη του $O(\log m+\log n)$ είναι ερκετή για να το υλοποιήσει και η ροή έχει πρόσβαση διαδοχικά, αυτή η μέθοδος περιγράφει στην πραγματικότητα έναν αλγόριθμο ροής χρησιμοποιώντας ένα πέρασμα.

Ωστόσο, η προηγούμενη γνώση του m είναι μια αρκετά ρεαλιστική υπόθεση. Αντίθετα, στα περισσότερα σενάρια ροής το μήκος της ροής είναι άγνωστο εκ των προτέρων ή - ακόμη χειρότερα - δεν υπάρχει προκαθορισμένο τέλος της ροής. Τέτοιες συνεχείς ροές μπορούν να προκύψουν από διαρκείς ενημερώσεις αισθητήρων. Η απρόβλεπτη στιγμή ενός ερωτήματος σηματοδοτεί το τέλος μιας ροής στην οποία το ερώτημα πρέπει να αξιολογηθεί.

Μπορεί να αποτελεί έκπληξη το γεγονός είναι δυνατός ο σχεδιασμός ενός ομοιόμορφου τυχαίου δείγματος χωρίς να είναι γνωστό το μήκος της ροής. Αυτή η

προσέγγιση ονομάζεται δεξαμενή δειγματοληψίας αναφέρθηκε στην εργασία του (Vitter, 1985). Για κάθε θέση l στη ροή $a_1, a_2, \ldots, a_m$, διατηρεί ένα στοιχείο s που είναι ένα ομοιόμορφο τυχαίο δείγμα για όλα τα στοιχεία a_i, $i \leq l$, δηλαδή, πάνω από όλα τα στοιχεία της ροής έως το a_i. Στο τέλος της ροής, το s είναι το τελικό δείγμα που αντλείται από ολόκληρη τη ροή εισόδου.

Ο αλγόριθμος ξεκινά ορίζοντας το a_1 ως s. Στο επόμενο βήμα, επιλέγεται το a_2 για να αντικαταστήσει το a_1 ως δείγμα με πιθανότητα 1/2. Στη συνέχεια, το a_3 επιλέγεται ως s με πιθανότητα 1/3. Γενικά, για το $i \geq 2$, το a_i απομνημονεύεται ως s - και ως εκ τούτου αντικαθιστά το προηγουμένως αποθηκευμένο αντικείμενο - με πιθανότητα 1/i.

Ο αλγόριθμος δειγματοληψίας δεξαμενής για μια ροή τεσσάρων στοιχείων εισαγωγής αποτυπώνεται ως δέντρο αποφάσεων. Κάθε κόμβος χωρίς φύλλα αντιστοιχεί σε μια τυχαία απόφαση του αλγορίθμου για το εάν ή όχι θα αντικαταστήσει το πραγματικό δείγμα s από το τρέχον στοιχείο εισαγωγής. Κάθε κόμβος επισημαίνεται από την πιθανότητα της προηγούμενης απόφασης. Ως εκ τούτου, το προϊόν όλων των ετικετών κατά μήκος μιας διαδρομής από τη ρίζα προς ένα φύλλο δίνει την πιθανότητα για τη συγκεκριμένη ακολουθία αποφάσεων που αντιστοιχούν σε αυτήν τη διαδρομή.

Ως παράδειγμα υπολογίζεται η πιθανότητα επιλογής του δεύτερου στοιχείου εισαγωγής a_2 ως το τελικό δείγμα από τη ροή a_1, a_2, a_3, a_4. Μετά την ανάγνωση του πρώτου στοιχείου εισαγωγής, ο αλγόριθμος επέλεξε το a_1 ως το πραγματικό δείγμα. Κατά την ανάγνωση του a_2 στο επόμενο βήμα, ο αλγόριθμος επιλέγει το a_2 ως s με πιθανότητα 1/2. Για να καταλήξει στο a_2 ως τελικό δείγμα, ο αλγόριθμος πρέπει να αποφασίσει να μην επιλέξει ούτε το a_3 ούτε το a_4 ως πραγματικό δείγμα στα επόμενα δύο βήματα. Το στοιχείο a_3 δεν επιλέγεται με πιθανότητα (1–1/3) = 2/3. Το a_4 δεν επιλέγεται με πιθανότητα (1–1/4) = 3/4. Το στοιχείο a_2 καταλήγει ως το τελικό δείγμα εάν και μόνο εάν συμβούν όλα τα αναφερόμενα συμβάντα που συμβαίνουν με πιθανότητα $\frac{1}{2}\frac{2}{3}\frac{3}{4} = \frac{1}{4}$. Ομοίως, μπορεί κανείς να υπολογίσει την ίδια πιθανότητα για την επιλογή του άλλου στοιχείου εισαγωγής ως τελικού δείγματος.

Μια άλλη άποψη για το προαναφερθέν παράδειγμα είναι η ακόλουθη. Η πιθανότητα που υπολογίστηκε μόλις για την επιλογή a_2 ως τελικού δείγματος s αντιστοιχεί στην ακολουθία αποφάσεων "ναι", "ναι", "όχι", "όχι" και τοιουτοτρόπως στη διαδρομή από τη ρίζα προς το φύλλο. Ωστόσο, ενώ τα a_1 και a_2 το καθένα έχουν μόνο ένα αντίστοιχο φύλλο, τα a_3 και a_4 συσχετίζονται με διάφορα φύλλα, δηλαδή ακολουθίες αποφάσεων. Φυσικά, η πιθανότητα να καταλήξει ο αλγόριθμος σε αντικείμενα όπως το τελικό s είναι το άθροισμα των πιθανοτήτων των αντίστοιχων ακολουθιών. Τελικά, η πιθανότητα για καθένα από τα τέσσερα στοιχεία εισαγωγής είναι 1/4.

Ο αλγόριθμος δειγματοληψίας δεξαμενών που περιγράφεται είναι σίγουρα ένας αλγόριθμος ροής καθώς διαβάζει διαδοχικά τη ροή εισόδου και απαιτεί την απομνημόνευση ενός μόνο στοιχείου. Για να επαληθευτεί ότι η επιλογή για το s στο τέλος της ροής αποδίδει ένα ομοιόμορφο τυχαίο δείγμα, εξετάζεται η πιθανότητα μερικά a_l , με $1 \leq l \leq m$, είναι τα τελικά s. Αυτό συμβαίνει εάν το a_l έχει επιλεγεί ως το πραγματικό s και επιπλέον κανένα από α_{l+1}, α_{l+2},…α_m αντικαθιστά το a_l ως το πραγματικό δείγμα. Για αυτήν την πιθανότητα, προκύπτει:

$P_r[a_l$είναι το τελικό s$] = P_r[a_l$ επιλέχθηκε ως το τελικό s$]\cdot\prod_{i=l+1}^{m} P_r[a_i \delta\varepsilon\nu\ \varepsilon\nu\tau\iota\kappa\alpha\theta\iota\sigma\tau\acute{\alpha}\ \alpha_l\ \omega\varsigma\ s] = \frac{1}{l}\cdot\prod_{i=l+1}^{m}(1-\frac{1}{i}) = \frac{1}{l}\cdot$

$\prod_{i=l+1}^{m}\frac{i-1}{i} = \frac{1}{m}$

που σημαίνει ότι οποιοδήποτε στοιχείο α_l καταλήγει ως το τελικό δείγμα με την ίδια πιθανότητα.

Έστω μία κατάσταση κατά την οποία είναι επιθυμητό ένα δείγμα από μία ροή που περιλαμβάνει περισσότερα από ένα στοιχεία. Μία φυσική προσέγγιση είναι να εκτελεστούν k παράλληλες στιγμές της περιγραφόμενης διαδικασίας για να ληφθεί ένα τυχαίο δείγμα που περιέχει k στοιχεία. Εφόσον το k είναι υπογραμμικό σε m, η αποθήκευση που απαιτείται για αυτήν τη μέθοδο είναι επίσης υπογραμμική σε m. Ωστόσο, είναι σημαντικό να σημειωθεί ότι μια τέτοια διαδικασία οδηγεί σε τυχαία δειγματοληψία με αντικατάσταση, όπου κάθε στοιχείο του δείγματος επιλέγεται από

ολόκληρη τη ροή των m αντικειμένων. Ως εκ τούτου, ένα στοιχείο από τη ροή μπορεί να επιλεγεί περισσότερες από μία φορές στο δείγμα

5.2. Μαζική Ομαδοποίηση Δεδομένων που Βασίζεται σε Σχέδια

Παρουσιάζεται στο DBMSTClu μια νέα μη παραμετρική μέθοδο βασισμένη σε πυκνότητα που λειτουργεί σε έναν περιορισμένο αριθμό γραμμικών μετρήσεων, δηλαδή μια σκιαγραφημένη έκδοση του γραφήματος ομοιότητας G μεταξύ των αντικειμένων N στο σύμπλεγμα. Σε αντίθεση με τους αλγόριθμους k-mean, k-medians ή k-medoids, η μέθοδος δεν αποτυγχάνει να διακρίνει συστάδες με συγκεκριμένες δομές. Δεν απαιτείται παράμετρος εισαγωγής σε αντίθεση με το DBSCAN ή τη μέθοδο Spectral Clustering. Το DBMSTClu ως τεχνική που βασίζεται σε γραφήματα βασίζεται στο γράφημα ομοιότητας G που κοστίζει θεωρητικά το $O(N^2)$ στη μνήμη. Ωστόσο, ο αλγόριθμός ακολουθεί το δυναμικό μοντέλο ημι-ροής με το χειρισμό του G ως ένα ρεύμα ενημερώσεων βάρους άκρης και το σκιαγραφεί με τη μία μετάδοση των δεδομένων σε μια συμπαγή δομή που απαιτεί χώρο O(poly log (N)). Χάρη στην ιδιότητα του Δένδρου ελαχίστων συνδέσεων (Minimum Spanning Tree-MST) για την έκφραση της υποκείμενης δομής ενός γραφήματος, ο αλγόριθμος ανιχνεύει με επιτυχία τον σωστό αριθμό μη κυρτών συστάδων, ανακτώντας ένα κατά προσέγγιση MST από το σχέδιο γραφήματος του G. Παρέχονται θεωρητικές εγγυήσεις για την ποιότητα του διαμερίσματος συμπλέγματος και επίσης αποδεικνύεται το πλεονέκτημά του έναντι του υπάρχοντος state-of-the-art σε πολλά σύνολα δεδομένων.

Έστω ένα σύνολο δεδομένων με σημεία N. Είτε υπάρχει ήδη το υποκείμενο δίκτυο, είτε θεωρείται ότι μπορεί να δημιουργηθεί ένα γράφημα ομοιότητας G μεταξύ σημείων όπου σημεία του συνόλου δεδομένων είναι οι κορυφές ενώ οι σταθμισμένες άκρες εκφράζουν αποστάσεις μεταξύ αυτών των αντικειμένων. Μπορεί π.χ. να πρόκειται για την Ευκλείδεια απόσταση. Και στις δύο περιπτώσεις, το γράφημα στο οποίο εφαρμόζεται το DBMSTClu πρέπει να ακολουθεί αυτόν τον ορισμό:

Ορισμός (γράφημα G = (V, E)): Ένα γράφημα G = (V, E) αποτελείται από ένα σύνολο κορυφών ή κόμβων V και ένα σύνολο ακμών $E \subseteq V \times V$. Δεν αποδίδονται χαρακτηριστικά σε κόμβους ή άκρα. Το γράφημα είναι μη κατευθυνόμενο αλλά

σταθμισμένο. Το βάρος w σε μια άκρη μεταξύ κόμβου i και j - εάν υπάρχει αυτό το άκρο - αντιστοιχεί στην κανονικοποιημένη προκαθορισμένη απόσταση μεταξύ i και j, με $0<w\leq 1$.

Στο υπόλοιπο μέρος της μελέτης δηλώνεται ότι [N]={1, ..., N}. |V| και |E| αντιπροσωπεύουν αντίστοιχα την cardinality των συνόλων V και E. Εν ολίγοις, |V| =N και |E| =M. Ένα πυκνό γράφημα, $M=N(N-1)/2$. $E=\{e_1, ..., e_M\}$ και για όλες τις άκρες e_i έχει ένα βάρος w_i που αντιστοιχεί σε απόσταση μεταξύ δύο κορυφών. Το E (G) χρησιμοποιείται για να περιγράψει το σύνολο των άκρων ενός γραφήματος G.

Ο αλγόριθμος ομαδοποίησης ροής χωρίζεται σε δύο βασικά βήματα: 1) Από τη ροή των βαρών άκρων, δημιουργείται ένα σχέδιο του γραφήματος και στη συνέχεια υπολογίζεται ένα κατά προσέγγιση MST χωρίς περισσότερες πληροφορίες από το σχέδιο που αποκτήθηκε προηγουμένως. 2) Η φάση ομαδοποίησης κόμβων πραγματοποιείται από το κατά προσέγγιση MST χωρίς να απαιτείται καμία παράμετρος.

Η επεξεργασία δεδομένων στο δυναμικό μοντέλο ροής (Muthukrishnan, 2005) συνεπάγεται τα ακόλουθα: 1) Το γράφημα πρέπει να αντιμετωπίζεται ως ροή s ενημερώσεων βάρους άκρου: $s = (a_1, ... a_j, ...)$ όπου το a_j είναι το j -οστή ενημέρωση στη ροή που αντιστοιχεί στην πλειάδα $a_j = (i, w_{old}, i, \Delta wi)$ με i να δηλώνει τον δείκτη της άκρης για ενημέρωση, w_{old}, το προηγούμενο βάρος του και Δw_i την ενημέρωση που θα εκτελεστεί. Κατά συνέπεια, μετά την ανάγνωση a_j στη ροή, στο i-ιοστό άκρο αντιστοιχεί το νέο βάρος $w_i = w_{old,\, i} + \Delta w_i$. 2) Ο αλγόριθμος πρέπει να κάνει μόνο ένα πέρασμα σε αυτήν τη ροή - ή το πολύ λίγα περάσματα. 3) Τα άκρα μπορούν και τα δύο να εισαχθούν ή να διαγραφούν (μοντέλο περιστροφικής πύλης). Έτσι, τα βάρη αλλάζουν τακτικά, όπως στα κοινωνικά δίκτυα όπου τα άτομα μπορούν να είναι φίλοι για κάποιο διάστημα.

Καθώς μια νέα ενημέρωση εμφανίζεται μόνο μία φορά στη ροή των ενημερώσεων, χρησιμοποιούνται σχέδια γραφημάτων από τους (Ahn et al.,2012a) οι οποίοι στηρίχθηκαν στην αρχή της «l_0-δειγματοληψίας» (Cormode & Firmani, 2014) για την παραγωγή περιορισμένου αριθμού γραμμικών μετρήσεων του γραφήματος. Η

βασική ιδέα είναι η συγκέντρωση πληροφοριών σχετικά με τα βάρη των άκρων μέσω λίγων γραμμικών μετρήσεων. Αυτά ενημερώνονται δυναμικά καθώς τα νέα δεδομένα a_j από τη ροή s εμφανίζονται μόνο μία φορά. Αυτή η συμπαγής δομή δεδομένων επιτρέπει το σχεδιασμό ενός σχεδόν ομοιόμορφου τυχαίου μη μηδενικού σταθμισμένου άκρου (εν συντομία, ένα μη μηδενικό άκρο) ανά πάσα στιγμή χάρη στη «l_0-δειγματοληψία» (Cormode & Firmani, 2014):

Ορισμός 3. 2 («l_0-sampling) Ένα (ε,δ) l_p-sampler για ένα μη μηδενικό διάνυσμα $x \in R^n$ αποτυγχάνει με πιθανότητα το πολύ δ ή επιστρέφει κάποια $i \in [n]$ με πιθανότητα

$$(1 \pm \varepsilon)\frac{|x_i|^p}{\|x\|_P^P}$$

Όπου $\|x\|_P^P = (\sum_{i\in[n]}|x_i|^P)^{1/p}$ είναι η p-νορμα του x. Συγκεκριμένα, εάν p = 0, σε περίπτωση μη αποτυχίας, επιστρέφει κάποια i με πιθανότητα

$$(1 \pm \varepsilon)\frac{1}{|suppx|}$$

Όπου suppx={i∈[n]‖ $x_i \neq 0$}

Το σχέδιο απαιτεί χώρο O(polylog(N)). Ο αλγόριθμός μας είναι ημι-ροή, αλλά στην πράξη το κόστος χώρου είναι σημαντικά χαμηλότερο από το θεωρητικό O (N^2) όριο και ο αλγόριθμος χρειάζεται μόνο μία μετάδοση των δεδομένων. Αυτό το σχέδιο είναι το πρώτο που υποστηρίζει τόσο την εισαγωγή όσο και τη διαγραφή των άκρων. Σε αυτό το πλαίσιο, ο αριθμός των κόμβων είναι γνωστός, ενώ τα άκρα, των οποίων τα βάρη μπορούν να αυξηθούν ή να μειωθούν (αλλά πρέπει πάντα να παραμείνουν θετικά) συνοψίζονται σε αυτά τα σχέδια.

5.3.Αλγόριθμος Ανάκτησης για Μεγάλα και Υψηλής Διαστάσεων Δεδομένα

Παρέχεται μια ενοποιημένη προβολή βελτιστοποίησης του επαναληπτικού Hessian σχεδίου (iterative Hessian sketch-IHS) και της επαναληπτικής διπλής τυχαίας προβολής (iterative dual random projection-IDRP). Καθιερώνεται μια πρωταρχική-διπλή σύνδεση μεταξύ του διαγράμματος Hessian και της διπλής τυχαίας προβολής και αποδεικνύεται ότι οι επαναληπτικές επεκτάσεις τους είναι διαδικασίες

βελτιστοποίησης με προετοιμασία. Αναπτύσσονται επιταχυνόμενες εκδόσεις του IHS και του IDRP με βάση αυτήν την εικόνα μαζί με την σύζευξη κλίσης καθόδου και προτείνεται μια μέθοδος διαγράμματος primal-dual που ταυτόχρονα μειώνει το μέγεθος και τη διάσταση του δείγματος.

Η πρόσφατη εργασία για το «επαναληπτικό διάγραμμα» Hessian (IHS) των (Pilanci & Wainwright, 2016) και η επαναληπτική διπλή τυχαία προβολή (IDRP) (Zhang et al., 2014), βελτίωσε την κατάσταση. Αυτές οι μέθοδοι είναι σε θέση να βελτιώσουν την ακρίβεια της λύσης τους με την επαναληπτική επίλυση ενός μικρού μεγέθους προβλήματος. Το Hessian διάγραμμα (Pilanci & Wainwright, 2016) έχει σχεδιαστεί για να μειώσει το μέγεθος του δείγματος του αρχικού προβλήματος, ενώ η διπλή τυχαία προβολή (Zhang et al., 2014) προτείνεται για τη μείωση της διαστατικότητας των δεδομένων. Κατά συνέπεια, όταν το μέγεθος του δείγματος και η διάσταση των χαρακτηριστικών είναι και τα δύο μεγάλα, το IHS και το IDRP πρέπει ακόμη να επιλύσουν σχετικά προβλήματα μεγάλης κλίμακας καθώς μπορούν να σκιαγραφήσουν το πρόβλημα μόνο από μία οπτική γωνία. Στην εργασία τους οι (Wang et al., 20117), αντιμετώπισαν το πρόβλημα της ανάκτησης της βέλτιστης λύσης για μεγάλα και πολυδιάστατα δεδομένα, επιλύοντας μικρά σκιαγραφημένα προβλήματα του αρχικού προβλήματος. Ακολουθούνται οι ακόλουθες παραδοχές: **Πρώτον**, προτάθηκε μια επιταχυνόμενη έκδοση του IHS που είναι υπολογιστικά εξίσου αποτελεσματική με το IHS σε κάθε επανάληψη, αλλά απαιτεί προφανώς λιγότερους αριθμούς επαναλαμβανόμενων διαγραμμάτων για την επίτευξη συγκεκριμένης ακρίβειας. Στη συνέχεια, αποκαλύφθηκε μια primal dual σύνδεση μεταξύ IHS και IDRP, που προτάθηκαν ανεξάρτητα. Φαίνεται ότι αυτές οι δύο μέθοδοι είναι ισοδύναμες με την έννοια ότι η διπλή τυχαία προβολή εκτελεί ουσιαστικά το Hessian διάγραμμα στο διπλό χώρο. Αυτή η σύνδεση επιτρέπει την υλοποίηση μιας ενοποιημένης ανάλυσης των IHS και IDRP, και ταυτόχρονα αναπτύσσει μια επιταχυνόμενη σχεδίαση. Τέλος, ανακουφίζονται τα υπολογιστικά ζητήματα που εγείρονται από μεγάλα και υψηλής διάστασης μαθησιακά προβλήματα. Προτάθηκε μια primal dual μέθοδο σχεδίασης που μπορεί ταυτόχρονα να μειώσει το μέγεθος του δείγματος και τη διάσταση του προβλήματος και να ανακτήσει τη βέλτιστη λύση στο

αρχικό μεγάλης κλίμακας πρόβλημα υψηλής διαστάσεων με αποδεδειγμένες εγγυήσεις σύγκλισης.

Ο επιταχυνόμενος επαναληπτικός αλγόριθμος Hessian διαγράμματος (accelerated iterative Hessian sketch-Acc-IHS) χρησιμοποιεί την ιδέα της προκαθορισμένης κλίσης σύζευξης. Η κλίση σύζευξης είναι γνωστό ότι έχει καλύτερες ιδιότητες σύγκλισης από την καθοδική κλίση στην επίλυση γραμμικών συστημάτων (Lu et al., 2013; Johnson & Zhang, 2013). Δεδομένου ότι το επαναληπτικό Hessian σχέδιο εκτελεί την καθοδική κλίση στον μετασχηματισμένο χώρο

$z = \widehat{\boldsymbol{H}^{1/2}}w$, μπορεί να επιταχυνθεί εκτελώντας τη συζευγμένη καθοδική κλίση. Ομοίως, είναι δυνατή η έμμεση μεταμόρφωση του χώρο ορίζοντας το εσωτερικό γινόμενο ως $< x, y >= x^T \tilde{\boldsymbol{H}} y$

Σε κάθε επανάληψη, ο επιλύτης καλείται για το ακόλουθο γραμμικό σύστημα:

$$\min_{\mathbf{u}} \mathbf{u}^\top \left(\frac{\mathbf{X}^\top \boldsymbol{\Pi}\boldsymbol{\Pi}^\top \mathbf{X}}{2n} + \frac{\lambda}{2}\mathbf{I}_p \right) \mathbf{u} - \left\langle \mathbf{r}^{(t)}, \mathbf{u} \right\rangle$$

Σε αντίθεση με το IHS, το οποίο χρησιμοποιεί $\widetilde{\boldsymbol{H}^{-1}\nabla P(\boldsymbol{w}_{HS}^{(t)}}$ ως κατεύθυνση ενημέρωσης κατά την επανάληψη t, το Acc-IHS χρησιμοποιεί το $\mathbf{p}^{(t)}$ ως κατεύθυνση ενημέρωσης όπου επιλέγεται το $\mathbf{p}^{(t)}$ για να ικανοποιήσει την κατάσταση σύζευξης $\forall\, t_1, t_2 \geq 0, t_1 \neq t_2$

$$\left(\mathbf{p}^{(t_1)}\right)^\top \tilde{\mathbf{H}}^{-1/2}\mathbf{H}\tilde{\mathbf{H}}^{-1/2}\mathbf{p}^{(t_2)} = 0.$$

Δεδομένου ότι η κατεύθυνση ενημέρωσης είναι συζευγμένη με τις προηγούμενες κατευθύνσεις, είναι εγγυημένο ότι μετά από p επαναλήψεις ο αλγόριθμος φθάνει στον ακριβή ελαχιστοποιητή, δηλαδή:

$$\widehat{\boldsymbol{w}}_{HS}^{(t)} = \boldsymbol{w}^*$$

Επιπλέον, το Acc-IHS έχει το ίδιο υπολογιστικό κόστος με το τυπικό IHS στην επίλυση κάθε σκιαγραφημένου υποπροβλήματος. Ωστόσο, το ποσοστό σύγκλισης του Αλγόριθμου 1 είναι πολύ ταχύτερο από το IHS, δηλαδή απαιτεί την επίλυση πολύ

μικρότερου αριθμού σχεδιασμένων υπο-προβλημάτων σε σύγκριση με το IHS για την επίτευξη της ίδιας ακρίβειας προσέγγισης.

5.4.Αλγόριθμοι για Μεθόδους Υπολογισμού Δεδομένων Streaming

5.4.1. Αλγόριθμοι και Διαγράμματα Γραφήματος Ροής

Τα γραφήματα μεγάλης κλίμακας είναι πλέον ένα ευρέως χρησιμοποιούμενο εργαλείο για την αναπαράσταση δεδομένων πραγματικού κόσμου. Πολλές σύγχρονες εφαρμογές, όπως μηχανές αναζήτησης ή κοινωνικά δίκτυα, απαιτούν αποτελεσματική υποστήριξη διαφόρων ερωτημάτων σε γραφήματα μεγάλης κλίμακας. Είναι δυνατή η αποθήκευση ενός τεράστιου γραφήματος σε μια μεγάλη συσκευή αποθήκευσης, αλλά η τυχαία πρόσβαση σε αυτές τις συσκευές είναι συχνά αρκετά αργή και το κόστος υπολογισμού θα είναι ακριβό.

Για να ξεπεραστούν οι προκλήσεις που προκύπτουν από τον υπολογισμό σε τεράστια γραφήματα, μια σημαντική προσέγγιση είναι η διατήρηση μιας συνοπτικής αναπαράστασης που διατηρεί ορισμένες ιδιότητες του γραφήματος (δηλαδή, coreset). Μια άλλη δημοφιλής προσέγγιση είναι η επεξεργασία τόσο μεγάλων γραφημάτων στο μοντέλο ροής δεδομένων χρησιμοποιώντας περιορισμένο χώρο. Το ζητούμενο εδώ είναι να δημιουργηθεί μια δομή δεδομένων σύνοψης η δημιουργία της οποίας είναι εύκολη με τη χρήση ροής μόδας, ενώ παράλληλα να αποδίδεται καλή προσέγγιση των ιδιοτήτων των δεδομένων του γραφήματος. Πολλές από αυτές τις συνόψεις υλοποιούνται χρησιμοποιώντας διαγράμματα (υπολογισμός γραμμικής προβολής των δεδομένων) και τεχνικές δειγματοληψίας.

Οι τεχνικές που αναπτύχθηκαν για ροές γραφημάτων βρίσκουν πλέον εφαρμογή σε άλλους τομείς, συμπεριλαμβανομένων δομών δεδομένων για δυναμικά γραφήματα, αλγορίθμους προσέγγισης και κατανεμημένους και παράλληλους υπολογισμούς. Μια σημαντική έρευνα για αλγόριθμους ροής γραφημάτων και σχετικές τεχνικές λεπτομέρειες αναφέρθηκε στην εργασία του (McGregor, 2014) .

5.4.2. Αλγόριθμοι Επαλήθευσης Ροής

Μία από τις κύριες προκλήσεις στον υπολογισμό ροής σε τεράστια δεδομένα είναι ο σχεδιασμός αλγορίθμων για δυνητικά δύσκολα προβλήματα στα οποία οι

υπολογισμοί ή οι απαιτήσεις χώρου είναι απαγορευτικές βάσει του μοντέλου ροής. Σε αυτήν την περίπτωση, γίνεται ανάθεση της αποθήκευσης και της επεξεργασίας της ροής δεδομένων σε ένα πιο ισχυρό τρίτο μέρος, το cloud. Ωστόσο ο κάτοχος των δεδομένων θα ήθελε να είναι σίγουρος ότι ο επιθυμητός υπολογισμός έχει εκτελεστεί σωστά. Σε αυτήν τη ρύθμιση, ο ελεγκτής περιορισμένης χρήσης πόρων (κάτοχος δεδομένων) βλέπει μια ροή δεδομένων και προσπαθεί να λύσει το πρόβλημα με τη βοήθεια ενός πιο ισχυρού υποστηρικτή (cloud) που βλέπει την ίδια ροή. Αυτό το μοντέλο μπορεί να θεωρηθεί ως τροποποίηση ροής ενός κλασικού διαδραστικού συστήματος απόδειξης (streaming IP ή SIP) και έχει αποτελέσει αντικείμενο πολλών εργασιών (Thaler, 2014; Daruki et al., 2015; Chakrabarti et al., 2015) που έχουν καθιερώσει υπογραμμικά όρια χώρου και επικοινωνιών για κλασικά προβλήματα στη ροή. Εδώ ο στόχος είναι να αναπτυχθούν αποτελεσματικά (όσον αφορά την επικοινωνία και το χώρο) διαδραστικά πρωτόκολλα απόδειξης για την επαλήθευση υπολογισμών που ρέουν στη φύση και να διερευνηθεί πώς η αλληλεπίδραση και η επικοινωνία μπορούν να βοηθήσουν στην επίλυση προβλημάτων στην ανάλυση δεδομένων καθώς και σε κλασικά και θεμελιώδη προβλήματα στη γεωμετρική και σε συνδυαστικούς αλγόριθμους και βελτιστοποίηση κάτω από εισόδους ροής.

Στη διατριβή της η (Daruki, 2018), συνέβαλε στη μελέτη των υπογραμμικών αλγορίθμων για μαζικούς υπολογισμούς δεδομένων με τους ακόλουθους τρόπους.

5.4.3. Επαλήθευση ΡΟΩΝ ΓΙΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΓΡΑΦΗΜΑΤΩΝ

Όλες οι προηγούμενες εργασίες για την επαλήθευση γραφήματος ροής έγιναν στο μοντέλο σχολιασμού, το οποίο στην πράξη μοιάζει με μία αλληλεπιδραστική απόδειξη ροής (streaming interactive proofs-SIP) ενός κύκλου (ένα μόνο μήνυμα από τον πάροχο στον επαληθευτή μετά την ανάγνωση της ροής). Παρουσιάζονται αλληλεπιδραστικές αποδείξεις ροής (SIP) για προβλήματα γραφημάτων που παραδοσιακά είναι δύσκολα για ροή, όπως για το μέγιστο πρόβλημα αντιστοίχισης (σε διμερή και γενικά γραφήματα, τόσο σταθμισμένα όσο και μη σταθμισμένα), καθώς και για την προσέγγιση του προβλήματος πωλητή. Ενώ το μοντέλο ροής υπολογισμού ήταν εξαιρετικά αποτελεσματικό για την επεξεργασία αριθμητικών δεδομένων και δεδομένων μήτρας, η ικανότητά του να χειρίζεται μεγάλα γραφήματα είναι

περιορισμένη, ακόμη και στο λεγόμενο μοντέλο ημι-ροής όπου ο αλγόριθμος ροής επιτρέπεται να χρησιμοποιεί χωρικό τετράγωνο στον αριθμό κορυφών. Οι πρόσφατες ανακαλύψεις στη σχεδίαση γραφημάτων (McGregor, 2014) οδήγησαν σε εξοικονόμηση χώρου για πολλά προβλήματα στο μοντέλο ημι-ροής, αλλά τα κανονικά προβλήματα γραφήματος όπως οι αντιστοιχίσεις έχουν αποδειχθεί ότι είναι αρκετά δύσκολα.

Είναι γνωστό (Kapralov et al., 2014) ότι δεν υπάρχει καλύτερη προσέγγιση από 1 - 1 / e στη μέγιστη αντιστοίχιση cardinality στο μοντέλο ροής, ακόμη και με το διάστημα $\tilde{o}(n)$. Ακόμη και η περιορισμένη επικοινωνία (ουσιαστικά ένα μόνο μήνυμα από τον πάροχο) απαιτούσε ένα προϊόν χωρικής επικοινωνίας $\Omega(n^2)$ (Cormode et al., 2013; Chakrabarti et al., 2009). Τα αποτελέσματά δείχνουν ότι ακόμη και αν επιτραπούν μερικοί ακόμη κύκλοι επικοινωνίας, βελτιώνεται δραματικά η ανταλλαγή χωρικών επικοινωνιών για αντιστοίχιση, καθώς και για μεγαλύτερης ακριβείας επαλήθευση. Σημειώνεται ότι οι αλγόριθμοι ροής για αντιστοίχιση ποικίλλουν σε μεγάλο βαθμό στην απόδοση και την πολυπλοκότητα ανάλογα με το αν το γράφημα είναι σταθμισμένο ή μη, διμερές ή μη. Αντίθετα, τα αποτελέσματά ισχύουν για όλες τις μορφές αντιστοίχισης. Είναι ενδιαφέρον ότι η ειδική περίπτωση του τέλειου ταιριάσματος, λόγω του ότι βρίσκεται στο RNC (Karp et al., 1986), αναγνωρίζει ένα αποτελεσματικό SIP μέσω αποτελεσμάτων των Goldwasser, Kalai και Rothblum (Goldwasser et al., 2015) και (Cormode et al., 2011). Ομοίως για την καταμέτρηση τριγώνων, ο καλύτερος αλγόριθμος ροής (Ahn et al., 2012) αποδίδει μια πρόσθετη εκτίμηση σφάλματος εn^3 στον πολυλογαριθμικό χώρο, και πάλι στο μοντέλο σχολιασμών (ουσιαστικά ένας κύκλος επικοινωνίας) το καλύτερο αποτέλεσμα αποδίδει ένα προϊόν χωρικής επικοινωνίας $n^2 \log^2 n$, που είναι σχεδόν εκθετικά χειρότερο από το καθορισμένο όριο. Σημειώνεται ότι η μέτρηση των τριγώνων είναι ένα κλασικό πρόβλημα στη βιβλιογραφία των υπογραμμικών αλγορίθμων και ο προσδιορισμός του βέλτιστου ορίου χώρου και επικοινωνίας τέθηκε ως ανοιχτό πρόβλημα στο εργαστήριο υπογραμμικών αλγορίθμων. Το όριό για την επαλήθευση μιας προσέγγισης 3/2+ε για το TSP σε δυναμικά γραφήματα είναι επίσης ενδιαφέρον: μια ασήμαντη 2-προσέγγιση

στο μοντέλο ημι-ροής πραγματοοιείται μέσω του MST, αλλά είναι ανοιχτή για βελτίωση αυτού του ορίου (ακόμη και σε ένα πλέγμα).

Γενικά, τα αποτελέσματά μπορούν να θεωρηθούν ότι παρέχουν περαιτέρω πληροφορίες σχετικά με την ανταλλαγή μεταξύ χώρου και επικοινωνίας σε υπογραμμικούς αλγόριθμους. Το μοντέλο σχολιασμού της επαλήθευσης παρέχει $\Omega(n^2)$ χαμηλότερα όρια στο προϊόν χωρικής επικοινωνίας για τα προς εξέταση προβλήματα: υπό αυτή την οπτική, το γεγονός είναι δυνατή η απόκτηση καλύτερων πολυωνυμικών ορίων με μόνο έναν σταθερό αριθμό κύκλων καταδεικνύει τη δύναμη μερικών μόνο κύκλων αλληλεπίδρασης. Σημειώνεται επίσης ότι σχεδόν όλα τα κανονικά σκληρά προβλήματα για τους αλγόριθμους ροής (Index (Chakrabarti et al., 2015), Disjointness (Bahmani et al., 2012), Boolean Hidden Matching (Kogan & Krauthgamer, 2015) αναγνωρίζουν αποτελεσματικά SIP. Ένα SIP για ευρετήριο αναλύθηκε στην εργασία των (Chakrabarti et al., 2015) και χρησιμοποιεί στην εργασία της (Daruki, 2018) SIPs για Disjointness/Αταξία και Boolean Hidden Matching

5.4.4. Επαλήθευση Ροής για Ανάλυση Δεδομένων

Η μετάβαση από τον άμεσο υπολογισμό σε εξωτερική ανάθεση στο cloud έχει οδηγήσει σε νέους τρόπους σκέψης για τον υπολογισμό μαζικής κλίμακας. Στη ρύθμιση επαλήθευσης, η υπολογιστική προσπάθεια χωρίζεται μεταξύ ενός υπολογιστικά αδύναμου πελάτη (του επαληθευτή) ο οποίος κατέχει τα δεδομένα και θέλει να λύσει ένα επιθυμητό πρόβλημα και ενός πιο ισχυρού διακομιστή (του παρόχου) που εκτελεί τους υπολογισμούς. Ο πελάτης έχει περιορισμένη πρόσβαση (ροή) στα δεδομένα, καθώς και περιορισμένη δυνατότητα συνομιλίας με τον διακομιστή (μετρούμενη από το μέγεθος της επικοινωνίας), αλλά επιθυμεί να επαληθεύσει την ορθότητα των απαντήσεων του παρόχου.

Υπάρχουν πολλές υπηρεσίες "cloud" τρίτων που μπορούν να εκτελέσουν εντατικές υπολογιστικές εργασίες σε μεγάλα δεδομένα. Στα παραδείγματα περιλαμβάνονται οι υπηρεσίες Amazon EC2, η πλατφόρμα cloud της Azure της Microsoft, η Google Compute Engine και ακόμη και μια σειρά από εξειδικευμένες πλατφόρμες για ανάλυση δεδομένων μεγάλης κλίμακας. Αυτοί οι διακομιστές δεν είναι

υπολογιστικά περιορισμένοι: συνήθως περιλαμβάνουν μεγάλες ομάδες κόμβων υπολογιστών.

Στη διατριβή της (Daruki, 2018), μελετήθηκε μια ροή διαδραστικών αποδείξεων για προβλήματα στην ανάλυση δεδομένων και παρουσίασε αποτελεσματικά SIP για ορισμένα βασικά προβλήματα. Παρουσιάστηκαν πρωτόκολλα για ομαδοποίηση και προσαρμογή σχήματος, καθώς και βελτιωμένο πρωτόκολλο για πολλαπλασιασμό ορθογωνικής μήτρας. Συγκεκριμένα, δόθηκαν SIP 3 μηνυμάτων που μπορούν να επαληθεύσουν μια ελάχιστη μπάλα εγκλεισμού (minimum enclosing ball-MEB) και το πλάτος ενός σημείου που έχει καθοριστεί ακριβώς με πολυλογαριθμικό χώρο και κόστος επικοινωνίας. Παρουσιάστηκαν πρωτόκολλα πολυλογαριθμικού κύκλου με χώρο πολυλογαριθμικής επικοινωνίας και επαληθευτή για την επαλήθευση βέλτιστων k-κέντρων και k-πλακών στον ευκλείδειο χώρο. Παρουσιάστηκε επίσης ένα απλό πρωτόκολλο 3 μηνυμάτων για την επαλήθευση μιας 2-προσέγγισης k-κέντρο σε έναν μετρικό χώρο, μέσω της απλής προσαρμογής της 2-προσέγγισης Gonzalez για το k-κέντρο. Παρουσιάστηκαν επίσης πρωτόκολλα για θεμελιώδη προβλήματα ανάλυσης μήτρας: παρέχεται ένα βελτιωμένο πρωτόκολλο για προβλήματα ορθογωνικής μήτρας, τα οποία με τη σειρά τους μπορούν να χρησιμοποιηθούν για την επαλήθευση k (κατά προσέγγιση) ιδιοδιανυσμάτων ενός n×n ακέραιου πίνακα A. Σε γενικές γραμμές, οι λύσεις χρησιμοποιούν πολυλογαριθμικούς κύκλους επικοινωνίας και πολυλογαριθμικό συνολικό χώρο επικοινωνίας και επαληθευτή. Για ειδικές περιπτώσεις (όταν τα πιστοποιητικά βελτιστοποίησης μπορούν να επαληθευτούν εύκολα), παρουσιάζονται σταθερά κυκλικά πρωτόκολλα με παρόμοιο κόστος. Για τον πολλαπλασιασμό ορθογώνιας μήτρας και την επαλήθευση ιδιοδιανυσμάτων, τα πρωτόκολλά λειτουργούν στο πιο περιορισμένο μοντέλο ροής δεδομένων με σχολιασμούς (το οποίο ουσιαστικά αντιστοιχεί σε SIP ενός μηνύματος) και χρησιμοποιούν υπογραμμική (αλλά όχι πολυλογαριθμική) επικοινωνία.

5.4.5. Υπολογισμοί Συρόμενου Παραθύρου σε Ροές Δεδομένων

Το μοντέλο συρόμενου παραθύρου (sliding window model), όπου τα στοιχεία δεδομένων φθάνουν συνεχώς και χρησιμοποιούνται μόνο τα πιο πρόσφατα N στοιχεία κατά την απάντηση ερωτημάτων, έχει κριθεί κατάλληλο για τις περισσότερες

εφαρμογές που χειρίζονται δεδομένα με τεχνικές ροών. Στην εργασία των (Babcock et al., 2002) παρουσιάζεται μια νέα τεχνική για την επίλυση δύο σημαντικών και συναφών προβλημάτων στο μοντέλο συρόμενου παραθύρου - διατηρώντας τη διακύμανση και διατηρώντας μια συσταδοποίηση k-medians. Η λύση τους στο πρόβλημα της διατήρησης της διακύμανσης παρέχει μια συνεχώς ενημερωμένη εκτίμηση της διακύμανσης των τελευταίων N τιμών σε μια ροή δεδομένων με σχετικό σφάλμα το πολύ ε χρησιμοποιώντας μνήμη $O\left(\frac{1}{\varepsilon^2}\log N\right)$. Παρουσιάζεται λοιπόν ένας προσεγγιστικός αλγόριθμος δύο κριτηρίων, σταθερού συντελεστή που διατηρεί κατά προσέγγιση k-medians για τα τελευταία N σημεία δεδομένων χρησιμοποιώντας μνήμη $O\left(\frac{k}{\tau^4}N^{2\tau}\log^2 N\right)$ όπου $\tau < 1/2$ είναι μια παράμετρος που ανταλλάσσει το χώρο μνήμης που δεσμεύεται με το συντελεστής προσέγγισης του $O\left(2^{O(1/\tau)}\right)$. Παρουσιάζεται επίσης ένας αλγόριθμος που χρησιμοποιεί ακριβώς k κέντρα, σε αντίθεση με τα $2k$ στο προηγούμενο αποτέλεσμα, και έχει την ίδια εγγύηση προσέγγισης του $O\left(2^{O(1/\tau)}\right)$. Ωστόσο, αυτός ο αλγόριθμος είναι πιο περίπλοκος και απαιτεί μνήμη $O\left(\frac{k^{1/\tau}}{\tau^4}N^{2\tau}\log^2 N\right)$

Στο *μοντέλο συρόμενου παραθύρου*, τα στοιχεία δεδομένων φτάνουν σε μια ροή και μόνο τα τελευταία N στοιχεία (μέγεθος παραθύρου) που έχουν φτάσει θεωρούνται συναφή ανά πάσα στιγμή. Αυτά τα πιο πρόσφατα N στοιχεία ονομάζονται *ενεργά* στοιχεία δεδομένων. Τα υπόλοιπα λέγονται *ληγμένα* και δεν συνεισφέρουν πλέον σε απαντήσεις ερωτημάτων ή στατιστικά στοιχεία για το σύνολο δεδομένων. Μόλις υποβληθεί σε επεξεργασία ένα στοιχείο δεδομένων, δεν μπορεί να ανακτηθεί για περαιτέρω υπολογισμό αργότερα, εκτός εάν αποθηκεύεται ρητά στη μνήμη. Η ποσότητα της διαθέσιμης μνήμης θεωρείται ότι είναι περιορισμένη, ειδικότερα, υπογραμμική στο μέγεθος του συρόμενου παραθύρου. Επομένως, αλγόριθμοι που απαιτούν αποθήκευση ολόκληρου του συνόλου ενεργών στοιχείων δεν είναι αποδεκτοί σε αυτό το μοντέλο.

Θα χρησιμοποιηθεί η έννοια της *χρονοσφραγίδας* (timestamp), η οποία αντιστοιχεί στη θέση ενός ενεργού στοιχείου δεδομένων στο τρέχον παράθυρο. Τα ενεργά στοιχεία δεδομένων χρονοσφραγίζονται από το πιο πρόσφατο στο παλαιότερο, με το πιο

πρόσφατο στοιχείο δεδομένων να έχει μια χρονοσφραγίδα 1. Έστω ότι το x_i δηλώνει το στοιχείο δεδομένων με χρονοσφραγίδα i. Είναι σαφές ότι οι χρονοσφραγίδες αλλάζουν με κάθε νέα άφιξη και δεν είναι επιθυμητό να γίνονται ρητές ενημερώσεις. Μια απλή λύση είναι η καταγραφή των χρόνων άφιξης σε έναν μετρητή των log N bits. Τότε η χρονοσφραγίδα μπορεί να εξαχθεί σε σύγκριση με την τιμή μετρητή της τρέχουσας άφιξης.

Στην εργασία των (Babcock et al., 2002) διατηρούνται ιστογράμματα για τα ενεργά στοιχεία δεδομένων στη ροή δεδομένων. Η ανθρώπινη αντίληψη για τα ιστογράμματα είναι πολύ πιο γενική από την παραδοσιακή που χρησιμοποιείται στη βιβλιογραφία των βάσεων δεδομένων. Συγκεκριμένα, κάθε κουβάς (bucket) στα ιστογράμματα αποθηκεύει κάποια δομή περίληψης / σύνοψης για ένα συνεχόμενο σύνολο στοιχείων δεδομένων, δηλαδή, το ιστόγραμμα χωρίζεται σε κουβάδες με βάση την ώρα άφιξης των στοιχείων δεδομένων. Μαζί με αυτήν τη σύνοψη, για κάθε κάδο, διατηρείται η χρονοσφραγίδα του πιο πρόσφατου στοιχείου δεδομένων σε αυτόν τον κουβά (η *χρονοσφραγίδα κουβά*). Όταν η χρονοσφραγίδα ενός κουβά φτάσει στο $N + 1$, όλα τα στοιχεία δεδομένων στον κουβά έχουν λήξει, οπότε μπορεί να αδειάσει αυτός ο κουβάς και να ανακτηθεί η μνήμη του. Όλοι οι κουβάδες, εκτός από τον τελευταίο, περιέχουν μόνο ενεργά στοιχεία, ενώ ο τελευταίος κουβάς μπορεί να περιέχει ορισμένα στοιχεία που έχουν λήξει εκτός από τουλάχιστον ένα ενεργό στοιχείο. Οι κουβάδες αριθμούνται B_1, B_2, ..., B_m, ξεκινώντας από το πιο πρόσφατο (B_1) έως το παλαιότερο (B_m). Περαιτέρω, t_1, t_2, ..., t_m δηλώνονται οι χρονοσφραγίδες των κουβάδων.

5.5.Σκιαγράφηση γραμμικών ταξινομητών μέσω ροών δεδομένων

Παρουσιάζεται ένα νέο υπογραμμικό χωρικό διάγραμμα - το Weight-MedianSketch - για την εκμάθηση συμπιεσμένων γραμμικών ταξινομητών σε ροές δεδομένων. Το μοντέλο αυτό υποστηρίζει επιπλέον την αποτελεσματική ανάκτηση βαρών μεγάλου μεγέθους. Αυτό επιτρέπει την εκτέλεση περιορισμένης μνήμης πολλών στατιστικών αναλύσεων σε ροές, συμπεριλαμβανομένης της επιλογής διαδικτυακών δυνατοτήτων, της εξήγησης δεδομένων ροής, της σχετικής ανίχνευσης δελτοειδούς και της εκτίμησης ροής των αμοιβαίων πληροφοριών κατά τη διάρκεια της ροής. Σε αντίθεση με τα σχετικά διαγράμματα που καταγράφουν τις πιο συχνά εμφανιζόμενες

λειτουργίες (ή στοιχεία) σε μια ροή δεδομένων, το Weight-Median Sketch καταγράφει τις λειτουργίες που είναι πιο διακριτικές για τη ροή (ή κατηγορία). Το Weight-Median Sketch υιοθετεί τη βασική χρήση δομής δεδομένων στο Count-Sketch, αλλά, αντί να μετράει το διάγραμμα, καταγράφει σχεδιασμένες ημερομηνίες κλίσης στις παραμέτρους του μοντέλου. Παρέχεται η θεωρητική ανάλυση που καθιερώνει εγγυήσεις ανάκτησης για μαζική και διαδικτυακή μάθηση και καταδεικνύει εμπειρικές βελτιώσεις στις ανταλλαγές ακρίβειας μνήμης έναντι εναλλακτικών μεθόδων υπολογισμού μνήμης, συμπεριλαμβανομένων σχεδίων βασισμένων σε μετρήσεις και κατακερματισμό χαρακτηριστικών.

Ακολουθεί περιγραφή της μεθόδου που ανέπτυξαν στην εργασία τους οι (Tai et al., 2018), το WeightMedian Sketch (WM-Sketch), μαζί με μια απλή παραλλαγή, το Active-Set Weight-Median Sketch (AWM-Sketch), που βελτιώνεται εμπειρικά στο βασικό WM-Sketch τόσο στην ταξινόμηση όσο και στην ακρίβεια ανάκτησης .

5.5.1. Weight-Median Sketch

Η κύρια δομή δεδομένων στο WM-Sketch είναι ίδια με αυτήν που χρησιμοποιείται στο Count-Sketch. Το διάγραμμα παραμετροποιείται κατά μέγεθος k, βάθος s και πλάτος k/s. Αρχικοποιείται το διάγραμμα με έναν πίνακα τάξης μεγέθους-k ρυθμισμένο στο μηδέν. Για δεδομένο βάθος s, ο πίνακας αυτός παρουσιάζεται να είναι διατεταγμένος σε s σειρές, καθεμία από τις οποίες έχει πλάτος από πλάτος k/s (έστω ότι το k είναι πολλαπλάσιο του s). Έστω z ο πίνακας αυτός ο οποίος εμφανίζεται ισοδύναμα ως διάνυσμα σε R^k.

Η συνολική ιδέα είναι ότι κάθε σειρά του σχεδίου είναι μια συμπιεσμένη έκδοση του διανύσματος βάρους μοντέλου $w \in R^d$, όπου κάθε δείκτης $i \in [d]$ αντιστοιχίζεται σε κάποια προσδιορισμένη μονάδα αποθήκευσης πληροφορίας (κάδος-bucket) $j \in [k/s]$. Εφόσον $k/s \ll d$, θα υπάρξουν πολλές συγκρούσεις μεταξύ αυτών των βαρών. Επομένως, θα πρέπει να διατηρούνται s-σειρές - κάθε μία με διαφορετικές εκχωρήσεις χαρακτηριστικών σε κάδους - προκειμένου να αποσαφηνιστεί το βάρος.

5.5.2. Χαρακτηριστικά Κατακερματισμού

Προκειμένου να αποφευχθεί η ρητή αποθήκευση της χαρτογράφησης από χαρακτηριστικά σε κάδους, η οποία θα απαιτούσε γραμμικό χώρο σε d, εφαρμόζεται η χαρτογράφηση χρησιμοποιώντας λειτουργίες κατακερματισμού όπως στο Count-Sketch. Για κάθε σειρά j ϵ [s], διατηρούνται κάποιες λειτουργίες κατακερματισμού, $h_j: [d] \to [k/s]$ και $s_j: [d] \to \{-1, +1\}$. Έστω ότι ο πίνακας A ϵ $\{1, +1\}^{kxd}$ δηλώνει την προβολή Count-Sketch που υποδηλώνεται έμμεσα από τις συναρτήσεις κατακερματισμού h_j και σ_j, και R να είναι μια κλιμακωτή έκδοση αυτής της προβολής, $R = \frac{1}{\sqrt{s}}A$. Χρησιμοποιείται η προβολή R για τη συμπίεση των διανυσματικών χαρακτηριστικών και την ενημέρωση του Διαγράμματος.

Ενημερώσεις. Το διάγραμμα ενημερώνεται εκτελώντας ενημερώσεις καθοδικής κλίσης απευθείας στον συμπιεσμένο ταξινομητή z. Οι κλίσεις υπολογίζονται σε μια «συμπιεσμένη» έκδοση $\widehat{L_t}$L της κανονικοποιημένης απώλειας:

$$\widehat{L_t(z)} = l(y_t z^T R x_t) + \frac{\lambda}{2}\|z\|_2^2$$

Αυτό αποδίδει την ακόλουθη ενημέρωση στο z:

$$\widehat{\Delta_t} := -\eta_t \nabla \widehat{L_t(z)} = -\eta_t (y_t \nabla l(y_t z^T R x_t) R x_t + \lambda z$$

Είναι χρήσιμο αυτή η ενημέρωση να συγκριθεί με τον κανόνα του Count-Sketch update (Charikar et al., 2002). Στη ρύθμιση συχνών αντικειμένων, η εισαγωγή x_t είναι μια μοναδική κωδικοποίηση για το στοιχείο που εμφανίζεται σε αυτό το χρονικό βήμα. Η ενημέρωση για την κατάσταση Count-Sketch z_{cs} είναι η ακόλουθη:

$$\widehat{\Delta_t^{cs}} = A x_t$$

όπου το A ορίζεται πανομοιότυπα όπως παραπάνω. Αγνοώντας τον όρο κανονικοποίησης, ο κανόνας ενημέρωσης είναι απλώς η ενημέρωση Count-Sketch που κλιμακώνεται από τη σταθερά $-\eta_t y_t s^{-1/2} \nabla l(y_t z^T R x_t$. Ωστόσο, μια σημαντική λεπτομέρεια που πρέπει να σημειωθεί είναι ότι η ενημέρωση Count-Sketch είναι ανεξάρτητη από την κατάσταση σχεδίου z_{cs}, ενώ η ενημέρωση WM-Sketch εξαρτάται

από το z. Αυτή η κυκλική εξάρτηση μεταξύ ενημερώσεων κατάστασης και κατάστασης είναι η κύρια πρόκληση στην ανάλυσή μας για το WM-Sketch.

Ερωτήματα. Για να ληφθεί μια εκτίμηση $\widehat{w_i}$ του i-οστού βάρους, χρησιμοποιείται η μέση τιμή των τιμών $\{\sqrt{s}\sigma_j(i)z_{j,h_j(i)}: j\epsilon[s]\}$. Αποθήκευση για τον παράγοντα $\sqrt{s}$, αυτό είναι πανομοιότυπο με τη διαδικασία ερωτήματος για το Count-Sketch.

6. Συμπεράσματα

Αρκετές σύγχρονες εφαρμογές απαιτούν χειρισμό δεδομένων τόσο μαζικά, ώστε τα παραδοσιακά αλγοριθμικά μοντέλα να μην παρέχουν ακριβή μέσα για το σχεδιασμό και την αξιολόγηση αποτελεσματικών αλγορίθμων. Τέτοια μοντέλα συνήθως υποθέτουν ότι όλα τα δεδομένα χωρούν στη μνήμη και ότι ο χρόνος εκτέλεσης διαμορφώνεται με ακρίβεια ως ο αριθμός των βασικών εντολών που εκτελεί ο αλγόριθμος. Ωστόσο, σε εφαρμογές όπως διαδικτυακά κοινωνικά δίκτυα, σύγχρονα επιστημονικά πειράματα μεγάλης κλίμακας, μηχανές αναζήτησης, παράδοση διαδικτυακού περιεχομένου, καθώς και παρακολούθηση προϊόντων και καταναλωτών για μεγάλους εμπόρους λιανικής όπως το Amazon και το Walmart, πρέπει να αναλυθούν δεδομένα πολύ μεγάλα για να χωρέσουν στη μνήμη. Αυτό το ζήτημα οδήγησε στην ανάπτυξη διαφόρων μοντέλων για την επεξεργασία τόσο μεγάλων ποσοτήτων δεδομένων. Πρόκειται για το μοντέλο εξωτερικής μνήμης (Vitter, 2006) και την προσωρινή μνήμη (Frigo et al., 2012) όπου κάποιος στοχεύει να ελαχιστοποιήσει τον αριθμό των μπλοκ που έχουν ληφθεί από το δίσκο, δοκιμή ιδιοκτησίας (Goldreich, 1997), όπου θεωρείται ότι τα δεδομένα είναι τόσο τεράστια που δεν θέλει κανείς ούτε καν να τα δει στο σύνολό τους ως εκ τούτου σκοπός είναι η ελαχιστοποίηση του αριθμού των ανιχνευτών που υπάρχουν στα δεδομένα, και η μαζικά παράλληλοι αλγόριθμοι που λειτουργούν σε συστήματα όπως MapReduce και Hadoop (Borthakur, 2007). Επίσης, σε ορισμένες εφαρμογές, τα δεδομένα φτάνουν με τρόπο ροής και πρέπει να υποβάλλονται σε επεξεργασία εν κινήσει καθώς φτάνουν. Τέτοιες περιπτώσεις προκύπτουν, για παράδειγμα, με ροές πακέτων στην παρακολούθηση της κυκλοφορίας του δικτύου ή με ροές ερωτήσεων που φθάνουν σε μια διαδικτυακή υπηρεσία, όπως μια μηχανή αναζήτησης.

Στην εργασία του ο Nelson εστίασε σε αυτό το τελευταίο μοντέλο ροής υπολογισμού, όπου ένας δεδομένος αλγόριθμος πρέπει να κάνει ένα πέρασμα πάνω από ένα σύνολο δεδομένων για να υπολογίσει κάποια λειτουργία. Τέτοιοι αλγόριθμοι ροής που χρησιμοποιούν μνήμη σημαντικά υπογραμμική στην ποσότητα δεδομένων είναι επιθυμητοί, καθώς θεωρείται ότι τα δεδομένα είναι πολύ μεγάλα για να χωρέσουν στη μνήμη. Μερικές φορές είναι επίσης χρήσιμο να ληφθούν υπόψη αλγόριθμοι που

επιτέπουν όχι μόνο ένα, αλλά μερικά περάσματα πάνω από τα δεδομένα, σε περιπτώσεις όπου για παράδειγμα το σύνολο δεδομένων υπάρχει στο δίσκο και ο αριθμός των περασμάτων μπορεί να κυριαρχήσει επί του συνολικού χρόνου εκτέλεσης. Συζητήθηκαν επίσης περιστασιακά σχεδιαγράμματα. Ένα σχεδιάγραμμα σχετίζεται με κάποια συνάρτηση f, και ένα σχεδιάγραμμα ενός συνόλου δεδομένων x αποτελεί μια συμπιεσμένη αναπαράσταση του x από την οποία μπορεί να υπολογιστεί το f(x). Φυσικά, κάτω από αυτόν τον ορισμό, το f(x) είναι ένα έγκυρο σχεδιάγραμμα του x, αλλά συχνά απαιτούνται πολλά περισσότερα από το αυτό εκτός του υπολογισμού του f (x). Για παράδειγμα, συνήθως απαιτείται να είναι δυνατή η ενημέρωση του σχεδιαγράμματος καθώς φθάνουν περισσότερα δεδομένα και μερικές φορές επίσης απαιτείται τα σχεδιαγράμματα δύο διαφορετικών συνόλων δεδομένων που προετοιμάστηκαν ανεξάρτητα να μπορούν να συγκριθούν για τον υπολογισμό συνάρτησης των συγκεντρωτικών δεδομένων ή των μέτρων ομοιότητας ή διαφοράς σε διαφορετικά σύνολα δεδομένων.

Βιβλιογραφία

Aggarwal, C. C. (2003, June). A framework for diagnosing changes in evolving data streams. In *Proceedings of the 2003 ACM SIGMOD international conference on Management of data* (pp. 575-586).

Aggarwal, C. C., Han, J., Wang, J., & Yu, P. S. (2004, August). On demand classification of data streams. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 503-508).

Aggarwal, C. C., Philip, S. Y., Han, J., & Wang, J. (2003, January). A framework for clustering evolving data streams. In *Proceedings 2003 VLDB conference* (pp. 81-92). Morgan Kaufmann.

Aggarwal, C. C., Wolf, J. L., Yu, P. S., Procopiuc, C., & Park, J. S. (1999). Fast algorithms for projected clustering. *ACM SIGMoD Record*, *28*(2), 61-72.

Agrawal, R., Gehrke, J., Gunopulos, D., & Raghavan, P. (1998, June). Automatic subspace clustering of high dimensional data for data mining applications. In *Proceedings of the 1998 ACM SIGMOD international conference on Management of data* (pp. 94-105).

Agrawal, R., Imieliński, T., & Swami, A. (1993, June). Mining association rules between sets of items in large databases. In *Proceedings of the 1993 ACM SIGMOD international conference on Management of data* (pp. 207-216).

Ahn, K. J., Guha, S., & McGregor, A. (2012, January). Analyzing graph structure via linear measurements. In Proceedings of the twenty-third annual ACM-SIAM symposium on Discrete Algorithms (pp. 459-467). Society for Industrial and Applied Mathematics.

Ahn, K. J., Guha, S., & McGregor, A. (2012, May). Graph sketches: sparsification, spanners, and subgraphs. In Proceedings of the 31st ACM SIGMOD-SIGACT-SIGAI symposium on Principles of Database Systems (pp. 5-14).

Babcock, B., Datar, M., Motwani, R., & O'Callaghan, L. (2002). *Sliding window computations over data streams*. Stanford InfoLab.

Bahmani, B., Kumar, R., & Vassilvitskii, S. (2012). Densest subgraph in streaming and mapreduce. *arXiv preprint arXiv:1201.6567*.

Borthakur, D. (2007). The hadoop distributed file system: Architecture and design. *Hadoop Project Website*, *11*(2007), 21.

Chakrabarti, A., Cormode, G., & Mcgregor, A. (2009, July). Annotations in data streams. In *International Colloquium on Automata, Languages, and Programming* (pp. 222-234). Springer, Berlin, Heidelberg.

Chakrabarti, A., Cormode, G., McGregor, A., Thaler, J., & Venkatasubramanian, S. (2015). Verifiable stream computation and Arthur–Merlin communication. In *30th Conference on Computational Complexity (CCC 2015)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.

Chakrabarti, A., Cormode, G., McGregor, A., Thaler, J., & Venkatasubramanian, S. (2015). Verifiable stream computation and Arthur–Merlin communication. In *30th Conference on Computational Complexity (CCC 2015)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.

Charikar, M., Chen, K., & Farach-Colton, M. (2004). Finding frequent items in data streams. *Theoretical Computer Science*, *312*(1), 3-15.

Chen, Y., Dong, G., Han, J., Wah, B. W., & Wang, J. (2002, January). Multi-dimensional regression analysis of time-series data streams. In *VLDB'02: Proceedings of the 28th International Conference on Very Large Databases* (pp. 323-334). Morgan Kaufmann.

Cormode, G., & Firmani, D. (2014). A unifying framework for ℓ 0-sampling algorithms. *Distributed and Parallel Databases*, *32*(3), 315-335.

Cormode, G., & Garofalakis, M. (2005, August). Sketching streams through the net: Distributed approximate query tracking. In *Proceedings of the 31st international conference on Very large data bases* (pp. 13-24).

Cormode, G., Mitzenmacher, M., & Thaler, J. (2013). Streaming graph computations with a helpful advisor. *Algorithmica*, *65*(2), 409-442.

Cormode, G., Thaler, J., & Yi, K. (2011). Verifying computations with streaming interactive proofs. *arXiv preprint arXiv:1109.6882*.

Daruki, S., Thaler, J., & Venkatasubramanian, S. (2015, December). Streaming verification in data analysis. In *International Symposium on Algorithms and Computation* (pp. 715-726). Springer, Berlin, Heidelberg.

Dasu, T., Krishnan, S., Venkatasubramanian, S., & Yi, K. (2006). An information-theoretic approach to detecting changes in multi-dimensional data streams. In *In Proc. Symp. on the Interface of Statistics, Computing Science, and Applications*.

Datar, M., Gionis, A., Indyk, P., & Motwani, R. (2002). Maintaining stream statistics over sliding windows. *SIAM journal on computing*, *31*(6), 1794-1813.

Desale, K. S., Kumathekar, C. N., & Chavan, A. P. (2015, February). Efficient intrusion detection system using stream data mining classification technique. In *2015 International Conference on Computing Communication Control and Automation* (pp. 469-473). IEEE.

Domingos, P., & Hulten, G. (2000, August). Mining high-speed data streams. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 71-80).

Dong, G., Han, J., Lam, J., Pei, J., & Wang, K. (2001, September). Mining multi-dimensional constrained gradients in data cubes. In *VLDB* (Vol. 1, pp. 321-330).

Duda, R. O., & Hart, P. E. (1973). *Pattern classification and scene analysis* (Vol. 3, pp. 731-739). New York: Wiley.

Frigo, M., Leiserson, C. E., Prokop, H., & Ramachandran, S. (1999, October). Cache-oblivious algorithms. In *40th Annual Symposium on Foundations of Computer Science (Cat. No. 99CB37039)* (pp. 285-297). IEEE.

Garofalakis, M., Gehrke, J., & Rastogi, R. (2002, June). Querying and mining data streams: you only get one look a tutorial. In *Proceedings of the 2002 ACM SIGMOD international conference on Management of data* (pp. 635-635).

Giannella, C., Han, J., Pei, J., Yan, X., & Yu, P. S. (2003). Mining frequent patterns in data streams at multiple time granularities. *Next generation data mining*, *212*, 191-212.

Gilbert, A. C., Kotidis, Y., Muthukrishnan, S., & Strauss, M. (2001). Surfing wavelets on streams: One-pass summaries for approximate aggregate queries. In *Vldb* (Vol. 1, pp. 79-88).

Goldreich, O. (1999). Combinatorial property testing (a survey). *Randomization Methods in Algorithm Design*, *43*, 45-59.

Goldwasser, S., Kalai, Y. T., & Rothblum, G. N. (2015). Delegating computation: interactive proofs for muggles. *Journal of the ACM (JACM)*, *62*(4), 1-64.

Greenwald, M., & Khanna, S. (2001). Space-efficient online computation of quantile summaries. *ACM SIGMOD Record*, *30*(2), 58-66.

Guha, S., Meyerson, A., Mishra, N., Motwani, R., & O'Callaghan, L. (2003). Clustering data streams: Theory and practice. *IEEE transactions on knowledge and data engineering*, *15*(3), 515-528.

Guha, S., Rastogi, R., & Shim, K. (1998). CURE: An efficient clustering algorithm for large databases. *ACM Sigmod record*, *27*(2), 73-84.

Hand, D. J., & Adams, N. M. (2014). Data mining. *Wiley StatsRef: Statistics Reference Online*, 1-7.

Henzinger, M. R., Raghavan, P., & Rajagopalan, S. (1998). Computing on data streams. *External memory algorithms*, *50*, 107-118.

Hulten, G., Spencer, L., & Domingos, P. (2001, August). Mining time-changing data streams. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 97-106).

Jain, A. K., & Dubes, R. C. (1988). *Algorithms for clustering data*. Prentice-Hall, Inc..

Jin, R., & Agrawal, G. (2005, November). An algorithm for in-core frequent itemset mining on streaming data. In *Fifth IEEE International Conference on Data Mining (ICDM'05)* (pp. 8-pp). IEEE.

Johnson, R., & Zhang, T. (2013). Accelerating stochastic gradient descent using predictive variance reduction. *Advances in neural information processing systems*, *26*, 315-323.

Kane, M. J., Emerson, J., & Weston, S. (2013). Scalable strategies for computing with massive data. *Journal of Statistical Software*, *55*(14), 1-19.

Kapralov, M., Khanna, S., & Sudan, M. (2014, December). Streaming lower bounds for approximating MAX-CUT. In *Proceedings of the twenty-sixth annual ACM-SIAM symposium on Discrete algorithms* (pp. 1263-1282). Society for Industrial and Applied Mathematics.

Karp, R. M., Upfal, E., & Wigderson, A. (1986). Constructing a perfect matching is in random NC. *Combinatorica*, *6*(1), 35-48.

Kifer, D., Ben-David, S., & Gehrke, J. (2004, August). Detecting change in data streams. In *VLDB* (Vol. 4, pp. 180-191).

Koedinger, K. R., D'Mello, S., McLaughlin, E. A., Pardos, Z. A., & Rose, C. P. (2015). Data mining and education. *Wiley Interdisciplinary Reviews: Cognitive Science*, *6*(4), 333-353.

Kogan, D., & Krauthgamer, R. (2015, January). Sketching cuts in graphs and hypergraphs. In *Proceedings of the 2015 Conference on Innovations in Theoretical Computer Science* (pp. 367-376).

Kollios, G., Byers, J. W., Considine, J., Hadjieleftheriou, M., & Li, F. (2005). Robust Aggregation in Sensor Networks. *IEEE Data Eng. Bull.*, *28*(1), 26-32.

Lu, Y., Dhillon, P. S., Foster, D. P., & Ungar, L. H. (2013). Faster ridge regression via the subsampled randomized hadamard transform.

Madduri, K., & Bader, D. A. (2009, May). Compact graph representations and parallel connectivity algorithms for massive dynamic network analysis. In *2009 IEEE International Symposium on Parallel & Distributed Processing* (pp. 1-11). IEEE.

Maia, J., Junior, C. A. S., Guimarães, F. G., de Castro, C. L., Lemos, A. P., Galindo, J. C. F., & Cohen, M. W. (2020). Evolving clustering algorithm based on mixture of typicalities for stream data mining. *Future Generation Computer Systems*, *106*, 672-684.

McGregor, A. (2014). Graph stream algorithms: a survey. *ACM SIGMOD Record*, *43*(1), 9-20.

Murray, G. R., & Scime, A. (2015). Data Mining. *Emerging Trends in the Social and Behavioral Sciences: An Interdisciplinary, Searchable, and Linkable Resource*, 1-15.

Muthukrishnan, S. (2005). *Data streams: Algorithms and applications*. Now Publishers Inc.

Muthukrishnan, S. (2005). *Data streams: Algorithms and applications*. Now Publishers Inc.

Muthukrishnan, S. (2005). *Data streams: Algorithms and applications*. Now Publishers Inc.

O'callaghan, L., Mishra, N., Meyerson, A., Guha, S., & Motwani, R. (2002, February). Streaming-data algorithms for high-quality clustering. In *Proceedings 18th International Conference on Data Engineering* (pp. 685-694). IEEE.

Pilanci, M., & Wainwright, M. J. (2016). Iterative Hessian sketch: Fast and accurate solution approximation for constrained least-squares. *The Journal of Machine Learning Research*, *17*(1), 1842-1879.

Poularakis, K., Iosifidis, G., & Tassiulas, L. (2013, December). Approximation caching and routing algorithms for massive mobile data delivery. In *2013 IEEE Global Communications Conference (GLOBECOM)* (pp. 3534-3539). IEEE.

Rajaraman, A., & Ullman, J. D. (2011). *Mining of massive datasets*. Cambridge University Press.

Sakurai, Y., Papadimitriou, S., & Faloutsos, C. (2005, June). Braid: Stream mining through group lag correlations. In *Proceedings of the 2005 ACM SIGMOD international conference on Management of data* (pp. 599-610).

Shrivastava, N., Buragohain, C., Agrawal, D., & Suri, S. (2004, November). Medians and beyond: new aggregation techniques for sensor networks. In *Proceedings of the 2nd international conference on Embedded networked sensor systems* (pp. 239-249).

Teng, S. H. (2010, June). The Laplacian paradigm: Emerging algorithms for massive graphs. In *International Conference on Theory and Applications of Models of Computation* (pp. 2-14). Springer, Berlin, Heidelberg.

Thaler, J. (2014). Semi-streaming algorithms for annotated graph streams. *arXiv preprint arXiv:1407.3462*.

Vitter, J. S. (1985). Random sampling with a reservoir. *ACM Transactions on Mathematical Software (TOMS)*, *11*(1), 37-57.

Vitter, J. S. (2008). *Algorithms and data structures for external memory*. Now Publishers Inc.

Weiss, G. M., & Davison, B. D. (2010). Data mining. In *TO APPEAR IN THE HANDBOOK OF TECHNOLOGY MANAGEMENT, H. BIDGOLI (ED.)*.

Wu, X., Zhu, X., Wu, G. Q., & Ding, W. (2013). Data mining with big data. *IEEE transactions on knowledge and data engineering*, *26*(1), 97-107.

Yi, B. K., Sidiropoulos, N. D., Johnson, T., Jagadish, H. V., Faloutsos, C., & Biliris, A. (2000, March). Online data mining for co-evolving time sequences. In *Proceedings of 16th International Conference on Data Engineering (Cat. No. 00CB37073)* (pp. 13-22). IEEE.

Zhang, J. (2010). A survey on streaming algorithms for massive graphs. In *Managing and Mining Graph Data* (pp. 393-420). Springer, Boston, MA.

Zhang, L., Mahdavi, M., Jin, R., Yang, T., & Zhu, S. (2014). Random projections for classification: A recovery approach. *IEEE Transactions on Information Theory*, *60*(11), 7300-7316.