
Εθνικο Μετσοβιο Πολυτεχνειο

Σχολη Ηλεκτρολογων Μηχανικων και Μηχανικων Υπολογιστων

Τομεας Τεχνολογιας, Πληροφορικης και Υπολογιστων


# GraKeL: μία Βιβλιοθήκη για Πυρήνες Γράφων

## Διπλωματικη Εργασια

του


**ΣΙΓΛΙΔΗ ΓΙΑΝΝΗ**

**Επιβλέπων Ε.Μ.Π.:** Ανδρέας Γεώργιος Σταφυλοπάτης
Καθηγητής Ε.Μ.Π.

**Εξωτερικός Επιβλέπων:** Μιχάλης Βαζιργιάννης
Καθηγητής Ecole Polytechnique, Καθηγητής Ο.Π.Α.



Αθήνα, Οκτώβριος 2018

Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Τομέας Τεχνολογίας, Πληροφορικής και Υπολογιστών

# GraKeL: μία Βιβλιοθήκη για Πυρήνες Γράφων

## Διπλωματικη Εργασια

του

**ΣΙΓΛΙΔΗ ΓΙΑΝΝΗ**

**Επιβλέπων Ε.Μ.Π.:** Ανδρέας Γεώργιος Σταφυλοπάτης
Καθηγητής Ε.Μ.Π.

**Εξωτερικός Επιβλέπων:** Μιχάλης Βαζιργιάννης
Καθηγητής Ecole Polytechnique, Καθηγητής Ο.Π.Α.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 18η Οκτωβρίου 2018.

| *(Υπογραφή)* | *(Υπογραφή)* | *(Υπογραφή)* |
|---|---|---|
| ..... | ..... | ..... |
| Αντρέας Γεώργιος Σταφυλοπάτης<br>Καθηγητής Ε.Μ.Π. | Μιχάλης Βαζιργιάννης<br>Καθηγητής Ο.Π.Α.<br>Καθηγητής Ecole Polytechnique | Γιώργος Στάμου<br>Καθηγητής Ε.Μ.Π. |

Αθήνα, Οκτώβριος 2018

*(Υπογραφή)*

.....

**Ιωαννης Σιγλιδης**
Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

# Περίληψη

Το πρόβλημα ακριβούς μέτρησης της ομοιότητας μεταξύ δεδομένων που έχουν αναπαρασταθεί με τη μορφή γράφων είναι στον πυρήνα πολλών εφαρμογών σε ένα μεγάλο εύρος επιστημονικών και τεχνολογικών κλάδων. Λόγω της πολυωνυμικής υπολογιστικής πολυπλοκότητας και της θεμελιώδους θεωρητικής τους βάσης, οι πυρήνες γράφων έχουν εμφανιστεί ως μία ελπιδοφόρα προσσέγγιση στην αντιμετώπιση αυτού του προβλήματος. Εστιάζοντας σε διαφορετικά δομικά χαρακτηρηστικά των γράφων, μπορούν στην πολυμορφία τους να παρέχουν μία λύση αιχμής σε πλήθος εφαρμογών του πραγματικού κόσμου. Σε αυτήν την διπλωματική παρουσιάζουμε την ανάπτυξη του GraKeL, μίας βιβλιοθήκης που ενοποιεί μία ικανή ποσότητα σημαντικών πυρήνων γράφων σε μία κοινή αντικειμενοστρεφή δομή. Η βιβλιοθήκη είναι υλοποιημένη σε γλώσσα προγραμματισμού Python και είναι κατασκευασμένη βάσει του προτύπου της βιβλιοθήκης scikit-learn. Είναι εύκολη στη χρήση και μπορεί να συνδυαστεί φυσικά με υπολογιστικά αντικείμενα του ίδιου του scikit-learn για να σχηματίσει μία πλήρη ακολουθία εφαρμογών μηχανικής μάθησης, για προβλήματα όπως αυτά της ταξινόμησης και της συσταδοποίησης γράφων.
Παρέχεται με άδεια BSD και μπορεί να βρεθεί στη διεύθυνση: https://github.com/ysig/GraKeL.

## Λέξεις Κλειδιά

Ομοιότητα Γράφων, Πυρήνες Γράφων, Βιβλιοθήκη Python, Βιοπληροφορική, Χημιοπληροφορική

# Abstract

The problem of accurately measuring the similarity between graphs is at the core of many applications in a variety of disciplines. Because of their polynomial complexity and their fundamental theoretical foundation, graph kernels have recently emerged as a promising approach to this problem. By focusing on different structural aspects of graphs, in their diversity they can provide state of the art solutions to real world applications. In this thesis, we present the development of GraKeL, a library that unifies a sufficient amount of influential graph kernels into a common object-oriented framework. The library is written in Python programming language and is build on top of the scikit-learn project template. It is simple to use and can be naturally combined with scikit-learn's modules to build a complete machine learning pipeline for tasks such as graph classification and clustering. It is BSD licensed and can be found at: https://github.com/ysig/GraKeL.

## Keywords

Graph Similarity, Graph Kernels, Python Library, Bioinformatics, Chemoinformatics

# Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή κ. Ανδρέα Σταφυλοπάτη για την υποστήριξη του. Επίσης θα ήθελα να ευχαριστήσω τον καθηγητή Μιχάλη Βαζιργιάννη για την κοπιώδη υποστήριξη και το ενδιαφέρον του όλο το διάστημα που εργαζόμουν μαζί του στο Εργαστήριο LiX, στο Παρίσι. Ακόμα ευχαριστώ ιδιαίτερα τον μεταδιδακτορικό φοιτητή Γιάννη Νικολέντζο δίχως τη βοήθεια, την καθοδήγηση και το ενδιαφέρον του οποίου δεν θα είχα φέρει εις πέρας το δύσκολο έργο ολοκλήρωσης αυτής της διπλωματικής.
Τέλος θα ήθελα να ευχαριστήσω τους γονείς μου Ελένη και Παναγιώτη για την αγάπη και την συνεχή παρουσία τους και τις φίλες και τους φίλους μους, δίχως την ύπαρξη των οποίων δεν θα υπήρχα.
Η παρούσα εργασία είναι αφιερωμένη στην μνήμη της γιαγιάς μου Μαρίνας και του ξαδέρφου μου Παντελή.

# Περιεχόμενα

# Κατάλογος Σχημάτων

# Κατάλογος Πινάκων

# Κεφάλαιο 1

# Εισαγωγή

Η αναπαράσταση γράφων αποτελεί έναν ευέλικτο και περιεκτικό τρόπο με τον οποίο σε διάφορα πεδία της επιστήμης και της τεχνολογίας αναπαρίστανται σύνολα οντοτήτων, ζευγάρια των οποίων συσχετίζονται. Τα τελευταία χρόνια η αναπαράσταση δεδομένων με την μορφή γράφων έχει γνωρίσει τρομερή άνθιση σε πολλά πεδία, από τα κοινωνικά δίκτυα μέχρι την βιοπληροφορική. Διάφορα προβλήματα που εγείρουν όλο και πιο πολύ το ενδιαφέρον, εγκαλούν την χρήση τεχνικών μηχανικής μάθησης σε δεδομένα που έχουν αναπαρασταθεί ως γράφοι. Η δυσκολία και συχνά η αναποτελεσματικότητα των πειραματικών μεθόδων, σε επιστήμες όπως η χημεία και η βιολογία, έχει οδηγήσει στην διερεύνηση τεχνικών στο χώρο της μηχανικής μάθησης, σαν μία πιο αποδοτική εναλλακτική λύση. Για παράδειγμα, η κατανόηση της λειτουργίας μίας πρωτεΐνης με γνωστή ακολουθία μέσα από αναλυτικές πειραματικές μεθόδους, είναι μία ιδιαίτερα επίπονη και χρονοβόρα διαδικασία. Αντί για αυτό μπορούμε να δοκιμάσουμε υπολογιστικές προσεγγίσεις προκειμένου να προβλέψουμε την πρωτεϊνική λειτουργία. Αναπαριστώντας τις πρωτεΐνες σαν γράφους, το πρόβλημα μπορεί να οριστεί σαν πρόβλημα ταξινόμησης γράφων (graph classification problem) όπου η λειτουργία μίας νεοανακαλυφθείσας πρωτεΐνης προβλέπεται βάσει ενός συνόλου πρωτεϊνών με γνωστή λειτουργία. Παράλληλα με την ανάγκη για μεθόδους μειωμένου υπολογιστικού κόστους, ένα μεγάλο πλήθος εργασιών είναι αδύνατο να έρθουν εις πέρας από ανθρώπους, λόγω του πολύ μεγάλου όγκου δεδομένων που απαιτούν επεξεργασία. Για παράδειγμα, ο αριθμός των κακόβουλων (malicious) εφαρμογών έχει αυξηθεί αρκετά τα τελευταία χρόνια. Η δια χειρός εξέταση δειγμάτων κώδικα με στόχο την ανίχνευση κακόβουλης λειτουργίας δεν είναι εφικτή, καθώς ο αριθμός των δειγμάτων αυξάνεται. Χάρη στο γεγονός ότι τα περισσότερα καινούργια δείγματα κακόβουλου κώδικα είναι παραλλαγές υπάρχοντος κακόβουλου λογισμικού, μπορούμε αναπαριστώντας τα ως γράφους κλήσης συναρτήσεων (function call graphs) να ανιχνεύσουμε αυτές τις παραλλαγές. Ως επακόλουθο το πρόβλημα ανίχνευσης κακόβουλου λογισμικού μπορεί να διατυπωθεί ως πρόβλημα ταξινόμησης γράφων, όπου τμήματα κώδικα των οποίων δεν γνωρίζουμε την συμπεριφορά μπορούν να συγκριθούν με κακόβουλα και μη-κακόβουλα δείγματα. Από τα παραπάνω γίνεται φανερό ότι η ταξινόμηση γράφων, εξελίσσεται σε 'ζωτικό' πρόβλημα σε ένα μεγάλο εύρος εφαρμογών. Παράλληλα η ταξινόμηση γράφων είναι πολύ στενά συνδεδεμένη με το πρόβλημα της σύγκρισης γράφων, ένα κομβικό πρόβλημα στην θεωρία γράφων. Παρόλο που το πρόβλημα

αυτό μελετάται έντονα για πολλά χρόνια, μία γενικά αποδεκτή λύση τόσο σε σχέση με την περιγραφικότητα της όσο και σε σχέση με την αποδοτικότητα της δεν έχει βρεθεί. Ως επακόλουθο μπορούμε κάλλιστα να συμπεράνουμε ότι μία τέτοια λύση μπορεί να μην υπάρχει, υπόθεση που μας οδηγεί στην αποδοχή της διαφορετικότητας τόσο στον τρόπο απόδοσης όσο και στον βαθμό προσέγγισης των υπαρχόντων μεθόδων στην ολότητα τους. Παράλληλα η επισήμανση και η επανανοηματοδότηση της έννοιας του υπολογισμού στην σύγχρονη επιστημονικοτεχνολογική ανάπτυξη, οδηγεί στην ανάγκη ύπαρξης υπολογιστικών προτύπων τα οποία θα έχουν οριακά την ίδια θέση την οποία καταλάμβανε το πρότυπο χιλιόγραμμο στην επιστημονικοτεχνολογική ανάπτυξη της εποχής του. Η δημιουργία μίας βιβλιοθήκης για την επίλυση ενός προβλήματος δεν αποτελεί κατ΄ αυτόν τον τρόπο μονάχα μία τεχνική λύση σε ένα πρόβλημα, αλλά μία προκείμενη σε ένα επιστημονικό επιχείρημα. Ταυτόχρονα η εξέλιξη των γλωσσών προγραμματισμού και η επικράτηση του λογισμικού ανοιχτού κώδικα και των ηλεκτρονικών αποθετηρίων στην σύγχρονη ερευνητική πρακτική, έχει δώσει την δυνατότητα οι βιβλιοθήκες να είναι ανοιχτές σε χρήση και σε αλλαγή, με μεγάλη ευκολία, χωρίς φυσικά η τελευταία να μην είναι δέσμια των αντίστοιχων περιορισμών που προκύπτουν με την ίδια την ύπαρξη μίας κοινότητας (όπως η επιστημονική). Επιστημονικά περιοδικά (journals), στα οποία οι ερευνήτριες/ητές συνοψίζουν τα αποτελέσματα της ερευνητικής πρακτικής σε σχέση με τα εκάστοτε πεδία τους, συμμετέχοντας έτσι στην διαδικασία ταξινόμησης της γνώσης, έχουν αρχίσει να δίνουν αντίστοιχο χώρο δημοσίευσης στο ίδιο το λογισμικό τοποθετώντας στην διαδικασία επιλογής του κριτήρια εγκυρότητας, διαύγειας, νομικής άδειας χρήσης, δυνατότητα συμμετοχής και προσβασιμότητας στο επίπεδο του κώδικα (βλ. jmlr/mloss) κλπ.

## 1.1 Αντικείμενο της διπλωματικής

Ένας πολύ δημοφιλής τρόπος σύγκρισης γράφων, που αρχίζει να επικρατεί τα τελευταία χρόνια στο χώρο της μηχανικής μάθησης είναι οι 'πυρήνες γράφων' (graph kernels). Αυτά τα μέτρα ομοιότητας έχουν αποκτήσει θετική φήμη στην βιβλιογραφία τόσο για την μαθηματική θεμελίωση τους, που τους αποδίδει εγγύηση υπολογιστικής σύγκλισης (computational convergence guarantee) σε ποικίλα προβλήματα μάθησης, όσο και για την ύπαρξη και παρούσα ανάπτυξη μίας πληθώρας τέτοιων μεθόδων που η υπολογιστική τους πολυπλοκότητα ανήκει στην πολυωνυμική κλάση P. Παρόλο που έχουν μελετηθεί πολύ τα τελευταία χρόνια και ένα μεγάλο πλήθος αυτών των τεχνικών θεωρούνται σημαίνουσες και καθιερωμένες, δεν επιχειρήθηκε με συστηματικό τρόπο η συλλογή τους σε ένα ελεύθερο λογισμικό, αντικειμενοστρεφούς δομής, με δυνατότητα συλλογικής επεξεργασίας χρήσης από όλη την επιστημονική κοινότητα που να περιλαμβάνει ένα πλήρες εγχειρίδιο χρήσης/κατανόησης των τεχνικών αυτών για το ευρύ επιστημονικό κοινό. Είτε για να καλύψει μία ανάγκη, είτε για να δημιουργήσει μία επιθυμία το **GraKeL** σχεδιάστηκε στο πλαίσιο αυτής της διπλωματικής, προκειμένου να ικανοποιεί αυτήν την απαίτηση. Σχετικές απόπειρες (βλ. [75]) είναι ελλιπείς τόσο ως προς το περιεχόμενο και το εύρος απεύθυνσης τους αλλά και λόγω του ίδιου του τρόπου που είναι συσκευασμένος (packaging) ο κώδικάς τους. Στο πλαίσιο ανάπτυξης του παραπάνω λογισμικού έλαβε χώρα η εκτενής μελέτη της υπάρχουσας βιβλιογραφίας των πυρήνων γράφων και η επιλογή των

πιο σημαντικών, ενώ δόθηκε έμφαση στην βελτιστοποίηση της υλοποίησης τους σε επίπεδο κώδικα, τόσο όσον αφορά την αλγοριθμική τους πολυπλοκότητα, όσο και την πολυπλοκότητα *υλικού* κατά την χρήση υπάρχοντων υπολογιστικών εργαλείων. Παράλληλα με σκοπό να απευθύνεται σε ένα ευρύ κοινό προγραμματιστών, επιχειρήθηκε η συγγραφή ενός συστηματικού εγχειριδίου (documentation) τόσο για την χρήση όσο και για την συμμετοχή στην ανάπτυξη της ίδιας της βιβλιοθήκης. Η γλώσσα προγραμματισμού που επιλέχθηκε ήταν η python, μία γλώσσα δημοφιλής τόσο σε επιστημονικούς όσο και σε τεχνολογικούς κύκλους. Παρόλο που είναι χαρακτηρισμένη ως γλώσσα που ακολουθεί το πρότυπο του αντικειμενοστρεφούς προγραμματισμού (OOP) από τους κατασκευαστές της, είναι γνωστή ως *γλώσσα σεναρίων* (script language) από την προγραμματιστική κοινότητα μιάς και ακολουθεί εκτέλεση με διερμηνέα και οκνηρό (lazy) σύστημα τύπων, που δίνουν την ευκολία στον προγραμματιστή να αναπτύσσει γρήγορα ad-hoc εφαρμογές, ευδιάκριτα γραμμένες σε υψηλό επίπεδο (high level). Παράλληλα το οικοσύστημα της python (python ecosystem), το σύνολο δηλαδή των βιβλιοθηκών που περιλαμβάνει η γλώσσα, τόσο από τους ίδιους τους κατασκευαστές της όσο και μέσω τρίτων στο επίσημο αποθετήριο βιβλιοθηκών γνωστό ως PyPi, επιτρέπει και προκρίνει τη συνύπαρξη τόσο γενικού όσο και ειδικού σκοπού βιβλιοθηκών, κάνοντας ταυτόχρονα πολύ εύκολη την εγκατάσταση τους. Ταυτόχρονα πολλές δημοφιλείς βιβλιοθήκες επιστημονικού υπολογισμού (scientific computing - βλ. Numpy, Scipy κλπ) είναι είτε υλοποιημένες είτε εκτελέσιμες σε επίπεδο διεπαφής (interface) μέσω περιτύλιξης κώδικα (code wrapping) άλλων γλωσσών, σε περιβάλλον προγραμματισμού python.

## 1.2 Οργάνωση της διπλωματικής

Η παρούσα διπλωματική εργασία χωρίζεται σε τέσσερα κεφάλαια. Μία θεωρητική εισαγωγή στο χώρο των πυρήνων γράφων, καθώς και η παρουσίαση όλων των επιλεγμένων πυρήνων δίνεται στο κεφάλαιο 2. Η ανάλυση της σχεδίασης του λογισμικού, καθώς και της συσκευασίας και διανομής του γίνεται στο κεφάλαιο 3, ενώ η πειραματική του αξιολόγηση παρουσιάζεται στο κεφάλαιο 4. Τέλος στο κεφάλαιο 5 παρουσιάζεται η πειραματική αξιολόγηση του λογισμικού καθώς και τα συμπεράσματα της διπλωματικής, ενώ παρέχονται ιδέες και κατευθύνσεις για την προοπτική μελλοντικής εξέλιξής του.

# Κεφάλαιο 2

# Πυρήνες Γράφων

Στο κεφάλαιο αυτό θα περιγράψουμε τα κίνητρα που μας οδήγησαν στην σχεδίαση μίας βιβλιοθήκης για πυρήνες γράφων, ενώ θα κάνουμε μία εισαγωγή στην θεωρία γράφων και στην θεωρία ταξινόμησης και πυρήνων, ολοκληρώνοντας με μία θεωρητική παρουσίαση όλων των πυρήνων γράφων που επιλέχθηκαν από την βιβλιογραφία για υλοποίηση. Αρχικά παρουσιάζονται τα προβλήματα της σύγκρισης γράφων (graph comparison) και της ταξινόμησης γράφων (graph classification) δύο θεμελιώδη προβλήματα στην μηχανική μάθηση μέσω γράφων. Έπειτα θα δοθούν ορισμοί θεμελιωδών εννοιών της θεωρίας γράφων. Εν συνεχεία, παρέχονται λεπτομέρειες σχετικά με το πρόβλημα της ταξινόμησης γράφων που αποτελεί τον βασικό τρόπο με τον οποίο θα συγκρίνουμε τα αποτελέσματα των πυρήνων μέσα στο εύρος της διπλωματικής. Θα παρουσιάσουμε τον ταξινομητή Μηχανών Διανυσμάτων Υποστήριξης (Support Vector Machine classifier) που αποτελεί τον βασικό ταξινομητή που χρησιμοποιήσαμε σε όλα τα πειράματα. Σημαντικό μέρος καταλαμβάνει στο τέλος αυτού του κεφαλαίου, η θεωρητική παρουσίαση όλων των πυρήνων γράφων που υλοποιήθηκαν στο σώμα της παρούσας εργασίας.

## 2.1 Κίνητρο

Τα τελευταία χρόνια έχουν πληθύνει ραγδαία τα διαθέσιμα δεδομένα που μπορούν να αποδοθούν φυσικά ως γράφοι. Τέτοιοι τύποι δεδομένων έχουν προκύψει σε πολλά πεδία εφαρμογών από τα κοινωνικά δίκτυα μέχρι την βιολογία και την χημεία. Τα κοινωνικά δίκτυα μπορούν να μοντελοποιήσουν μία μεγάλη πληθώρα καταστάσεων, και χρησιμοποιούνται για να αναπαραστήσουν τις σχέσεις μεταξύ ζευγαριών από ένα σύνολο ατόμων. Τέτοιες σχέσεις μπορεί να αποτελούν την φιλία σε μία κοινωνική ιστοσελίδα, ή ακόμα τις συνεργασίες με ιδρύματα και άλλους ακαδημαϊκούς σε μία ιστοσελίδα που καταγράφει την δραστηριότητα της ακαδημαϊκής κοινότητας. Στη χημεία, στην παραδοσιακή προσέγγιση, οι γράφοι έχουν χρησιμοποιηθεί κατά παράδοση, προκειμένου να αναπαραστήσουν μόρια όπου συνήθως οι κόμβοι αντιστοιχούν στα άτομα και οι ακμές στους χημικούς δεσμούς. Άλλο ένα πεδίο πλούσιο σε δεδομένα γράφων είναι η βιολογία, όπου οι γράφοι χρησιμοποιούνται συχνά για να αναπαραστήσουν DNA ακολουθίες, φυλογενετικά δίκτυα, μεταβολικά δίκτυα, γενετικά ρυθμιστικά δίκτυα, και δίκτυα πρωτεϊνικών αλληλεπιδράσεων. Αναπαραστάσεις με τη μορφή γράφων εντοπίζουμε και

σε αρκετά τεχνολογικά δίκτυα. Το πιο άμεσο παράδειγμα είναι το διαδίκτυο, στο οποίο οι κόμβοι ανταποκρίνονται σε ιστοσελίδες και οι ακμές σε υπερσυνδέσμους (hyperlinks) μεταξύ ιστοσελίδων.

Από το παραπάνω γίνεται εύλογο, ότι η αναπαράσταση γράφων προτιμάται για δεδομένα με δομή που συμπίπτει με αυτή των γράφων. Ωστόσο παρόλο που συμπίπτει, θα μπορούσαν να αναπαρασταθούν από διανύσματα χαρακτηριστικών (feature vectors). Μία ερώτηση που φαίνεται λοιπόν δόκιμη σε αυτό το σημείο είναι: 'ποια είναι τα προτερήματα της αναπαράστασης με γράφους;' και μία δεύτερη είναι: 'γιατί θα έπρεπε να προτιμήσει κάποιος να αναπαραστήσει τα δεδομένα του ως γράφους και όχι ως διανυσματικές αναπαραστάσεις;', οι οποίες αποτελούν μία πολύ κοινή αναπαράσταση στις κοινότητες της εξόρυξης γνώσης από τα δεδομένα (data mining) και της μηχανικής μάθησης (machine learning).

### 2.1.1 Η Αναπαράσταση Γράφου

Δεν υπάρχει αμφιβολία ότι οι γράφοι χαρακτηρίζονται από πολύ υψηλή αναπαραστατική δυνατότητα. Ένας γράφος δεν αναπαριστά μονάχα οντότητες αλλά και τις σχέσεις τους. Με άλλα λόγια, οι γράφοι περιγράφουν αναλυτικά τις σχέσεις μεταξύ διαφόρων μερών ενός αντικειμένου. Μερικές ευρέως διαδεδομένες δομές δεδομένων μπορούν να αναπαρασταθούν με τη μορφή γράφων [12]. Για παράδειγμα ένα διάνυσμα μπορεί να αναπαρασταθεί σαν ένας γράφος, όπου οι ακμές αντιστοιχούν στα μέρη του διανύσματος ενώ η αλληλουχία τους συμβολίζεται με ακμές. Ένας πίνακας αντιστοίχισης (associative array ή αλλιώς map) μπορεί να μοντελοποιηθεί σαν ένας γράφος όπου οι κόμβοι αντιστοιχούν στα κλειδιά (keys) και στις τιμές (values) και κάθε κλειδί συνδέεται με μία τιμή μέσω μία κατευθυνόμενης (directed) ακμής. Οι συμβολοσειρές (strings) μπορούν επίσης να αναπαρασταθούν σαν γράφοι, όπου κάθε κόμβος είναι ένας χαρακτήρας και οι ακμές συμβολίζουν την σειρά με την οποία συνδέονται. Λόγω της εκφραστικότητας των γράφων ακόμα και δεδομένα που δεν εμφανίζουν εγγενώς δομή που μοιάζει με αυτή του γράφου, είναι χρήσιμο να απεικονίζονται κατά αυτόν τον τρόπο. Ένα πολύ κοινό παράδειγμα είναι δεδομένα κειμένου, στα οποία οι γράφοι χρησιμοποιούνται για να αναπαραστήσουν σχέσεις μεταξύ γλωσσολογικών τμημάτων.

### 2.1.2 Γράφοι ή Διανύσματα Χαρακτηριστικών

Στον χώρο της εξόρυξης γνώσης από τα δεδομένα και στη μηχανική μάθηση τα δεδομένα αναπαρίστανται συνήθως μέσω διανυσμάτων. Σε αντίθεση με τις διανυσματικές αναπαραστάσεις οι γράφοι προσφέρουν μεγάλη ευελιξία. Συγκεκριμένα, οι γράφοι ξεπερνούν μία σειρά από όρια εγγενή στις διανυσματικές αναπαραστάσεις. Όπως σχολιάστηκε παραπάνω, οι γράφοι περιγράφουν ταυτόχρονα και τις οντότητες και τις σχέσεις τους. Ως επακόλουθο δεν περιγράφουν απλώς *τιμές* (διαφόρων οντοτήτων) σε ένα χώρο, αλλά *δομή*. Ακόμα σε αντίθεση με την ανάγκη των διανυσμάτων να έχουν το ίδιο μέγεθος για όλα τα αντικείμενα που λαμβάνονται υπόψη, στους γράφους είναι δυνατό να έχουμε διακυμάνσεις στον αριθμό των κόμβων και στον αριθμό των ακμών, δίνοντας έτσι τη δυνατότητα στο μέγεθος του γράφου να προσαρμόζεται καλύτερα στο μέγεθος και την πολυπλοκότητα κάθε τμήματος των δεδομένων (που

αποτελεί ξεχωριστή οντότητα). Με βάση τα παραπάνω, οι γράφοι μοιάζουν με την ιδανική αναπαράσταση δεδομένων σε διάφορους επιστημονικούς τομείς. Εντούτοις η αναπαράσταση μέσω γράφων έχει τα μειονεκτήματα της: οι γράφοι δεν μπορούν να ενταχθούν στον πολύ πλούσιο φορμαλισμό των διανυσματικών χώρων, ενώ ακόμα πολλοί τελεστές (operators) μεταξύ γράφων είναι υπολογιστικά ακριβείς, αντίθετα με την απλή μαθηματική τους διατύπωση. Το πιο χαρακτηριστικό παράδειγμα αυτών των τελεστών αφορά αυτόν που υπολογίζει αν δύο αντικείμενα είναι ταυτόσημα. Στην περίπτωση δύο διανυσμάτων η πράξη της ισότητας, μεταφράζεται στην ισότητα όλων των συστατικών τους στοιχείων, υπολογισμός γραμμικός ως προς το μέγεθός τους. Για την αντίστοιχη πράξη στην θεωρία γράφων γνωστή ως ισομορφισμός (graph isomorphism, βλέπε ορισμό 2.13), δεν έχουν βρεθεί μέχρι στιγμής αλγόριθμοι πολυωνυμικού χρόνου. Γενικότερα το πρόβλημα της σύγκρισης δύο αντικειμένων, είναι πολύ πιο ασθενώς διατυπωμένο στους γράφους σε σχέση με τα διανύσματα. Στους κοινούς διανυσματικούς χώρους που συναντάμε στην πράξη, η απόσταση μπορεί να υπολογιστεί αποδοτικά χρησιμοποιώντας την γενικά αποδεκτή μετρική της ευκλείδειας απόστασης. Δυστυχώς μία αντίστοιχα 'κοινή' μετρική δεν υπάρχει στην περίπτωση των γράφων. Σ' αυτό οφείλεται το γεγονός ότι δεν υπάρχει κανονική διάταξη (canonical ordering) μεταξύ των κόμβων του γράφου και άρα απουσιάζει μία ¨ένα προς ένα' αντιστοίχιση μεταξύ των κόμβων δύο οποιωνδήποτε γράφων. Επιπροσθέτως, η αναγνώριση κοινών μερών μεταξύ δύο γράφων, δεν είναι εξίσου υπολογιστικά προσβάσιμη, μιάς και αν ένας γράφος έχει $n$ κόμβους υπάρχουν $2^n$ δυνατά υποσύνολα αυτών. Συνεπώς ο χώρος αναζήτησης είναι, καταρχήν, εκθετικός αν θεωρήσουμε ότι ελέγχουμε όλα τα δυνατά υποσύνολα. Ακόμα πιο ενδιαφέρον παρουσιάζει το γεγονός ότι ακόμα και για κάποιες διαισθητικά και υπολογιστικά κοινές πράξεις σε διανυσματικούς χώρους όπως το άθροισμα και η διαφορά, δεν ορίζονται οι αντίστοιχες πράξεις στους γράφους. Παρόλο λοιπόν που οι γράφοι αποτελούν έναν πολύ 'άμεσο' τρόπο αναπαράστασης δεδομένων δεν επικράτησαν για όλους τους παραπάνω λόγους, ως ο κύριος τρόπος αναπαράστασης δεδομένων στην επιστήμη υπολογιστών.

### 2.1.3 Μάθηση με Αναπαραστάσεις Γράφων

Ο τρόπος με τον οποίο αναπαριστώνται τα δεδομένα είναι κομβικός στους χώρους της εξόρυξης γνώσης από τα δεδομένα και της μηχανικής μάθησης. Κάθε αλγόριθμος είναι σχεδιασμένος για δεδομένα μίας συγκεκριμένης αναπαράστασης. Λόγω της ευελιξίας των γράφων, κάποιος θα περίμενε ότι θα υπάρχει μεγάλη πρόοδος στην ανάπτυξη αλγορίθμων που μπορούν να δέχονται, ως είσοδο, δεδομένα που έχουν αναπαρασταθεί ως γράφοι. Αντίθετα κάτι τέτοιο δεν συμβαίνει, μιας και λόγω της συνδυαστικής (combinatorial) φύσης των γράφων, αυτή η πρόκληση δεν λήφθηκε ποτέ σοβαρά υπόψη. Ως επακόλουθο η έρευνα σε αυτές τις περιοχές εστιάστηκε κυρίως σε αλγορίθμους μεταξύ διανυσμάτων, μιας και τα διανύσματα παρουσιάζουν πολλές ενδιαφέροντες μαθηματικές ιδιότητες, ενώ οι πράξεις και ο χειρισμός παρουσιάζει πολύ μικρότερη υπολογιστική πολυπλοκότητα. Στους σύγχρονους υπολογιστές οι στοιχειώδεις πράξεις μεταξύ διανυσμάτων, όπως η πρόσθεση και ο πολλαπλασιασμός γίνονται σε επίπεδο επεξεργαστή. Συνεπώς, δεν μας παραξενεύει το γεγονός ότι οι πιο δημοφιλείς αλγόριθμοι μηχανικής μάθησης είναι σχεδιασμένοι ώστε να λαμβάνουν στην είσοδο τους διανυσματικές

αναπαραστάσεις των εκάστοτε δεδομένων. Ακόμα και σε εφαρμογές που οι γράφοι είναι η φυσική αναπαράσταση των δεδομένων, γίνονται απόπειρες να αναπαρασταθούν σαν διανύσματα χαρακτηριστικών, προκειμένου να χρησιμοποιηθούν υπάρχουσες τεχνικές, αντί να σχεδιαστούν αλγόριθμοι που δέχονται στην είσοδο τους γράφους. Ιδανικά, θα θέλαμε να υπάρχει ένας τρόπος να μετασχηματίσουμε τους γράφους σε διανύσματα χαρακτηριστικών, χωρίς να χάνουμε το αναπαραστατικό τους πλεονέκτημα. Τούτη η επιθυμία, μπορεί εύκολα να απορριφθεί αν σκεφτούμε πως αν μπορούσαμε με έναν προφανή ή 'υπολογιστικά εύλογο' τρόπο να παραστήσουμε γράφους σε ένα διανυσματικό χώρο, όλα τα προαναφερθέντα σημασιολογικά και υπολογιστικά προβλήματα δεν θα λάμβαναν χώρα. Συγκεκριμένα, η άμεση αναπαράσταση δεδομένων σαν διανύσματα υστερεί της πλούσιας τοπολογικής πληροφορίας που κωδικοποιούν οι γράφοι και ως επακόλουθο αποτελεί μία υποδεέστερη λύση. Σε γενικές γραμμές αληθεύει το γεγονός ότι ο σχεδιασμός ενός αλγορίθμου με γράφους αυξάνει είτε την σημασιολογική πολυπλοκότητα κατά την σχεδίαση είτε την υπολογιστική πολυπλοκότητα κατά την εκτέλεση, ιδιαιτέρως όταν αυτό γίνεται για δεδομένα με μεγάλο μέγεθος. Αντίθετα, τέτοιοι αλγόριθμοι έχουν μεγαλύτερη αποτελεσματικότητα, συμπεριλαμβανομένης και της ικανότητας *γενίκευσης*, σε σχέση με ανταγωνιστικούς αλγορίθμους, στην αντιμετώπιση προβλημάτων που δέχονται πολλές προσεγγίσεις, όπως π.χ. η κατηγοριοποίηση κειμένων (text classification). Συνεπώς η έξυπνη σχεδίαση, η αποδοτική υλοποίηση και η εύληπτη διάδοση τεχνικών για γράφους, μπορούν άμεσα να ωφελήσουν εφαρμογές στις οποίες τα δεδομένα εμφανίζονται κατά κύριο λόγο σε μορφή γράφων.

## 2.2 Εισαγωγικά Σημεία από την Θεωρία Γράφων

**Ορισμός 2.1** (Γράφος)**.** *Ένας γράφος $G = (V, E)$ αποτελείται από ένα σύνολο κόμβων (vertices ή nodes) $V$ και ένα σύνολο από ακμές (edges) $E = e \subseteq V, |e| = 2$ μεταξύ τους.*

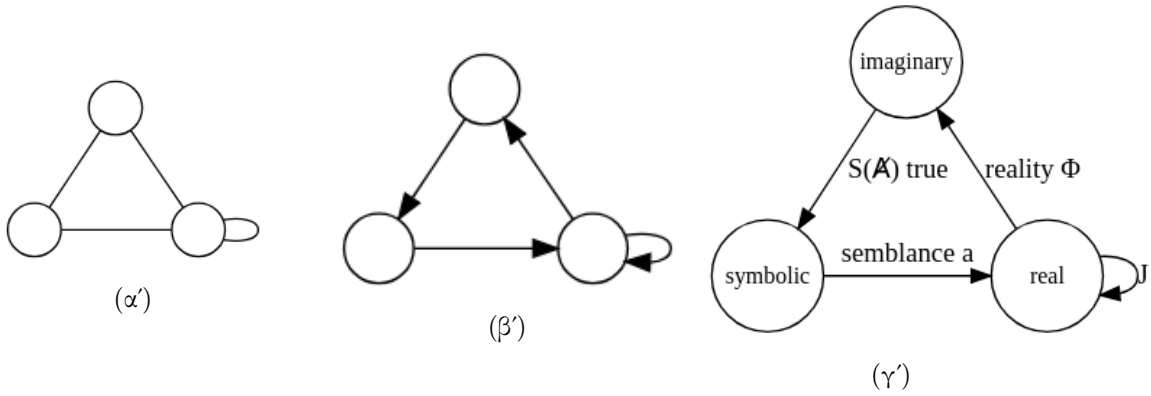
Το μέγεθος ενός γράφου $|V|$ αντιστοιχεί σε ένα σύνολο κόμβων συμβολίζουμε με $n$. Όσον αφορά τον αριθμό ακμών $|E|$ του γράφου, θα τον συμβολίζουμε με $m$. Ένα απλό παράδειγμα γράφου δίνεται στο σχήμα 2.1α′. Αν τώρα οι ακμές του γράφου θέλουμε να έχουν κατεύθυνση (βλ. σχήμα 2.1β′) φτάνουμε στον ορισμό των κατευθυνόμενων γράφων.

**Ορισμός 2.2** (Κατευθυνόμενος και μη Κατευθυνόμενος Γράφος)**.** *Ένας γράφος $G_d = (V_d, E_d)$ είναι κατευθυνόμενος αν οι ακμές του έχουν μία κατεύθυνση, δηλαδή όταν το $E_d$ είναι ένα σύνολο από διατεταγμένα ζευγάρια κόμβων. Οι ακμές ενός κατευθυνόμενου γράφου λέγονται τόξα (arcs). Ένας μη κατευθυνόμενος γράφος είναι κατευθυνόμενος $G = (V, E)$ όπου:*

$$\forall v_i, v_j \in V : (v_i, v_j) \in E \Leftrightarrow (v_j, v_i) \in E$$

Καθ′ όλη την θεωρητική ανάλυση αυτής της διπλωματικής, η λέξη γράφος θα αναφέρεται αποκλειστικά σε μη-κατευθυνόμενους γράφους. Θέλοντας να συμπεριλάβουμε περαιτέρω πληροφορία κατά την αναπαράσταση ενός γράφου φτάνουμε στον ορισμό του γράφου με επισημειώσεις.

**Ορισμός 2.3** (Γράφοι με Επισημειώσεις (Labels) ). *Επισημειωμένος λέγεται ένας γράφος $G = (V, E)$ συνδεδεμένος με μία συνάρτηση $\mathcal{L} : V \cup E \to L$ που αντιστοιχίζει κάθε κόμβο και ακμή του γράφου με μία επισημείωση από το σύνολο των επισημειώσεων $L$.*




(α′) (β′) (γ′)

Σχήμα 2.1: Παράδειγμα: Η *Λακανική τριάδα* σε διαδοχικές πιο πλούσιες σε πληροφορία αναπαραστάσεις: α′: γράφος, β′: κατευθυνόμενος γράφος, γ′: γράφος με επισημειώσεις στις ακμές και στους κόμβους

Ένας γράφος με επισημειώσεις στους κόμβους του λέγεται επίσης επισημειωμένος ανά κόμβο (node-labeled). Παρόμοια ένας γράφος με επισημειώσεις στις ακμές του λέγεται επισημειωμένος ανά ακμή (edge-labeled). Ένας γράφος με επισημειώσεις και στους κόμβους και στις ακμές λέγεται πλήρως επισημειωμένος (βλ. σχήμα 2.1γ′). Στην παρούσα διπλωματική θα μας απασχολήσουν και τα τρία παραπάνω είδη επισημειωμένων γράφων. Οι επισημειώσεις μπορούν να είναι είτε διακριτές (discrete) είτε συνεχής (continuous). Οι συνεχείς επισημειώσεις ονομάζονται και χαρακτηριστικά (attributes).
Μία εναλλακτική και πολύ κοινή αναπαράσταση της δομής ενός γράφου είναι ο πίνακας γειτνίασης (adjacency matrix).

**Ορισμός 2.4** (Πίνακας γειτνίασης). *Δεδομένου ενός γράφου $G = (V, E)$, ο πίνακας γειτνίασης του $A_G$ του γράφου $G$ είναι ένας δισδιάστατος πίνακας $|V| \times |V|$, όπου με $A_{ij}$ συμβολίζουμε το στοιχείο στην $i$-οστή γραμμή και $j$-οστή στήλη του πίνακα τότε:*

$$A_{ij} = \begin{cases} 1, \{v_i, v_j\} \in E \\ 0, otherwise \end{cases} \tag{2.1}$$

Μία έννοια πολύ κοινή στην ορολογία των γράφων, είναι αυτή της γειτονιάς.

**Ορισμός 2.5** (Γειτονιά). *Δεδομένου ενός μη κατευθυνόμενου γράφου $G = (V, E)$ και ενός κόμβου $v_i \in V$, η γειτονιά $\mathcal{N}(v_i)$ του $v_i$, ορίζεται ως $\mathcal{N}(v_i) = \{v_j : \{v_i, v_j\} \in E\}$, όπου $\{v_i, v_j\}$ είναι μία ακμή μεταξύ δύο κόμβων $v_i$ και $v_j$ του $V$ και αντιστοιχεί στο σύνολο των κόμβων που συνδέονται με το $v_i$ μέσα στο $G$.*

Μία έννοια στενά συνδεδεμένη με την γειτονιά είναι ο βαθμός (degree).

**Ορισμός 2.6** (Βαθμός). *Δεδομένου ενός μη κατευθυνόμενου γράφου $G = (V, E)$ και ενός κόμβου $v_i \in V$, ο βαθμός του $v_i$ στο $G$ ορίζεται ως*

$$d_G(v_i) = |\{v_j : \{v_i, v_j\} \in E\}| = |\mathcal{N}(v_i)|$$

*και αντιστοιχεί στον αριθμό των κόμβων που συνδέονται με το $v_i$,*

Για κατευθυνόμενους γράφους ορίζουμε στην ίδια λογική, εισερχόμενος-βαθμός (in-degree) και εξερχόμενος-βαθμός (out-degree) για τις ακμές που εισέρχονται και εξέρχονται αντίστοιχα. Κομβική για την θεωρία γράφων, είναι και η έννοια του υπογραφήματος.

**Ορισμός 2.7** (Υπογράφημα). *Δεδομένου ενός γράφου $G = (V, E)$, ένας γράφος $G' = (V', E')$ θα λέγεται υπογράφημα (subgraph) του $G$ και θα συμβολίζεται με $G' \subseteq G$, αν ισχύει $V' \subseteq V$ και $E' \subseteq E$. Στην περίπτωση που το παραπάνω συνδέεται άμεσα με ένα υποσύνολο των κόμβων και όχι των ακμών, φτάνουμε στον ορισμό του επαγόμενου υπογραφήματος.*

**Ορισμός 2.8** (Επαγόμενο Υπογράφημα). *Δεδομένου ενός γράφου $G = (V, E)$ και ενός υποσυνόλου κόμβων $U \subseteq V$, το υπογράφημα $G(U) = (U, E(U))$ θα λέγεται επαγόμενο (induced) δεδομένου ότι οι ακμές του $E(U)$ είναι ακριβώς όλες εκείνες οι ακμές που ανήκουν στο $E$ και συνδέουν κόμβους που ανήκουν και οι δύο στο $U$, ως εξής:*

$$E(U) = \{\{v_i, v_j\} \in E : v_i, v_j \in U\}\} \tag{2.2}$$

Ο βαθμός ενός κόμβου $v_i \in U, d_G(U)(v_i)$, ισούται με τον αριθμό κόμβο που είναι γειτονικοί του $v_i$ στον $G(U)$. Μία μετρική που περιγράφει συνολικά αν το πλήθος των βαθμών προσεγγίζουν το μέγιστο, είναι αυτή της πυκνότητας.

**Ορισμός 2.9** (Πυκνότητα). *Δεδομένου ενός γράφου $G = (V, E)$ ορίζουμε ως πυκνότητα (density) του γράφου $G$ τον αριθμό $\delta(G) = \frac{|E|}{|V|^2}$.*

Ένας γράφος στον οποίο ισχύει $\delta(G) = 1$, λέγεται πλήρης γράφος. Σε ένα πλήρη γράφο κάθε ζευγάρι κορυφών συνδέονται. Με βάση αυτή τη συνθήκη και της έννοιας του επαγόμενου υπογράφου, φτάνουμε στον ορισμό της κλίκας.

**Ορισμός 2.10** (Κλίκα). *Δεδομένου ενός γράφου $G = (V, E)$ θα ονομάζουμε κλίκα (clique) ένα υποσύνολο κόμβων $C \subseteq V$, τέτοιο ώστε $\delta(G(C)) = 1$.*

Θεμελιώδες έννοιες για την ανάλυση και εξερεύνηση ενός γράφου, είναι αυτές του περιπάτου, του μονοπατιού και του κύκλου.

**Ορισμός 2.11** (Περίπατος, Μονοπάτι, Κύκλος). *Ένας περίπατος (walk) σε ένα γράφος $G = (V, E)$ είναι μία ακολουθία από κόμβους $v_1, v_2, \ldots, v_{l+1}$ όπου $v_i \in V$ για όλα τα $1 \leq i \leq l + 1$ και $\{v_i, v_{i+1}\} \in E$ για όλα τα $1 \leq i \leq k$. Το μήκος ενός περιπάτου είναι ίσο με τον αριθμό των ακμών στην ακολουθία, δηλαδή $l$. Ένας περίπατος στον οποίο ισχύει*

$$v_i = v_j \Leftrightarrow i = j$$

*ονομάζεται μονοπάτι (path). Ένας κύκλος (cycle) είναι ένα μονοπάτι για το οποίο ισχύει $\{v_{k+1}, v_1\} \in E$.*

Ενώ με βάση αυτές μπορούμε να ορίσουμε μία έννοια απόστασης για δύο κόμβους, το κοντινότερο μονοπάτι.

**Ορισμός 2.12** (Κοντινότερο Μονοπάτι). *Κοντινότερο μονοπάτι (shortest path) μεταξύ δύο κόμβων $v_i, v_j$, ενός γράφου $G$ είναι ένα μονοπάτι από το $v_i, v_j$, τέτοιο ώστε δεν υπάρχει άλλο μονοπάτι μεταξύ αυτών των δύο κορυφών με μικρότερο μήκος.*

*Διάμετρος* ενός γράφου $G$ είναι το μήκος του μεγαλύτερου ελαχίστου μονοπατιού μεταξύ κάθε ζευγάρι κόμβων στον γράφο $G$. Τελευταία και μη εξαιρετέα είναι η σχέση που προσδιορίζει αν δύο γράφοι ταυτίζονται, γνωστή ως ισομορφισμός.

**Ορισμός 2.13** (Ισομορφισμός). *Ένας ισομορφισμός μεταξύ δύο επισημειωμένων γράφων $G = (V, E)$ και $G' = (V', E')$ είναι μία "1-1' απεικόνιση $\phi : V \to V'$ που διατηρεί τη γειτνίαση, δηλαδή $\forall v, u \in V : (v, u) \in E \Leftrightarrow (\phi(v), \phi(u)) \in E'$ και τις επισημειώσεις, δηλαδή αν $\psi \in V \times V \to V' \times V'$ είναι η αντιστοίχιση των ζευγαριών κόμβων όπως προκύπτει από την "1-1' απεικόνιση $\phi$ τέτοια ώστε $\psi((v, u)) = (\phi(v), \phi(u))$, τότε οι συνθήκες $\forall v \in V : \ell(v) \equiv \ell(\phi(v))$ και $\forall e \in E : \ell(e) \equiv \ell(\psi(e))$ πρέπει να ικανοποιούνται, όπου με $\equiv$ συμβολίζουμε ότι δύο επισημειώσεις ταυτίζονται.*

## 2.3 Προβλήματα Μηχανικής Μάθησης

Το σύνολο των προβλημάτων στον χώρο της μηχανικής μάθησης είναι τριχοτομημένο στις εξής μεγάλες κατηγορίες (1) επιβλεπόμενη (supervised) μάθηση, (2) μη-επιβλεπόμενη (unsupervised) μάθηση και (3) ενισχυτική (reinforcement) μάθηση. Στην *επιβλεπόμενη* μάθηση, ο στόχος είναι η μάθηση μίας αντιστοίχισης μεταξύ εισόδου εξόδου, δεδομένου ενός συνόλου ζευγαριών τιμών εισόδου - εξόδου, γνωστό ως σύνολο εκπαίδευσης. Στην περίπτωση που έχουμε διακριτή έξοδο, το πρόβλημα αυτό ονομάζεται *πρόβλημα ταξινόμησης*. Ένα γνωστό πρόβλημα ταξινόμησης είναι αυτό της αναγνώρισης χειρόγραφων ψηφίων. Οι είσοδοι αντιστοιχούν σε εικόνες χειρόγραφων ψηφίων και οι έξοδοι ή αλλιώς οι *επισημειώσεις* των κατηγοριών στο νούμερο του ψηφίου από το οποίο έχουν προέλθει οι αναπαραστάσεις. Δεδομένου ενός εκπαιδευτικού συνόλου εικόνων και των κατηγορικών επισημειώσεων τους, ο στόχος είναι η μάθηση μία αντιστοίχισης από εικόνες σε κατηγορικές επισημειώσεις, που μπορούν να χρησιμοποιηθούν για να αναγνωρίσουμε σε ποια κατηγορία ανήκουν νέες εικόνες. Αν η έξοδος αποτελείται από μία ή παραπάνω συνεχείς μεταβλητές, το πρόβλημα είναι γνωστό ως πρόβλημα παλινδρόμησης (regression). Ένα παράδειγμα ενός τέτοιου προβλήματος μπορεί να είναι το εξής: δεδομένου ψυχοφυσικών επισημειώσεων σχετικά με το αν μία μουσική μελωδία είναι αγχωτική με συνεχείς μετρήσεις από το 0 έως το 5, πρόβλεψη του βαθμού άγχους που προκαλεί μία μελωδία, έχοντας ως είσοδο ένα συνόλο χαρακτηριστικών που εξάγουμε από το ηχητικό (ή μελωδικό) της σήμα. Στην *μη-επιβλεπόμενη* μάθηση, μας δίνονται μόνο είσοδοι και ο στόχος μας είναι να εξάγουμε χρήσιμα πρότυπα (patterns) από αυτές. Ένα παράδειγμα μη επιβλεπόμενης μάθησης είναι η συσταδοποίηση (clustering), που αφορά τον διαχωρισμό των δεδομένων εισόδου σε ομάδες, έτσι ώστε τα δεδομένα εισόδου που καταλήγουν να είναι στην ίδια ομάδα, να είναι κατά μία έννοια πιο όμοια μεταξύ τους από αυτά που ανήκουν σε

άλλες ομάδες. Άλλο ένα πρόβλημα, γνωστό ως εκτίμηση πυκνότητας (density estimation) αφορά τον προσδιορισμό μίας 'κρυφής' συνάρτησης πυκνότητας πιθανότητας, δεδομένου ενός συνόλου δεδομένων εισόδου. Τέλος, όσον αφορά την *ενισχυτική* μάθηση, στόχος είναι να προσδιορίσουμε ποιες δράσεις πρέπει να λάβουμε σε μία δεδομένη κατάσταση προκειμένου να μεγιστοποιήσουμε μία μετρική που ποσοτικοποιεί κάποια έννοια συνολικής ανταμοιβής. Σε αντίθεση με την επιτηρούμενη μάθηση, οι ιδανικές έξοδοι δεν είναι γνωστές εξαρχής, αλλά αποκαλύπτονται την ώρα της ίδιας της διαδικασίας μάθησης καθώς το σύστημα αλληλεπιδρά με ένα άλλο σύστημα-περιβάλλον, το οποίο το ανταμείβει ή όχι με βάση την κατάσταση του κάθε στιγμή.

### 2.3.1 Προβλήματα Μάθησης με Γράφους

Η μάθηση σε γράφους έχει συγκεντρώσει μεγάλο ερευνητικό ενδιαφέρον τα τελευταία χρόνια. Όπως σχολιάστηκε παραπάνω, λόγω των υψηλών αναπαραστατικών δυνατοτήτων τους, οι γράφοι χρησιμοποιούνται για να αναπαραστήσουν δεδομένα από πολύ διαφορετικές πηγές. Ως επακόλουθο δεν μας εκπλήσσει το γεγονός ότι μία πληθώρα προβλημάτων μάθησης είναι ορισμένα για γράφους. Τα ακόλουθα τέσσερα προβλήματα είναι ίσως αυτά που έχουν μελετηθεί περισσότερο στην βιβλιογραφία:

- Ταξινόμηση Κόμβων: δεδομένου ενός γράφου με επισημειώσεις σε ένα γνήσιο υποσύνολο των κόμβων, επισημείωσε αποτελεσματικά τους υπόλοιπους.
- Πρόβλεψη σύνδεσης: δεδομένου ενός συνόλου κόμβων, πρόβλεψε αν πρέπει να συνδεθούν με μία ακμή.
- Ταξινόμηση Γράφων: δεδομένου ενός συνόλου γράφων με γνωστή κατηγοριοποίηση για ένα γνήσιο υποσύνολο τους, βρες σε ποιές κατηγορίες ανήκουν οι υπόλοιποι.
- Συσταδοποίηση Γράφων: δεδομένου ενός γράφου, ομαδοποίησε όλους τους κόμβους του σε συστάδες, λαμβάνοντας υπόψιν την δομή των ακμών του κατά τέτοιο τρόπο ώστε να υπάρχουν πολλές ακμές εσωτερικά κάθε συστάδας, και σχετικά λίγες μεταξύ τους.

Τα τρία πρώτα προβλήματα είναι προβλήματα *επιτηρούμενης* μάθησης, ενώ το τελευταίο είναι ένα πρόβλημα *μη-επιτηρούμενης* μάθησης. Στο εύρος αυτής της διπλωματικής θα μας απασχολήσει μόνο το τρίτο πρόβλημα, συγκεκριμένα αυτό της ταξινόμησης γράφων.

### 2.3.2 Το Πρόβλημα Ταξινόμησης Γράφων

Το πρόβλημα ταξινόμησης είναι το πιο συχνά εμφανιζόμενο πρόβλημα στο χώρο της μηχανικής μάθησης. Σε ένα πρόβλημα *ταξινόμησης*, στόχος είναι η μάθηση μίας αντιστοίχισης μεταξύ εισόδου-εξόδου, δεδομένου ενός συνόλου εκπαίδευσης. Στα παρακάτω, θα συμβολίζουμε την είσοδο με $x$ και την έξοδο με $y$. Οι είσοδοι συχνά αποκαλούνται και παραδείγματα ή στιγμιότυπα του προβλήματος, τη στιγμή που οι έξοδοι συνήθως αποκαλούνται επισημειώσεις κατηγοριών. Στις περισσότερες περιπτώσεις, κάθε είσοδος $x$ αναπαρίσταται σαν ένα διάνυσμα

πραγματικών αριθμών. Από την άλλη θα μπορούσε να είναι οτιδήποτε π.χ. μία εικόνα, ένα κείμενο ή ένας γράφος.

Έστω ότι το $\mathcal{X}$ είναι ένα σύνολο αντικειμένων που θα θέλαμε να ταξινομήσουμε. Ως επακόλουθο $x \in \mathcal{X}$. Οι έξοδοι παίρνουν τις τιμές τους από ένα πεπερασμένο σύνολο $y \in \mathcal{Y} = C_1, \ldots, C_{|Y|}$, όπου το $C$ είναι ένα πλήθος από κατηγορίες. Αν το $|\mathcal{U}| = 2$ τότε το προκύπτον πρόβλημα ονομάζεται πρόβλημα *δυαδικής* ταξινόμησης. Εναλλακτικά, αν $|\mathcal{Y}| > 2$, λέγεται *πολυ-κατηγορική* ταξινόμηση. Πάντοτε σε ένα πρόβλημα ταξινόμησης μας δίνεται ένα σύνολο $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$ όπου $(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$, $N$ παρατηρήσεων μαζί με τις επισημειώσεις των κατηγοριών στις οποίες ανήκουν. Θα υποθέσουμε ότι τα δεδομένα εκπαίδευσης $\mathcal{D}$ προέκυψαν από μία άγνωστη κατανομή. Με βάση αυτό το σύνολο εκπαίδευσης, ο στόχος είναι να μάθουμε μία συνάρτηση $f : \mathcal{X} \to \mathcal{Y}$ που ελαχιστοποιεί το σφάλμα λανθασμένης αντιστοίχισης, δεδομένης της άγνωστης κατανομής. Έχοντας υπολογίσει την συνάρτηση $f$, μπορούμε να την χρησιμοποιήσουμε για να κάνουμε προβλέψεις σε νέα δεδομένα. Συνήθως μας δίνεται και ένα άλλο σύνολο ζευγαριών εισόδων-εξόδων, που περιέχει διαφορετικές εξόδους από αυτές που χρησιμοποιήθηκαν κατά την εκπαίδευση. Αυτό το σύνολο λέγεται πειραματικό σύνολο και συνήθως χρησιμοποιείται για την εκτίμηση της ποιότητας της μάθησης. Ιδανικά θα θέλαμε η $f$ να μπορεί να ταξινομεί σωστά νέα δείγματα.

Στην περίπτωση που τα δεδομένα εισόδου είναι γράφοι, το πρόβλημα λέγεται ταξινόμηση γράφων. Συγκεκριμένα, δεδομένου ενός συνόλου εκπαίδευσης $\mathcal{D} = \{(G_i, y_i)\}_{i=1}^N$ που αποτελείται από $N$ γράφους, στόχος είναι να μάθουμε μία συνάρτηση $f : \mathcal{G} \leftarrow \mathcal{Y}$, όπου $\mathcal{G}$ είναι ο χώρος των γράφων που μπορούν να δοθούν ως είσοδοι σε αυτήν την συνάρτηση και $\mathcal{Y}$ είναι το σύνολο των δυνατών επισημειώσεων που μπορούν να αποδοθούν στους γράφους. Η συνάρτηση αυτή μπορεί να χρησιμοποιηθεί ύστερα για την ταξινόμηση νέων γράφων, δηλαδή γράφων που δεν εμφανίστηκαν στο σύνολο εκπαίδευσης, όπως συνήθως είναι αυτοί του πειραματικού συνόλου, σε κατηγορίες.

Το πρόβλημα της ταξινόμησης γράφων έχει αποτελέσει μία δημοφιλή περιοχή έρευνας τα τελευταία χρόνια, μιάς και έχει συγκεντρώσει πολλές εφαρμογές σε ένα μεγάλο εύρος περιοχών. Τέτοιες συνοπτικά εμφανίζονται σε ένα εύρος από τον χαρακτηρισμό ενός χημικού μορίου ως μεταλλαξιογόνου [76] και την πρόβλεψη της λειτουργίας μίας πρωτεΐνης δεδομένου της νουκλεοτιδικής του ακολουθίας [11], μέχρι τον εντοπισμό του αν ένα λογισμικό είναι κακόβουλο [85].

### 2.3.3 Σύγκριση Γράφων

Το πρόβλημα της ταξινόμησης γράφων συνδέεται άμεσα με αυτό της σύγκρισης. Πολλοί αλγόριθμοι στο χώρο της μηχανικής μάθησης λαμβάνουν αποφάσεις, βάσει ενός μέτρου *ομοιότητας* (similarity metric) ή ενός μέτρου *απόστασης* (distance metric) μεταξύ δύο στιγμιοτύπων εισόδου. Για παράδειγμα δημοφιλείς ταξινομητές, όπως ο ταξινομητής $k$-κοντινότερων γειτόνων ή ο ταξινομητής μηχανών διανυσμάτων υποστήριξης (βλέπε ενότητα 2.5), μπορούν να επιλύσουν προβλήματα μάθησης πολύ διαφορετικών δεδομένων, προσδιορίζοντας μονάχα μία μετρική ομοιότητας μεταξύ τους. Κατ' αυτόν τον τρόπο προσδιορίζοντας μία συνάρτηση απόστασης $d : \mathcal{G} \times \mathcal{G} \to \mathbb{R}^+$ μεταξύ γράφων, μπορούμε άμεσα να χρησιμοποιήσουμε έναν από

τους παραπάνω αλγορίθμους, για να ταξινομήσουμε δεδομένα που έχουν αναπαρασταθεί ως γράφοι.

Η σύγκριση γράφων είναι από μόνη της ένα πολύ δύσκολο πρόβλημα. Συγκεκριμένα, δεδομένου δύο γράφων $G_i, G_j$ σε ένα χώρο γράφων $\mathcal{G}$, το πρόβλημα της σύγκρισης γράφων αφορά τον προσδιορισμό μίας συνάρτησης αντιστοίχισης: $f : \mathcal{G} \times \mathcal{G} \to \mathbb{R}$, τέτοια ώστε να 'ποσοτικοποιεί' είτε μία έννοια ομοιότητας μεταξύ των γράφων $G_i, G_j$ είτε μία έννοια απόστασης μεταξύ τους (η έννοια της απόστασης είναι η αντίστροφη της ομοιότητας και συνεπώς συμπληρωματική). Η σύγκριση γράφων είτε από μόνη της είτε στον χώρο της ταξινόμησης γράφων βρίσκει εφαρμογή σε πολλά πεδία. Συγκεκριμένα στην χημιοπληροφορική, χρησιμοποιήθηκε για την πρόβλεψη κινητικών μοντέλων μεταξύ χημικών αντιδράσεων [80] και για την ταξινόμηση χημικών μορίων με στόχο π.χ. την ανίχνευση φαρμακευτικής δράσης [86]. Στην βιοπληροφορική, χρησιμοποίηθηκε για την κατάταξη γονιδίων και γονιδιωμάτων σε μία βάση γνώσης[43, 35] και στην κατανόηση μηχανισμών γονιδιακής ρύθμισης [20]. Πέρα από τους δύο αυτούς τομείς, η ομοιότητα γράφων έχει χρησιμοποιηθεί για την σύγκριση ηλεκτρικών κυκλωμάτων [77], προγραμμάτων γλώσσας C [30], κειμένων [68], ειδησεογραφικών γεγονότων [31] αλλά και για την αυτοματοποιημένη συλλογιστική [79] και για την αποσαφήνιση σχεσιακών οντοτήτων [38].

## 2.4 Πυρήνες Γράφων

Οι πυρήνες γράφων (graph kernels), αποτελούν μία από τις πιο μοντέρνες τεχνικές στην ταξινόμηση γράφων. Τεχνικές σαν αυτή παρουσιάζουν μερικές πολύ ελκυστικές στατιστικές ιδιότητες. Παράλληλα δύνανται να συνδυάσουν την αναπαραστατική δύναμη των γράφων και τις δυνατότητες διαχωρισμού της πληροφορίας που επιτυγχάνουν οι μέθοδοι που βασίζονται σε *πυρήνες*. Συνεπώς, αποτελούν πολύ ισχυρά εργαλεία τόσο για την αντιμετώπιση του προβλήματος της ομοιότητας γράφων όσο και για την επίλυση των προβλημάτων μάθησης. Αυτός είναι και ο κύριος λόγος που δημιουργήθηκε το συγκεκριμένο λογισμικό, για να κάνει άμεσα προσβάσιμες αυτές τις τεχνικές στο εκάστοτε στάδιο ενός προβλήματος μηχανικής μάθησης, προς επίλυση.

### 2.4.1 Συναρτήσεις Πυρήνα

Αρχικά παρουσιάζεται μία θεμελιώδης εισαγωγή στις συναρτήσεις πυρήνα.

**Ορισμός 2.14** (Gram Μήτρα)**.** *Δεδομένου ενός συνόλου εισόδων $x_1, \ldots x_N \in \mathcal{X}$ και μία συνάρτηση $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, η $N \times N$ μήτρα $K$ που ορίζεται σαν:*

$$K_{ij} = k(x_i, x_j)$$

*ονομάζεται Gram μήτρα (Gram Matrix) ή μήτρα πυρήνα του $k$ με βάση τις εισόδους $x_1, \ldots, x_N$*

Στη συνέχεια θα αναφερόμαστε στις μήτρες Gram σαν μήτρες πυρήνα.

**Ορισμός 2.15** (Θετικά Ημιορισμένος Πυρήνας)**.** *Έστω $\mathcal{X}$ ένα μη κενό σύνολο. Μία συμμετρική μήτρα $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, που για όλα τα $N \in \mathbb{N}$ και όλα τα $x_1, \ldots, x_N \in \mathcal{X}$, παράγει μία θετικά ημιορισμένη μήτρα, θα λέγεται θετικά ημιορισμένος πυρήνας, ή απλώς πυρήνας.*

Πρακτικά η συνάρτηση πυρήνα είναι ένα μέτρο ομοιότητας μεταξύ δύο αντικειμένων. Οι συναρτήσεις πυρήνα μπορούν να ειδωθούν, αλλιώς σαν εσωτερικά γινόμενα μεταξύ αναπαραστάσεων αυτών των δεδομένων. Συγκεκριμένα, για κάθε έγκυρο πυρήνα $k$ στο $\mathcal{X} \times \mathcal{X}$, υπάρχει μία συνάρτηση $\phi : \mathcal{X} \to \mathcal{H}$ που αναπαριστά τα δεδομένα σε ένα χώρο Hilbert, τέτοια ώστε:

$$\forall x_i, x_j \in \mathcal{X} : k(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle \tag{2.3}$$

όπου το $\langle \, . \, , \, . \, \rangle$ συμβολίζει το εσωτερικό γινόμενο σε αυτόν τον χώρο Hilbert. Ένας χώρος Hilbert, είναι ένας χώρος με εσωτερικό γινόμενο, που ικανοποιεί την συνθήκη πληρότητας ότι 'κάθε ακολουθία σημείων Cauchy που προκύπτει από αυτόν τον χώρο, συγκλίνει σε ένα σημείο σε αυτόν τον χώρο'. Ακόμα ένας χώρος Hilbert ικανοποιεί την ακόλουθη ιδιότητα γνωστή σαν ιδιότητα αναπαραγωγής (reproducing):

$$\forall f \in \mathcal{H}, \forall x \in \mathcal{X} : f(x) = \langle f, k(x, \dot{}) \rangle \tag{2.4}$$

Λόγω αυτής της ιδιότητας, ο $\mathcal{H}$ λέγεται χώρος Hilbert αναπαραγωγικού πυρήνα (reproducing kernel Hilbert space - RKHS) και συνδέεται με τον πυρήνα $k$. Είναι ενδιαφέρον να σημειώσουμε ότι κάθε συνάρτηση στο $\mathcal{X} \times \mathcal{X}$ συνδέεται με έναν RKHS και αντίστροφα [4].

### 2.4.2 Μέθοδοι Πυρήνα

Οι μέθοδοι πυρήνα (Kernel methods) είναι ένα σύνολο από αλγορίθμους μηχανικής μάθησης, που λειτουργούν σε ένα σύνολο δεδομένων εισόδου τα οποία έχουν έμμεσα αναπαρασταθεί σε ένα χώρο χαρακτηριστικών (feature space) χρησιμοποιώντας μία συνάρτηση πυρήνα. Ένα από τα σημαντικότερα πλεονεκτήματα αυτών των μεθόδων είναι ότι λειτουργούν σε πολλά και διαφορετικά είδη δεδομένων [70]. Ο χώρος εισόδων $\mathcal{X}$, δεν χρειάζεται να είναι ένας διανυσματικός χώρος, αλλά μπορεί να αναπαριστά οποιαδήποτε κατηγορία δεδομένων, όπως τον χώρο των συμβολοσειρών (string) ή τον χώρο των γράφων [26]. Οι μέθοδοι πυρήνα μπορούν ακόμα να εφαρμοσθούν σε πολλούς τύπους δεδομένων, από την στιγμή που μπορούμε να βρούμε μία αναπαράσταση $\phi : \mathcal{X} \to \mathcal{H}$, όπου το $\mathcal{H}$ είναι ένας RKHS. Μία τέτοια αναπαράσταση δεν είναι αναγκαίο να είναι φανερά ορισμένη. Οι μέθοδοι αυτοί κατά την εκτέλεση των υπολογισμών τους, αναπαριστούν έμμεσα τα δεδομένα σε ένα χώρο χαρακτηριστικών, υπολογίζοντας την τιμή του εσωτερικού γινομένου που θα είχαν αυτές οι αναπαραστάσεις αν ήταν φανερές σε αυτό τον χώρο, κάνοντας χρήση μίας συνάρτησης πυρήνα. Τέτοια εσωτερικά γινόμενα μπορούν να ερμηνευτούν ως μέτρα ομοιότητας μεταξύ των αντίστοιχων αντικειμένων, που συγκρίνουν. Προβλήματα μηχανικής μάθησης όπως η ταξινόμηση και η συσταδοποίηση, μπορούν να έρθουν εις πέρας χρησιμοποιώντας μόνο εσωτερικά γινόμενα που έχουν υπολογιστεί σε τέτοιους μη-φανερούς χώρους χαρακτηριστικών.

Οι μέθοδοι πυρήνα είναι πολύ δημοφιλείς και έχουν φανεί ιδιαίτερα επιτυχημένες σε ένα μεγάλο σύνολο εφαρμογών. Για να διασαφηνίσουμε λίγο περισσότερο τα παραπάνω, ας θεωρήσουμε

ένα πρόβλημα δυαδικής ταξινόμησης, με ένα σύνολο εκπαίδευσης $D = \{(x_i, y_i)\}_{i=1}^N$ όπου $(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$ και ο $\mathcal{X}$ είναι ένας χώρος με εσωτερικό γινόμενο (π.χ. $R^d$) και $Y = \{-1, +1\}$. Όπως αναφέρθηκε και παραπάνω, δεδομένου ενός συνόλου εκπαίδευσης $D$, στόχος είναι η *μάθηση* μίας συνάρτησης $f : \mathcal{X} \to \mathcal{Y}$, τέτοια ώστε το σφάλμα γενίκευσης της $f$ είναι όσο το δυνατόν χαμηλότερο. Συνεπώς αυτό που μας ενδιαφέρει είναι να ελαχιστοποιήσουμε το σφάλμα εκπαίδευσης, διατηρώντας ταυτόχρονα μία καλή απόδοση στα νέα δεδομένα (αυτά που δεν εμφανίστηκαν στο σύνολο εκπαίδευσης). Η κύρια τέτοια μέθοδος που θα μας απασχολήσει (βλ. ενότητα 2.5), ανήκει στις μεθόδους μεγάλου περιθωρίου (large margin methods), οι οποίες αναζητούν ένα υπερεπίπεδο που διαχωρίζει το μέρος των δειγμάτων εκπαίδευσης που ανήκουν στην κατηγορία $-1$ από αυτά που ανήκουν στην κατηγορία 1. Εν συνεχεία, η $f$ μπορεί να πάρει την μορφή $f(x) = \text{sign}(\langle w, x\rangle + b)$, όπου η συνάρτηση sign() αντιστοιχεί στην συνάρτηση προσήμου (η οποία παίρνει την τιμή 1 αν το όρισμα της είναι θετικό και την τιμή $-1$ αλλιώς). Δεδομένου ενός $x$, η *συνάρτηση απόφασης* $f$ δίνει στην έξοδο της μία πρόβλεψη που εξαρτάται από την θέση του $x$ σε σχέση με το υπερεπίπεδο $\langle w, x\rangle + b = 0$.

Αρχικά, ας υποθέσουμε ότι τα δεδομένα εκπαίδευσης είναι γραμμικά διαχωρίσιμα. Αν κάτι τέτοιο συμβαίνει, υπάρχει ένα υπερεπίπεδο τέτοιο ώστε τα σημεία που ανήκουν σε διαφορετικές κλάσεις βρίσκονται στις αντίθετες πλευρές του. Στην πραγματικότητα, παρουσιάζεται απειρία τέτοιων υπερεπιπέδων. Οι ταξινομητές μεγάλου περιθωρίου, επιλέγουν μεταξύ αυτών των υπερεπιπέδων αυτό που μεγιστοποιεί το περιθώριο μεταξύ των δύο κατηγοριών των σημειακών δεδομένων εκπαίδευσης, αυτό δηλαδή για το οποίο η απόσταση από τα κοντινότερα δεδομένα των δύο κατηγοριών είναι μέγιστη. Για την εύρεση αυτού του βέλτιστου υπερεπιπέδου, αναγκαία είναι η λύση ενός προβλήματος κυρτής τετραγωνικής βελτιστοποίησης. Το διάνυσμα $w$ που προκύπτει ως λύση σε αυτό το πρόβλημα είναι ένας γραμμικός συνδυασμός των παραδειγμάτων εκπαίδευσης:

$$w = \sum_{i=1}^{N} a_i y_i x_i \tag{2.5}$$

όπου το $\alpha_i \in \mathbb{R}^+$.

Χρησιμοποιώντας το παραπάνω αποτέλεσμα, που είναι γνωστό ως το θεώρημα αναπαράστασης (representer theorem) [69], ο γραμμικός ταξινομητής $f$ μπορεί να γραφεί ως:

$$f(x) = \text{sign}(\sum_{i=1}^{N} a_i y_i \langle x_i, x\rangle + b) \tag{2.6}$$

Στην περίπτωση, τώρα, που τα δεδομένα εκπαίδευσης δεν είναι γραμμικά διαχωρίσιμα, αναζητούμε ένα υπερεπίπεδο που μεγιστοποιεί το περιθώριο και την ίδια στιγμή ελαχιστοποιεί μία ποσότητα αντιστρόφως ανάλογη στο πλήθος των λαθών ταξινόμησης. Ο υπολογισμός αυτού του υπερεπιπέδου μπορεί και πάλι να διατυπωθεί ως ένα πρόβλημα κυρτής τετραγωνικής βελτιστοποίησης, όπου το διάνυσμα λύσης $w$ συνεχίζει να προκύπτει ως ένας γραμμικός συνδυασμός των δεδομένων εισόδου.

Σε σύνθετα προβλήματα ταξινόμησης, είναι πιθανό να μην υπάρχουν υπερεπίπεδα τέτοια ώστε να διαχωρίζουν τα θετικά επισημειωμένα παραδείγματα από τα αρνητικά, με τέτοιο τρόπο ώστε να παρέχουν ένα καλό αποτέλεσμα ταξινόμησης. Η απάντηση σε αυτό το πρόβλημα, που

συχνά είναι γνωστή ως τέχνασμα πυρήνα (kernel trick: [2, 14]), είναι η απεικόνιση των δεδομένων εισόδου σε έναν άλλο χώρο χαρακτηριστικών $\mathcal{H}$ (συνήθως υψηλότερων διαστάσεων) και η εύρεση ενός διαχωριστικού υπερεπιπέδου σε αυτόν τον χώρο. Έστω $\phi : \mathcal{X} \to \mathcal{H}$ μία απεικόνιση από το $\mathcal{X}$ στο χώρο χαρακτηριστικών $\mathcal{H}$, που διαθέτει εσωτερικό γινόμενο. Για να υπολογίσουμε το βέλτιστο υπερεπίπεδο στο χώρο χαρακτηριστικών, μπορούμε να χρησιμοποιήσουμε την προηγούμενη μαθηματική διατύπωση, απλώς αντικαθιστώντας το $\langle x_i, x\rangle$ με $\langle \phi(x_i), f(x)\rangle$. Έστω $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ μία συνάρτηση πυρήνα με την ακόλουθη ιδιότητα:

$$k(x_i, x_j) = \langle \phi(x_i), \phi(x_j)\rangle \tag{2.7}$$

Η συνάρτηση απόφασης $f$ μπορεί, τώρα, να γραφτεί ως:

$$f(x) = \text{sign}(\sum_{i=1}^{N} \alpha_i y_i \langle \phi(x_i), \phi(x)\rangle + b) = \text{sign}(\sum_{i=1}^{N} \alpha_i y_i k(x_i, x) + b) \tag{2.8}$$

Από την παραπάνω ανάλυση, είναι φανερό ότι ορίζοντας μία συνάρτηση πυρήνα $k$, απεικονίζουμε έμμεσα όλα τα σημεία των δεδομένων σε ένα χώρο χαρακτηριστικών $\mathcal{H}$. Ως επακόλουθο οι μέθοδοι πυρήνα μπορούν να λύσουν προβλήματα μηχανικής μάθησης, όπως η ταξινόμηση, χωρίς να χρειάζεται πουθενά, να διατυπώσουν ρητά μία τέτοια αντιστοίχιση.

## 2.5 Μηχανή Διανυσμάτων Υποστήριξης

Στη συνέχεια θα αναλύσουμε την μέθοδο πυρήνα του ταξινομητή μηχανής διανυσμάτων υποστήριξης (support vector machine ή αλλιώς SVM), ένα από τους πιο δημοφιλείς αλγορίθμους μηχανικής που έχει ως βάση του πυρήνες. Ο κλασσικός ταξινομητής SVM, θέτει το ακόλουθο πρόβλημα: δεδομένου ενός συνόλου $N$ αντικειμένων εκπαίδευσης, μαζί με τις κατηγορίες τους $D = \{(x_i, y_i)\}_{i=1}^{N}, x_i \in \mathcal{X} = \mathbb{R}^d, y_i \in \mathcal{Y} = \{-1, +1\}$, υπολόγισε ένα ταξινομητή $f : \mathcal{X} \to \mathcal{Y}$ που προβλέπει τις κατηγορίες νέων δεδομένων. Στην διατύπωση των προβλημάτων *άκαμπτου ορίου* (hard margin), θεωρούμε πώς τα προβλήματα μας είναι γραμμικά διαχωρίσιμα.
Όντας στην οικογένεια των ταξινομητών μεγάλου περιθωρίου, ο SVM αναζητά ένα υπερεπίπεδο που διαχωρίζει στιγμιότυπα της κατηγορίας $-1$ από την κατηγορία $+1$ [81]. Έστω $(w, b)$ οι παράμετροι του υπερεπιπέδου. Τότε, η απόσταση μεταξύ ενός σημείου $x \in \mathbb{R}^d$ και του υπερεπιπέδου υπολογίζεται ως:

$$\frac{|w^T x + b|}{||w||} \tag{2.9}$$

Αξίζει σε αυτό το σημείο να επισημάνουμε πως, ένα υπερεπίπεδο είναι αναλλοίωτο σε έναν μη-μηδενικό βαθμωτό πολλαπλασιασμό. Ως επακόλουθο μπορούμε να θέσουμε τις παραμέτρους του υπερεπιπέδου σε συγκεκριμένες τιμές έτσι ώστε να ισχύει ως προς το υπερεπίπεδο: $w^T x + b = 1$ και $w^T x + b = -1$ για το κοντινότερο θετικό και κοντινότερο αρνητικό δείγμα, αντίστοιχα. Τότε, η απόσταση μεταξύ δύο σημείων από το υπερεπίπεδο (δηλ. το περιθώριο) είναι ίσο με:

$$\frac{1}{||w||} \tag{2.10}$$

Ως επακόλουθο, η μεγιστοποίηση του περιθωρίου συνεπάγεται ελαχιστοποίηση του $||w||$, πράγμα που είναι ισοδύναμο με την ελαχιστοποίηση του $\frac{1}{2}||w||^2$. Συνεπώς στην περίπτωση που τα δεδομένα είναι διαχωρίσιμα, η λύση του SVM είναι η λύση του ακόλουθου προβλήματος ελαχιστοποίησης:

$$\begin{aligned} \underset{w,b}{\text{minimize}} \quad & \frac{1}{2}||w||^2 \\ \text{subject to} \quad & y_i(\langle w,\ x_i\rangle + b) \geq 1\ \forall i \in \{1,\ldots,N\} \end{aligned}$$

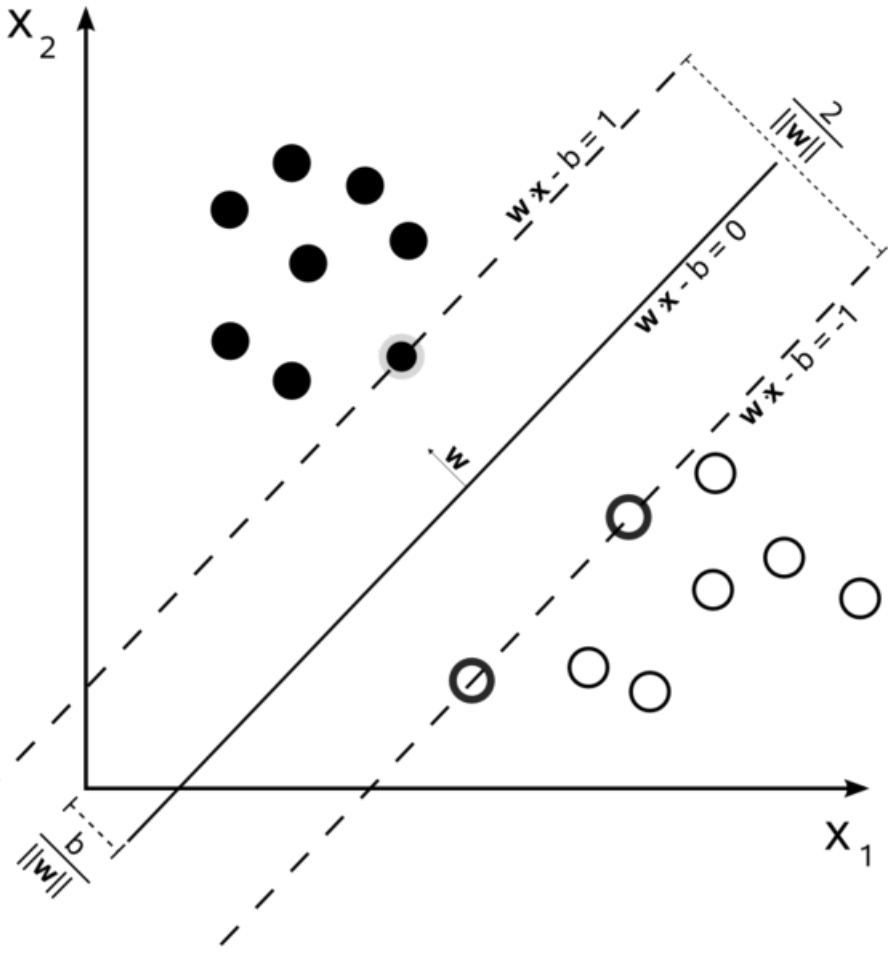
που αποτελεί ένα πρόβλημα κυρτού προγραμματισμού με περιορισμούς [15]. Αν τώρα στραφούμε στο δυϊκό (κατά Lagrange) αυτού του προβλήματος:

$$\begin{aligned} \underset{\alpha}{\text{maximize}} \quad & \sum_{i=1}^{N}\alpha_i - \frac{1}{4}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha \\ \text{subject to} \quad & \sum_{i=1}^{N}\alpha_i y_i = 0 \\ & \alpha_i \geq 0\ \forall i \in \{1,\ldots,N\} \end{aligned} \tag{2.11}$$

βλέπουμε ότι είναι δέσμιο γραμμικών περιορισμών και ως επακόλουθο έχει αποδοτική λύση, μέσω της χρήσης υπαρχόντων αλγορίθμων τετραγωνικού προγραμματισμού.
Ακόμα ισχύει το εξής:

$$w = \sum_{i=1}^{N} a_i y_i x_i \tag{2.12}$$

Η παράμετρος $w$ του υπερεπιπέδου (δηλ. η λύση του SVM) είναι ένας γραμμικός συνδυασμός των σημείων εκπαίδευσης $x_1 \ldots x_N$. Ένα σημείο $x_i$ συμβάλλει στην διαμόρφωση του $w$ αν και μόνον αν $a_i > 0$. Για εκείνα τα σημεία δεδομένων $x_i$ που ικανοποιούν το $a_i > 0$, ισχύει ότι $y_i(\langle w, x_i\rangle + b) = 0$. Κάτι τέτοιο σημαίνει ότι αυτά τα σημεία βρίσκονται πάνω στο περιθώριο και κατ' αυτόν τον τρόπο *υποστηρίζουν* το υπερεπίπεδο. Τα σημεία αυτά ονομάζονται *διανύσματα υποστήριξης* (βλέπε σχήμα 2.2).
Μέχρι στιγμής, έχουμε υποθέσει ότι τα δεδομένα εκπαίδευσης είναι γραμμικά διαχωρίσιμα. Από την άλλη, σε μία πληθώρα πραγματικών εφαρμογών κάτι τέτοιο δεν συμβαίνει. Για την επίλυση αυτού του προβλήματος, η συχνά αποκαλούμενη διατύπωση 'εύκαμπτου ορίου' (soft margin) επιτρέπει κάποια σημεία εκπαίδευσης να μην ταξινομηθούν σωστά [8, 17]. Συγκεκριμένα στην συνάρτηση που πρόκειται να ελαχιστοποιηθεί προστίθεται ένας όρος ποινής (penalty term) για κάθε σημείο που έχει ταξινομηθεί λάθος. Έτσι, όσο μεγαλύτερη είναι η απόσταση ενός σημείου από το όριο, τόσο μεγαλύτερος θα είναι και ο όρος ποινής. Επιτρέποντας λανθασμένη ταξινόμηση των δεδομένων εισόδου, είναι προφανές ότι δεν μπορούμε να χρησιμοποιήσουμε την διατύπωση του SVM όπως δόθηκε στην εξίσωση 2.11, μιας και ο περιορισμός της δεν ικανοποιείται για τα δεδομένα που κατηγοριοποιούνται λανθασμένα. Ως επακόλουθο, εισάγουμε μία βοηθητική μεταβλητή $\zeta_i$ για κάθε δεδομένο εκπαίδευσης $x_i$. Η βοηθητική αυτή μεταβλητή, υπολογίζει το βαθμό που ένα σημειακό δεδομένο παραβιάζει το όριο του περιθωρίου. Για τα σημεία των δεδομένων που βρίσκονται πάνω ή μέσα στο περιθώριο, ισχύει ότι




Σχήμα 2.2: Υπερεπίπεδο μέγιστου περιθωρίου στην περίπτωση δύο διαστάσεων για ένα SVM. Τα δείγματα που βρίσκονται στο όριο του περιθωρίου λέγονται **διανύσματα υποστήριξης**.

$\zeta = 0$, ενώ για τα άλλα ισχύει:

$$\zeta_i = |y_i - (\langle w, x_i \rangle + b)| \tag{2.13}$$

Συνεπώς, ένα σημειακό δεδομένο $x_i$ που βρίσκεται πάνω στο διαχωριστικό υπερεπίπεδο θα έχει $\zeta_i = 1$ και τα σημεία που είναι λανθασμένα ταξινομημένα θα έχουν $\zeta_i > 1$ (παράβαλε σχήμα 2.2). Σε τούτη την διατύπωση, στόχος μας είναι να μεγιστοποιήσουμε το περιθώριο, ενώ ταυτόχρονα μας ενδιαφέρει να ελέγξουμε την απόσταση μεταξύ των σημείων που έχουν ταξινομηθεί λάθος και του ορίου του περιθωρίου. Η απόσταση αυτή αντιστοιχεί στην συνολική συνεισφορά των βοηθητικών μεταβλητών, που είναι ίση με $\sum_{i=1}^{N} \zeta_i$. Κάτι τέτοιο μας οδηγεί στο ακόλουθο πρόβλημα βελτιστοποίησης:

$$\begin{aligned} \underset{w,b,\zeta}{\text{minimize}} \quad & \frac{1}{2}||w||^2 + C\sum_{i=1}^{N}\zeta_i \\ \text{subject to} \quad & y_i(\langle w, x_i \rangle + b) \geq 1 - \zeta_i \;\; \forall i \in \{1, \ldots, N\} \\ & \zeta_i \geq 0 \;\; \forall i \in \{1, \ldots, N\} \end{aligned} \tag{2.14}$$

όπου η παράμετρος $C > 0$ ελέγχει το βαθμό που ο ταξινομητής πρέπει να αποφύγει να ταξινομήσει λανθασμένα ένα δείγμα εκπαίδευσης, ελέγχοντας το αντίβαρο μεταξύ μεγιστοποίησης του περιθωρίου και ελαχιστοποίησης του βάρους των βοηθητικών μεταβλητών. Η βέλτιστη παράμετρος $C$ προσδιορίζεται συνήθως από αναζήτηση σε ένα εύρος διακριτών τιμών (grid-search) μέσω *n-πλάσιας διασταυρωμένης επικύρωσης* (n-fold cross-validation). Σε αυτήν την περίπτωση το δυϊκό (κατά Lagrange) πρόβλημα μπορεί να διατυπωθεί ως εξής:

$$\begin{aligned} \underset{\alpha}{\text{maximize}} \quad & \sum_{i=1}^{N} \alpha_i - \frac{1}{4} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha \\ \text{subject to} \quad & \sum_{i=1}^{N} \alpha_i y_i = 0 \\ & C \geq \alpha_i \geq 0 \;\; \forall i \in \{1, \ldots, N\} \end{aligned} \tag{2.15}$$

Το παραπάνω πρόβλημα βελτιστοποίησης είναι κατά ενδιαφέρον τρόπο το ίδιο με την εξίσωση 2.11, αν προσθέσουμε απλώς το $C$ ως άνω φράγμα στις παραμέτρους $\alpha_i$.

## 2.6 Πυρήνες Γράφων

Οι πυρήνες γράφων έχουν πρόσφατα προκύψει σαν μία πολλά υποσχόμενη προσέγγιση στην μηχανική μάθηση σε δεδομένα που εμφανίζονται με την μορφή των γράφων. Αυτές οι μέθοδοι καταφέρνουν να επεκτείνουν την εφαρμοσιμότητα των μεθόδων πυρήνα στον χώρο των γράφων. Οι πυρήνες γράφων μπορούν να χωριστούν σε δύο κατηγορίες: (1) αυτούς που συγκρίνουν κόμβους του ίδιου γράφου και (2) αυτούς που συγκρίνουν γράφους. Οι πυρήνες που αναπτύχθηκαν στο λογισμικό της παρούσας διπλωματικής βρίσκονται αποκλειστικά στην δεύτερη κατηγορία και έτσι οπουδήποτε συναντάμε αυτόν τον όρο στα παρακάτω αυτός θα περιγράφει συναρτήσεις πυρήνα μεταξύ γράφων.

Από τα προηγούμενα πρέπει να είναι πλέον φανερό ότι η εφαρμογή των συναρτήσεων πυρήνα αποτελείται από δύο στάδια. Πρώτα, σχεδιάζεται μία συνάρτηση πυρήνα, και βάση αυτής κατασκευάζεται ο πίνακας πυρήνα. Έπειτα, ένας αλγόριθμος μάθησης χρησιμοποιείται για τον υπολογισμό μίας βέλτιστης πολλαπλότητας (manifold) στον χώρο χαρακτηριστικών (π.χ. ένα υπερεπίπεδο σε ένα πρόβλημα δυαδικού προγραμματισμού). Από την στιγμή που υπάρχουν διάφοροι ταξινομητές ώριμοι και διαθέσιμοι στη βιβλιογραφία, οι οποίοι χρησιμοποιούν στην βάση τους πυρήνες, η έρευνα στον χώρο των πυρήνων γράφων έχει επικεντρωθεί στο πρώτο στάδιο. Σαν επακόλουθο, το ερευνητικό έργο εστιάστηκε στην ανάπτυξη εκφραστικών και αποδοτικών πυρήνων γράφων, ικανών να μετρήσουν με ακρίβεια μία έννοια ομοιότητας. Και στην περίπτωση αυτών των πυρήνων οι γράφοι προβάλλονται έμμεσα σε ένα χώρο χαρακτηριστικών $\mathcal{H}$. Όσον αφορά το δεύτερο στάδιο, για το σώμα της παρούσας διπλωματικής και για την πειραματική αξιολόγηση του λογισμικού (βλ. Κεφάλαιο 4), θα μας απασχολήσει μόνο ο ταξινομητής $SVM$ που αναφέρθηκε παραπάνω.

Ο κύριος στόχος στην εφαρμογή μεθόδων πυρήνα σε γράφους είναι ο προσδιορισμός κατάλληλων θετικά ημιορισμένων συναρτήσεων πυρήνα σε ένα σύνολο δεδομένων εισόδου που

δύνανται να υπολογίσουν την ομοιότητα μεταξύ τους. Πόσο εκφραστικός μπορεί όμως να είναι ένας πυρήνας στην πράξη· Ας υποθέσουμε αρχικά ότι ένας πυρήνας δύναται να ξεχωρίσει μεταξύ όλων των (μη-ισομορφικών) γράφων στον χώρο χαρακτηριστικών. Ένας τέτοιος πυρήνας λέγεται *πλήρης*.

**Ορισμός 2.16** (Πλήρης Πυρήνας Γράφων)**.** *Ένας πυρήνας γράφων $k(G_i, G_j) = \langle \phi(G_i), \phi(G_j) \rangle$ είναι πλήρης αν η $\phi$ είναι "1-1".*

Οι Gärtner, Flach και Wrobel έδειξαν ότι υπολογίζοντας οποιονδήποτε πλήρη πυρήνα γράφων είναι τουλάχιστον τόσο δύσκολο όσο να αποφασίσουμε αν δύο γράφοι είναι ισομορφικοί (βλέπε ορισμό 2.13) [27]. Ως αποτέλεσμα, είναι αδύνατη η χρήση πυρήνων γράφων που είναι πλήρεις σε πρακτικές εφαρμογές. Αντίθετα, χρησιμοποιώντας πυρήνες που δεν είναι πλήρεις, δεν υπάρχει περαιτέρω εγγύηση ότι δύο μη-ισομορφικοί γράφοι δεν θα απεικονιστούν στο ίδιο σημείο στον χώρο χαρακτηριστικών.
Μία από τις πιο δημοφιλείς μεθόδους για να ορίσουμε πυρήνες μεταξύ σύνθετων αντικειμένων είναι να τα αποσυνθέσουμε σε πιο απλά αντικείμενα άλλου τύπου (π.χ. συμβολοσειρές, διανύσματα, ...) με γνωστούς πυρήνες και συγκρίνοντας έπειτα αυτά και συλλέγοντας τα αποτελέσματα της σύγκρισης τους με ένα κατάλληλο τρόπο, να φτιάξουμε τελικά έναν πυρήνα για τα αρχικά αντικείμενα. Από τα πιο κοινά ήδη πυρήνων στη βιβλιογραφία, που προκύπτουν ακολουθώντας την παραπάνω μεθοδολογία λέγονται R-συνελικτικοί πυρήνες (R-convolutional kernels) [36]. Αυτοί οι πυρήνες αποσυνθέτουν τους γράφους σε ένα σύνολο υπο-δομών και προσθέτουν τις τιμές ομοιότητας, όπως υπολογίζονται μεταξύ κάθε ζευγαριού τους. Φυσικά θα περίμενε κανείς πως μία κατάλληλη τέτοια υπο-δομή θα ήταν οι ίδιοι οι υπογράφοι (βλέπε ορισμό 2.7). Οι Gärtner, Flach και Wrobel έδειξαν επιπρόσθετα ότι το πρόβλημα υπολογισμού ενός πυρήνα που συγκρίνει όλους τους υπογράφους είναι και αυτό NP-δύσκολο. Κατά συνέπεια, γίνεται σαφές ότι χρειάζεται να σκεφτούμε μία εναλλακτική: λιγότερο δυνατούς πυρήνες γράφων, που είναι υπολογίσιμοι σε πολυωνυμικό χρόνο.
Από την άλλη, είναι απαραίτητο ότι αυτοί οι πυρήνες θα αποτελούν μία εκφραστική μετρική της *ομοιότητας* μεταξύ των γράφων. Είναι κοινό στην βιβλιογραφία οι πυρήνες να οργανώνονται σε μεγάλες κατηγορίες, που η κάθε μία μελετάει διαφορετικά δομικά χαρακτηριστικά ενός γράφου. Συγκεκριμένα, υπάρχουν πυρήνες που συγκρίνουν γράφους, βάση τυχαίων περιπάτων [25, 13, 84], υποδέντρων [66, 5, 54], κύκλων [40], μονοπατιών [11, 29] και μικρούς υπογράφους [19, 39, 49, 72]. Άλλοι αλγόριθμοι αποσυνθέτουν τους γράφους σε σύνολα από κατευθυνόμενα ακυκλικά γραφήματα (ΚΑΓ) και έπειτα χρησιμοποιώντας υπάρχοντες πυρήνες δέντρων συγκρίνουν αυτά τα ΚΑΓ μεταξύ τους [1]. Συμπληρωματικά έχουν εμφανιστεί στην βιβλιογραφία πυρήνες γράφων, που χρησιμοποιούν άλλους πυρήνες (όπως τους προαναφερθείς) που είναι γνωστοί ως σκελετοί πυρήνα (kernel frameworks). Ο σκελετός πυρήνα Weisfeiler-Lehman βελτιώνει έτσι την απόδοση υπαρχόντων πυρήνων χρησιμοποιώντας μία επαναληπτική διαδικασία επισημείωσης των κόμβων που βασίζεται στο τεστ ισομορφισμού Weisfeiler-Lehman [73]. Ο σκελετός πυρήνα k-core αποσυνθέτει κάθε γράφο σε ιεραρχίες ένθετων υπογραφημάτων, καθένα από τα οποία παρουσιάζει μεγαλύτερο βαθμό *συνδεσιμότητας* (connectivity) σε σχέση με το προηγούμενο και έπειτα κάνοντας χρήση ενός άλλου πυρήνα γράφων υπολογίζει την ομοιότητα ως

το άθροισμα των ομοιοτήτων μεταξύ των υπογράφων που προκύπτουν σε κάθε επίπεδο της ιεραρχίας [61].

Οι περισσότεροι από τους προαναφερθέντες πυρήνες γράφων συγκρίνουν συγκεκριμένες υποδομές των γράφων (π.χ. γραφίδια (graphlets), κύκλους, υποδέντρα κλπ). Αυτές οι υπο-δομές αντιστοιχούν είτε σε μικρούς υπογράφους είτε σε σχέσεις μεταξύ πολύ μικρών υποσυνόλων από κόμβους. Κατά συνέπεια οι αλγόριθμοι αυτοί εστιάζουν σε *τοπικές ιδιότητες* των γράφων. Κάποιοι πυρήνες γράφων που ξεφεύγουν από αυτήν την προσέγγιση και προσπαθούν να συγκεντρώσουν γενικά χαρακτηριστικά ή ιδιότητες των γράφων, έχουν προταθεί στη βιβλιογραφία. Για παράδειγμα πυρήνες που βασίζονται στο αριθμό Lovász και την αντίστοιχη ορθοκανονική αναπαράσταση του γράφου [42] ή πυρήνες που χρησιμοποιούν μετρικές από την θεωρία πληροφοριών όπως την απόκλιση (divergence) Jensen Shannon [6]. Άλλοι πυρήνες όπως ο πολυκλιμακωτός Λαπλασιανός (Multiscale Laplacian) ξεκινούν από τοπικές ιδιότητες του γράφου για να φτάσουν σε γενικές [47]

Οι περισσότεροι πυρήνες γράφων έχουν σχεδιαστεί για να λειτουργούν ταυτόχρονα σε γράφους με και χωρίς επισημειώσεις. Αυτό παρακάμπτεται εύκολα στις περιπτώσεις που ένας γράφος δεν έχει επισημείωσεις, όπου στην περίπτωση των κόμβων μπορούμε να πάρουμε ως επισημείωση ένα τοπικό χαρακτηριστικό όπως για παράδειγμα τον βαθμό του κόμβου, ενώ στις περιπτώσεις των ακμών να επισημειώσουμε κάθε ακμή με μία δυάδα με τις επισημειώσεις των κόμβων στα άκρα. Οι περισσότεροι πυρήνες θεωρούν πως οι επισημειώσεις είναι διακριτές (discrete) όπως π.χ. ένα νούμερο ή μία συμβολοσειρά κλπ. Παρόλο που έχουν μελετηθεί λιγότερο, έχουν σχεδιαστεί πυρήνες για γράφους με συνεχείς (continuous) επισημειώσεις, π.χ. διανύσματα. Ακόμα, υπάρχοντες πυρήνες με διακριτές επισημειώσεις όπως ο πυρήνας *κοντινότερου-μονοπατιού*, μπορεί να επεκταθεί για να υποστηρίζει συνεχείς επισημειώσεις, έχοντας σαν αντίβαρο την σημαντική αύξηση υπολογιστικής πολυπλοκότητας. Πρόσφατες ερευνητικές απόπειρες προσπάθησαν να αναπτύξουν πυρήνες γράφων που συνεχίζουν να λειτουργούν αποδοτικά όσο το μέγεθος των γράφων [22, 63, 58], χωρίς βέβαια να είναι και σε αυτή την περίπτωση υπολογιστικά συγκρίσιμοι με μεγάλο πλήθος πυρήνων βαθμωτών επισημειώσεων. Στη συνέχεια θα παρουσιαστεί μία εισαγωγή και ανάλυση όλων των πυρήνων γράφων που αναπτύχθηκαν στο GraKeL (μέχρι την παρούσα έκδοση 0.1a4), πληθώρα εκ των οποίων αναφέρθηκε παραπάνω.

### 2.6.1 Πυρήνες Τυχαίων Περιπάτων

Η πιο πολυμελετημένη οικογένεια πυρήνων γράφων είναι οι *πυρήνες τυχαίων περιπάτων* που υπολογίζουν την ομοιότητα μεταξύ ενός ζευγαριού γράφων βάση του αριθμού των κοινών τους περιπάτων [44, 25, 55, 13, 84, 74]. Πυρήνες που ανήκουν σε αυτή την οικογένειά έχουν εστιάσει κυρίως στην μέτρηση του αριθμού ταυτόσημων μονοπατιών μεταξύ δύο γράφων. Υπάρχουν πολλές διαφοροποιήσεις των πυρήνων τυχαίων μονοπατιών. Ο $k$-βημάτων αλγόριθμος τυχαίων περιπάτων συγκρίνει τυχαία μονοπάτια μήκους $k$ μεταξύ δύο γράφων. Ο πιο ευρέως χρησιμοποιημένος πυρήνας από αυτήν την οικογένεια, είναι ο πυρήνας τυχαίων περιπάτων γεωμετρικής προόδου [25], ο οποίος συγκρίνει σταδιακά περιπάτους μέχρι και απείρου μήκους αναθέτοντας ένα βάρος $\lambda^k$ ($\lambda < 1$) σε περιπάτους με μήκος $k$, προκειμένου να διασφαλίσει

σύγκλιση της αντίστοιχης γεωμετρικής σειρά που προκύπτει κατά τον υπολογισμό του μέτρου ομοιότητας. Στη συνέχεια θα δώσουμε ένα τυπικό ορισμό του γεωμετρικού πυρήνα τυχαίων περιπάτων. Δεδομένου δύο γράφων με επισημειώσεις στους κόμβους $G_i = (V_i, E_i)$ και $G_j = (V_j, E_j)$, το *γινόμενο* τους είναι $G_\times = (V_\times, E_\times)$ είναι ένας γράφος με σύνολο κόμβων:

$$V_\times = \{(v_i, v_j) : v_i \in V_i \wedge v_j \in V_j \wedge \ell(v_i) = \ell(v_j)\} \tag{2.16}$$

και σύνολο ακμών:

$$E_\times = \{\{(v_i, v_j), (u_i, u_j)\} : \{v_i, u_i\} \in E_i \wedge \{v_j, u_j\} \in E_j\} \tag{2.17}$$

Η εκτέλεση ενός τυχαίου περιπάτου στο $G_\times$ είναι ισοδύναμη με την εκτέλεση ταυτόχρονων τυχαίων περιπάτων στο $G_i$ και $G_j$. Ο γεωμετρικός πυρήνας τυχαίων μονοπατιών μετράει κοινούς περιπάτους (που μπορούν να εκτείνονται έως το άπειρο) μεταξύ δύο γράφους και ορίζεται ως εξής.

**Ορισμός 2.17** (Γεωμετρικός Πυρήνας Τυχαίων Περιπάτων)**.** *Έστω $G_i$ και $G_j$ δύο γράφοι και έστω ότι το $A_\times$ αναπαριστά τον πίνακα γειτνίασης του γινομένου τους $G_\times$ και έστω $V_\times$ το σύνολο των κόμβων του. Τότε, ο γεωμετρικός πυρήνας τυχαίων περιπάτων ορίζεται ως*

$$K_\times^\infty(G_i, G_j) = \sum_{p,q=1}^{|V_\times|} \Big[\sum_{l=0}^{\infty} \lambda^l A_\times^l\Big]_{pq} = e^T(I - \lambda A_\times)^{-1} e \tag{2.18}$$

*όπου $I$ είναι ο ταυτοτικός πίνακας, $e$ είναι ένα διάνυσμα που περιέχει μόνο άσσους και $\lambda$ είναι ένα θετικό, βάρος πραγματικής τιμής. Ο γεωμετρικός πυρήνας τυχαίων περιπάτων συγκλίνει μόνο αν $\lambda < \frac{1}{\lambda_\times}$ όπου $\lambda_\times$ είναι η μεγαλύτερη ιδιοτιμή του $A_\times$.*

Ο ευθύς υπολογισμός του γεωμετρικού πυρήνα τυχαίων μονοπατιών, έχει πολυπλοκότητα $\mathcal{O}(n^6)$, μιας και αυτή είναι η πολυπλοκότητα υπολογισμού του $A_\times = A_i \otimes A_j$ (όπου $\otimes$ είναι το *γινόμενο Kronecker* μεταξύ δύο πινάκων). Η υπολογιστική πολυπλοκότητα της μεθόδου, αποτελεί έναν αυστηρό περιορισμό για την εφαρμογή του σε πραγματικές εφαρμογές. Σαν λύση σε αυτό το πρόβλημα ο Vishwanathan κ.α. πρότειναν τέσσερεις αποτελεσματικές μεθόδους για τον αποδοτικό υπολογισμό των πυρήνων τυχαίων μονοπατιών που μειώνουν την πολυπλοκότητα του υπολογισμού του πυρήνα από $\mathcal{O}(n^6)$ σε $\mathcal{O}(n^3)$ [84]. Συγκεκριμένα αυτός που υλοποιήσαμε αφορά την φασματική αποσύνθεση ενός πίνακα γειτνίασης $A$. Συγκεκριμένα, αν γράψουμε τον πίνακα γειτνίασης κάθε γράφου σαν $A = PDP^{-1}$, όπου $D$ είναι ένας διαγώνιος πίνακας με τις ιδιοτιμές του πίνακα $A$ και ο $P$, είναι ο πίνακας που στην i-οστή του στήλη του φέρει το ιδιοδιάνυσμα που αντιστοιχεί στην i-οστή ιδιοτιμή της διαγωνίου του $D$. Επειδή ο πίνακας $A$ θεωρείται συμμετρικός $P^{-1} = P^T$. Σαν συνέπεια έχουμε:

$$\begin{aligned}
e^T(I - \lambda A_\times)^{-1} e &= e^T(I - \lambda A_i \otimes A_j)^{-1} e \\
&= e^T(I - \lambda(P_i D_i P_i^{-1}) \otimes (P_j D_j P_j^{-1}))^{-1} e \\
&= e^T(I - \lambda(P_i \otimes P_j)(D_i \otimes D_j)(P_i^{-1} \otimes P_j^{-1}))^{-1} e \\
&= (e^T P_i^{-1}) \otimes (e^T P_j^{-1})(I - \lambda D_i \otimes D_j)^{-1}(P_i e) \otimes (P_j e) \\
&= ((P_i e) \otimes (P_j e))^T (I - \lambda D_i \otimes D_j)^{-1}(P_i e) \otimes (P_j e)
\end{aligned} \tag{2.19}$$

Ως αποτέλεσμα το γινόμενο Kronecker γίνεται μεταξύ πινάκων μεγέθους $n$ και η αντιστροφή ανάγεται σε αντιστροφή διαγώνιου πίνακα, η οποία είναι γραμμική ως προς το μέγεθος του. Άλλος ένας δημοφιλής πυρήνας τυχαίου μονοπατιού που υλοποιήθηκε μέσα στο παρόν λογισμικό είναι ο εκθετικός πυρήνας τυχαίων περιπάτων.

**Ορισμός 2.18** (Εκθετικός Πυρήνας Τυχαίων Περιπάτων)**.** *Έστω $G_i$ και $G_j$ δύο γράφοι και έστω ότι το $A_\times$ αναπαριστά τον πίνακα γειτνίασης του γινόμενου τους γράφου $G_\times$ και έστω $V_\times$ το σύνολο των κόμβων του. Τότε, ο εκθετικός πυρήνας τυχαίων περιπάτων ορίζεται ως*

$$K_\times^\infty(G_i, G_j) = \sum_{p,q=1}^{|V_\times|} \Big[ \sum_{l=0}^{\infty} \frac{\lambda^l}{l!} A_\times^l \Big]_{pq} = e^T \exp(\lambda A_\times) e \tag{2.20}$$

*όπου $I$ είναι ο ταυτοτικός πίνακας, $e$ είναι ένα διάνυσμα που περιέχει μόνο άσσους και $\lambda$ είναι ένα θετικό, βάρος πραγματικής τιμής και* exp *η εκθετική συνάρτηση.*

Και σε αυτήν την περίπτωση η υπολογιστική πολυπλοκότητα του πυρήνα μπορεί να μειωθεί δείχνοντας το παρακάτω:

$$e^T \exp(\lambda A_\times) e = (e^T P_i) \otimes (e^T P_j) \exp(\lambda D_i \otimes D_j)(P_i^{-1} e) \otimes (P_j^{-1} e) \tag{2.21}$$

Όσον αφορά την περίπτωση επισημειωμένων γράφων στους κόμβους, ο πίνακας $A_\times$ μεταξύ δύο γράφων $i, j$ αντικαθίσταται σε αυτήν την περίπτωση με ένα $W_\times = \sum_{k=1}^{|\Sigma|} A_i^{l^k} \otimes A_j^{l^k}$ όπου ο πίνακας $A_i^{l^k}$ αντιστοιχεί στον πίνακα με άσσους στις ακμές που και οι δύο κόμβοι τους έχουν επισημείωση $l^k$. Στην περίπτωση αυτή, δεν υπάρχει τρόπος ώστε να βελτιστοποιήσουμε τον υπολογισμό του εκθετικού πυρήνα, ενώ για την βελτιστοποίηση του γεωμετρικού χρησιμοποιούνται οι μέθοδοι των συζυγών παραγώγων (conjugate gradients) που είναι ικανές να λύνουν αποδοτικά συστήματα της μορφής: $Mx = b$ [9].
Ο Mahé κ.ά. πρότειναν περαιτέρω επεκτάσεις των πυρήνων τυχαίων μονοπατιών [55]. Συγκεκριμένα, πρότειναν μία μέθοδο εμπλουτισμού (enrichment) των επισημειώσεων που ως προσέγγιση αυξάνει την διακριτική ακρίβεια (specificity) του πυρήνα και στις περισσότερες φορές ελαττώνει την υπολογιστική πολυπλοκότητα. Λόγω του γεγονότος ότι οι τυχαίοι περίπατοι μπορούν να συμπεριλαμβάνουν ξανά και ξανά τους ίδιους κόμβους, σε όμοιους γράφους ένα κύκλος μεταξύ κόμβων θα προσμετράται πολύ περισσότερο απ΄ότι οι υπόλοιποι. Το πρόβλημα αυτό είναι γνωστό στην βιβλιογραφία ως "tottering". Προκειμένου να το αντιμετωπίσουν ο Mahé κ.ά. πρότειναν ένα δευτέρας τάξης τυχαίο περίπατο Markov. Οι Sugiyama και Borgwardt έδειξαν ότι στην περίπτωση του γεωμετρικού πυρήνα, ο υποσταθμισμός των μονοπατιών με μήκος μεγαλύτερο του 1 από ένα από τον παράγοντα $l_k$ με $l < 1$ (προκειμένου να εξασφαλίζεται η σύγκλιση), έχει ως αποτέλεσμα το μέτρο ομοιότητας να κυριαρχείται από περιπάτους μήκους 1, ένα φαινόμενο γνωστό ως "halting" [74].

### 2.6.2 Πυρήνας Κοντινότερων Μονοπατιών

Ο πυρήνας κοντινότερων μονοπατιών shortest-path kernel μετατρέπει κάθε γράφο, σε γράφο κοντινότερων μονοπατιών και έπειτα συγκρίνει ένα ζευγάρι γράφων σε σχέση με τα μήκη

και τις επισημειώσεις που έχουν οι κόμβοι στα άκρα τους. Δεδομένου ενός γράφου εισόδου $G = (V, E)$, κατασκευάζουμε έναν γράφο $S = (V, E_s)$, που περιέχει το ίδιο σύνολο κόμβων με τον αρχικό και οι ακμές τους είναι ένα υπερσύνολο του αρχικού περιλαμβάνοντας όλα τα ζευγάρια κόμβων μεταξύ των οποίων υπάρχει μονοπάτι. Επιπλέον προσθέτουμε σε κάθε ακμή μία επισημείωση ίση με το βάρος του κοντινότερου μονοπατιού μεταξύ των δύο αυτών κόμβων. Στη συνέχεια δεδομένου του γράφου κοντινότερων μονοπατιών, ο πυρήνας κοντινότερων μονοπατιών ορίζεται ως εξής:

**Ορισμός 2.19** (Πυρήνας Κοντινότερων Μονοπατιών)**.** *Έστω $G_i$, $G_j$ δύο γράφοι και $S_i$, $S_j$ οι αντίστοιχοι γράφοι κοντινότερων μονοπατιών. Ο πυρήνας κοντινότερων μονοπατιών ορίζεται στα $S_i = (V_i, E_i)$ και $S_j = (V_j, E_j)$ ως*

$$k(S_i, S_j) = \sum_{e_i \in E_i} \sum_{e_j \in E_j} k^{(1)}_{walk}(e_i, e_j) \tag{2.22}$$

*όπου $k^{(1)}_{walk}(e_i, e_j)$ είναι ένας θετικά ημιορισμένος πυρήνας μεταξύ περιπάτων στις ακμές με μήκος* 1*.*

Σε γράφους με επισημειώσεις ο πυρήνας $k^{(1)}_{walk}(e_i, e_j)$ σχεδιάστηκε για να συγκρίνει όλα τα μήκη των κοντινότερων μονοπατιών που αντιστοιχούν σε ακμές $e_i$ και $e_j$, που οι επισημειώσεις των ακριανών τους κόμβων ταυτίζονται. Έστω $e_i = \{v_i, u_i\}$ και $e_j = \{v_j, u_j\}$. Τότε ο $k^{(1)}_{walk}(e_i, e_j)$ ορίζεται συνήθως ως:

$$\begin{aligned} k^{(1)}_{walk}(e_i, e_j) = {} & k_v(\ell(v_i), \ell(v_j))\ k_e(\ell(e_i), \ell(e_j))\ k_v(\ell(u_i), \ell(u_j)) \\ & + k_v(\ell(v_i), \ell(u_j))\ k_e(\ell(e_i), \ell(e_j))\ k_v(\ell(u_i), \ell(v_j)) \end{aligned} \tag{2.23}$$

όπου $k_v$ είναι ένας πυρήνας που συγκρίνει επισημειώσεις κόμβων και $k_e$ ένας πυρήνας που συγκρίνει μήκη κοντινότερων μονοπατιών. Οι επισημειώσεις κόμβων συχνά συγκρίνονται μέσω ενός πυρήνα Dirac, ενώ τα μήκη των συντομότερων μονοπατιών συχνά συγκρίνονται με ένα πυρήνα Dirac και πιο σπάνια με ένα πυρήνα brownian bridge [10]. Ο πυρήνας Dirac ορίζεται ως:

**Ορισμός 2.20** (Πυρήνας Dirac)**.** *Έστω δύο αντικείμενα $o_i, o_j \in \mathcal{O}$ και μία πράξη ισότητας $\backprime=\backprime$ στον χώρο $\mathcal{O}$. Τότε ο πυρήνας Dirac ορίζεται ως:*

$$d(o_i, o_j) = \begin{cases} 1, \text{αν } o_i = o_j \\ 0, \text{αλλιώς} \end{cases}$$

ενώ ο πυρήνας brownian bridge ορίζεται ως:

**Ορισμός 2.21** (Πυρήνας Brownian Bridge)**.** *Έστω δύο αριθμοί $l_i, l_j \in \mathbb{R}$. Τότε ο πυρήνας Brownian Bridge ορίζεται ως:*

$$bb_c(l_i, l_j) = \max(0, c - |l_i - l_j|)$$

*με το $c$ να αποτελεί μία ελεύθερη παράμετρο του πυρήνα.*

Η υπολογιστική πολυπλοκότητα του παραπάνω αλγορίθμου είναι της τάξης του $\mathcal{O}(n^4)$, που όμως μπορεί να μειωθεί σημαντικά στην πράξη αν δημιουργήσουμε διανύσματα χαρακτηριστικών με θέσεις που κρατούν την συχνότητα εμφάνισης για όλες τις δυνατές τιμές που μπορούν να λάβουν τα μονοπάτια και οι επισημειώσεις (αν υπάρχουν). Κάτι τέτοιο είναι ιδιαίτερα αποτελεσματικό σε μία συλλογή γράφων, με τον συνολικό υπολογισμό της μήτρας πυρήνα να ανάγεται σε έναν πολλαπλασιασμό πινάκων $n \times |\Sigma|$, όπου $\Sigma$ είναι το σύνολο όλων των δυνατών συνδυασμών μήκους κοντινότερων μονοπατιών και επισημειώσεων, πράγμα που αποτελεί και τον κυρίαρχο όρο στην υπολογιστική πολυπλοκότητα.

### 2.6.3 Πυρήνας Γραφιδίων

Ο πυρήνας γραφιδίων αποσυνθέτει ένα γράφο σε γραφίδια (δηλ. μικρού μεγέθους υπογράφους με $k$ κόμβους, όπου συνήθως $k \in \{3, 4, 5\}$) [65] και μετράει την συχνότητα εμφάνισης των γραφιδίων σε υπογράφους των γράφων εισόδου. Έστω $\mathcal{G} = \{graphlet_1, graphlet_2, \ldots, graphlet_r\}$ το σύνολο γραφιδίων μεγέθους $k$. Έστω ακόμα $f_G \in \mathbb{N}^r$ ένα διάνυσμα τέτοιο ώστε το $i$-οστό του στοιχείο ισούται με την συχνότητα εμφάνισης του $graphlet_i$ στο $G$, $f_{G,i} = \#(graphlet_i \sqsubseteq G)$. Τότε, ο πυρήνας γραφιδίων ορίζεται ως εξής.

**Ορισμός 2.22** (Πυρήνας Γραφιδίων μεγέθους $k$)**.** *Έστω $G_i$, $G_j$ δύο γράφοι μεγέθους $n \geq k$, και $f_{G_i}, f_{G_j}$ διανύσματα τα οποία μετρούν την συχνότητα εμφάνισης κάθε γραφιδίου μεγέθους $k$ σε δύο γράφους. Τότε ο πυρήνας γραφιδίων ορίζεται ως*

$$k(G_i, G_j) = f_{G_i}^{\top} \, f_{G_j} \tag{2.24}$$

Όπως φαίνεται από τον παραπάνω ορισμό, ο πυρήνας γραφιδίων υπολογίζεται με άμεσα ορισμένες αναπαραστάσεις χαρακτηριστικών. Αρχικά, υπολογίζουμε την αναπαράσταση κάθε γράφου στο χώρο χαρακτηριστικών και έπειτα η τιμή του πυρήνα υπολογίζεται ως το εσωτερικό γινόμενο μεταξύ δύο διανυσμάτων χαρακτηριστικών. Το υπολογιστικό φράγμα του πυρήνα γραφιδίων εισάγεται από το γεγονός ότι μία εξαντλητική απαρίθμηση των γραφιδίων είναι υπολογιστικά ακριβή. Από την στιγμή που υπάρχουν $\binom{n}{k}$ υπογράφοι μεγέθους $k$ στον γράφο, ο υπολογισμός του διανύσματος χαρακτηριστικών για ένα γράφο μεγέθους $n$ απαιτεί χρόνο $\mathcal{O}(n^k)$. Για να λύσει αυτό το πρόβλημα ο Shervashidze κ.α. κατέφυγε στην δειγματοληψία [72]. Χρησιμοποιώντας τις ανισότητες του Weissman κ.α. [88], έδειξαν ότι δειγματοληπτώντας ένα δεδομένο αριθμό γραφιδίων, οι εμπειρικές κατανομές των γραφιδίων θα είναι ικανοποιητικά κοντά στην πραγματική κατανομή τους στο γράφο. Μία εναλλακτική στρατηγική που μειώνει την εκφραστικότητα του πυρήνα είναι η απαρίθμηση μόνο συνδεδεμένων γραφιδίων $k$ κόμβων και όχι όλων των δυνατών.

### 2.6.4 Σκελετός Πυρήνα Weisfeiler-Lehman

Ο σκελετός πυρήνα Weisfeiler-Lehman λειτουργεί στην κορυφή υπάρχοντων πυρήνων γράφων και είναι εμπνευσμένος από το τεστ Weisfeiler-Lehman για τον ισομορφισμό γράφων [87]. Η κύρια ιδέα του αλγορίθμου Weisfeiler-Lehman είναι η αντικατάσταση των επισημειώσεων κάθε κόμβου με ένα πολυσύνολο (multiset) επισημειώσεων που αποτελείται από την επισημείωση

του αρχικού κόμβου και των διατεταγμένων επισημειώσεων των γειτόνων. Το προκύπτον πολυσύνολο συμπιέζεται σε μία νέα επισημείωση, που δεν έχει ξαναεμφανιστεί. Αυτή η διαδικασία επανεπισημείωσης επαναλαμβάνεται για $h$ επαναλήψεις, ταυτόχρονα για όλους τους κόμβους και τους γράφους της εισόδου. Ως επακόλουθο, οι συμπιεσμένες επισημειώσεις δύο κόμβων από διαφορετικούς γράφους θα ταυτίζονται αν και μόνον αν, ταυτίζονται οι επισημειώσεις των πολυσυνόλων τους. Πιο τυπικά, δεδομένου ενός γράφου $G = (V, E)$ που είναι συνδεδεμένος με μία συνάρτηση επισημειώσεων $\ell = \ell_0$, ο γράφος Weisfeiler-Lehman του $G$ σε ύψος $i$ είναι ένας γράφος $G_i = (V, E)$ συνδεδεμένος με μία συνάρτηση επισημειώσεων $\ell_i$ η οποία έχει προκύψει μετά από $i$ επαναλήψεις της διαδικασίας επανεπισημείωσης που περιγράφηκε παραπάνω. Η ακολουθία Weisfeiler-Lehman μέχρι το ύψος $h$ του $G$ αποτελείται από τους γράφους Weisfeiler-Lehman του $G$ σε ύψη από το 0 έως το $h$, $\{G_0, G_1, \ldots, G_h\}$.

**Ορισμός 2.23** (Σκελετός Weisfeiler-Lehman)**.** *Έστω οποιοσδήποτε πυρήνας $k$ μεταξύ γράφων, που θα ονομάσουμε πυρήνα βάσης (base kernel). Τότε ο σκελετός Weisfeiler-Lehman σε $h$ επαναλήψεις με τον πυρήνα βάσης $k$ μεταξύ δύο γράφων $G$ και $G'$ ορίζεται ως*

$$k_{WL}(G, G') = k(G_0, G'_0) + k(G_1, G'_1) + \ldots + k(G_h, G'_h) \tag{2.25}$$

*όπου $h$ το πλήθος των επαναλήψεων Weisfeiler-Lehman και $\{G_0, G_1, \ldots, G_h\}$ και $\{G'_0, G'_1, \ldots, G'_h\}$ οι ακολουθίες Weisfeiler-Lehman του $G$ και του $G'$ .*

Από τον παραπάνω ορισμό, είναι φανερό ότι κάθε πυρήνας γράφων που λαμβάνει υπόψιν διακριτές επισημειώσεις κόμβων μπορεί να ενταχθεί στον σκελετό πυρήνα Weisfeiler-Lehman και να συγκρίνει τους γράφους βάση ολόκληρης της ακολουθίας Weisfeiler-Lehman.
Όταν ο υπολογισμός του πυρήνα βάσης αφορά την αρίθμηση κοινών αρχικών και συμπιεσμένων επισημειώσεων στους δύο γράφους, τότε ο πυρήνας είναι ισοδύναμος με ένα πυρήνα που συγκρίνει υποδέντρα που εξήχθησαν από τους δύο γράφους. Ο πυρήνας Weisfeiler-Lehman-υποδέντρων είναι ένας πολύ δημοφιλής αλγόριθμος, που θεωρείται από τους πιο αποτελεσματικούς στην παρούσα βιβλιογραφία, τόσο στην ποιότητα όσο και στην ταχύτητα του, όσον αφορά την ταξινόμηση γράφων.

**Ορισμός 2.24** (Πυρήνας Weisfeiler-Lehman υποδέντρων)**.** *Έστω $G$, $G'$ δύο γράφοι. Ορίζουμε ως $\Sigma_i \subseteq \Sigma$ το σύνολο των γραμμάτων που προκύπτουν σαν επισημειώσεις των κόμβων, τουλάχιστον μία φορά στα $G$ και $G'$ στο τέλος της $i$-οστής επανάληψης του αλγορίθμου Weisfeiler-Lehman. Έστω $\Sigma_0$ το σύνολο των αρχικών επισημειώσεων των γράφων $G$ και $G'$. Θα υποθέσουμε ότι όλα τα $\Sigma_i$ δεν έχουν κοινά στοιχεία μεταξύ τους. Χωρίς βλάβη της γενικότητας, υποθέτουμε ότι κάθε $\Sigma_i = \{\sigma_{i1}, \ldots, \sigma_{i|\Sigma_i|}\}$ είναι ταξινομημένο. Θα ορίσουμε μία απεικόνιση $c_i : \{G, G'\} \times \Sigma_i \to \mathbb{N}$ τέτοια ώστε $c_i(G, \sigma_{ij})$ να είναι ο αριθμός των εμφανίσεων του γράμματος $\sigma_{ij}$ στον γράφο $G$.*
*Ο πυρήνας υποδέντρων Weisfeiler-Lehman (Weisfeiler-Lehman subtree kernel)μεταξύ δύο γράφων $G$ και $G'$ με $h$ επαναλήψεις ορίζεται ως*

$$k(G, G') = \langle \phi(G), \phi(G') \rangle \tag{2.26}$$

*όπου*

$$\phi(G) = (c_0(G, \sigma_{01}), \ldots, c_0(G, \sigma_{0|\Sigma_0|}), \ldots, c_h(G, \sigma_{h1}), \ldots, c_h(G, \sigma_{h|\Sigma_h|})) \tag{2.27}$$

*και*

$$\phi(G') = (c_0(G', \sigma_{01}), \ldots, c_0(G', \sigma_{0|\Sigma_0|}), \ldots, c_h(G', \sigma_{h1}), \ldots, c_h(G', \sigma_{h|\Sigma_h|})) \tag{2.28}$$

Μπορούμε να δείξουμε ότι ο παραπάνω ορισμός είναι ισοδύναμος με την σύγκριση των αριθμών κοινών υποδέντρων μεταξύ των δύο γράφων [73]. Ο πυρήνας υποδέντρων ονομάζεται σε άλλες περιπτώσεις πυρήνας ιστογράμματος κόμβων (vertex histogram kernel).
Τέλος η πολυπλοκότητα του σκελετού Weisfeiler-Lehman είναι ίση με $\mathcal{O}(hm\mathcal{O}(T_{\text{base-kernel}}))$ όπου με $\mathcal{O}(T_{\text{base-kernel}})$ συμβολίζουμε την πολυπλοκότητα ενός πυρήνα βάσης.

### 2.6.5 Πυρήνας Πυραμιδικού Ταιριάσματος

Ο πυρήνας πυραμιδικού ταιριάσματος (pyramid match είναι πολύ δημοφιλής στον χώρο της όρασης υπολογιστών και έχει αποδειχθεί ιδιαίτερα χρήσιμος σε πολλές εφαρμογές από την αναγνώριση αντικειμένων μέχρι την ανάκτηση εικόνας [33, 50]. Ο πυρήνας πυραμιδικού ταιριάσματος επεκτείνει την εφαρμοσιμότητά του σε δεδομένα με τη μορφή γράφου [62]. Αυτός ο πυρήνας μπορεί να διαχειριστεί τόσο γράφους με διακριτές επισημειώσεις όσο και γράφους χωρίς επισημειώσεις.
Ο πυρήνας πυραμιδικού ταιριάσματος πρώτα παριστά όλους τους κόμβους ενός γράφου σε έναν διανυσματικό χώρο χαμηλών διαστάσεων, χρησιμοποιώντας τα ιδιοδιανύσματα των $d$ μεγαλύτερων σε μέγεθος ιδιοτιμών του πίνακα γειτνίασης του γράφου. Από την στιγμή που τα πρόσημα αυτών των ιδιοδιανυσμάτων είναι αυθαίρετα, αντικαθιστά όλα τους τα συστατικά στοιχεία με τις απόλυτες τιμές τους. Κάθε κόμβος είναι συνεπώς ένα σημείο σε έναν $d$-διάστατο μοναδιαίο υπερκύβο. Για να βρεθεί μία προσεγγιστική αντιστοιχία μεταξύ των συνόλων των κόμβων των δύο γράφων, ο πυρήνας απεικονίζει τα σημεία σε ιστογράμματα πολλαπλών αναλύσεων (multi-resolution) και συγκρίνει τα προκύπτοντα ιστογράμματα με μία σταθμισμένη συνάρτηση τομής ιστογραμμάτων.
Αρχικά, ο πυρήνας διαμερίζει των χώρο χαρακτηριστικών σε περιοχές με αυξάνον μέγεθος και παίρνει το σταθμισμένο άθροισμα όλων των αντιστοιχίσεων που προκύπτουν σε κάθε επίπεδο. Δύο σημεία ταιριάζουν αν πέφτουν στην ίδια περιοχή, ενώ ταιριάσματα μεταξύ περιοχών μεγαλύτερου μεγέθους, σταθμίζονται λιγότερο από ταιριάσματα μικρότερων περιοχών. Ο πυρήνας επαναληπτικά προσαρμόζει ένα πλέγμα κελιών αυξανόμενου μεγέθους στον $d$-διάστατο μοναδιαίο υπερκύβο. Κάθε κελί συνδέεται με μία συγκεκριμένη διάσταση και το μέγεθος του σε κάθε διάσταση διπλασιάζεται σε κάθε επανάληψη, ενώ το μέγεθος του στις άλλες διαστάσεις παραμένει σταθερό και ίσο με 1. Δεδομένης μία ακολουθίας επιπέδων από το 0 ως το $L$, τότε στο επίπεδο $l$, ο $d$-διάστατος μοναδιαίος υπερκύβος έχει $2^l$ κελιά σε κάθε διάσταση και $D = 2^l d$ κελιά στο σύνολο. Δεδομένου ενός ζευγαριού γράφων $G, G'$, έστω $H_G^l$ και $H_{G'}^l$ που συμβολίζουν τα ιστογράμματα των $G$ και $G'$ στα επίπεδα $l$, και $H_G^{l(i)}$, $H_{G'}^{l(i)}$, ο αριθμός των κόμβων στα $G$, $G'$ που βρίσκονται στο $i$-οστό κελί. Ο αριθμός των σημείων μεταξύ δύο συνόλων, τα

οποία ταυτίζονται στο επίπεδο $l$ υπολογίζεται έπειτα χρησιμοποιώντας την συνάρτηση τομής ιστογραμμάτων

$$I(H_G^l, H_{G'}^l) = \sum_{i=1}^{D} \min\left(H_G^l(i), H_{G'}^l(i)\right) \tag{2.29}$$

Τα ταιριάσματα που προκύπτουν στο επίπεδο $l$ συμβαίνουν ακόμα στα επίπεδα $0, \ldots, l-1$. Μας ενδιαφέρουν μόνο εκείνα τα ταιριάσματα που είναι καινούργια σε κάθε νέο ταίριασμα που δίνονται από την διαφορά $I(H_{G_1}^l, H_{G_2}^l) - I(H_{G_1}^{l+1}, H_{G_2}^{l+1})$ για $l = 0, \ldots, L-1$. Ο αριθμός των νέων ταιριασμάτων που προκύπτει σε κάθε επίπεδο στην πυραμίδα σταθμίζεται με βάση το μέγεθος των κελιών αυτού του επιπέδου. Ταιριάσματα που προκύπτουν μεταξύ μικρότερων κελιών σταθμίζονται περισσότερο από αυτά που προκύπτουν σε μεγαλύτερα κελιά. Συγκεκριμένα, το βάρος για το επίπεδο $l$ είναι ίσο με $\frac{1}{2^{L-l}}$. Συνεπώς, τα βάρη είναι εκθετικά αντιστρόφως ανάλογα του μήκους της πλευράς των κελιών, η οποία αλλάζει όσο το μέγεθος των κελιών αυξάνεται. Ο πυρήνας πυραμιδικού ταιριάσματος ορίζεται ως εξής

$$k(G, G') = I(H_G^L, H_{G'}^L) + \sum_{l=0}^{L-1} \frac{1}{2^{L-l}} \left(I(H_G^l, H_{G'}^l) - I(H_G^{l+1}, H_{G'}^{l+1})\right) \tag{2.30}$$

Η πολυπλοκότητα του είναι ίση με $\mathcal{O}(dnL)$ όπου $n$ ο αριθμός των κόμβων στους γράφους που συγκρίνονται.

Στην περίπτωση γράφων με επισημειώσεις, ο πυρήνας περιορίζει τα ταιριάσματα σε αυτά μεταξύ κόμβων που μοιράζονται την ίδια επισημείωση. Αναπαριστά κάθε γράφο σαν ένα σύνολο συνόλων διανυσμάτων και ταιριάζει ζευγάρια κόμβων από τα σύνολα κόμβων δύο γράφων με κοινές επισημειώσεις, χρησιμοποιώντας τον πυρήνα πυραμιδικού ταιριάσματος. Ο προκύπτων πυρήνας για επισημειωμένους γράφους αντιστοιχεί στο άθροισμα των ξεχωριστών πυρήνων

$$k(G, G') = \sum_{i=1}^{c} k^i(G, G') \tag{2.31}$$

όπου $c$ είναι ο αριθμός των διαφορετικών επισημειώσεων και $k^i(G_1, G_2)$ ο πυρήνας πυραμιδικού ταιρίασματος μεταξύ συνόλων κόμβων μεταξύ δύο γράφων, που και στα δύο έχει δοθεί η επισημείωση $i$.

### 2.6.6 Πυρήνας Lovász $\vartheta$

Ο αριθμός Lovász $\vartheta(G)$ ενός γράφου $G = (V, E)$ είναι ένας πραγματικός αριθμός που αποτελεί το άνω φράγμα στην χωρητικότητα Shannon ενός γράφου. Εισήχθη από τον László Lovász το 1979 [52]. Ο αριθμός Lovász είναι πολύ στενά συνδεδεμένος με την έννοια της ορθοκανονικής αναπαραστάσης γράφων. Η ορθοκανονική αναπαράσταση ενός γράφου $G$ αποτελείται από ένα σύνολο μοναδιαίων διανυσμάτων $U_G = \{\mathbf{u}_i \in \mathbb{R}^d : ||\mathbf{u}_i|| = 1\}_{i \in V}$ όπου σε κάθε κόμβο $i$ ανατίθεται ένα μοναδιαίο διάνυσμα $\mathbf{u}_i$ τέτοιο ώστε $(i, j) \notin E \implies \mathbf{u}_i^\top \mathbf{u}_j = 0$. Συγκεκριμένα, ο αριθμός Lovász ενός γράφου $G$ ορίζεται σαν

$$\vartheta(G) = \min_{\mathbf{c}, U_G} \max_{i \in V} \frac{1}{(\mathbf{c}^\top \mathbf{u}_i)^2} \tag{2.32}$$

όπου $\mathbf{c} \in \mathbb{R}^d$ είναι ένα μοναδιαίο διάνυσμα και $U_G$ μία ορθοκανονική αναπαράσταση του $G$. Γεωμετρικά το $\vartheta(G)$ ορίζεται σαν τον μικρότερο κώνο που εσωκλείει μία έγκυρη ορθοκανονική αναπαράσταση $U_G$. Ο αριθμός Lovász $\vartheta(G)$ ενός γράφου $G$ μπορεί να υπολογιστεί σε οποιαδήποτε επιθυμητή ακρίβεια σε πολυωνυμικό χρόνο, λύνοντας ένα πρόβλημα βελτιστοποίησης ημιορισμένου προγραμματισμού.

Ο πυρήνας Lovász $\vartheta$ χρησιμοποιεί τις ορθοκανονικές αναπαραστάσεις που συνδέονται με τον αριθμό Lovász για να συγκρίνει δύο γράφους [42]. Αυτός ο πυρήνας αφορά γράφους χωρίς επισημειώσεις. Δεδομένου μίας συλλογής γράφων, πρώτα αναπαριστά τις ορθοκανονικές αναπαραστάσεις των κόμβων κάθε γράφου, υπολογίζοντας τον αριθμό Lovász $\vartheta$. Έτσι, $U_G$ είναι το σύνολο που περιέχει όλες τις ορθοκανονικές αναπαραστάσεις του $G$. Έστω $S \subseteq V$ ένα υποσύνολο των κόμβων του $G$. Τότε, ο αριθμός Lovász ενός συνόλου κόμβων του $S$ ορίζεται ως εξής

$$\vartheta_S(G) = \min_{\mathbf{c}} \max_{i \in S} \frac{1}{(\mathbf{c}^\top \mathbf{u}_i)^2} \tag{2.33}$$

όπου $\mathbf{c} \in \mathbb{R}^d$ είναι ένα μοναδιαίο διάνυσμα και $\mathbf{u}_i$ είναι η αναπαράσταση του κόμβου $i$ η οποία προκύπτει από τον υπολογισμό του αριθμού Lovász $\vartheta(G)$ του $G$. Η τιμή Lovász ενός συνόλου κόμβων $S$ αναπαριστά την γωνία του μικρότερου κώνου που εσωκλείει το σύνολο των ορθοκανονικών αναπαραστάσεων αυτών των κόμβων (δηλ. το υποσύνολο του $U_G$ που ορίζεται σαν $\{\mathbf{u}_i : \mathbf{u}_i \in U_G, i \in S\}$).

Ο πυρήνας Lovász $\vartheta$ μεταξύ δύο γράφων $G, G'$ ορίζεται ως:

$$k_{\text{Lovász}}(G, G') = \sum_{S \subseteq V} \sum_{S' \subseteq V'} \delta(|S|, |S'|) \frac{1}{Z_{|S|}} k(\vartheta_S(G), \vartheta_{S'}(G')) \tag{2.34}$$

όπου $Z_{|S|} = \binom{|V|}{|S|}\binom{|V'|}{|S|}$, $\delta(|S|, |S'|)$ είναι ένας πυρήνας dirac (βλέπε ορισμό 2.20) και $k$ είναι ένας θετικά ημιορισμένος πυρήνας μεταξύ πραγματικών αριθμών (π.χ. γραμμικός ή γκαουσιανός).

Ο πυρήνας Lovász $\vartheta$ αποτελείται από δύο κύρια βήματα: (1) υπολογισμός του αριθμού Lovász $\vartheta$ του κάθε γράφου και η εξαγωγή των αντίστοιχων ορθοκανονικών αναπαραστάσεων και (2) ο υπολογισμός του αριθμού Lovász για όλους τους υπογράφους (δηλ. του υποσυνόλου των κόμβων $S \subseteq V$) του κάθε γράφου. Ο ακριβής υπολογισμός του πυρήνα Lovász $\vartheta$, δεν είναι κάτι εφικτό στις περισσότερες πραγματικές εφαρμογές από την στιγμή που απαιτεί τον υπολογισμό των ελάχιστων κώνων που εσωκλείουν $2^n$ σύνολα κόμβων.

Όταν λοιπόν ασχολούμαστε με μεγάλους γράφους, είναι σημαντικό να καταφεύγουμε στην δειγματοληψία. Έτσι, δεδομένου ενός γράφου $G$, αντί να υπολογίσουμε την τιμή Lovász σε όλα τα $2^n$ σύνολα κόμβων, την υπολογίζουμε σε ένα μικρότερο πλήθος από υπογράφους $\mathfrak{S} \in 2^V$. Τότε, ο πυρήνας Lovász $\vartheta$ ορίζεται ως εξής:

$$\hat{k}_{\text{Lovász}}(G, G') = \sum_{S \subseteq \mathfrak{S}} \sum_{S' \subseteq \mathfrak{S}'} \delta(|S|, |S'|) \frac{1}{\hat{Z}_{|S|}} k(\vartheta_S(G), \vartheta_{S'}(G'))$$

όπου $\hat{Z}_{|S|} = |\mathfrak{S}_{|S|}||\mathfrak{S}'_{|S|}|$ και $\mathfrak{S}_{|S|} = \{B \in \mathfrak{S} : |B| = |S|\}$ είναι το υποσύνολο του $\mathfrak{S}$ που αποτελείται από όλα τα σύνολα του με πληθικότητα ίση με του $S$.

Η χρονική πολυπλοκότητα υπολογισμού του $\hat{k}_{\text{Lovász}}(G, G')$ είναι $\mathcal{O}(n^2 m\epsilon^{-1} + s^2 T(k) + sn)$ όπου $T(k)$ είναι η πολυπλοκότητα υπολογισμού του πυρήνα βάσης $k$, $n = |V|$, $m = |E|$ και $s = \max(|\mathfrak{S}|, |\mathfrak{S}'|)$. Ο πρώτος όρος αναπαριστά το κόστος επίλυσης ενός προβλήματος βελτιστοποίησης ημιορισμένου προγραμματισμού που υπολογίζει τον αριθμό Lovász $\vartheta$. Ο δεύτερος όρος αντιστοιχεί στην πολυπλοκότητα χειρότερης περίπτωσης για τον υπολογισμό του αθροίσματος των τιμών Lovász. Τέλος, ο τρίτος όρος είναι το κόστος υπολογισμού των τιμών Lovász των δειγματοληπτημένων υποσυνόλων κόμβων.

### 2.6.7 Πυρήνας SVM-$\vartheta$

Ο πυρήνας SVM-$\vartheta$ συνδέεται άμεσα με τον πυρήνα Lovász $\vartheta$ [42]. Ο πυρήνας Lovász $\vartheta$ υποφέρει από μεγάλη υπολογιστική πολυπλοκότητα και ο πυρήνας SVM-$\vartheta$ σχεδιάστηκε σαν μία πιο αποδοτική εναλλακτική. Όπως και ο πυρήνας Lovász $\vartheta$, αφορά γράφους χωρίς επισημειώσεις.
Δοθέντος ενός γράφου $G = (V, E)$ τέτοιου ώστε $|V| = n$, ο αριθμός Lovász του $G$ μπορεί να οριστεί ως

$$\vartheta(G) = \min_{\mathbf{K} \in L} \omega(\mathbf{K}) \tag{2.35}$$

όπου $\omega(\mathbf{K})$ είναι το SVM μίας κατηγορίας (one-class) που δίνεται από

$$\omega(\mathbf{K}) = \max_{\alpha_i > 0} 2 \sum_{i=1}^{n} \alpha_i - \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j \mathbf{K}_{ij} \tag{2.36}$$

και το $L$ είναι ένα σύνολο από θετικά ημιορισμένους πίνακες που ορίζεται ως

$$L = \{\mathbf{K} \in S_n^+ : \mathbf{K}_{ii} = 1, \mathbf{K}_{ij} = 0 \, \forall (i, j) \notin E\} \tag{2.37}$$

όπου $S_n^+$ είναι το σύνολο όλων $n \times n$ των θετικά ημιορισμένων πινάκων.
Ο πυρήνας SVM-$\vartheta$ πρώτα υπολογίζει τον πίνακα $\mathbf{K}_{LS}$ που είναι ίσος με

$$\mathbf{K}_{LS} = \frac{\mathbf{A}}{\rho} + \mathbf{I} \tag{2.38}$$

όπου $\mathbf{A}$ είναι ο πίνακας γειτνίασης του $G$, $\mathbf{I}$ είναι ο $n \times n$ ταυτοτικός πίνακας και $\rho \geq -\lambda_n$ με $\lambda_n$ να είναι η μικρότερη ιδιοτιμή του $\mathbf{A}$. Ο πίνακας $\mathbf{K}_{LS}$ είναι κατασκευασμένος έτσι ώστε να είναι θετικά ημιορισμένος και αποδεικνύεται ότι:

$$\omega(\mathbf{K}_{LS}) = \sum_{i=1}^{n} \alpha_i \tag{2.39}$$

όπου $\alpha_i$ είναι οι όροι που μεγιστοποιούν την εξίσωση 2.36 [41]. Ακόμα, έχει αποδειχθεί ότι σε συγκεκριμένες οικογένειες γράφων (π.χ. τυχαίοι γράφοι Erdös Rényi), το $\omega(\mathbf{K}_{LS})$ προσεγγίζει με μεγάλη πιθανότητα κατά ένα σταθερό παράγοντα το $\vartheta(G)$.
Τότε, ο πυρήνας SVM-$\vartheta$ ορίζεται ως εξής:

$$k_{\text{SVM}}(G, G') = \sum_{S \subseteq V} \sum_{S' \subseteq V'} \delta(|S|, |S'|) \frac{1}{Z_{|S|}} k\Big( \sum_{i \in S} \alpha_i, \sum_{j \in S'} \alpha_j \Big) \tag{2.40}$$

όπου $Z_{|S|} = \binom{|V|}{|S|}\binom{|V'|}{|S|}$, $\delta(|S|,|S'|)$ είναι ένας πυρήνας Dirac (βλέπε ορισμό 2.20) και $k$ ένας θετικά ημιορισμένος πυρήνας μεταξύ πραγματικών τιμών (π.χ. γραμμικός, γκαουσιανός).
Ο πυρήνας SVM-$\vartheta$ απαρτίζεται από τρία κύρια βήματα: (1) κατασκευή του πίνακα $\mathbf{K}_{LS}$ του $G$ που απαιτεί χρόνο $\mathcal{O}(n^3)$ (2) λύση του προβλήματος SVM μίας κλάσης σε χρόνο $\mathcal{O}(n^2)$, προκειμένου να χρησιμοποιήσουμε τις τιμές των $\alpha_i$ και (3) υπολογισμό του αθροίσματος των τιμών $\alpha_i$ για όλους τους υπογράφους (δηλαδή τα υποσύνολα κόμβων $S \subseteq V$) του κάθε γράφου. Όπως και στον πυρήνα Lovász $\vartheta$ ο υπολογισμός της παραπάνω ποσότητας για όλα τα σύνολα $2^n$ των κόμβων δεν είναι υπολογιστικά εφικτή σε πραγματικά δεδομένα. Για την επίλυση αυτού του προβλήματος, χρησιμοποιούμε και στην περίπτωση SVM-$\vartheta$ την μέθοδο της δειγματοληψίας. Δεδομένου ενός γράφου $G$, ο πυρήνας δειγματοληπτεί ένα δεδομένο αριθμό υπογράφων $\mathfrak{S} \in 2^V$. Τότε, ο πυρήνας SVM-$\vartheta$ ορίζεται ως εξής

$$\hat{k}_{\mathrm{SVM}}(G,G') = \sum_{S\subseteq\mathfrak{S}} \sum_{S'\subseteq\mathfrak{S}'} \delta(|S|,|S'|) \frac{1}{\hat{Z}_{|S|}} \Big( \sum_{i\in S} \alpha_i, \sum_{j\in S'} \alpha_j \Big)$$

όπου $\hat{Z}_{|S|} = |\mathfrak{S}_{|S|}||\mathfrak{S}'_{|S|}|$, ενώ με το $\mathfrak{S}_{|S|}$ συμβολίζουμε το υποσύνολο του $\mathfrak{S}$ που αποτελείται από όλα τα σύνολα με πληθικό αριθμό $|S|$, συγκεκριμένα $\mathfrak{S}_{|S|} = \{B \in \mathfrak{S} : |B| = |S|\}$.
Η χρονική πολυπλοκότητα του υπολογισμού $\hat{k}_{\mathrm{SVM}}(G,G')$ είναι $\mathcal{O}(n^3 + s^2T(k) + sn)$ όπου $T(k)$ είναι η πολυπλοκότητα υπολογισμού του πυρήνα βάσης $k$ και $s = \max(|\mathfrak{S}|,|\mathfrak{S}'|)$. Ο πρώτος όρος αναπαριστά το κόστος υπολογισμού $\mathbf{K}_{LS}$ (στον οποίο υπερισχύει το κόστος της αποσύνθεσης ιδιοτιμών). Ο δεύτερος όρος αντιστοιχεί στην πολυπλοκότητα χειρότερης περίπτωσης για την σύγκριση των αθροισμάτων των τιμών $\alpha_i$. Ο τρίτος και τελευταίος όρος αφορά το κόστος υπολογισμού του αθροίσματος των τιμών $\alpha_i$ για τα δειγματοληπτημένα υποσύνολα κόμβων.

### 2.6.8 Πολυκλιμακωτός Λαπλασιανός Πυρήνας

Ο πολυκλιμακωτός Λαπλασιανός πυρήνας μπορεί να χειριστεί επισημειωμένους γράφους, είτε με διακριτές επισημειώσεις είτε με χαρακτηριστικά [47]. Λαμβάνει υπόψη την δομή των γράφων σε ένα εύρος διαφορετικών κλιμάκων κατασκευάζοντας μία ιεραρχία εμφωλευμένων υπογράφων. Αυτοί οι υπογράφοι συγκρίνονται ο ένας με τον άλλον κάνοντας χρήση ενός άλλου πυρήνα, που αποκαλείται Λαπλασιανός πυρήνας στον χώρο χαρακτηριστικών. Αυτός ο πυρήνας δύναται να καταστήσει έναν πυρήνα ανάμεσα στους κόμβους δύο ή περισσοτέρων γράφων, έναν πυρήνα μεταξύ των ίδιων των γράφων. Από τη στιγμή που ο ακριβής υπολογισμός του πολυκλιμακωτού Λαπλασιανού είναι μία πολύ υπολογιστικά ακριβή διαδικασία, ο πυρήνας χρησιμοποιεί μία πιθανοτική διαδικασία παρόμοια με την πολύ γνωστή προσέγγιση μέθοδο Nyström για τον υπολογισμό μητρών πυρήνα [89].
Έστω $G = (V,E)$ ένας μη-κατευθυνόμενος γράφος τέτοιος ώστε $n = |V|$. Η Λαπλασιανή του $G$ είναι ένας $n \times n$ πίνακας που ορίζεται ως

$$\mathbf{L} = \mathbf{D} - \mathbf{A}$$

όπου $\mathbf{A}$ είναι ένας πίνακας γειτνίασης του $G$ και $\mathbf{D}$ είναι ένας διαγώνιος πίνακας τέτοιος ώστε $\mathbf{D}_{ii} = \sum_j \mathbf{A}_{ij}$.

Δεδομένου δύο γράφων $G_1$ και $G_2$ $n$ κόμβων, μπορούμε να ορίσουμε έναν πυρήνα μεταξύ τους ως ένα πυρήνα μεταξύ των αντίστοιχων κανονικών κατανομών $p_1 = \mathcal{N}(\mathbf{0}, {\mathbf{L_1}}^{-1})$ και $p_2 = \mathcal{N}(\mathbf{0}, {\mathbf{L_2}}^{-1})$ όπου $\mathbf{0}$ είναι το $n$-διάστατο διάνυσμα μηδενικών. Πιο συγκεκριμένα, δοθέντος δύο γράφων $G_1$ και $G_2$ $n$ κόμβων με Λαπλασιανούς πίνακες $\mathbf{L_1}$ και $\mathbf{L_2}$ αντίστοιχα, ο Λαπλασιανός πυρήνας γράφων με παράμετρο $\gamma$ μεταξύ δύο γράφων είναι

$$k_{LG}(G_1, G_2) = \frac{|(\frac{1}{2}\mathbf{S}_1^{-1} + \frac{1}{2}\mathbf{S}_2^{-1})^{-1}|^{1/2}}{|\mathbf{S}_1|^{1/4}|\mathbf{S}_2|^{1/4}}$$

όπου $\mathbf{S}_1 = \mathbf{L}_1^{-1} + \gamma\mathbf{I}$, $\mathbf{S}_2 = \mathbf{L}_2^{-1} + \gamma\mathbf{I}$ και $\mathbf{I}$ είναι ο $n \times n$ ταυτοτικός πίνακας. Ο Λαπλασιανός πυρήνας γράφων αποδίδει την ομοιότητα μεταξύ των συνολικών σχημάτων των δύο γράφων. Εντούτοις, θεωρεί ότι και οι δύο γράφοι έχουν το ίδιο μέγεθος και ότι αυτό δεν εξαρτάται από τις μεταθέσεις των κόμβων.

Για να εξασφαλίσει ανεξαρτησία από τις μεταθέσεις των κόμβων, ο πολυκλιμακωτός Λαπλασιανός πυρήνας γράφων αναπαριστά κάθε κόμβο σαν ένα $m$-διάστατο διάνυσμα του οποίου τα στοιχεία αντιστοιχούν σε *τοπικά* και *ανεξάρτητα από μεταθέσεις* χαρακτηριστικά των κόμβων. Τέτοια χαρακτηριστικά μπορούν για παράδειγμα να συμπεριλαμβάνουν το βαθμό ενός κόμβου ή το νούμερο των τριγώνων στα οποία συμμετέχει. Τότε, εκτελεί έναν γραμμικό μετασχηματισμό με τον οποίο αναπαριστά κάθε γράφο σαν μία κατανομή των χαρακτηριστικών που λάβαμε υπόψιν αντί για την κατανομή των κόμβων. Έστω $\mathbf{U}_1, \mathbf{U}_2 \in \mathbb{R}^{m \times n}$ οι μήτρες απεικόνισης των χαρακτηριστικών αυτών των γράφων, όπου οι μήτρες των οποίων οι στήλες περιέχουν τις διανυσματικές αναπαραστάσεις των κόμβων αυτών των δύο γράφων. Τότε, η αναπαράσταση χαρακτηριστικών του Λαπλασιανού πυρήνα γράφων ορίζεται ως

$$k_{FLG}(G_1, G_2) = \frac{|(\frac{1}{2}\mathbf{S}_1^{-1} + \frac{1}{2}\mathbf{S}_2^{-1})^{-1}|^{1/2}}{|\mathbf{S}_1|^{1/4}|\mathbf{S}_2|^{1/4}}$$

όπου $\mathbf{S}_1 = \mathbf{U}_1\mathbf{L}_1^{-1}\mathbf{U}_1^\top + \gamma\mathbf{I}$, $\mathbf{S}_2 = \mathbf{U}_2\mathbf{L}_2^{-1}\mathbf{U}_2^\top + \gamma\mathbf{I}$ και $\mathbf{I}$ είναι ο $m \times m$ ταυτοτικός πίνακας. Από την στιγμή που τα χαρακτηριστικά των κόμβων είναι τοπικά και ανεξάρτητα από την αναδιάταξη των κόμβων, ο χώρος χαρακτηριστικών του Λαπλασιανού πυρήνα γράφων είναι αναλλοίωτος στις μεταθέσεις. Ακόμα, από την στιγμή που οι κατανομές αυτές βρίσκονται σε ένα χώρο χαρακτηριστικών αντί για ένα χώρο διανυσμάτων, ο Λαπλασιανός πυρήνας γράφων για χώρους χαρακτηριστικών μπορεί να εφαρμοστεί σε γράφους διαφορετικών μεγεθών.

Έστω $\phi(v)$ η αναπαράσταση ενός κόμβου $v$ από τοπικά χαρακτηριστικά των κόμβων όπως περιγράφηκε παραπάνω. Ο πυρήνας βάσης $\kappa$ μεταξύ δύο κόμβων $v_1$ και $v_2$ αντιστοιχεί στο εσωτερικό γινόμενο των διανυσμάτων χαρακτηριστικών τους:

$$\kappa(v_1, v_2) = \phi(v_1)^\top \phi(v_2) \tag{2.41}$$

Έστω $G_1$ και $G_2$ δύο γράφοι με σύνολα κόμβων $V_1 = \{v_1, \ldots, v_{n_1}\}$ και $V_2 = \{u_1, \ldots, u_{n_2}\}$ αντίστοιχα και έστω $\bar{V} = \{\bar{v}_1, \ldots, \bar{v}_{n_1+n_2}\}$ η ένωση των δύο συνόλων κόμβων. Έστω ακόμα $\mathbf{K} \in \mathbb{R}^{(n_1+n_2)\times(n_1+n_2)}$ μία μήτρα πυρήνα που ορίζεται ως

$$\mathbf{K}_{ij} = \kappa(\bar{v}_i, \bar{v}_j) = \phi(\bar{v}_i)^\top \phi(\bar{v}_j) \tag{2.42}$$

Έστω $\mathbf{u}_1, \ldots, \mathbf{u}_p$ το μεγιστοτικό ορθοκανονικό σύνολο (maximal orthonormal set) των μη-μηδενικών διανυσμάτων ιδιοτιμών του $\mathbf{K}$ με τις αντίστοιχες ιδιοτιμές $\lambda_1, \ldots, \lambda_p$. Τότε τα διανύσματα:

$$\xi_i = \frac{1}{\sqrt{\lambda_i}} \sum_{l=1}^{n_1+n_2} [\mathbf{u}_i]_l \phi(\bar{v}_l) \tag{2.43}$$

όπου $[\mathbf{u}_i]_l$ είναι το $l^{th}$ στοιχείο του διανύσματος $\mathbf{u}_i$, σχηματίζουν μία ορθοκανονική βάση του υποχώρου $\{\phi(\bar{v}_1), \ldots, \phi(\bar{v}_{n_1+n_2})\}$. Ακόμα, έστω $\mathbf{Q} = [\lambda_1^{1/2}\mathbf{u}_1, \ldots, \lambda_p^{1/2}\mathbf{u}_p] \in \mathbb{R}^{p\times p}$ και $\mathbf{Q}_1, \mathbf{Q}_2$ η πρώτη $n_1$ και τελευταία $n_2$ γραμμή της μήτρας $\mathbf{Q}$ αντίστοιχα. Τότε, ο γενικευμένος χώρος χαρακτηριστικών του Λαπλασιανού πυρήνα γράφων που δημιουργείται από τον πυρήνα βάσης $\kappa$ ορίζεται ως:

$$k^{\kappa}_{FLG}(G_1, G_2) = \frac{|(\frac{1}{2}\mathbf{S}_1^{-1} + \frac{1}{2}\mathbf{S}_2^{-1})^{-1}|^{1/2}}{|\mathbf{S}_1|^{1/4}|\mathbf{S}_2|^{1/4}} \tag{2.44}$$

όπου $\mathbf{S}_1 = \mathbf{Q}_1\mathbf{L}_1^{-1}\mathbf{Q}_1^{\top} + \gamma\mathbf{I}$ και $\mathbf{S}_2 = \mathbf{Q}_2\mathbf{L}_2^{-1}\mathbf{Q}_2^{\top} + \gamma\mathbf{I}$ όπου $\mathbf{I}$ είναι ο $p \times p$ ταυτοτικός πίνακας. Ο πολυκλιμακωτός Λαπλασιανός πυρήνας γράφων χτίζει μία ιεραρχία εμφωλευμένων υπογράφων, όπου κάθε υπογράφος έχει κέντρο γύρω από έναν κόμβο και υπολογίζει τον γενικευμένο χώρο χαρακτηριστικών του Λαπλασιανού πυρήνα γράφων μεταξύ κάθε ζευγαριού υπογράφων. Έστω $G$ ο πυρήνας με σύνολο κόμβων $V$ και $\kappa$ ένας θετικά ημιορισμένος πίνακας στο $V$. Ας υποθέσουμε ότι για κάθε $v \in V$, έχουμε μία εμφωλευμένη ακολουθία από $L$ γειτονιές

$$v \in N_1(v) \subseteq N_2(v) \subseteq \ldots \subseteq N_L(v) \tag{2.45}$$

και για κάθε $N_l(v)$, έστω $G_l(v)$ ο αντίστοιχος επαγόμενος υπογράφος του $G$. Οι πολυκλιμακωτοί Λαπλασιανοί πυρήνες υπογράφων ορίζονται σαν $\mathfrak{K}_1, \ldots, \mathfrak{K}_L : V \times V \to \mathbb{R}$ ως εξής

1. $\mathfrak{K}_1$ είναι ο γενικευμένος χώρος χαρακτηριστικών του Λαπλασιανού πυρήνα χαρακτηριστικών $k^{\kappa}_{FLG}$ που επάγεται από τον πυρήνα βάσης $\kappa$ μεταξύ των υπογράφων του χαμηλότερου επιπέδου (δηλ. των κόμβων)

$$\mathfrak{K}_1(v, u) = k^{\kappa}_{FLG}(v, u) \tag{2.46}$$

2. Για $l = 2, 3, \ldots, L$, το $\mathfrak{K}_l$ είναι ο γενικευμένος χώρος χαρακτηριστικών του Λαπλασιανού πυρήνα γράφων ο οποίος επάγεται από το $\mathfrak{K}_{l-1}$ μεταξύ $G_l(v)$ και $G_l(u)$

$$\mathfrak{K}_l(v, u) = k^{\mathfrak{K}_{l-1}}_{FLG}(G_l(v), G_l(u)) \tag{2.47}$$

Τότε, ο πολυκλιμακωτός Λαπλασιανός πυρήνας γράφων μεταξύ δύο γράφων $G_1, G_2$ ορίζεται ως

$$k_{MLG}(G_1, G_2) = k^{\mathfrak{K}_L}_{FLG}(G_1, G_2)$$

Ο πολυκλιμακωτός Λαπλασιανός πυρήνας γράφων υπολογίζει το $\mathfrak{K}_1$ για όλα τα ζευγάρια κόμβων, έπειτα υπολογίζει $\mathfrak{K}_2$ για όλα τα ζευγάρια κόμβων κ.ο.κ. Συνεπώς, απαιτεί $\mathcal{O}(Ln^2)$ υπολογισμούς πυρήνων. Στο πιο υψηλό επίπεδο της ιεραρχίας κάθε υπογράφος που έχει κέντρο γύρω από έναν κόμβο $G_l(v)$ μπορεί να έχει το πολύ $n$ κόμβους. Συνεπώς, το κόστος ενός

απλού υπολογισμού του γενικευμένου Λαπλασιανού πυρήνα χώρου χαρακτηριστικών μπορεί να χρειαστεί $\mathcal{O}(n^3)$ χρόνο. Αυτό σημαίνει ότι στη χειρότερη περίπτωση, το κόστος υπολογισμού του $k_{MLG}$ είναι $\mathcal{O}(Ln^5)$. Δεδομένου ενός συνόλου $N$ γράφων, ο υπολογισμός της μήτρας πυρήνα απαιτεί την επανάληψη αυτής της διαδικασίας για όλα τα ζευγάρια γράφων, που συνολικά χρειάζεται $\mathcal{O}(LN^2n^5)$ χρόνο, πράγμα που αποτελεί έναν πολύ σημαντικό περιορισμό για την χρήση αυτού του πυρήνα σε πραγματικές εφαρμογές.

Η λύση σε αυτό το πρόβλημα είναι να υπολογίσουμε για κάθε επίπεδο $l = 1, 2, \ldots, L+1$ μία κοινή βάση για όλους τους υπογράφους όλων των γράφων ταυτόχρονα σε ένα δεδομένο επίπεδο. Έστω $G_1, G_2, \ldots, G_N$ μία συλλογή από γράφους με $V_1, V_2, \ldots, V_N$ τα σύνολα κόμβων τους και έστω ότι $V_1, V_2, \ldots, V_N \subseteq \mathcal{V}$ ενός γενικού χώρου κόμβων $\mathcal{V}$. Ο κοινός χώρος χαρακτηριστικών των κόμβων για όλη τη συλλογή γράφων είναι $W = span\{\bigcup_{i=1}^{N}\bigcup_{v\in V_i}\{\phi(v)\}\}$. Έστω $c = \sum_{i=1}^{N}|V_i|$ ο συνολικός αριθμός κόμβων και $\bar{V} = (\bar{v}_1, \ldots, \bar{v}_c)$ η αλληλουχία όλων των συνόλων κόμβων για όλους τους γράφους. Έστω $\mathbf{K}$ η αντίστοιχη από κοινού μήτρα πυρήνα και $\mathbf{u}_1, \ldots, \mathbf{u}_p$ το μεγιστοτικό ορθοκανονικό σύνολο όλων των ιδιοδιανυσμάτων με μη-μηδενικές ιδιοτιμές του $\mathbf{K}$ με τις αντίστοιχες ιδοτιμές $\lambda_1, \ldots, \lambda_p$ και $p = dim(W)$. Τότε τα διανύσματα

$$\xi_i = \frac{1}{\sqrt{\lambda_i}}\sum_{l=1}^{c}[\mathbf{u}_i]_l\phi(\bar{v}_l) \qquad i = 1, \ldots, p$$

αποτελούν μία ορθοκανονική βάση του $W$. Ακόμα, έστω ότι $\mathbf{Q} = [\lambda_1^{1/2}\mathbf{u}_1, \ldots, \lambda_p^{1/2}\mathbf{u}_p] \in \mathbb{R}^{p\times p}$ και το $\mathbf{Q}_1$ συμβολίζει τις πρώτες $n_1$ γραμμές του πίνακα $\mathbf{Q}$, το $\mathbf{Q}_2$ τις επόμενες $n_2$ γραμμές του πίνακα $\mathbf{Q}$ κ.ο.κ. Για κάθε ζευγάρι γράφων $G_i, G_j$ της συλλογής, ο γενικευμένος Λαπλασιανός πυρήνας χαρακτηριστικών γράφων που προκύπτει από το $\kappa$ μπορεί να εκφραστεί ως

$$k_{FLG}^{\kappa}(G_i, G_j) = \frac{|(\frac{1}{2}\bar{\mathbf{S}}_i^{-1} + \frac{1}{2}\bar{\mathbf{S}}_j^{-1})^{-1}|^{1/2}}{|\bar{\mathbf{S}}_i|^{1/4}|\bar{\mathbf{S}}_j|^{1/4}}$$

όπου $\bar{\mathbf{S}}_i = \mathbf{Q}_i\mathbf{L}_i^{-1}\mathbf{Q}_i^{\top} + \gamma\mathbf{I}$, $\bar{\mathbf{S}}_j = \mathbf{Q}_j\mathbf{L}_j^{-1}\mathbf{Q}_j^{\top} + \gamma\mathbf{I}$ και $\mathbf{I}$ είναι ο $p \times p$ ταυτοτικός πίνακας.

#### 2.6.8.1 Προσέγγιση Χαμηλής Τάξης

Ο υπολογισμός της μήτρας πυρήνα μεταξύ όλων των κόμβων όλων των γράφων ($c$ κόμβων στο σύνολο) και η αποθήκευση των τιμών τους είναι μία διαδικασία με μεγάλο κόστος. Από την άλλη η διάσπαση ιδιοδιανυσμάτων ιδιοτιμών είναι ακόμα χειρότερη όσον αφορά την υπολογιστική της πολυπλοκότητα, ενώ το $p$ είναι επίσης πολύ μεγάλο. Το πρόβλημα της διαχείρισης των μητρών $\bar{\mathbf{S}}_1, \ldots, \bar{\mathbf{S}}_N$ (καθένας εκ των οποίων έχει μέγεθος $p \times p$) γίνεται υπολογιστικά ανέφικτο. Ως επακόλουθο, ο πολυκλιμακωτός Λαπλασιανός πυρήνας γράφων αντικαθιστά το $W$ με ένα μικρότερο, προσεγγιστικό κοινό χώρο χαρακτηριστικών. Έστω $\tilde{V} = (\tilde{v}_1, \ldots, \tilde{v}_{\tilde{c}})$ βε $\tilde{c} \ll c$ κόμβοι που δειγματοληπτούνται από το κοινό σύνολο κόμβων. Τότε, ο προκύπτων υποδειγματοληπτημένος χώρος χαρακτηριστικών των κόμβων είναι $\tilde{W} = span\{\phi(v) : v \in \tilde{V}\}$. Έστω $\tilde{p} = dim(\tilde{W})$. Παρόμοια με προηγουμένως, ο πυρήνας κατασκευάζει μία ορθοκανονική βάση $\{\xi_1, \ldots, \xi_{\tilde{p}}\}$ για το $\tilde{W}$ σχηματίζοντας τώρα (την πολύ μικρότερη) μήτρα πυρήνα $\mathbf{K}_{ij} = \kappa(\tilde{v}_i, \tilde{v}_j)$, υπολογίζοντας τις ιδιοτιμές και τα ιδιοδιανύσματα

και θέτοντας $\xi_i = \frac{1}{\sqrt{\lambda_i}} \sum_{l=1}^{\tilde{c}} [\mathbf{u}_i]_l \phi(\tilde{v}_l)$. Ο προκύπτων προσεγγιστικός Λαπλασιανός πυρήνας γενικευμένου χώρου χαρακτηριστικών είναι

$$k^{\kappa}_{FLG}(G_1, G_2) = \frac{|(\frac{1}{2}\tilde{\mathbf{S}}_1^{-1} + \frac{1}{2}\tilde{\mathbf{S}}_2^{-1})^{-1}|^{1/2}}{|\tilde{\mathbf{S}}_1|^{1/4}|\tilde{\mathbf{S}}_2|^{1/4}}$$

όπου $\tilde{\mathbf{S}}_1 = \tilde{\mathbf{Q}}_1 \mathbf{L}_1^{-1} \tilde{\mathbf{Q}}_1^{\top} + \gamma \mathbf{I}$, $\tilde{\mathbf{S}}_2 = \tilde{\mathbf{Q}}_2 \mathbf{L}_2^{-1} \tilde{\mathbf{Q}}_2^{\top} + \gamma \mathbf{I}$ είναι οι προβολές του $\bar{\mathbf{S}}_1$ και $\bar{\mathbf{S}}_2$ στο $\tilde{W}$ και $\mathbf{I}$ είναι ο $\tilde{p} \times \tilde{p}$ ταυτοτικός πίνακας. Τέλος, ο πυρήνας εισάγει άλλο ένα βαθμό προσέγγισης περιορίζοντας το $\tilde{W}$ να είναι ο χώρος που προκύπτει από τα πρώτα $\hat{p} < \tilde{p}$ διανύσματα βάσης (ταξινομημένα κατά φθίνουσα ιδιοτιμή), εφαρμόζοντας αποτελεσματικά την τεχνική του PCA πυρήνα (kernel PCA) στο $\{\phi(\tilde{v})\}_{\tilde{v} \in \tilde{V}}$. Ο συνδυασμός αυτών των παραγόντων κάνει τον υπολογισμό της πλήρους ακολουθίας πυρήνων υπολογιστικά εφικτή, μειώνοντας την πολυπλοκότητα του υπολογισμού της μήτρας πυρήνα για μία συλλογή $N$ γράφων σε $\mathcal{O}(NL\tilde{c}^2\hat{p}^3 + NL\tilde{c}^3 + N^2\hat{p}^3)$.

### 2.6.9 Σκελετός Πυρήνα Core

Ο σκελετός πυρήνα core, είναι ένα τέχνασμα για την αύξηση της εκφραστικής δυνατότητας υπάρχοντων πυρήνων μεταξύ γράφων [61]. Ο σκελετός αυτός δεν περιορίζεται σε πυρήνες μεταξύ γράφων, αλλά μπορεί να εφαρμοσθεί σε κάθε αλγόριθμο σύγκρισης γράφων. Στηρίζεται στην $k$-core αποσύνθεση, η οποία είναι ικανή να αποκαλύπτει τοπολογικά και ιεραρχικά χαρακτησηστικά εσωτερικά κάθε γράφου. Συγκεκριμένα, η $k$-core αποσύνθεση είναι ένα ισχυρό εργαλείο για ανάλυση δικτύων και χρησιμοποιείται ευρέως για την σαν ένα μέτρο της σημασίας και ως μέτρο καλής *συνεκτικότητας* (connectedness) για κόμβους σε ένα μεγάλο εύρος εφαρμογών. Η έννοια του $k$-core πρωτοεισήχθει από τον Seidman για να μελετήσει την συνοχή (cohesion) των κοινωνικών δικτύων [71]. Τα τελευταία χρόνια, η αποσύνθεση $k$-core έχει καθιερωθεί σαν ένα βασικό εργαλείο σε πολλές εφαρμογές, όπως η αναπαράσταση δικτύων [3], στην πρόβλεψη πρωτεϊνικής λειτουργίας [90] και στη συσταδοποίηση γράφων [28].

#### 2.6.9.1 Core Αποσύνθεση

Έστω $G = (V, E)$ ένας μη κατευθυνόμενος γράφος, χωρίς βάρη. Έστω ότι τα $n$ και $m$ συμβολίζουν τον αριθμό των κόμβων και τον αριθμό των ακμών, αντίστοιχα. Δεδομένου ενός υποσυνόλου κόμβων $S \subseteq V$, έστω $E(S)$ το σύνολο ακμών του επαγόμενου γράφου $G' = (S, E(S))$ (βλέπε ορισμό 2.8) των κόμβων $S$. Έστω $G$ ένας γράφος και $G'$ ο υπογράφος του $G$ (βλέπε ορισμό 2.7) που επάγεται από ένα σύνολο κόμβων. Τότε, το $G'$ ορίζεται σαν η $k$-core αποσύνθεση του $G$ (για την οποία θα χρησιμοποιούμε καταχρηστικά τον όρο $k$-*κόρα*), που συμβολίζεται με $C_k$, αν είναι ένα μεγιστοτικό υπογράφημα του $G$ στο οποίο όλοι οι κόμβοι έχουν βαθμό τουλάχιστον $k$. Συνεπώς, αν ο $G'$ είναι η $k$-κόρα του $G$, τότε $\forall v \in S$, $d_{G'}(v) \geq k$. Κάθε $k$-κόρα είναι ένας μοναδικός υπογράφος του $G$, ο οποίος δεν είναι κατ' ανάγκη συνδεδεμένος. Ο αριθμός κόρας $c(v)$ ενός κόμβου $v$ ισούται με την μεγαλύτερη τάξη κόρας στην οποία ανήκει ο $v$. Με άλλα λόγια, ο $v$ έχει αριθμό κόρας $c(v) = k$, αν ανήκει στην $k$-κόρα αλλά όχι στην $(k+1)$-κόρα. Ο *εκφυλισμός* (degeneracy) $\delta^*(G)$ ενός γράφου $G$ ορίζεται σαν το μέγιστο $k$ για το οποίο ο γράφος $G$ περιέχει έναν μη-κενό $k$-core υπογράφο,

$\delta^*(G) = \max_{v \in V} c(v)$. Ακόμα, υποθέτοντας ότι $\mathcal{C} = \{C_0, C_1, \ldots, C_{\delta^*(G)}\}$ είναι το σύνολο όλων των $k$-cores, τότε τα $\mathcal{C}$ διαμορφώνουν μία εμφωλευμένη αλυσίδα

$$C_{\delta^*(G)} \subseteq \ldots \subseteq C_1 \subseteq C_0 = G \tag{2.48}$$

Συνεπώς, η $k$-core αποσύνθεση είναι ένα πολύ χρήσιμο εργαλείο για την ανακάλυψη ιεραρχικών δομών σε γράφους. Η $k$-core αποσύνθεση ενός γράφου μπορεί να υπολογιστεί σε χρόνο $\mathcal{O}(n+m)$ [57, 7]. Η βασική ιδέα είναι ότι μπορούμε να πάρουμε την $i$-κόρα ενός γράφου αν αναδρομικά αφαιρέσουμε όλους τους κόμβους με βαθμό μικρότερο του $i$ και τις συνδεόμενες ακμές του γράφου, μέχρι το σημείο που κανένας άλλος κόμβος να μην μπορεί να αφαιρεθεί.

#### 2.6.9.2 Core Πυρήνες

Η $k$-core αποσύνθεση δημιουργεί μία ιεραρχία εμφωλευμένων υπογράφων, όπου ο καθένας έχει ισχυρότερες ιδιότητες συνεκτικότητας σε σχέση με τους προηγούμενους. Ο core σκελετός πυρήνας υπολογίζει την ομοιότητα μεταξύ των αντίστοιχων υπογράφων σύμφωνα με την core ιεραρχία, συνοψίζοντας τα αποτελέσματα. Έστω $G = (V, E)$ και $G' = (V', E')$ δύο γράφοι. Έστω ακόμα $k$ ένας οποιοσδήποτε πυρήνας γράφων. Τότε, ο σκελετός πυρήνα core με πυρήνα βάσης $k$ ορίζεται ως

$$k_c(G, G') = k(C_0, C_0') + k(C_1, C_1') + \ldots + k(C_{\delta^*_{min}}, C'_{\delta^*_{min}}) \tag{2.49}$$

όπου $\delta^*_{min}$ είναι ο ελάχιστος βαθμός εκφυλισμού των δύο γράφων, και $C_0, C_1, \ldots, C_{\delta^*_{min}}$ και $C_0', C_1', \ldots, C'_{\delta^*_{min}}$ είναι οι υπογράφοι 0ης-κόρας, 1ης-κόρας,..., $\delta^*_{min}$οστής-κόρας των $G$ και $G'$, αντίστοιχα. Αποσυνθέτοντας τους γράφους σε υπογράφους αυξανόμενης βαρύτητας ο αλγόριθμος είναι ικανός να αποδώσει με μεγαλύτερη ακρίβεια *υφέρπουσα* ομοιότητα στην δομή δύο γράφων.

Η υπολογιστική πολυπλοκότητα του σκελετού πυρήνα core εξαρτάται από την πολυπλοκότητα του πυρήνα βάσης και τον εκφυλισμό των υπό σύγκριση γράφων. Δεδομένου ενός ζευγαριού γράφων $G, G'$ και ενός πυρήνα βάση $k$ για την σύγκριση των δύο γράφων, έστω $\mathcal{O}_k$ η χρονική πολυπλοκότητα του αλγορίθμου $k$. Έστω ακόμα $\delta^*_{min} = \min\left(\delta^*(G), \delta^*(G')\right)$ οι ελάχιστοι βαθμοί εκφυλισμού των δύο γράφων. Τότε, η πολυπλοκότητα υπολογισμού του σκελετού πυρήνα core με πυρήνα βάσης $k$ είναι $\mathcal{O}_c = \delta^*_{min}\mathcal{O}_A$. Είναι ευρέως γνωστό ότι ο βαθμός εκφυλισμού ενός γράφου έχει άνω φράγμα τον μέγιστο βαθμό των κόμβων του και την μέγιστη ιδιοτιμή του πίνακα γειτνίασης $\lambda_1$. Από την στιγμή που για τους περισσότερους *πραγματικούς* γράφους ισχύει ότι $\lambda_1 \ll n$ και $\delta^*_{max} \ll n$, η επαύξηση στην χρονική πολυπλοκότητα του πυρήνα βάσης δεν είναι σημαντική.

### 2.6.10 Πυρήνας Αποστάσεων Ζευγαριών Γειτονικών Υπογράφων

Ο πυρήνας αποστάσεων ζευγαριών γειτονικών υπογράφων (neighborhood subgraph pairwise distance kernel) εξάγει ζευγάρια *ριζωμένων* υπογράφων από κάθε γράφο που οι ρίζες τους βρίσκονται σε συγκεκριμένη απόσταση και οι οποίοι περιέχουν κόμβους έως και ένα συγκεκριμένο *ύψος* από την ρίζα [18]. Έπειτα συγκρίνει τους γράφους βάσει αυτού του συνόλου ζευγαριών ριζωμένων υπογράφων. Για να αποφευχθεί ο έλεγχος ισομορφισμού χρησιμοποιούνται

σταθερά στοιχεία των γράφων προκειμένου να παράγουν μία αντιπροσωπευτική κωδικοποίηση για καθένα από τους ριζωμένους υπογράφους.

Έστω $G = (V, E)$ ένας γράφος. Η απόσταση μεταξύ δύο κόμβων $u, v \in V$, που συμβολίζεται ως $D(u, v)$, είναι το μήκος του ελαχίστου μονοπατιού μεταξύ τους. Η γειτονιά ακτίνας $r$ κάθε κόμβου $v$ είναι το σύνολο των κόμβων σε απόσταση μικρότερη ίση του $r$ από το $v$, δηλαδή $\{u \in V : D(u, v) \leq r\}$. Δεδομένου ενός υποσυνόλου κόμβων $S \subseteq V$, έστω $E(S)$ οι ακμές επαγόμενου υπογράφου του $S$. Ο υπογράφος γειτονιάς ακτίνας $r$ ενός κόμβου $v$ είναι ο υπογράφος που επάγεται από μία γειτονιά ακτίνας $r$ του $v$ και συμβολίζεται με $N_r^v$. Έστω ακόμα $R_{r,d}(A_v, B_u, G)$, μία σχέση μεταξύ δύο ριζωμένων $A_u, B_u$ και ενός γράφου $G = (V, E)$ που είναι αληθής αν και μόνον αν, τα $A_v$, $B_u$ βρίσκονται στο $\{N_r^v : v \in V\}$, όπου απαιτούμε τα $A_v, B_u$ να είναι ισομορφικά με μία $N_r^v$ προκειμένου να επαληθεύσουμε ότι ανήκουν στο σύνολο, καθώς και ότι $D(u, v) = d$. Θα συμβολίσουμε με $R^{-1}(G)$ το ανάστροφο σχεσιακό κατηγόρημα που αποδίδει όλα τα ζευγάρια ριζωμένων γράφων $A_v$, $B_u$ που ικανοποιούν την παραπάνω συνθήκη. Συνεπώς το $R^{-1}(G)$ επιστρέφει όλα τα ζευγάρια από γειτονικούς γράφους ακτίνας $r$ που οι ρίζες τους είναι σε απόσταση $d$ σε ένα δεδομένο γράφο $G$. Ο πυρήνας αποστάσεων ζευγαριών γειτονικών υπογράφων χρησιμοποιεί τον παρακάτω πυρήνα:

$$k_{r,d}(G, G') = \sum_{A_v, B_v \in R_{r,d}^{-1}(G)} \sum_{A'_{v'}, B'_{v'} \in R_{r,d}^{-1}(G')} \delta(A_v, A'_{v'})\delta(B_v, B'_{v'}) \tag{2.50}$$

όπου η συνάρτηση $\delta$ είναι 1 αν οι υπογράφοι της εισόδου είναι ισομορφικοί (βλέπε ορισμό 2.13) και 0 αλλιώς. Ο παραπάνω πυρήνας μετράει τον αριθμό των ζευγαριών γειτονιάς ακτίνας $r$ και απόστασης $d$ που ταυτίζονται μεταξύ των δύο γράφων. Τότε, ο πυρήνας αποστάσεων ζευγαριών υπογράφων γειτονιάς ορίζεται ως:

$$k(G, G') = \sum_{r=0}^{r^*} \sum_{d=0}^{d^*} \hat{k}_{r,d}(G, G') \tag{2.51}$$

όπου $\hat{k}_{r,d}$ είναι μία κανονικοποιημένη εκδοχή του $k_{r,d}$, δηλαδή

$$\hat{k}_{r,d}(G, G') = \frac{k_{r,d}(G, G')}{\sqrt{k_{r,d}(G, G)k_{r,d}(G', G')}}$$

Η παραπάνω εκδοχή εξασφαλίζει ότι σε όλες τις σχέσεις όλων των τάξεων δίνεται το ίδιο βάρος, αδιάφορα από το μέγεθος των επαγόμενων συνόλων.

Ο πυρήνας αποστάσεων ζευγαριών γειτονικών υπογράφων περιλαμβάνει έναν πυρήνα ακριβούς ταιριάσματος μεταξύ δύο γράφων (δηλ. τον $\delta$ πυρήνα) που είναι ισοδύναμος με την επίλυση του προβλήματος ισομορφισμού. Η λύση του προβλήματος ισομορφισμού γράφων δεν είναι υπολογιστικά αποδοτική. Συνεπώς, ο πυρήνας καλείται να υποκατασταθεί με μία προσέγγιση. Δεδομένου ενός υπογράφου $G_S$ που προκύπτει από ένα σύνολο κόμβων $S$, ο πυρήνας υπολογίζει μία αναλλοίωτη κωδικοποίηση του υπογράφου μέσω μία συνάρτησης επισημειώσεων $\mathcal{L}^g : \mathcal{G} \to \Sigma^*$, όπου $\mathcal{G}$ είναι το σύνολο ριζωμένων γράφων και $\Sigma^*$ είναι το σύνολο συμβολοσειρών σε ένα πεπερασμένο αλφάβητο $\Sigma$. Η συνάρτηση $\mathcal{L}^g$ χρησιμοποιεί δύο άλλες συναρτήσεις επισημειώσεων: (1) μία συνάρτηση $\mathcal{L}^n$ για κόμβους και (2) μία συνάρτηση $\mathcal{L}^e$ για ακμές. Η

$\mathcal{L}^n$ είναι για κάθε κόμβο $v$ ίση με την αλληλουχία της λεξικογραφικά ταξινομημένης λίστας από τριάδες αποστάσης-απόστασης και επισημείωσης ρίζας $\langle D(v,u), D(v,h), \mathcal{L}(u)\rangle$, για όλα τα $u \in S$, όπου $h$ είναι η ρίζα του υπογράφου και $\mathcal{L}$ είναι μία συνάρτηση που απεικονίζει κόμβους και ακμές στην επισημείωση τους. Συνεπώς, η παραπάνω συνάρτηση επανεπισημειώνει κάθε κόμβο με μία συμβολοσειρά που κωδικοποιεί την αρχική επισημείωση του κόμβου, την απόσταση του από όλους τους άλλους επισημειωμένους κόμβους και την απόσταση του από τον κόμβο ρίζα. Η συνάρτηση $\mathcal{L}^e(u,v)$ είναι για κάθε ακμή $(u,v)$ ίση με την επισημείωση $\langle \mathcal{L}^n(u), \mathcal{L}^n(v), \mathcal{L}((u,v))\rangle$. Συνεπώς η συνάρτηση $\mathcal{L}^e(u,v)$ επισημειώνει κάθε ακμή βάσει των νέων επισημειώσεων των τερματικών της κόμβων και την αρχικής της επισημείωση (αν υπάρχει). Τέλος, η συνάρτηση $\mathcal{L}^g(G_S)$ επισημειώνει κάθε ριζωμένο γράφο που επάγεται από το $S$ με την αλληλουχία της λεξικογραφικά ταξινομημένης λίστας του $\mathcal{L}^e(u,v)$ για όλα τα $(u,v) \in E(S)$. Ο πυρήνας χρησιμοποιεί έπειτα μία συνάρτηση κατακερματισμού που δέχεται συμβολοσειρές και επιστρέφει φυσικούς αριθμούς $H : \Sigma^* \to \mathbb{N}$ προκειμένου να εξάγει ένα μοναδικό αναγνωριστικό για κάθε υπογράφο. Συνεπώς, αντί να ελέγχει για όλα τα ζευγάρια υπογράφων αν είναι ισομορφικά (βλέπε ορισμό 2.13), ο πυρήνας ελέγχει απλώς αν το αναγνωριστικό όλων των ζευγαριών ταυτίζεται.

Η υπολογιστική πολυπλοκότητα του πυρήνα neighborhood subgraph pairwise distance kernel είναι $\mathcal{O}(|V||S||E(S)|\log|E(S)|)$ και κυριαρχείται από τις επαναλαμβανόμενες επαναλήψεις υπολογισμού της αναλλοίωτης κάθε γράφου, για κάθε κόμβο του. Συνεπώς για μικρές τιμές των $d^*$ και $r^*$ ο υπολογισμός της αναλλοίωτης είναι μία διαδικασία σταθερού χρόνου και συνεπώς η συνολική πολυπλοκότητα του αλγορίθμου είναι στην πράξη γραμμική, ως προς το μέγεθος του γράφου.

### 2.6.11 Πυρήνας Κατακερματισμού Γειτονιών

Ο πυρήνας κατακερματισμού γειτονιών (neighborhood hash kernel) δέχεται ως είσοδο γράφους με επισημειώσεις [39]. Ανανεώνοντας τις επισημειώσεις των κόμβων τους και μετρώντας τον κοινό αριθμό τους, καταφέρνει να εξάγει ένα πολύ απλό και αποτελεσματικό μέτρο ομοιότητας μεταξύ τους. Ο πυρήνας αντικαθιστά τις διακριτές επισημειώσεις των κόμβων με δυαδικούς πίνακες δεδομένου μήκους και έπειτα χρησιμοποιεί λογικές πράξεις με τις οποίες ανανεώνει τις τιμές τους, προκειμένου να περιέχουν πληροφορία που αφορά την δομή που βρίσκεται στη γειτονιά κάθε κόμβου.

Έστω $\ell : \mathcal{V} \to \Sigma$ μία συνάρτηση που απεικονίζει τους κόμβους του γράφου σε ένα αλφάβητο $\Sigma$, το οποίο αποτελεί το σύνολο των δυνατών διακριτών επισημειώσεων. Συνεπώς, δεδομένου ενός κόμβου $v$, $\ell(v) \in \Sigma$ είναι η επισημείωση του κόμβου $v$. Ο αλγόριθμος πρώτα μετασχηματίζει κάθε διακριτό κόμβο σε μία δυαδική επισημείωση. Μία διακριτή επισημείωση είναι ένας δυαδικός πίνακας που αποτελείται από $d$ bits ως εξής

$$s = \{b_1, b_2, \ldots, b_d\}$$

όπου η σταθερά $d$ ικανοποιεί την συνθήκη $2^d - 1 \gg |\Sigma|$ και $b_1, b_2, \ldots, b_d \in \{0,1\}$.

Το πιο σημαντικό βήμα του αλγορίθμου αφορά μία διαδικασία που ανανεώνει τις επισημειώσεις των κόμβων. Για να επιτύχει κάτι τέτοιο ο πυρήνας, ο πυρήνας κάνει χρήση δύο πολύ γνωστών

δυαδικών τελεστών: (1) το *αποκλειστικό ή* ($XOR$) και (2) την *δυαδική περιστροφή* ($ROT$). Έστω ότι με $XOR(s_i, s_j) = s_i \oplus s_j$ συμβολίζουμε την πράξη $XOR$ μεταξύ δύο επισημειώσεων με bit $s_i$ και $s_j$ (δηλ. η πράξη $XOR$ εφαρμόζεται σε όλα τα μέλη του). Η έξοδος αυτής της πράξης είναι ένας δυαδικός πίνακας του οποίου τα μέλη αναπαριστούν μία τιμή $XOR$ μεταξύ των αντίστοιχων στοιχείων στους πίνακες $s_i$ και $s_j$. Η πράξη $ROT_o$ παίρνει ως είσοδο ένα δυαδικό πίνακα και μετατοπίζει τα τελευταία $o$ bits στα αριστερά κατά $o$ bits και μεταφέρει τα πρώτα κατά $o$ στο δεξιά όπως φαίνεται παρακάτω

$$ROT_o(s) = \{b_{o+1}, b_{o+2}, \ldots, b_d, b_1, \ldots, b_o\} \tag{2.52}$$

Στη συνέχεια, θα παρουσιάσουμε λεπτομερώς δύο διαδικασίες για την ανανέωση των επισημειώσεων των κόμβων: (1) τον απλό κατακερματισμό γειτονιών και (2) τον ευαίσθητο στο μέτρημα κατακερματισμό γειτονιών.

#### 2.6.11.1 Απλός Κατακερματισμός Γειτονιών

Δεδομένου ενός γράφου $G = (V, E)$ με δυαδικές επισημειώσεις, η διαδικασία ανανέωσης των επισημειώσεων του απλού κατακερματισμού γειτονιών κατακερματίζει για κάθε κόμβο την γειτονιά του χρησιμοποιώντας τις λογικές πράξεις $XOR$ και $ROT$. Συγκεκριμένα, δεδομένου ενός κόμβου $v \in V$, έστω $\mathcal{N}(v) = \{u_1, \ldots, u_d\}$ το σύνολο όλων των γειτόνων του $v$. Τότε, ο πυρήνας υπολογίζει τον απλό κατακερματισμό γειτονιάς ως:

$$NH(v) = ROT_1\big(\ell(v)\big) \oplus \big(\ell(u_1) \oplus \ldots \oplus \ell(u_d)\big) \tag{2.53}$$

Ο προκύπτων κατακερματισμός $NH(v)$ είναι και πάλι ένας δυαδικός πίνακας μήκους $d$ ο οποίος χρησιμοποιείται ως η νέα επισημείωση του $v$. Αυτή η νέα επισημείωση αντιπροσωπεύει την κατανομή από κόμβους γύρω από τον $v$. Συνεπώς, αν $v_i$ και $v_j$ είναι δύο κόμβοι με την ίδια επισημείωση (δηλ. $\ell(v_i) = \ell(v_j)$) και τα σύνολα επισημειώσεων των γειτόνων τους ταυτίζονται, οι τιμές κατακερματισμού τους θα είναι ίδιες (δηλ. $NH(v_i) = NH(v_j)$). Διαφορετικά, θα διαφέρουν εκτός από την περίπτωση των συμπτωματικών συγκρούσεων των κατακερματισμένων τιμών (accidental hash collisions). Η κύρια ιδέα πίσω από την διαδικασία ανανέωσης των επισημειώσεων, είναι ότι η τιμή του κατακερματισμού είναι ανεξάρτητη της σειράς εμφάνισης των κόμβων σε μία γειτονιά λόγω της αντιμεταθετικής ιδιότητας της πράξης $XOR$. Συνεπώς, κάποιος μπορεί να ελέγξει αν οι κατανομές των κόμβων των γειτόνων ταυτίζονται χωρίς να χρειάζεται να ταξινομήσει ή να ταιριάξει δύο σύνολα επισημειώσεων (παράβαλε την διαδικασία ανανέωσης επισημειώσεων του σκελετού πυρήνα Weisfeiler Lehman ενότητα 2.6.4).

#### 2.6.11.2 Ευαίσθητος στο Μέτρημα Κατακερματισμός Γειτονιών

Ο απλός κατακερματισμός γειτονιών όπως περιγράφηκε παραπάνω υποφέρει από ένα πλήθος συμπτωματικών συγκρούσεων των κατακερματισμένων τιμών. Συγκεκριμένα, υπάρχει πιθανότητα οι τιμές κατακερματισμού των γειτονιών για δύο ανεξάρτητους κόμβους να συκρούονται, ακόμα και στην περίπτωση που δεν υπάρχουν κατακερματισμοί που συγκρούονται κατά σύμπτωση. Από την άλλη και μόνο το γεγονός ύπαρξης τέτοιων ταυτίσεων, μπορεί να επηρεάσει το γεγονός ότι η μήτρα πυρήνα είναι θετικά ημιορισμένη. Ως λύση σε αυτό το πρόβλημα, η

διαδικασία ανανέωσης επισημειώσεων του ευαίσθητου στο μέτρημα κατακερματισμού γειτονιάς μετράει το πλήθος ύπαρξης μίας επισημείωσης στο σύνολο επισημειώσεων. Συγκεκριμένα, χρησιμοποιεί πρώτα ένα αλγόριθμο ταξινόμησης (συγκεκριμένα radix sort με πολυπλοκότητα $\mathcal{O}(dn)$ γραμμική ως προς $n$, δεδομένου ενός σταθερού $d$) για να ευθυγραμμίσει τις δυαδικές επισημειώσεις των κόμβων των γειτόνων και έπειτα εξάγει μοναδικές επισημειώσεις (σύνολα $\{\ell_1, \ldots, \ell_l\}$ στην περίπτωση $l$ στο πλήθος μοναδικών επισημειώσεων) και για κάθε επισημείωση μετράει τον αριθμό των εμφανίσεων. Έπειτα, ανανεώνει κάθε μοναδική επισημείωση ενός κόμβου βάση του αριθμού εμφανίσεων ως εξής

$$\ell'_i = ROT_o\big(\ell_i \oplus o\big) \tag{2.54}$$

όπου $\ell_i, \ell'_i$ είναι αντίστοιχα η αρχική και η ανανεωμένη επισημείωση και $o$ είναι ο αριθμός των εμφανίσεων αυτής της επισημείωσης στο σύνολο των γειτόνων. Η παραπάνω διαδικασία κάνει τις τιμές των κατακερματισμών μοναδικές, όντας εξαρτημένη του αριθμού επαναλήψεων κάθε επισημείωσης. Τελικά, ο ευαίσθητος στο μέτρημα κατακερματισμός γειτονιών υπολογίζεται ως:

$$CSNH(v) = ROT_1\big(\ell(v)\big) \oplus \big(\ell'_1 \oplus \ldots \oplus \ell'_l\big) \tag{2.55}$$

Συνολικά, τόσο ο απλός και ο ευαίσθητος στο μέτρημα καταμερισμός γειτονιών μπορούν να ειδωθούν ως γενικές προσεγγίσεις για τον εμπλουτισμό των επισημειώσεων των κόμβων βάσει της κατανομής επισημειώσεων των γειτονικών κόμβων.

#### 2.6.11.3 Υπολογισμός Πυρήνα

Οι διαδικασίες ανανέωσης επισημειώσεων κατακερματισμού γειτονιάς που παρουσιάζονται παραπάνω συνοψίζουν την πληροφορία γειτονιάς των γειτόνων για κάθε κόμβο. Εν συνεχεία, δεδομένου δύο γράφων $G$ και $G'$, οι ανανεωμένες επισημειώσεις των κόμβων τους συγκρίνονται βάσει της συνάρτησης:

$$\kappa(G, G') = \frac{c}{|V| + |V'| - c} \tag{2.56}$$

όπου $c$ είναι ο αριθμός των επισημειώσεων που μοιράζονται οι δύο γράφοι. Η συνάρτηση αυτή είναι ισοδύναμη με τον συντελεστή *Tanimoto* που χρησιμοποιείται σαν μέτρο ομοιότητας μεταξύ συνόλων με διακριτές τιμές και για το οποίο έχει αποδειχθεί ότι είναι θετικά ημιορισμένος [32].

Οι διαδικασίες ανανέωσης επισημειώσεων, δεν εφαρμόζονται κατ' ανάγκη μία και μοναδική φορά, αλλά μπορούν να εφαρμοστούν επαναληπτικά. Ανανεώνοντας τις δυαδικές επισημειώσεις αρκετές φορές, οι επισημειώσεις μπορούν να αιχμαλωτίσουν σχέσεις υψηλότερου βαθμού μεταξύ των κόμβων. Για παράδειγμα, αν η διαδικασία ανανέωσης εφαρμοστεί συνολικά $h$ φορές, η ανανεωμένη επισημείωση $\ell(v)$ του κόμβου $v$ αναπαριστά την κατανομή επισημειώσεων των $h$-γειτόνων του. Συνεπώς, δύο κόμβοι $v_i, v_j$ με ίδιες επισημειώσεις και συνδέσεις μεταξύ των $r$-γειτόνων τους θα έχουν την ίδια επισημείωση. Δεδομένου ενός γράφου $G = (V, E)$, έστω ότι οι $G_1, \ldots, G_h$ συμβολίζουν τους ανανεωμένους γράφους, όπου οι επισημειώσεις των κόμβων έχουν ανανεωθεί $1, \ldots, h$ φορές αντίστοιχα. Τότε, δεδομένου δύο γράφων $G$ και $G'$, ο

πυρήνας κατακερματισμού γειτονιών ορίζεται ως:

$$k(G, G') = \frac{1}{h} \sum_{i=1}^{h} \kappa(G_i, G'_i) \tag{2.57}$$

Η υπολογιστική πολυπλοκότητα του πυρήνα κατακερματισμού γειτονιών είναι $\mathcal{O}(dhn\bar{D})$, όπου $n = |V|$ είναι ο αριθμός των κόμβων του γράφου και $\bar{D}$ είναι ο μέσος βαθμός των κόμβων.

### 2.6.12 Πυρήνας Ταιριάσματος Υπογράφων

Ο πυρήνας ταιριάσματος υπογράφων μετράει τον αριθμό των ταιριασμάτων υπογράφων φραγμένου μεγέθους μεταξύ δύο γράφων [49]. Ο πυρήνας είναι πολύ γενικός από τη στιγμή που μπορεί να εφαρμοστεί σε γράφους που περιέχουν τόσο διακριτές όσο και συνεχείς επισημειώσεις κόμβων και ακμών.

Έστω $\mathcal{G}$ ένα σύνολο γράφων. Θα υποθέσουμε ότι όλοι οι γράφοι που περιέχονται στο σύνολο έχουν διακριτές ή συνεχείς επισημειώσεις. Συγκεκριμένα, έστω $\ell : \mathcal{V} \cup \mathcal{E} \to \mathcal{L}$ μία συνάρτηση επισημείωσης που αποδίδει είτε διακριτές είτε συνεχείς επισημειώσεις στους κόμβους και τις ακμές του γράφου. Δεδομένου δύο γράφων $G = (V, E)$ και $G' = (V', E')$, έστω $\mathcal{B}(G, G')$ συμβολίζει το σύνολο όλων των ¨1-1' απεικονίσεων μεταξύ συνόλων $S \subseteq V$ και $S' \subseteq V'$, και έστω $\lambda : \mathcal{B}(G, G') \to \mathbb{R}^+$ μία συνάρτηση βάρους. Τότε, ο πυρήνας ταιριάσματος υπογράφων ορίζεται ως:

$$k(G, G') = \sum_{\phi \in \mathcal{B}(G,G')} \lambda(\phi) \prod_{v \in S} \kappa_V(v, \phi(v)) \prod_{e \in S \times S} \kappa_E(e, \psi(e)) \tag{2.58}$$

όπου $S = dom(\phi)$ και $\kappa_V, \kappa_E$ είναι συναρτήσεις πυρήνα που ορίζονται σε κόμβους και ακμές, αντίστοιχα.

Ένα στιγμιότυπο του πυρήνα ταιριάσματος υπογράφων προκύπτει άμα θέσουμε τις συναρτήσεις $\kappa_V, \kappa_E$ ως εξής:

$$\kappa_V(v, v') = \begin{cases} 1, & \text{αν } \ell(v) \equiv \ell(v'), \\ 0, & \text{αλλιώς} \end{cases}$$

$$kai \tag{2.59}$$

$$\kappa_E(e, e') = \begin{cases} 1, & \text{αν } e \in E \wedge e' \in E' \wedge \ell(e) \equiv \ell(e') \text{ ή } e \notin E \wedge e' \notin E', \\ 0, & \text{αλλιώς} \end{cases}$$

που είναι γνωστός σαν ο *πυρήνας κοινών ισομορφικών υπογράφων*. Αυτός ο πυρήνας μετράει τον αριθμό των ισομορφικών υπογράφων που περιέχονται μεταξύ των δύο γράφων.

Για να μετρήσει τον αριθμό των ισομορφισμών (βλέπε ορισμό 2.13) μεταξύ υπογράφων, ο πυρήνας στηρίζεται πάνω στο αποτέλεσμα του Levi [51] που συνδέει υπογράφους που είναι κοινοί στο αρχικό υπογράφημα με κλίκες στο γινόμενο γράφημα. Πιο συγκεκριμένα, κάθε μέγιστη κλίκα στον γινόμενο γράφο συνδέεται με τον μέγιστο κοινό υπογράφο μεταξύ των παραγόντων (γράφων). Αυτό επιτρέπει σε κάποιον να υπολογίσει τον κοινό πυρήνα ισομορφισμού υπογράφων, απλώς μετρώντας, τις κλίκες στον γινόμενο γράφο.

Ο γενικός πυρήνας ταιριάσματος υπογράφων επεκτείνει την θεωρία του Levi και κατασκευάζει ένα σταθμισμένο γινόμενο γράφο προκειμένου να επιτρέψει μία πιο ευέλικτη βαθμονόμηση των ¨1-1' απεικονίσεων. Δεδομένου δύο γράφων $G = (V, E)$, $G' = (V', E')$, και πυρήνες κόμβων και ακμών $\kappa_V$ και $\kappa_E$, ο σταθμισμένος γινόμενος γράφος $G_P = (V_P, E_P)$ των $G$ και $G'$ ορίζεται ως:

$$\begin{aligned} V_P &= \{(v, v') \in V \times V' : \kappa_V(v, v') > 0\} \\ E_P &= \{((v, v'), (u, u')) \in V_P \times V_P : v \neq u \wedge v' \neq u' \wedge \kappa_E((v, v'), (u, u')) > 0\} \\ c(u) &= \kappa_V(v, v') \quad \forall u = (v, v') \in V_P \\ c(e) &= \kappa_E((v, u), (v', u')) \quad \forall e \in E_P, \\ \text{όπου } e &= ((v, v'), (u, u')) \end{aligned} \tag{2.60}$$

Αφού δημιουργήσει τον σταθμισμένο γινόμενο γράφο, ο πυρήνας απαριθμεί τις κλίκες. Συγκεκριμένα ξεκινάει με μία κενή κλίκα και την επεκτείνει βήμα-βήμα μέσω όλων των κόμβων, διατηρώντας την ιδιότητα της κλίκας. Έστω $w$ το βάρος μίας κλίκας $C$. Όποτε μία κλίκα $C$ επεκτείνεται σε ένα νέο κόμβο $v$, το βάρος της κλίκας ανανεώνεται ως εξής: πρώτα πολλαπλασιάζεται από το βάρος του κόμβου $w' = w \cdot c(v)$ και έπειτα πολλαπλασιάζεται με όλες τις ακμές που συνδέουν το $v$ με ένα κόμβο στο $C$, δηλαδή $w' = \sum_{u \in C} w \cdot c((v, u))$. Ο αλγόριθμος αποφεύγει αποτελεσματικά διπλότυπα αφαιρώντας ένα κόμβο από το σύνολο υποψηφίων, αφού όλες οι κλίκες που το περιέχουν έχουν εξερευνηθεί εξαντλητικά.
Η πολυπλοκότητα του πυρήνα ταιριάσματος υπογράφων εξαρτάται του πλήθους από κλίκες στο γινόμενο γράφο. Η πολυπλοκότητα χειρότερης περίπτωσης του πυρήνα όταν λαμβάνει υπόψιν υπογράφους το πολύ μεγέθους $k$ είναι $\mathcal{O}(kn^{k+1})$, όπου $n = |V| + |V'|$ είναι το άθροισμα του αριθμού των κόμβων των δύο γράφων.

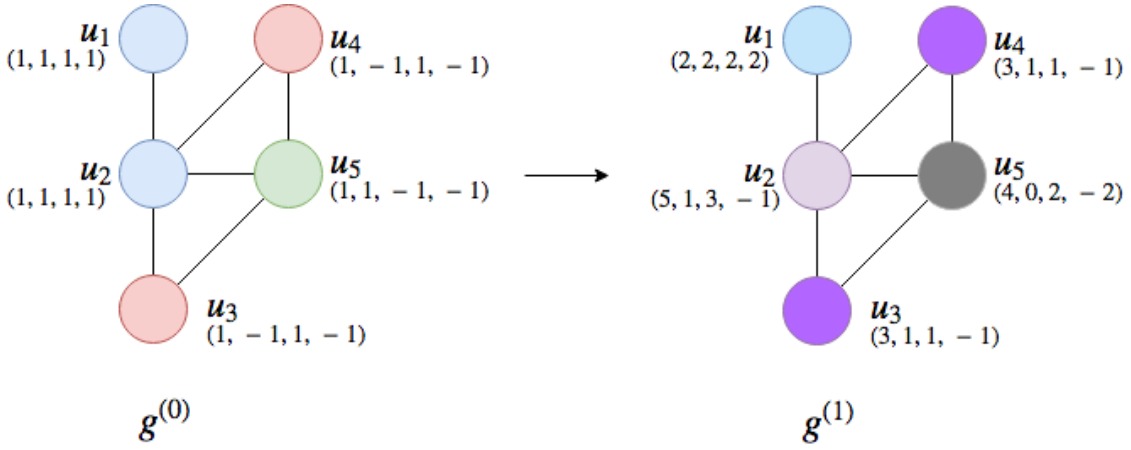
### 2.6.13 Πυρήνας Κώδικα Hadamard

Μία τεχνική εμπλουτισμού των επισημειώσεων σαν αυτή του πυρήνα κατακερματισμού γειτόνων και του σκελετού πυρήνα Weisfeiler-Lehman, εισήχθη από τους Tetsuya Kataoka και Akihito Inokuchi στο [45], γνωστός ως πυρήνας κώδικα Hadamard.
Δεδομένου ενός συνόλου διακριτά επισημειωμένων γράφων $\mathbf{G} = [G]_{i=1}^{N}$, συλλέγουμε το σύνολο $\Sigma$ όλων των διαφορετικών επισημειώσεων του $\mathbf{G}$. Η $2^k$-οστή μήτρα κώδικα Hadamard $H_{2^k}$, ορίζεται ως εξής:

$$H_{2^{k+1}} = \begin{cases} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \text{ ιφ } k = 0 \\ \\ \begin{pmatrix} H_{2^k} & H_{2^k} \\ H_{2^k} & -H_{2^k} \end{pmatrix}, \text{ιφ } k > 0 \end{cases} \tag{2.61}$$

Τώρα ορίζοντας την μήτρα Hadamard $\mathbb{H} = H_{2^{\lceil \log_2 |\Sigma| \rceil}}$, δίνουμε σε κάθε κόμβο ως αρχική επισημείωση την:

$$l^{(0)}(v) = \texttt{row}_i \mathbb{H}, \ \textbf{ανν} \ label(v) = \Sigma_i \tag{2.62}$$




Σχήμα 2.3: Ένα παράδειγμα της διαδικασίας επανεπισημείωσης για τον πυρήνα κώδικα Hadamard

Με βάση αυτήν την επισημείωση προτείνεται ο παρακάτω κανόνας ανανέωσης των επισημειώσεων:

$$l^{(k+1)}(v) = l^{(k)}(v) + \sum_{u \in N(v)} l^{(k)}(u) \tag{2.63}$$

όπου με $N(v)$ συμβολίζουμε το σύνολο των κόμβων που είναι γείτονες του κόμβου $v$.
Ακολουθώντας τον παραπάνω κανόνα επανεπισημείωσης (βλέπε σχήμα 2.3) επαναληπτικά για έναν δεδομένο αριθμό επαναλήψεων, μπορούμε να χρησιμοποιήσουμε ένα κοινό πυρήνα βάσης $\kappa$ για διακριτές επισημειώσεις όπως στην περίπτωση του σκελετού πυρήνα Weisfeiler-Lehman, αθροίζοντας τις τιμές του από όλες τις επαναλήψεις (βλέπε εξίσωση 2.57). Η πολυπλοκότητα επανεπισημείωσης του πυρήνα Hadamard είναι επίσης γραμμική (όπως και του πυρήνα κατακερματισμού γειτονιών) αλλά έχει αποδειχθεί πειραματικά πως δύναται να ανταποκριθεί αποτελεσματικά σε δεδομένα κλιμακούμενου μεγέθους.

### 2.6.14 Πυρήνας Αλμάτων Γράφων

Δεδομένου δύο γράφων, ο πυρήνας αλμάτων γράφων (Graph Hopper) συγκρίνει τα ελάχιστα μονοπάτια μεταξύ ζευγαριών κόμβων των δύο γράφων [23]. Ο πυρήνας λαμβάνει υπόψιν τόσο τα μήκη των μονοπατιών όσο και τους κόμβους που συναντά, ενώ κάνει 'άλματα' μεταξύ συντομότερων μονοπατιών. Ο πυρήνας είναι ισοδύναμος με ένα σταθμισμένο άθροισμα από πυρήνες κόμβων.
Έστω $G = (V, E)$ ένας γράφος. Ο γράφος αποτελείται είτε από διακριτές επισημειώσεις είτε από συνεχείς. Έστω $\ell : \mathcal{V} \to \mathcal{L}$ μία συνάρτηση επισημείωσης που αποδίδει είτε διακριτές, είτε συνεχείς επισημειώσεις στους κόμβους του γράφου $G$. Ο πυρήνας συγκρίνει επισημειώσεις κόμβων (διακριτές ή συνεχείς) χρησιμοποιώντας έναν πυρήνα $k_n$ (π.χ. του πυρήνα dirac στην περίπτωση διακριτών επισημειώσεων και ενός γραμμικού ή γκαουσιανού στην περίπτωση συνεχών επισημειώσεων). Δεδομένου δύο κόμβων $v, u \in V$ και ενός μονοπατιού (βλέπε ορισμό 2.11) $\pi$ από το $v$ στο $u$, θα συμβολίζουμε με $\pi(i) = v_i$ τον $i$οστό κόμβο που συναντάμε ενώ κάνουμε άλματα από τον έναν κόμβο στον άλλο μέσα στο μονοπάτι. Αν συμβολίσουμε με

$l(\pi)$ το σταθμισμένο μήκος του $\pi$ και με $|\pi|$ το μήκος του όσον αφορά τον αριθμό των κόμβων που περιέχει $\pi$. Το ελάχιστο μονοπάτι $\pi_{ij}$ από $v_i$ στο $v_j$ ορίζεται με βάση το σταθμισμένο μήκος. Η διάμετρος $\delta(G)$ του $G$ είναι ο μέγιστος δυνατός αριθμός κόμβων σε ένα ελάχιστο μονοπάτι του $G$, όσον αφορά το σταθμισμένο μήκος.
Ο πυρήνας αλμάτων γράφων ορίζεται ως το άθροισμα των πυρήνων μονοπατιών $k_p$ στις οικογένειες ελαχίστων μονοπατιών $P, P'$ των $G, G'$

$$k(G, G') = \sum_{\pi \in P} \sum_{\pi' \in P'} k_p(\pi, \pi') \tag{2.64}$$

Ο πυρήνας μονοπατιών $k_p(\pi, \pi')$ είναι το άθροισμα των πυρήνων κόμβων $k_n$ που συναντώνται ταυτόχρονα ενώ κάνουν άλματα $\pi$ ανδ $\pi'$ μεταξύ κόμβων ίδιου μήκος στον αριθμό τους, δηλαδή:

$$k_p(\pi, \pi') = \begin{cases} \sum_{j=1}^{|\pi|} k_n(\pi(j), \pi'(j)), & \text{ιφ } |\pi| = |\pi'|, \\ 0, & \text{οτηερωισε.} \end{cases} \tag{2.65}$$

Ο πυρήνας $k(G, G')$ μπορεί να γραφεί ως ένα άθροισμα από πυρήνες κόμβων:

$$k(G, G') = \sum_{v \in V} \sum_{v' \in V'} w(v, v') k_n(v, v') \tag{2.66}$$

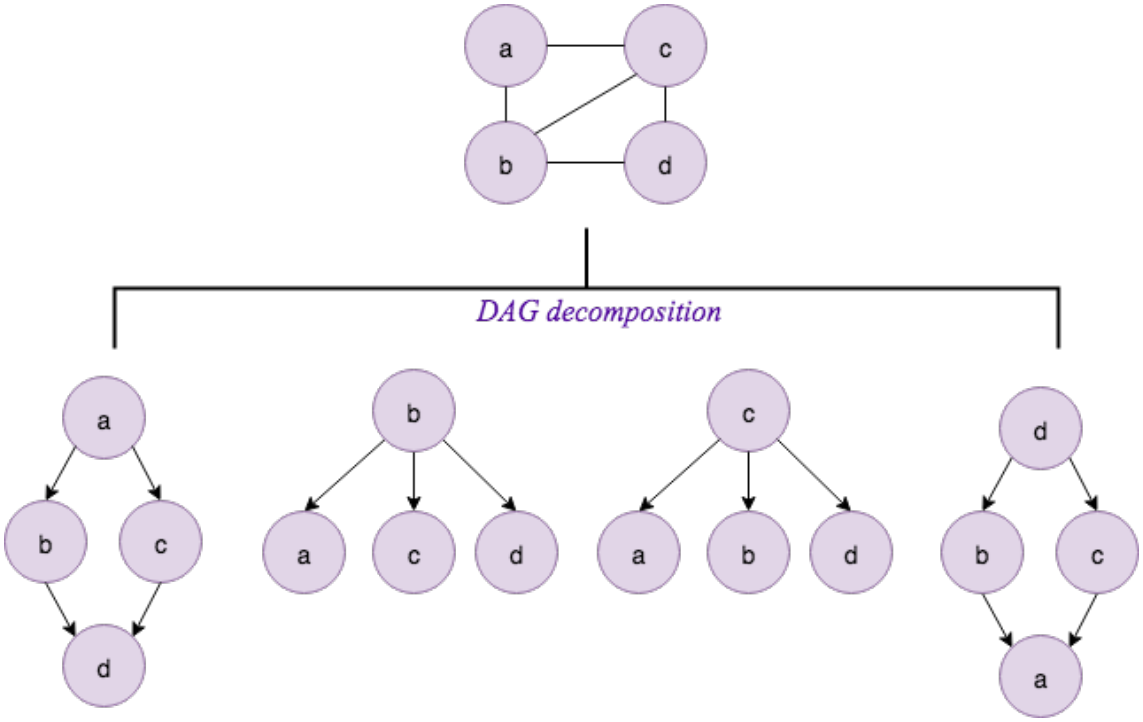
όπου το $w(v, v')$ μετράει τον αριθμό των φορών κατά τις οποίες τα $v$ και $v'$ εμφανίζονται στο ίδιο άλμα (ή συντεταγμένη) $i$ κοντινότερων μονοπατιών $\pi, \pi'$ ίδιου αριθμού κόμβων $|\pi| = |\pi'|$. Μπορούμε να γράψουμε το βάρος $w(v, v')$ ως:

$$w(v, v') = \sum_{j=1}^{\delta} \sum_{i=1}^{\delta} |\{(\pi, \pi') : \pi(i) = v, \pi'(i) = v', |\pi| = |\pi'| = j\}| = \sum_{j=1}^{\delta} \sum_{i=1}^{\delta} [\mathbf{M_v}]_{ij} [\mathbf{M_{v'}}]_{ij} \tag{2.67}$$

όπου $\mathbf{M_v}$ είναι ένας $\delta \times \delta$ πίνακας στον οποίο το στοιχείο $[\mathbf{M_v}]_{ij}$ μετράει πόσες φορές το $v$ εμφανίζεται στην $i$οστή συντεταγμένη του ελαχίστου μονοπατιού στο $G$ διακριτού μήκους $j$ και $\delta = \max(\delta(G), \delta(G'))$. Τα στοιχεία αυτών των πινάκων μπορούν να υπολογιστούν αποδοτικά χρησιμοποιώντας αναδρομικούς αλγορίθμους περάσματος μηνυμάτων. Η συνολική πολυπλοκότητα υπολογισμού του $k(G, G')$ είναι $\mathcal{O}(n^2(m + \log n + d + \delta^2))$ όπου $n$ είναι ο αριθμός των κόμβων, όπου $m$ είναι ο αριθμός των ακμών και $d$ είναι η διάσταση των χαρακτηριστικών κόμβων ($d = 1$ στην περίπτωση των διακριτών επισημειώσεων κόμβων).

### 2.6.15 Πυρήνας ODD-STh

Ο πυρήνας ODD-STh είναι ένας πυρήνας μεταξύ γράφων με επισημειώσεις. Η ιδέα στην οποία στηρίζεται η σχεδίαση αυτού του πυρήνα, είναι αυτή της χρήσης πυρήνων για ταξινομημένα δέντρα με ρίζα, δηλαδή πυρήνων για γράφους που ακολουθούν τις εξής ιδιότητες: (1) είναι κατευθυνόμενοι, (2) όλοι οι κόμβοι ανάγονται με σχέση προγόνου σε ένα κοινό κόμβο *ή ρίζα* και (3) οι κόμβοι του δέντρου είναι δομικά *διατεταγμένοι*. Πυρήνες που αφορούν τέτοιους γράφους, έχουν μελετηθεί στη βιβλιογραφία λόγω της μεγάλης εκφραστικότητας τους και τις χαμηλής υπολογιστικής τους πολυπλοκότητας [37, 59, 82].




Σχήμα 2.4: Ένα παράδειγμα αποσύνθεσης ενός γράφου σε ένα σύνολο ακυκλικών γραφημάτων μέσω εξερευνήσεων BFS

Η ιδέα πίσω από τον πυρήνα ODD-STh που προτείνεται στο [56], ξεκινά από την αποσύνθεση δύο γράφων σε ένα σύνολο ταξινομημένων ακυκλικών γραφημάτων και την πρόσθεση όλων των τιμών ενός πυρήνα $K_{DAG}$ για ακυκλικά γραφήματα μεταξύ όλων των ζευγαριών τους ως εξής:

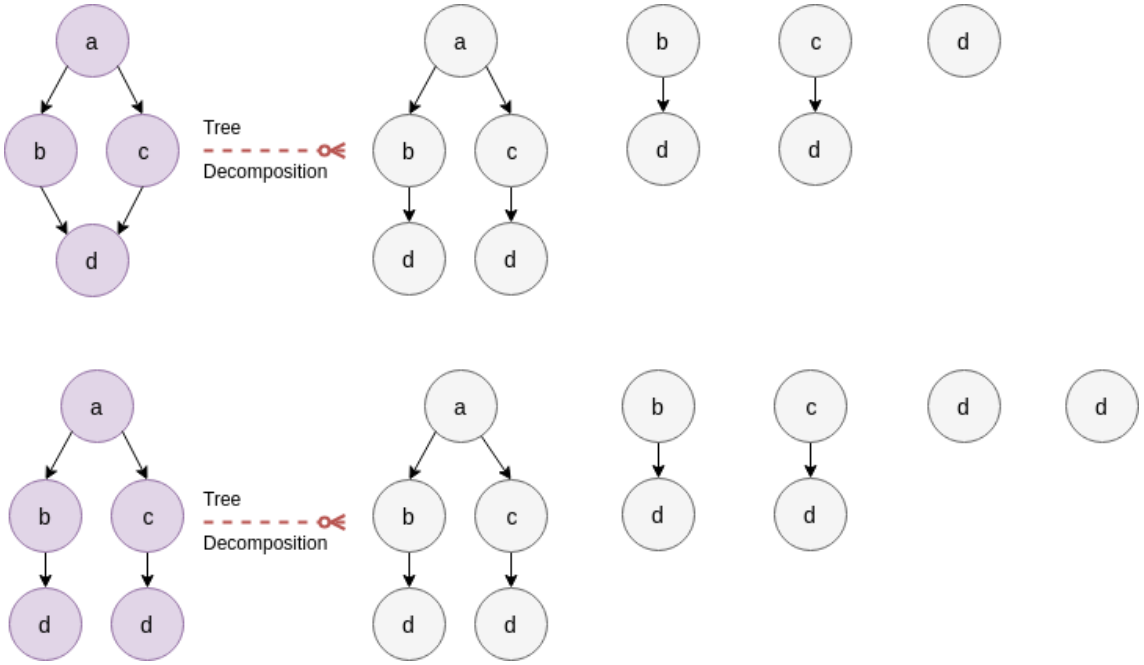
$$K_{K_{DAG}}(G_1, G_2) = \sum_{\substack{D_1 \in DD(G_1) \\ D_2 \in DD(G_2)}} K_{DAG}(D1, D2) \tag{2.68}$$

όπου με $DD(G_i)$ θα συμβολίζουμε την αποσύνθεση ενός γράφου σε ακυκλικά γραφήματα. Συγκεκριμένα για κάθε γράφο $G_i$, το $DD(G_i)$ θα είναι ίσο με το σύνολο όλων των κατευθυνόμενων εξερευνήσεων BFS που ξεκινούν από κάθε κόμβο του γράφου (βλέπε σχήμα 2.4) Εν συνεχεία, ο πυρήνας $K_{DAG}$ ανάγεται στο άθροισμα ενός πυρήνα ριζωμένων δέντρων με διάταξη $C()$ ως εξής:

$$K_{DAG} = \sum_{\substack{v_1 \in V(D_1) \\ v_2 \in V(D_2)}} C(root(T(v_1)), root(T(v_2))) \tag{2.69}$$

Ως πυρήνα $C()$ θα θεωρούμε από εδώ και στο εξής τον πυρήνα υποδέντρων (**S**ub-**T**ree kernel), όπως ορίζεται στο [83]. Το $T()$ αντιστοιχεί στο σύνολο *επισκέψεων δέντρων* (βλέπε σχήμα 2.5) σε ένα κατευθυνόμενο ακυκλικό γράφημα οι οποίες ακολουθούν τον ακόλουθο κανόνα *διάταξης* μεταξύ των κόμβων τους.

**Ορισμός 2.25** (Αυστηρή σχέση μερικής διάταξης $\dot{>}$ μεταξύ των κόμβων ενός ΚΑΓ)**.** *Ας*



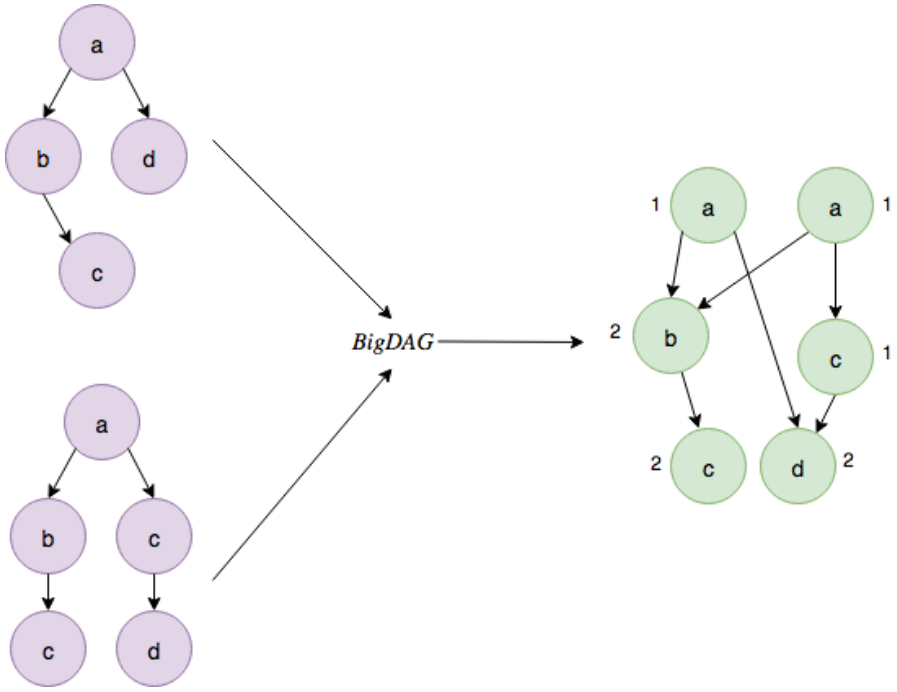

Σχήμα 2.5: Επισκέψεις ταξινομημένων δέντρων μεταξύ δύο ΚΑΓ. Προσέξτε ότι η δεύτερη περίπτωση διαφέρει από την πρώτη στη συχνότητα εμφάνισης του δέντρου-κόμβου d.

*υποθέσουμε ότι μεταξύ των κόμβων ενός γράφου έχουμε ολική διάταξη (π.χ. λεξικογραφική). Τότε η σχέση $\dot{>}$ είναι αληθής αν μία από τις επόμενες συνθήκες αληθεύουν:*

1. $L(v_i) < L(v_j)$
2. $L(v_i) = L(v_j) \wedge [\delta^+(v_i) < \delta^+(v_j)]$
3. *Έστω* $[L(v_i) = L(v_j)] \wedge [\delta^+(v_i) = \delta^+(v_j)]$ *και τα παιδιά* $ch_l[v_i]$ *του* $v_i$ *είναι μερικώς ταξινομημένα από την σχέση που ορίζουμε, δηλ. σχηματίζουν δύο μερικώς ταξινομημένα σύνολα.*
   *Τότε μπορούμε να ορίσουμε δύο ακολουθίες:*
   $$ch_1[v_j], ch_2[v_j], \ldots, ch_m[v_j] ch_1[v_j], ch_2[v_j], \ldots, ch_m[v_j]$$
   *όπου* $m = \delta^+(v_i) = \delta^+(v_j)$ *και κάθε* $ch_k[v_i]$ *είναι ένα (όχι απαραίτητα μοναδικό) ελάχιστο στοιχείο στο σύνολο που ορίζεται ως* $\{ch_l[v_i] | l \in \{k, \ldots, m\}\}$. *Τότε η σχέση αληθεύει αν:*
   $$(\exists l. ch_l[v_i] \dot{<} ch_l[v_j]) \wedge (\neg \exists k < l. (ch_k[v_i] \dot{<} ch_k[v_j] \vee ch_k[v_j] \dot{<} ch_k[v_i]))$$

*ορίζει μία μερική διάταξη (βλ. [56, Θεώρημα 5.1]).*

Δεδομένης της παραπάνω σχέσης μερικής διάταξης μεταξύ των κόμβων ενός ΚΑΓ, μπορούμε να ορίσουμε μία νέα αποσύνθεση με διάταξη ODD (Ordered Dag Decomposition - Διατεταγμένη Αποσύνθεση ΚΑΓ), βάσει της οποίας όλα τα ΚΑΓ μπορούν να συνοψιστούν σε ένα μεγάλο ΚΑΓ που θα συμβολίζεται ως $BigDAG$. Η μέθοδος αυτή που εισήχθει στο [1, MinimalDAG: Σχήμα 2, p. 3], συνοψίζει κόμβους με κοινές επισημειώσεις (μέσω ενός μετρητή




Σχήμα 2.6: Κατασκευή ενός $BigDAG$ από δύο επιμέρους ΚΑΓ. Οι ακέραιοι αριθμοί στους κόμβους του $BigDAG$ αντιστοιχούν στις συχνότητες εμφάνισης τους.

συχνοτήτας εμφάνισης) αν ανήκουν στο ίδιο μονοπάτι του κάθε ΚΑΓ, ενώ συντηρεί και ενσωματώνει στο πλήρες γράφημα κόμβους για τους οποίους δεν βρέθηκε τρόπος να συνοψιστούν (βλέπε σχήμα 2.6). Είναι δόκιμο να πούμε ότι το συνολικό $BigDAG$ μετράει τον αριθμό των υποδομών που ταυτίζονται μεταξύ των ODD ενός γράφου. Λόγω της σχέσης μερικής διάταξης το παραγόμενο $BigDAG$ μας επιτρέπει να οδηγηθούμε στην παρακάτω ισοδυναμία:

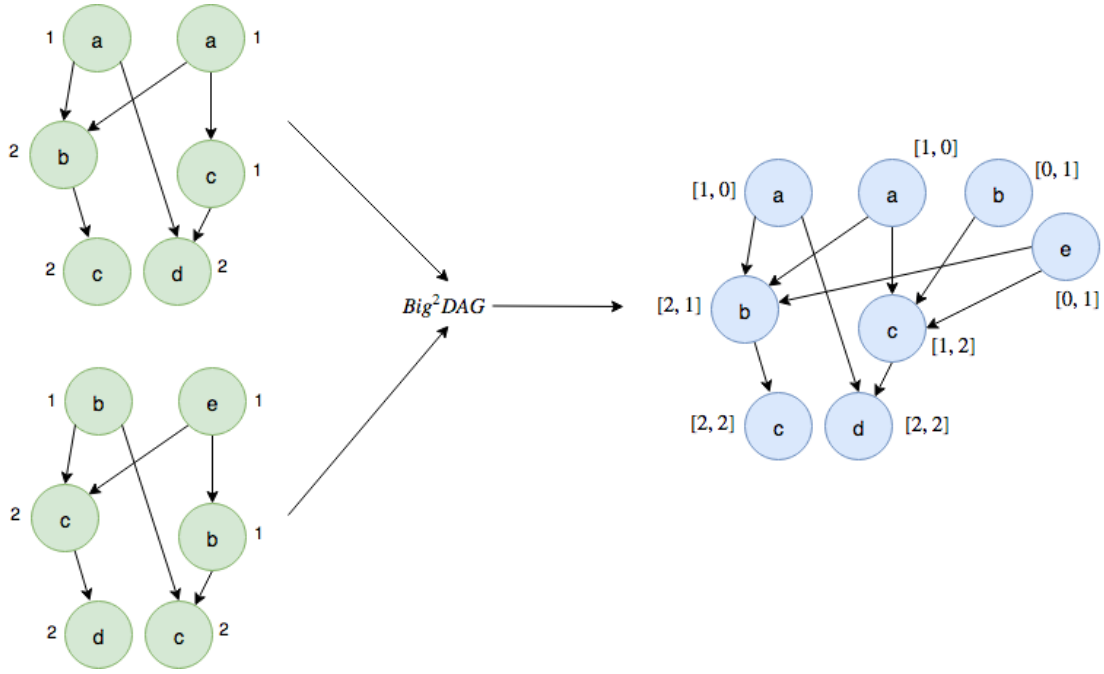
$$K_{K_{DAG}}(G_1, G_2) = K_{BigDAG}(G_1, G_2) = \sum_{\substack{u_1 \in V(BigDAG(G_1)) \\ u_2 \in V(BigDAG(G_2)}} f_{u_1} f_{u_2} C(u_1, u_2) \tag{2.70}$$

όπου $f_u$ είναι ο μετρητής συχνότητας εμφάνισης του κόμβου $u$ και $C(u, v)$ είναι το πλήθος των γνήσιων υποδέντρων των δέντρων που ξεκινούν από τους κόμβους $u$ και $v$.

Τέλος έχοντας αναπαραστήσει κάθε γράφο ως ένα $BigDAG$, μπορούμε να συνοψίσουμε όλους τους γράφους σε ένα κοινό $Big^2DAG$ αν αντί να συνεχίσουμε να προσθέτουμε τις συχνότητες εμφάνισης των κόμβων όταν αυτοί τατίζονται, τις αντικαταστήσουμε με ένα διάνυσμα συχνοτήτων το οποίο έχει στην $i$-οστή θέση την τιμή 0 αν ο κόμβος δεν εμφανίζεται στο $i$-οστό $BigDAG$ ενώ αλλιώς έχει για τιμή την συχνότητα εμφάνισης του (βλέπε σχήμα 2.7).

Στο τελικό $Big^2DAG$ γράφημα, ο υπολογισμός της μήτρας πυρήνα ανάγεται στον παρακάτω υπολογισμό:

$$K_{Big^2DAG}(G_i, G_j) = \sum_{u_1, u_2 \in V(Big^2DAG)} F_{u_1}[i] \star F_{u_2}[j] C(u_1, u_2) \tag{2.71}$$




Σχήμα 2.7: Κατασκευή ενός $Big^2DAG$ από δύο επιμέρους $BigDAG$. Οι ακέραιοι αριθμοί αντικαθίστανται από διανύσματα συχνότητας που συγκρατούν την τιμή συχνότητας του κάθε κόμβου στο αρχικό $BigDAG$ ή δίνουν την τιμή 0, στην περίπτωση που δεν υπήρχε.

που είναι ισοδύναμος με:

$$K_{Big^2DAG}(G_i, G_j) = \sum_{u \in V(Big^2DAG)} F_u[i] \star F_u[j] C(u, u) \tag{2.72}$$

επειδή ο πυρήνας υποδέντρων θα ταιριάξει μόνο στην περίπτωση που αυτά ταυτίζονται, δηλαδή:

$$C(u_1, u_2) \neq 0 \leftrightarrow T(u_1) = T(u_2) \tag{2.73}$$

Τέλος προκειμένου να κατασκευάσουμε το $Big^2DAG$ κάθε κόμβος θα πρέπει να αναπαρασταθεί σαν μία τούπλα $\langle D_u, F_u[\cdot], ID_u \rangle$ όπου με $D_u$ συμβολίζουμε το μέγεθος του υποδέντρου που ξεκινάει από τον κόμβο $u$, με $F_u[\cdot]$ το διάνυσμα συχνότητας εμφάνισης αυτού του κόμβου και με $ID_u$ ένα αλφαριθμητικό που μας χρησιμεύει προκειμένου να αναπαριστούμε αυτόν τον κόμβο **μοναδικά** (ιδιότητα που προκύπτει από την σχέση μερικής διάταξης). Συγκεκριμένα:

$$ID_u = \begin{cases} L_u, \text{ αν } \delta^+(u) = 0 \\ L_u(ID_{ch_1[u]}, \cdots, ID_{ch_{\delta^+(u)}[u]}), \text{ αλλιώς} \end{cases} \tag{2.74}$$

Προκειμένου να μειωθεί ακόμα περισσότερο ο χρόνος εκτέλεσης του αλγορίθμου μία προσέγγιση προτάθηκε, συγκεκριμένα αυτή του περιορισμού του βάθους εξερεύνησης των δέντρων BFS κατά την αποσύνθεση ODD, σε μία μέγιστη τιμή $h$. Η θεωρητική πολυπλοκότητα του αλγορίθμου, θεωρώντας ότι το $h$ είναι σταθερό και παίρνει μικρές τιμές, είναι της τάξης του $\mathcal{O}(n \log n)$ [56, Υποενότητα 5.5].

### 2.6.16 Πυρήνας Διάδοσης

Οι πυρήνες διάδοσης εισήχθησαν ως ένα γενικός σκελετός πυρήνα στο [60]. Βασίζονται στην ιδέα της διάδοσης της πληροφορίας επισημείωσης μεταξύ κόμβων του γράφου, με βάση τη δομή του. Κάθε γράφος θεωρείται ότι έχει συνεχείς επισημειώσεις στους κόμβους, όπου στην περίπτωση ύπαρξης διακριτών, αυτές μετασχηματίζονται σαν διανύσματα σταθερού μήκους όσο το μέγεθος του συνόλου των δυνατών επισημειώσεων και με άσους στην θέση που αντιστοιχεί στην αντίστοιχη επισημείωση (ήτοι One-Hot-Vectors). Το σύνολο όλων των κόμβων ενός γράφου, μπορεί να ειδωθεί σαν μία κατανομή πιθανότητας $P$ μεγέθους $n \times d$, όπου το $n$ αντιστοιχεί στο νούμερο των κόμβων και το $d$ στη διάσταση των χαρακτηριστικών. Έπειτα η ιδέα της διάδοσης εφαρμόζεται προκειμένου να κατασκευάσουμε ένα αλγοριθμικό σκελετό για τους πυρήνες διάδοσης. Στην γενική του μορφή, ένας πυρήνας διάδοσης ακολουθεί τον σκελετό του αλγορίθμου 1.

**Algorithm 1:** Υπολογισμός του γενικού πυρήνα διάδοσης

**δεδομένα**: ένα σύνολο γράφων $\{G^{(i)}\}_i$, ένας αριθμός επαναλήψεων $t_{MAX}$, επαναληπτικό σχήμα διάδοσης, ένας πυρήνας βάσης $\langle \, . \, , \, . \rangle$

**αρχικοποίηση**: $K \leftarrow 0$, αρχικοποίηση των κατανομών $P_0^{(i)}$.

**for** $t \leftarrow 0 \dots t_{MAX}$ **do**

  **forall** *τους γράφους* $G^{(i)}$ **do**

    **forall** *τους κόμβους* $u \in G^{(i)}$ **do**

      κατακερμάτισε το $p_{t,u}$ σε ομάδες, ; ▷ διάνυσμα με αριθμούς ομάδας

      όπου $p_{t,u}$ είναι γραμμή του $P_t^{(i)}$ που αντιστοιχεί στον κόμβο $u$

    υπολόγισε το $\Phi_i = \phi(G_t^i)$ ; ▷ υπολόγισε πόσα στοιχεία ανήκουν σε κάθε ομάδα

  $K \leftarrow K + \langle \Phi, \Phi \rangle$ ; ▷ προσέθεσε την συνεισφορά αυτής της επανάληψης στον υπολογισμό πυρήνα

**forall** *τους γράφους* $G^{(i)}$ **do**

  $P_{t+1}^{(i)} \leftarrow P_t^{(i)}$ ; ▷ διάδωσε την πληροφορία του κάθε κόμβου

Ο υπολογισμός πυρήνα $\langle \Phi, \Phi \rangle_{ij}$, στην επανάληψη $t$ μεταξύ δύο γράφων $i, j$ είναι ισοδύναμος με το ακόλουθο διπλό άθροισμα:

$$K(G_t^{(i)}, G_t^{(j)}) = \sum_{u \in G_t^{(i)}} \sum_{v \in G_t^{(j)}} k(u, v) \tag{2.75}$$

όπου ο υπολογισμός του πυρήνα μεταξύ κόμβων $k(u, v)$, γίνεται μέσω ομαδοποίησης (binning). Προκειμένου να ομαδοποιηθούν αποτελεσματικά οι κόμβοι, έπρεπε να βρεθεί μία μέθοδος που να είναι τόσο υπολογιστικά αποδοτική, όσο και εκφραστική. Μία απλή συνάρτηση κατακερματισμού απορρίφθηκε, καθώς θα διαχώριζε τιμές που ήταν πολύ πιο *όμοιες* μεταξύ τους σε σχέση με άλλες. Μία έννοια *τοπικότητας* έπρεπε να προστεθεί στην διαδικασία ομαδοποίησης, προκειμένου παρόμοια μοτίβα διάχυσης να μαζεύονται σε ίδιες ομάδες. Για το σκοπό αυτό, χρησιμοποιήθηκε η τεχνική του τοπικά ευαίσθητου κατακερματισμού (Locally Sensitive Hashing - **LSH**) όπως φαίνεται στον αλγόριθμο 2, για διάφορες μετρικές εγγύτητας, η οποία εφαρμόζεται συνολικά στην κατανομή πιθανότητας όλων των κόμβων, όλων των γράφων της εισόδου.

Όσο αφορά τον τρόπο διάδοσης, βάσει ενός πίνακα μετάβασης $T$ για κάθε γράφο, ο οποίος είναι κανονικοποιημένος κατά γραμμή, ένα επαναληπτικό σχήμα διάδοσης σχεδιάστηκε στη

**Algorithm 2:** Υπολογισμός του LSH

**δεδομένα**: πίνακας $X \in \mathcal{R}^{N \times D}$, μέγεθος ομάδας $w$, μετρική $M$
**if** $M = H$ **then**
  $X \leftarrow \sqrt{X}$ ; ▷ μετασχηματισμός τετραγωνικής ρίζας
**if** $M = H \; or \; M = L2$ ; ▷ δημιούργησε ένα τυχαίο διάνυσμα προβολής
**then**
  $v \leftarrow \text{RAND-NORM}(D)$ ; ▷ πάρε τυχαία δείγματα από την $\mathcal{N}(0,1)$
**else if** $M = TV \; or \; M = L1$ **then**
  $v \leftarrow \text{RAND-NORM}(D)/\text{RAND-NORM}(D)$ ; ▷ πάρε τυχαία δείγματα από την $Cauchy(0,1)$
$b = w * \text{RAND-UNIF}()$ ; ▷ τυχαίο διάνυσμα μετατόπισης $b \sim \mathcal{U}[0, w]$
$h(X) = \text{floor}((X * v + b)/w)$ ; ▷ υπολόγισε τους κατακερματισμούς

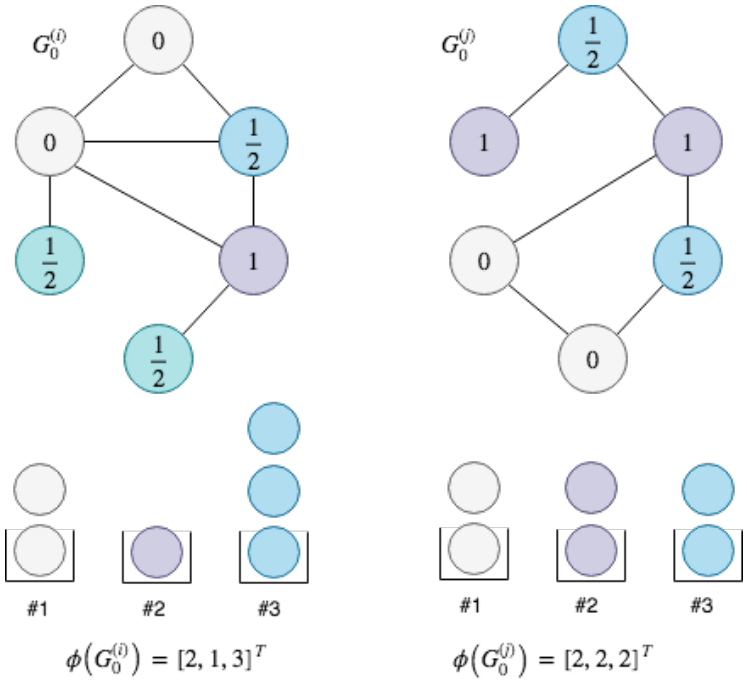
βάση του επόμενου απλού νόμου αντικατάστασης:

$$P_{t+1} \leftarrow TP_t \tag{2.76}$$

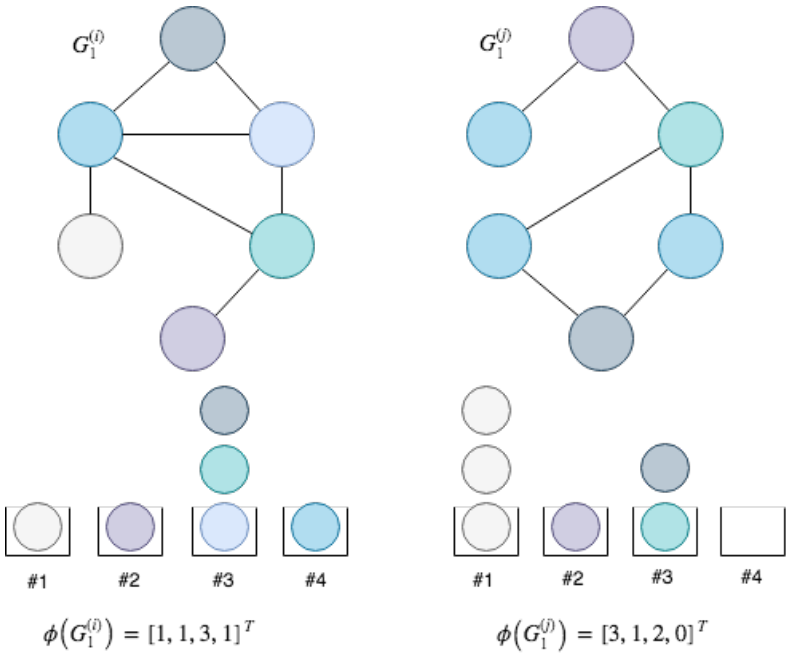
Ο πίνακας μετάδοσης $T$ είναι κατά κανόνα ίσος με $D^{-1}A$ για κάθε γράφο, όπου με $A$ συμβολίζουμε τον πίνακα γειτνίασης (βλέπε ορισμό 2.4) αυτού του γράφου και $D = \text{diag}(\sum_j A_{ij})$. Σχηματικό παράδειγμα τέτοιας προώθησης μπορεί να φανεί στο σχήμα 2.8. Ο πυρήνας διάδοσης που τελικά υλοποιήσαμε στο πλαίσιο του grakel ακολουθεί τον αλγόριθμο 3, ενώ στην περίπτωση μας θεωρούμε ότι οι γράφοι που μας δίνονται είναι *πλήρως επισημειωμένοι*.

**Algorithm 3:** Υπολογισμός του γενικού πυρήνα διάδοσης για πλήρως επισημειωμένους πυρήνες

**δεδομένα**: ένα σύνολο γράφων $\{G^{(i)}\}_i$, ένας αριθμός επαναλήψεων $t_{MAX}$, πίνακας μετάδοσης $T$, μέγεθος ομάδας $w$, μετρική $M$, ένας πυρήνας βάσης $\langle . \, , \, . \rangle$
**αρχικοποίηση**: $K \leftarrow 0$, αρχικοποίηση των κατανομών $P_0 \leftarrow \delta_{l(V)}$ (στην περίπτωση διακριτών επισημειώσεων) ή $P_0 \leftarrow attr(V)$ στην περίπτωση των συνεχών.
**for** $t \leftarrow 0 \cdots t_{MAX}$ **do**
  CALCULATE-LSH$(P_t, w, M)$ ; ▷ ομαδοποίησε τους κόμβους
  **forall** *τους γράφους* $G^{(i)}$ **do**
    Υπολόγισε το $\Phi_t = \phi(G_t^{(i)})$ ; ▷ μέτρησε το πλήθος των στοιχείων σε κάθε ομάδα
  $P_{t+1} \leftarrow TP_t$ ; ▷ διάχυση επισημειώσεων
  $K \leftarrow K + \langle \Phi, \, \Phi \rangle$ ; ▷ προσέθεσε την συνεισφορά αυτής της επανάληψης στον υπολογισμό πυρήνα




(α′) Αρχική κατανομή επισημειώσεων ($t = 0$)




(β′) Ανανεωμένη κατανομή επισημειώσεων ($t = 1$)

Σχήμα 2.8: Παράδειγμα εφαρμογής του αλγορίθμου προώθησης, με τοπικά ευαίσθητο κατακερματισμό, για δύο επαναλήψεις 2.8α′, 2.8β′.

# Κεφάλαιο 3

# Ανάπτυξη του GraKeL

Η σχεδίαση μίας μοντέρνας βιβλιοθήκης προγραμματισμού δεν είναι μία απλή υπόθεση. Ο προγραμματιστής καλείται να συνδυάσει 'κοινωνικές' ιδιότητες της βιβλιοθήκης, όπως η ευχρηστία και η υψηλού επιπέδου οργάνωση, με ιδιότητες 'υλικού' που προκύπτουν από το στόχο της υπολογιστικής αποτελεσματικότητας. Σε αυτό το κεφάλαιο θα ασχοληθούμε με την βιβλιοθήκη από την σκοπιά της ανάπτυξης λογισμικού. Πρώτα θα περιγράψουμε την βασική μεθοδολογία σχεδίασης που ακολουθήθηκε, περιγράφοντας τον θεμελιώδη λειτουργικό σκελετό της βιβλιοθήκης, μαζί με τα βασικά της αντικείμενα. Έπειτα με το πιο θεμελιώδες αντικείμενο αυτού του πακέτου που αφορά τον υπολογισμό πυρήνα, προκειμένου να περιγράψουμε πιο αναλυτικά τον τρόπο με τον οποίο σχεδιάστηκε και λειτουργεί. Τέλος θα δοθούν συμπληρωματικές πληροφορίες σχετικά με το πως μία σύγχρονη βιβλιοθήκη προγραμματισμού συσκευάζεται (packaging), διανέμεται και δοκιμάζεται.

## 3.1 Σχεδιαστικές Αποφάσεις

Το GraKeL επιλέχθηκε να αναπτυχθεί σε γλώσσα προγραμματισμού Python. Η γλώσσα αυτή έχει αποδείξει την αξία της τόσο στην έρευνα όσο και σε εφαρμογές [16]. Διαθέτει εκτέλεση με διερμηνέα (interpreter), που διευκολύνει τον προγραμματιστή να αναπτύσσει εφαρμογές πολύ γρήγορα, καθώς λόγω του οκνηρού (lazy) συστήματος τύπων που διαθέτει, τα σημασιολογικά λάθη προκύπτουν μόνο την στιγμή που θα αποτελέσουν πρόβλημα. Κάτι τέτοιο φέρνει την διαδικασία διόρθωσης του προγράμματος (debugging) στον ίδιο *χρόνο* με την ίδια του την εκτέλεση. Ταυτόχρονα υποστηρίζει το μοντέλο του αντικειμενοστρεφούς προγραμματισμού, υποστηρίζοντας έτσι και την σχεδίαση μίας σύγχρονης βιβλιοθήκης. Στο μοντέλο αυτό η στοιχειώδης δομή δεδομένων καλείται *αντικείμενο* και αποτελεί πραγματικό στιγμιότυπο στη μνήμη ενός σύνθετου, και πιθανώς οριζόμενου από τον χρήστη, τύπου δεδομένων ονόματι κλάση. Κάθε κλάση αποτελείται από *ιδιότητες* (attributes), που αποτελούν ένα είδος εσωτερικής μεταβλητής και μεθόδους, που αποτελούν ένα είδος εσωτερικής συνάρτησης. Π.χ. μία κλάση που ορίζει ένα γράφος, θα διαθέτει ιδιότητες όπως το σύνολο των κόμβων, το σύνολο των ακμών και τις επισημειώσεις και μεθόδους όπως ο υπολογισμός της πυκνότητας του, του πίνακα κοντινότερων μονοπατιών ή ακόμα και πράξεις μεταξύ αντικει-

μένων αυτής της κλάσης γράφων, όπως το γινόμενο. Το μοντέλο αυτό, δίνει την ευελιξία στην/ον προγραμματίστρια/τιστή, τόσο να επεκτείνει τρομερά αποτελεσματικά τις υπάρχουσες δομές δεδομένων που υπάρχουν από την δημιουργία της και εν συνεχεία τις καινούργιες που δημιουργεί αυτός ή άλλοι προγραμματιστές, όσο και να ενσωματώνει όλη την πληροφορία που αφορά το στιγμιότυπο ενός αντικειμένου (δεδομένα, συναρτήσεις) στο περιεχόμενο μίας και μόνο μεταβλητής. Ακόμα οι συντακτικοί κανόνες της γλώσσας είναι διαμορφωμένοι με τέτοιο τρόπο που η στοίχιση κώδικα χρησιμοποιείται, προκειμένου να μειώσει την χρήση συντακτικών συμβόλων, κάνοντας πολύ ευκολότερη την ανάγνωση του κώδικα και κατ΄ επέκταση την περαιτέρω ανάπτυξη ή ενσωμάτωση υπάρχοντος κώδικα σε εφαρμογές.
Βέβαια, ο σημαντικότερος λόγος για τον οποίο η γλώσσα προγραμματισμού Python έχει επικρατήσει, που είναι τόσο αποτέλεσμα όσο και η αιτία του σχεδιασμού της, είναι το μεγάλο *οικοσύστημα* βιβλιοθηκών, εργαλείων και πλαισίων λογισμικού, τα οποία έχουν αναπτυχθεί σε αυτήν τα τελευταία χρόνια ταυτόχρονα με την επικράτηση της ελεύθερης διάθεσης και τροποποίησης τους μέσω των αδειών ανοιχτού λογισμικού. Για να απλοποιήσουν αυτή τη διαδικασία οι σχεδιαστές της python δημιούργησαν ένα package manager γνωστό ως pip υπεύθυνο για την εγκατάσταση βιβλιοθηκών καθώς και μία πλατφόρμα γνωστή ως PyPi στην οποία οποιοσδήποτε μπορεί να ανεβάσει πακέτα προς εγκατάσταση. Πολύ σημαντικός παράγοντας στην διάδοση, τον διαμοιρασμό και την επεξεργασία του ανοιχτού κώδικα αποτέλεσε η ύπαρξη ηλεκτρονικών αποθετηρίων (repositories) όπως το GitHub. Χρησιμοποιώντας ένα λογισμικό που είχε αρχικά αναπτυχθεί για τον συνεπή έλεγχο των εκδόσεων (version control) στην ανάπτυξη του πυρήνα του Linux, γνωστό ως git, ηλεκτρονικά αποθετήρια αυτής της μορφής καθιστούν ιδιαίτερα εύκολη την προβολή, χρήση και την συνεισφορά ή επέκταση του κώδικα οποιουδήποτε χρήστη τους από οποιονδήποτε άλλο.

### 3.1.1 Το Πρότυπο του scikit-learn

Μία πολύ γνωστή βιβλιοθήκη μηχανικής μάθησης στην Python είναι γνωστή ως scikit-learn. Εκτός από τις πολύ γρήγορες υλοποιήσεις ενός μεγάλου πλήθους αλγορίθμων σε ένα μεγάλο εύρος τεχνικών στο χώρο της μηχανικής μάθησης, η βιβλιοθήκη αυτή συνδυάζει τρομερή ευχρηστία ταυτόχρονα με αναλυτικά εγχειρίδια για όλους τους διαφορετικούς υποψήφιους χρήστες της (οι οποίοι είναι της τάξης των εκατομμυρίων) [64]. Προκειμένου διάφοροι ενδιαφερόμενοι προγραμματιστές να μπορούν να προτείνουν δυνατές επεκτάσεις της, καθώς και για να καθιερώσει ένα πρότυπο κλάσεων μηχανικής μάθησης για τον αντικειμενοστρεφή προγραμματισμό, μία *φόρμα* αυτού του λογισμικού δημιουργήθηκε στο GitHub, ως sklearn-template. Λόγω των παραπάνω και με βάση το γεγονός ότι δεν υπήρχε υποστήριξη για το είδος των τεχνικών πυρήνα και συγκεκριμένα των πυρήνων γράφων, το παραπάνω σχεδιαστικό πρότυπο επιλέχθηκε για την ανάπτυξη του GraKeL. Σαν κληρονομική σχέση δύο θεμελιωδών κλάσεων του `sklearn`, συγκεκριμένα της κλάσης `sklearn.base.BaseEstimator` και της κλάσης `sklearn.base.TransformerMixin`, σχεδιάστηκε η βασική κλάση του grakel, γνωστή ως `grakel.Kernel`. Κάθε υλοποίηση ενός πυρήνα γράφων, αποτελεί μία κλάση που κληρονομεί την κλάση `Kernel`. Η κλάση αυτή αποτελεί κάτι ενδιάμεσο μίας διεπαφής (δηλ. ενός συνόλου δηλώσεων ονομάτων μεθόδων και χαρακτηριστικών με κενό περιεχόμενο) και μία κανονικής

κλάσης. Συγκεκριμένα περιέχει κατάλληλες μεθόδους που αν υλοποιηθούν σωστά από τον προγραμματιστή, η ανάπτυξη του πυρήνα συντομεύεται. Ταυτόχρονα κάθε αντικείμενο που είναι ένας έγκυρος πυρήνας μεταξύ γράφων πρέπει να υλοποιεί κάποιες βασικές μεθόδους και συνεπώς να υλοποιεί μία διεπαφή.
Όλοι οι πυρήνες τοποθετήθηκαν σε ένα υπο-πακέτο του grakel υπό την διεύθυνση `grakel.kernels`. Κάθε αντικείμενο που υλοποιεί το πρότυπο `TransformerMixin` υλοποιεί τρεις μεθόδους:

- `fit`: Προσαρμογή του μοντέλου σε ένα σύνολο δεδομένων γνωστό ως *σύνολο εκπαίδευσης* (εξαγωγή χαρακτηριστικών, παραμετροποίηση κ.α.)
- `transform`: Υπολογισμός των τιμών του μοντέλου σε ένα *πειραματικό σύνολο*, βάσει του αποτελέσματος της παραμετροποίησης στο σύνολο εκπαίδευσης
- `fit_transform`: Προσαρμογή και υπολογισμός του μοντέλου στο *σύνολο εκπαίδευσης* (κάποιες φορές μπορεί να προσφέρει μία γρηγορότερη υλοποίηση από την ακολουθία `fit` - `transform` στα ίδια δεδομένα)

Ταυτόχρονα το πρότυπο `BaseEstimator` αποτελείται από δύο μεθόδους `set_params`, `get_params`, οι οποίες αν υλοποιηθούν σωστά καθιστούν δυνατή την εξαγωγή παραμέτρων αρχικοποίησης (initialization parameters), καθώς και την εξωτερική επανάθεση αυτών των παραμέτρων σε ένα ήδη αρχικοποιημένο αντικείμενο μιας κλάσης, προκειμένου να μπορεί να ξαναχρησιμοποιηθεί (αρχικοποιημένο) με διαφορετικές παραμετροποιήσεις χωρίς να χρειάζεται να δημιουργείται κάθε φορά ένα νέο στιγμιότυπο αυτής της κλάσης. Κάθε κλάση πρέπει περαιτέρω να διατυπώνει όλες τις παραμέτρους που απαιτούνται για την αρχικοποίηση της, ρητά. Τα παραπάνω είναι σημαντικά προκειμένου ένας Transformer που σχεδιάζει ο προγραμματιστής να μπορεί να εισαχθεί στο λεγόμενο scikit-learn `Pipeline`. Στην περίπτωση αυτή η υπολογιστική μονάδα του πυρήνα γράφων, μπορεί και να προσαρμοστεί εύκολα και αφηρημένα σε μία δομή υψηλού επιπέδου βημάτων επεξεργασίας - ταξινόμησης - αξιολόγησης μίας αρχικής εισόδου δεδομένων, σχεδιάζοντας σχεδόν σε διανοητικό επίπεδο μία εφαρμογή ή ένα πείραμα μηχανικής μάθησης. Από τους παραπάνω περιορισμούς, φαίνεται ότι η σχεδίαση της κλάσης πυρήνα δεν είναι μία τετριμμένη διαδικασία.

### 3.1.2 Σχεδίαση της Κλάσης `Kernel`

Όπως είδαμε στο κεφάλαιο 2, ένας πυρήνας μεταξύ γράφων εμφανίζεται συνήθως στη βιβλιογραφία σαν μία συνάρτηση: $k : \mathcal{G} \times \mathcal{G} \to \mathbb{R}$ για την οποία υπάρχει μία απεικόνιση:

$$\phi : \mathcal{G} \to \mathbb{H}, \text{για έναν χώρο Hilbert } \mathbb{H}$$

όπου κάθε τιμή πυρήνα μπορεί να υπολογιστεί ως $k(G_i, G_j) = \langle G_i, G_j \rangle$ όπου $\langle \ . \ , \ . \ \rangle$ αναπαριστά ένα εσωτερικό γινόμενο σε αυτόν τον χώρο. Η μήτρα $[\mathcal{K}]_{ij} = k(G_i, G_j)$ που προκύπτει από όλα τα ζευγάρια γράφων μίας συλλογής, ονομάζεται μήτρα πυρήνα (βλ. 2.14). Οποιαδήποτε μήτρα που προκύπτει από ένα μέτρο ομοιότητας, είναι μήτρα πυρήνα αν για κάθε συλλογή εισόδων είναι θετικά ημιορισμένη (βλ. 2.15), δηλαδή αν $\forall \mathcal{K} : \ \lambda_{min}(\mathcal{K}) \geq 0$, όπου

$\lambda_{min}(\mathcal{K})$ η μικρότερη ιδιοτιμή του πίνακα $K$. Μελετώντας υπάρχουσες υλοποιήσεις πυρήνων στη βιβλιογραφία αυτό που διαπιστώσαμε ήταν ότι αν αντί να σχεδιάζαμε την κλάση πυρήνα ώστε να υπολογίζεται μεταξύ ζευγαριών, την σχεδιάζαμε για μία συλλογή γράφων $[G]_{i=1}^{N}$ θα είχαμε σημαντικά υπολογιστικά πλεονεκτήματα.
Συνολικά, ο τρόπος με τον οποίο η κλάση `Kernel` σχεδιάστηκε πάνω στο πρότυπο του `Transformer` είναι ο ακόλουθος:

- `fit`: Εξαγωγή χαρακτηριστικών για ένα σύνολο γράφων εκπαίδευσης
- `transform`: Υπολογισμός του πίνακα πυρήνα μεταξύ ενός συνόλου γράφων πειραματισμού και των γράφων εκπαίδευσης, είτε εξάγοντας και συγκρίνοντας όμοια χαρακτηριστικά με αυτά του `fit`, είτε υπολογίζοντας τιμές βάσει των χαρακτηριστικών του `fit`, είτε τέλος επεκτείνοντας τα υπάρχοντα και υπολογίζοντας μία γενική μετρική (χωρίς βέβαια την αποθήκευση της επέκτασης).

Τέλος για το `fit_transform` έχουμε ένα συνδυασμό των παραπάνω, πράγμα που συνήθως αποτελεί την μόνη λειτουργία που υλοποιούν οι υπάρχοντες πυρήνες της βιβλιογραφίας, από τους ίδιους τους σχεδιαστές τους.
Για να γίνει πιο σαφές το παραπάνω με βάση την μήτρα πυρήνα, δεδομένου δύο συλλογών γράφων (εκπαίδευσης/πειράματος): $G^n, G^m$, θεωρούμε την πλήρη μήτρα πυρήνα $\mathcal{K}$ ως:

$$\mathcal{K} = \left[\begin{array}{c||c} \mathcal{K}^{n\times n} & \mathcal{K}^{n\times m} \\ \hline\hline \mathcal{K}^{m\times n} & \mathcal{K}^{m\times m} \end{array}\right] \tag{3.1}$$

Τότε κάθε κλάση ενός πυρήνα που υλοποιεί την κλάση `Kernel`, θα πρέπει να έχει την ακόλουθη συμπεριφορά:

- $\mathcal{K}^{n\times n}$ = `<KernelName>.fit_transform`$(\mathcal{G}^n)$
- $\mathcal{K}^{m\times n}$ = `<KernelName>.fit`$(\mathcal{G}^{\nu})$.τρανσφορμ$(\mathcal{G}^{\mu})$
- $\mathcal{K}$ = `<KernelName>.fit_transform`$([\mathcal{G}^n\ \mathcal{G}^m])$

Σε ένα πρόβλημα ταξινόμησης γράφων (βλ. 2.3.2), αυτό που χρειάζεται να υπολογίσουμε είναι οι πυρήνες $\mathcal{K}^{n\times n}$ και $\mathcal{K}^{m\times n}$. Μία τέτοια συμπεριφορά προέκυψε και ως αναγκαιότητα για την ένταξη κάθε `Kernel` στο `Pipeline`.
Εν συνεχεία κάθε πυρήνας σχεδιάστηκε με την ακόλουθη στοιχειώδη *παραμετροποίηση*:

- `verbose` Μία λογική (`bool`) παράμετρος για να δίνει την δυνατότητα στον προγραμματιστή να παρέχει πληροφορία σχετικά με την πορεία εκτέλεσης του πυρήνα, σε περίπτωση επιθυμίας του χρήστη.
- `normalize` Η κανονικοποίηση είναι μία πολύ σημαντική ιδιότητα που πρέπει να ακολουθεί ένας πυρήνας προκειμένου να είναι χρήσιμος σε πειράματα ταξινόμησης. Αυτή η λογική (`bool`) παράμετρος αναγκάζει τον προγραμματιστή να μπορεί να εξασφαλίζει στον χρήστη της βιβλιοθήκης, ότι σε περίπτωση που το επιθυμεί ο πίνακας πυρήνας

θα είναι κανονικοποιημένος τόσο στα αποτελέσματα του `fit_transform` όσο και του `transform`. Η κανονικοποίηση είναι μία πολύ απλή πράξη διαίρεσης στοιχείο προς στοιχείο της μήτρας πυρήνα με τις τιμές της διαγωνίου της, ως εξής:

$$[\hat{\mathcal{K}}]_{ij} = \frac{[\mathcal{K}]_{ij}}{\sqrt{[\mathcal{K}]_{ii} * [\mathcal{K}]_{jj}}} \tag{3.2}$$

- `n_jobs` Μία ακέραια `int` παράμετρος που προσδιορίζει το πλήθος των παράλληλων εργασιών στις οποίες επιθυμεί ο χρήστης να διαμοιραστούν οι παραλληλοποιήσιμες εργασίες του συγκεκριμένου πυρήνα, αν υπάρχουν.

Όσον αφορά την υλοποίηση των λίγων ως τώρα σκελετών πυρήνα, δεν ορίστηκε μία ξεχωριστή κλάση. Εντούτοις, η σχεδίαση τους είχε ως κοινό χαρακτηριστικό την προσθήκη μίας παραμέτρου αρχικοποίησης (στο όνομα `base_kernel`) η οποία μπορούσε να είναι είτε μία κλάση τύπου `Kernel`, είτε μία τούπλα (`tuple`) δύο στοιχείων με πρώτο μία κλάση τύπου `Kernel` και δεύτερο ένα σύνολο ορισμάτων, με σκοπό την δυνατότητα παραμετροποίησης του πυρήνα βάσης.
Για να είναι δυνατή η κανονικοποίηση του αποτελέσματος των framework μία νέα μέθοδος χρειάστηκε να προστεθεί, σχεδιαστικά, σε κάθε αντικείμενο της κλάσης `Kernel`: η μέθοδος `diagonal`. Η μέθοδος αυτή δεν δέχεται ορίσματα και πρέπει να έχει πάντοτε την ακόλουθη συμπεριφορά: Αν ένας πυρήνας έχει γίνει `fit` αλλά όχι `transform`, τότε επιστρέφει την διαγώνιο $[\mathcal{K}^{n\times n}]_{ii}$ (παράβαλε εξίσωση 3.1). Αν αντίθετα ένας πυρήνας έχει γίνει `fit` και `transform` τότε επιστρέφει την διαγώνιο του $[\mathcal{K}^{n\times n}]_{ii}$ και την διαγώνιο του $[\mathcal{K}^{m\times m}]_{jj}$ από το τελευταίο `transform`. Σημαντικό εδώ είναι να σημειώσουμε ότι τα στοιχεία της διαγωνίου $[\mathcal{K}^{m\times m}]_{jj}$ δεν υπολογίζονται κατά το `transform` (δηλ. οι τιμές πυρήνα όλων των στοιχείων με τον εαυτό τους), αν ο χρήστης δεν έχει επιλέξει κανονικοποίηση.
Με σκοπό την ύπαρξη μίας κύριας κλάσης - αφηρημένης διεπαφής στην οποία ο χρήστης να απευθύνεται προκειμένου να αρχικοποιήσει έναν πυρήνα, να δημιουργήσει εύκολα ιεραρχίες `framework/base-kernel` και να μπορεί να εκτελεί γενικότερες πρόσθετες εξωτερικές λειτουργίες που χρησιμοποιούν τον πίνακα πυρήνα (π.χ. να υπολογίσει προσεγγίσεις του, όπως η προσέγγιση Nyström), ένα αντικείμενο σχεδιάστηκε στο σχεδιαστικό πρότυπο του decorator ονόματι `GraphKernel`.

### 3.1.3 Γενική Μορφή Εισόδου

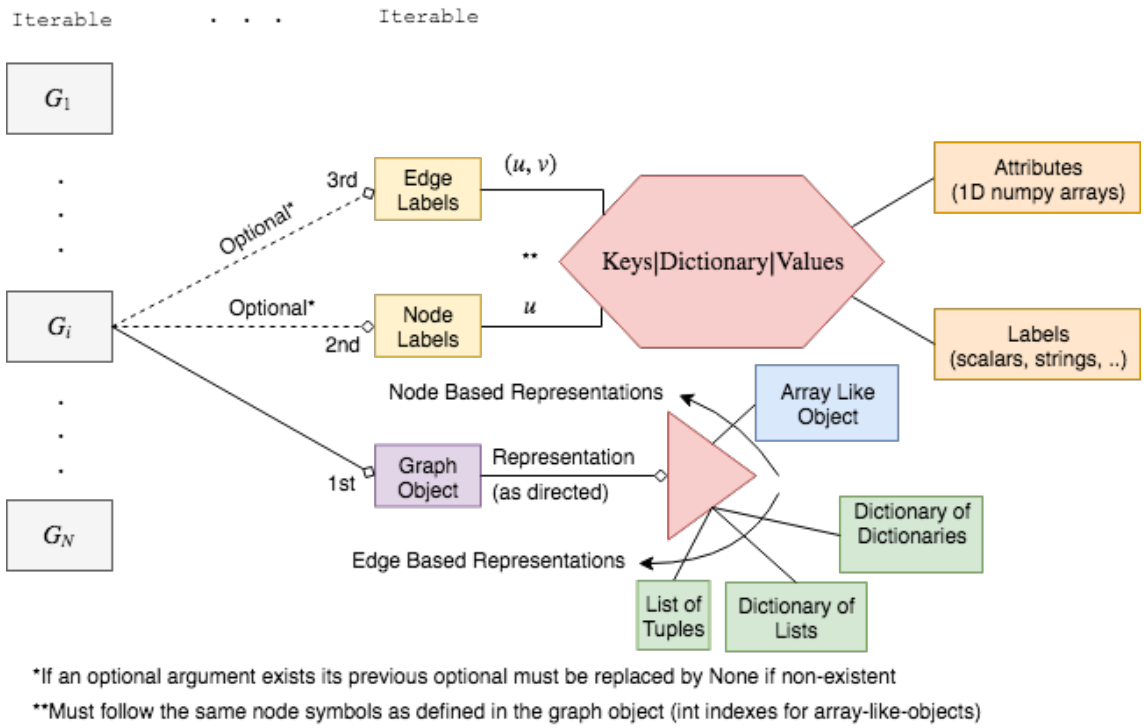
Η ανάγκη σχεδιαστικής ενοποίησης όλων των πυρήνων οδήγησε και στην ανάγκη δημιουργίας ενός προτύπου αναπαράστασης της εισόδου των μεθόδων `fit`, `fit_transform` και `transform` καθενός αντικειμένου τύπου `Kernel`. Ακολουθώντας το πρότυπο του scikit-learn για άλλους `Transformer` όπως ο `tf-idf`, η είσοδος θεωρήθηκε σαν ένας *graph vectorizer* ή αλλιώς ένα `Iterable` από γράφους (παράβαλε σχήμα 3.1). Κάθε γράφος μπορεί να αναπαρασταθεί από ένα `Iterable` τουλάχιστον ενός και το πολύ τριών στοιχείων.
Πρώτο στοιχείο κάθε γράφου, είναι ένα αντικείμενο που αναπαριστά τον γράφο ως δομή. Οι υπάρχουσες αναπαραστάσεις γράφων στην βιβλιογραφία χωρίζονται σε αυτές που βασίζονται

στις ακμές του γράφου (1) και σε αυτές που βασίζονται στους κόμβους (2). Οι πρώτες, μπορούν να είναι από μία λίστα κόμβων μέχρι ένα λεξικό, ενώ οι δεύτερες περιγράφονται κυρίως με έναν πίνακα γειτνίασης. Ως δεύτερο στοιχείο μπορούμε να έχουμε ένα λεξικό που αναπαριστά τις επισημειώσεις των κόμβων του γράφου. Οι επισημειώσεις μπορούν να είναι είτε σύμβολα (πχ. αριθμός ή συμβολοσειρά) είτε διανύσματα πραγματικών τιμών (χαρακτηριστικά, παράβαλε ορισμό 2.3). Τρίτο και τελευταία στοιχείο είναι ένα λεξικό μεταξύ ζευγαριών κόμβων για όλες τις υπάρχουσες ακμές, είτε συμβολικών επισημειώσεων, είτε διανυσμάτων πραγματικών τιμών. Τα δύο τελευταία στοιχεία μπορούν να παραληφθούν ή να αντικατασταθούν από κενά ορίσματα (τύπου `None`) στην περίπτωση που δεν υπάρχουν.

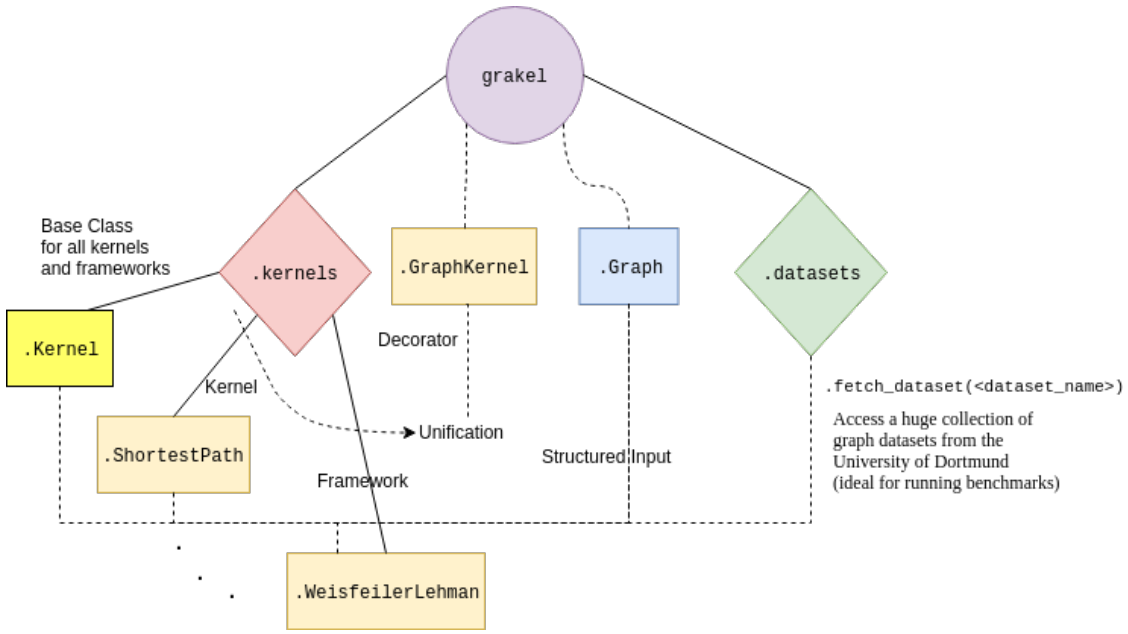
Ο τρόπος εσωτερικής αναπαράστασης των γράφων, φάνηκε ιδιαίτερα σημαντικός, όσον αφορά την υλοποίηση ενός πυρήνα. Πολλοί πυρήνες όπως ο *πυρήνας τυχαίων περιπάτων* (βλέπε υποενότητα 2.6.1) χρησιμοποιούν άμεσα τον πίνακα γειτνίασης, ενώ πυρήνες όπως ο *πυρήνας κοντινότερων μονοπατιών* (βλέπε υποενότητα 2.6.2), υπολογίζονται ταχύτερα αν έχουμε αναπαράσταση ακμών. Άλλοι πυρήνες, όπως για παράδειγμα ο ODD-STh χρησιμοποιούν δευτερεύουσα πληροφορία του γράφου που αν κωδικοποιηθεί σωστά εξάγεται γρηγορότερα σε μία από τις δύο αναπαραστάσεις. Συνεπώς χρειαζόταν ένας τρόπος, προκειμένου ταυτόχρονα η αναπαράσταση των γράφων να χρειάζεται την ελάχιστη δυνατή μνήμη και να μας δίνεται η δυνατότητα είτε να επιλέγουμε είτε να αδιαφορούμε για το είδος της εσωτερικής τους αναπαράστασης κατά τη χρήση τους τόσο σε συναρτήσεις μεταξύ γράφων όσο και όσον αφορά την ανάπτυξη ενός πυρήνα. Ως αποτέλεσμα δημιουργήθηκε η κλάση `grakel.Graph` η οποία ενοποίησε την είσοδο του χρήστη σε δύο βασικές εσωτερικές αναπαραστάσεις των οποίων την ύπαρξη (ή συνύπαρξη) καθορίζει ο προγραμματιστής (των πυρήνων) και από τις οποίες μπορεί να εξάγει χρήσιμη πληροφορία (για τους πυρήνες), χωρίς να ασχολείται αναγκαστικά με την μορφή των δεδομένων εισόδου και κατ΄ επέκταση της ίδιας της εσωτερικής τους αναπαράστασης. Παρόλο που μία τέτοια σχεδιαστική προσέγγιση φαίνεται πολύ απλοϊκή, προτιμήθηκε σε σχέση με την χρήση μίας υπάρχουσας βιβλιοθήκης όπως η `networkx`, μίας και χρειαζόταν για την επίλυση ενός στενά καθορισμένου προβλήματος με το λιγότερο δυνατό κόστος.

Σημαντικό κομμάτι της ίδιας της ανάπτυξης του λογισμικού είναι η δυνατότητα εκτέλεσης benchmarks. Για το λόγο αυτό υπήρχε η ανάγκη παροχής υπηρεσιών εύκολης εισαγωγής γνωστών dataset που χρησιμοποιούνται στο χώρο των πυρήνων γράφων. Για να καλύψουμε αυτήν την ανάγκη οδηγηθήκαμε στην ανάπτυξη μίας συνάρτησης εν ονόματι `fetch_dataset` σε ένα υποπακέτο του `grakel` με το όνομα `datasets`. Η συνάρτηση αυτή είναι υπεύθυνη για το κατέβασμα (downloading), την φόρτωση (loading) και την τοπική απόθεση (caching) ενός dataset από μία τεράστια συλλογή όπως αυτή συντηρείται και ελέγχεται από την ερευνητική ομάδα για τους πυρήνες γράφων στο πανεπιστήμιο του Dortmund [46]. Κάθε dataset αποθηκεύεται σε μία τοπική διεύθυνση, ώστε να μπορεί να χρησιμοποιηθεί στο μέλλον χωρίς την ύπαρξη σύνδεσης στο διαδίκτυο.

Η συνολική οργάνωση του grakel όπως περιγράφηκε παραπάνω, συνοψίζεται στο σχήμα 3.2.




Σχήμα 3.1: Σχηματική απεικόνιση του τρόπου αναπαράστασης της εισόδου για τις μεθόδους `fit`, `fit_transform` και `transform` κάθε αντικειμένου τύπου `Kernel`




Σχήμα 3.2: Σχηματική απεικόνιση της οργάνωσης του λογισμικού grakel. Οι ρόμβοι αναπαριστούν υποπακέτα (submodules) ενώ τα παραλληλόγραμμα κλάσεις.

## 3.2 Ανάπτυξη ενός Πυρήνα: Η κλάση `Kernel`

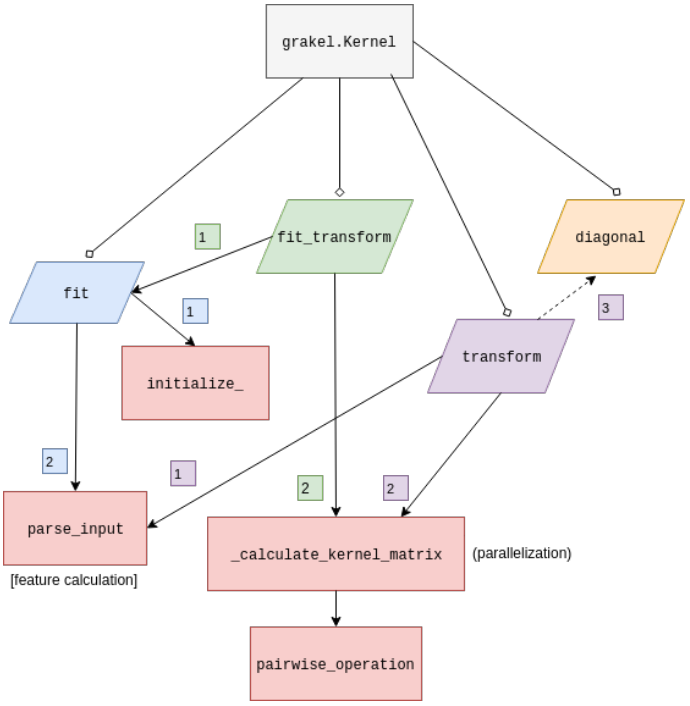
Ας δούμε τώρα πιο αναλυτικά την σχεδίαση της κλάσης `Kernel` που αποτελεί την κύρια και σημαντικότερη οντότητα αυτής της βιβλιοθήκης (παράβαλε σχήμα 3.3). Από την μελέτη της σχετικής βιβλιογραφίας ο υπολογισμός ενός γράφου πυρήνα μπόρεσε να αποδομηθεί στα εξής δύο αφηρημένα βήματα: (1) ανάγνωση της εισόδου και εξαγωγή χαρακτηριστικών και (2) υπολογισμός του πίνακα πυρήνα.

### 3.2.1 Η Μέθοδος `fit`

Κατά την κλήση της μεθόδου `fit` η βασική συνάρτηση η οποία καλείται είναι η `parse_input` σχεδιασμένη για να φέρει εις πέρας το (1) και να αποθηκεύσει τα χαρακτηριστικά σε μία ιδιότητα της κλάσης. Παράλληλα προκειμένου η κλάση `Kernel` να κληροομεί σωστά τον `BaseEstimator` ήταν αναγκαίο η μέθοδος `set_params` να δουλεύει αποτελεσματικά, έπειτα από την αρχικοποίηση ενός αντικειμένου, ενώ ταυτόχρονα στην μέθοδο `__init__` όλες οι παράμετροι εισόδου έπρεπε να αρχικοποιούν ιδιότητες με το ίδιο όνομα, για να δουλεύει η μέθοδος `get_params`. Ο έλεγχος μεταβλητών και η αρχικοποίηση δευτερευόντων χαρακτηριστικών ανατέθηκε στην συνάρτηση `initialize_` η οποία καλείται στην πρώτη γραμμή της μεθόδου `fit`.

### 3.2.2 Η Μέθοδος `fit_transform`

Για τον υπολογισμό της μήτρας πυρήνα είναι υπεύθυνες δύο μέθοδοι:
(1) η `_calculate_kernel_matrix` και (2) η `pairwise_operation`. Στην περίπτωση του `fit_transform` η πρώτη μέθοδος καλείται στο σύνολο των δεδομένων που έχουν αποθηκευθεί μετά την κλήση της μεθόδου `parse_input` σε μία εσωτερική ιδιότητα της κλάσης `self.X`. Τα δεδομένα αναμένεται να έχουν την μορφή ενός `Iterable`, το οποίο σε κάθε στοιχείο του περιέχει τα εξαχθέντα χαρακτηριστικά που αφορούν τον κάθε πυρήνα. Χρησιμοποιώντας αυτά τα χαρακτηριστικά υπολογίζουμε την μήτρα πυρήνα, εφαρμόζοντας την μέθοδο `pairwise_operation` μεταξύ κάθε ζευγαριού της άνω διαγωνίου (καθώς η μήτρα πυρήνα είναι πάντα συμμετρική). Συνεπώς ο υπολογισμός της μήτρας πυρήνα κατανέμεται μεταξύ του υπολογισμού χαρακτηριστικών και της εφαρμογής μίας μετρικής μεταξύ τους. Στις ακραίες περιπτώσεις (όπως συμβαίνει π.χ. με τα *R-frameworks*) τα χαρακτηριστικά είναι τέτοια που ο υπολογισμός της μήτρας πυρήνα μπορεί να είναι ισοδύναμος με ένα γινόμενο πινάκων. Όταν συμβαίνει κάτι τέτοιο, κατά την ανάπτυξη του πυρήνα είναι προτιμότερο να παραληφθεί η μέθοδος `pairwise_operation` και να επανεγγραφεί η μέθοδος `_calculate_kernel_matrix`. Όσον αφορά την μέση περίπτωση που το υπολογιστικό κόστος κατανέμεται μεταξύ του συνήθως σειριακού `parse_input` και του `pairwise_operation` μία θεμελιώδης μέθοδος παραλληλοποίησης υλοποιήθηκε για τον υπολογισμό του πίνακα πυρήνα. Συγκεκριμένα δεδομένου του πλήθους των στοιχείων της άνω διαγωνίου, ίσο με $\frac{N(N+1)}{2}$ και ενός πλήθους παράλληλων εργασιών `n_jobs`, αν χωρίσουμε ομοιόμορφα τη λίστα δεικτών $K = [0, \ldots, \frac{N(N+1)}{2} - 1]$, μπορούμε έπειτα να πάρουμε τους δείκτες του ζητούμενου ζευγαριού γράφων $(i, j)$ που αντι-




Σχήμα 3.3: Σχηματική απεικόνιση του τρόπου οργάνωσης των μεθόδων της κλάσης `Kernel`. Τα νούμερα συμβολίζουν την σειρά κλήσης άλλων μεθόδων από την εκάστοτε μέθοδο. Η διακοπτόμενη κλήση αφορά την περίπτωση που κατά την αρχικοποίηση η παράμετρος normalization είναι `True`.

στοιχούν σε ένα $k \in K$ ως:

$$i = \left\lfloor N - 1 - \left\lfloor \frac{\sqrt{4N(N+1) - 8k - 7} - 1}{2} \right\rfloor \right\rfloor \tag{3.3}$$

$$j = \left\lfloor k + i - \left\lfloor \frac{N(N+1)}{2} \right\rfloor + \left\lfloor \frac{(N-i)(N-i+1)}{2} \right\rfloor \right\rfloor \tag{3.4}$$

Η υπολογιστική ωφελιμότητα της παραλληλοποίησης εξαρτάται από το υπολογιστικό κόστος της μεθόδου `pairwise_operation` (σε σχέση με τις υπόλοιπες), πράγμα που μας απαγορεύει να την εγγυηθούμε στην γενική περίπτωση.

Στην περίπτωση που ο χρήστης επιθυμεί μία κανονικοποιημένη μήτρα πυρήνα η μέθοδος `fit_transform` διαιρεί στοιχείο προς στοιχείο τις τιμές του πίνακα με την τετραγωνική ρίζα κάθε στοιχείου του εξωτερικού γινομένου της διαγωνίου με τον εαυτό της. Η διαγώνιος του πίνακα πυρήνα αποθηκεύεται σε κάθε περίπτωση σε μία εσωτερική μεταβλητή προκειμένου να μπορεί να χρησιμοποιηθεί χωρίς να επανυπολογιστεί κατά την κλήση της μεθόδου `diagonal`.

### 3.2.3 Η Μέθοδος `transform`

Όσον αφορά τώρα την περίπτωση του `transform` όλες οι μέθοδοι που αναφέρθηκαν παραπάνω θα πρέπει να προσαρμοστούν. Συγκεκριμένα σε πρώτη φάση, το `parse_input` καλείται να σχηματίσει χαρακτηριστικά συγκρίσιμα με τα δεδομένα του `fit`. Σε πολλές περιπτώσεις κάτι

τέτοιο απαιτεί την αποθήκευση μετα-πληροφορίας κατά το `fit` (π.χ. ενός λεξικού που λειτουργεί ως αρίθμηση των επισημειώσεων) αλλά και της δυνατότητας του `parse_input` να γνωρίζει αν καλείται από το `transform`, το `fit` ή και σε κάποιες περιπτώσεις από το `fit_transform`. Κάτι τέτοιο επιλύεται φυσικά με την χρήση μία ιδιωτικής ιδιότητας `self._method_calling` της κλάσης, που λαμβάνει τιμές $1, 2, 3$ κατά τα `fit`, `fit_transform` και `transform` αντίστοιχα. Εν συνεχεία κατά την κλήση της συνάρτησης `_calculate_kernel_matrix` μεταξύ όλων των ζευγαριών των δεδομένων του `transform` και του `fit`, τρέχουμε την μέθοδο `pairwise_operation`. Σε αυτήν την περίπτωση, δεδομένου ενός πλήθους παράλληλων εργασιών `n_jobs` η παραλληλοποίηση επιτυγχάνεται αν χωρίσουμε ομοιόμορφα την λίστα δεικτών $K = [0, \ldots, NM-1]$ (όπου $M$ το πλήθος των δεδομένων κατά το `transform`), και έπειτα για κάθε $k \in K$ εξάγουμε για κάθε επεξεργαστή τα ζευγάρια δεικτών $(i, j)$ ως $(k \mod N, \lfloor \frac{k}{N} \rfloor)$.

Στην περίπτωση που ο χρήστης επιθυμεί μία κανονικοποιημένη μήτρα πυρήνα η μέθοδος `transform` καλεί την μέθοδο `diagonal` η οποία επιστρέφει το διάνυσμα της τιμής πυρήνα όλων των στοιχείων του `fit` με τον εαυτό τους $d_x$, καθώς και το διάνυσμα της τιμής πυρήνα όλων των στοιχείων του `transform` με τον εαυτό τους $d_y$. Προκειμένου να προκύψει ο κανονικοποιημένος πίνακας, διαιρεί στοιχείο προς στοιχείο της $M \times N$ μήτρας πυρήνα με την τετραγωνική ρίζα κάθε στοιχείου του εσωτερικού γινομένου $d_y \times d_x$.

## 3.3 Packaging

Έκτος από την ίδια την σχεδίαση (design) και την ανάπτυξη (development) της, η ολοκλήρωση μίας σύγχρονης βιβλιοθήκης προγραμματισμού απαιτεί την *συσκευασία* της (packaging). Αν μία βιβλιοθήκη μπορεί να παρομοιαστεί με ένα μηχανισμό, σε επίπεδο ανάπτυξης ένας προγραμματιστής πρέπει να μπορεί να αναγνωρίσει τα δομικά της μέρη, να μπορεί να καταλάβει από τι αποτελούνται, πως κατασκευάζονται και τον τρόπο με τον οποίο συναρμολογούνται, καθώς και να μπορεί να ανιχνεύσει τις αλλαγές της από το παρελθόν, ενώ δύναται γνωρίζοντας λεπτομερώς μέρη από την παρούσα μορφή της να την διορθώσει ή να την επεκτείνει. Σε επίπεδο χρήσης, η γνώση της έκδοσης της και η γενική *επαλήθευση λειτουργίας* της, η ευκολία εγκατάστασης της, η δυνατότητα ελέγχου λειτουργίας χωρίς την γνώση χρήσης της, καθώς και η διάθεση ενός εγχειριδίου που περιέχει πληροφορίες σχετικές με την εγκατάσταση, την χρήση και την κατασκευή της είναι εξίσου απαραίτητα. Παρόλο που συχνά στο εύρος των ατόμων που απευθύνονται τέτοιες βιβλιοθήκες οι δύο αυτοί ρόλοι, *προγραμματιστή* και *χρήστη* είναι ζήτημα ¨εστίασης της προσοχής¨, είναι χρήσιμο να συντηρούνται ως πόλοι ανάπτυξης του λογισμικού.

### 3.3.1 Ανάπτυξη Κώδικα

Ξεκινώντας από την ανάπτυξη κώδικα σημαντικό είναι να ξεκινήσουμε με τους βασικούς κανόνες συγγραφής του.

#### 3.3.1.1 Το Πρότυπο PEP-8

Προκειμένου ο κώδικας να είναι ευανάγνωστος, η ανάπτυξη κάθε πακέτου προγραμματισμού python καλείται να ακολουθεί συγκεκριμένους 'αισθητικούς' κανόνες τόσο στο επίπεδο της σύνταξης όσο και της σημασιολογίας. Το πρότυπο PEP-8 είναι το παρόν πρότυπο σύνταξης για την γλώσσα προγραμματισμού Python. Περιέχει κανόνες που αφορούν το μέγιστο δυνατό μήκος μίας γραμμής κώδικα και την στοίχιση των ορισμάτων κατά την κλήση ή τον ορισμό μίας συνάρτησης, που επεκτείνονται πέραν της μίας γραμμής, μέχρι κανόνες για τον τρόπο ελέγχου ενός τύπου, τον τρόπο χειρισμού εξαιρέσεων (exception handling) και την χρήση συναρτήσεων αντί για συναρτησιακών (lamda's) [34]. Το πρότυπο αυτό μπορεί να παραμετροποιείται από τον εκάστοτε προγραμματιστή, αλλά εξασφαλίζει μία συνοχή στον τρόπο με τον οποίο τελικά συντάσσει κώδικα. Για τον αυτόματο έλεγχο του παραπάνω, έχει αναπτυχθεί ένα αντίστοιχο πακέτο γνωστό ως flake8.

#### 3.3.1.2 PyPI

Κάθε πακέτο python απαιτεί να μπορεί να εγκατασταθεί οικουμενικά σε όλο το σύνολο των μηχανημάτων για τα οποία προορίζεται. Κάτι τέτοιο επιλύεται μέσω της ίδιας της python (βλ. βιβλιοθήκη setuptools) και απαιτεί από τον προγραμματιστή μονάχα ένα στοιχειώδη τρόπο οργάνωσης της βιβλιοθήκης, καθώς και την συγγραφή ενός αρχείου εγκατάστασης setup.py. Διαθέτοντας κάτι τέτοιο η βιβλιοθήκη μπορεί να τοποθετηθεί στο ηλεκτρονικό αποθετήριο βιβλιοθηκών της Python, γνωστό ως PyPI: **Py**thon **P**ackage **I**ndex, μέσω του οποίου μπορεί να εγκατασταθεί από το κύριο εργαλείο εγκατάστασης βιβλιοθηκών, γνωστό ως pip. Η python ως γλώσσα με διερμηνέα δεν διαθέτει την έννοια των εκτελεσίμων όπως αυτή υπάρχει από την C (binaries) ή την Java (bytecode), ταυτίζοντας το πακέτο εκτέλεσης με τον ίδιο τον κώδικα, μέσω των λεγόμενων *eggs*. Για να καλυφθεί αυτή και άλλες ανάγκες, τα λεγόμενα wheels εισήχθησαν από την κοινότητα που αναπτύσσει την γλώσσα python.

Περαιτέρω, κάθε σύγχρονη βιβλιοθήκη επιστημονικού υπολογισμού (scientific-computing) καλείται να μπορεί συνταιριάξει, το ευχάριστο και μή-περιοριστικό προγραμματιστικό της περιβάλλον, με τα άγρια και άκομψα πλάσματα των γλωσσών C, C++, Fortran λόγω της αμεσότητας και της αποδοτικότητάς τους. Τόσο στο επίπεδο του setup.py όσο και στο επίπεδο των wheels κάτι τέτοιο δεν αποτελεί μία δυσπρόσιτη πρακτική, ειδικά με την χρήση πακέτων όπως το Cython. Τέλος σημαντικό για τον έλεγχο κάθε πακέτου είναι η ύπαρξη ενός συνόλου από στοιχειώδη δοκιμαστικά προγράμματα (unit-tests) προκειμένου πριν την δημοσίευση του κώδικα, αλλά και κατά την συντήρηση και επέκταση του να επαληθεύεται η σωστή λειτουργία του. Πακέτα όπως το nose έχουν αναπτυχθεί, προκειμένου η εκτέλεση και η καταγραφή των προβλημάτων που προκύπτουν από την εκτέλεση τους να παρουσιάζεται συνοπτικά, ενώ ενσωματώνουν άλλα όπως το coverage που είναι υπεύθυνο να παρουσιάζει στατιστικά στοιχεία, σχετικά με τον βαθμό στον οποίο τα δοκιμαστικά προγράμματα δοκιμάζουν τον πλήρη κώδικα της βιβλιοθήκης.

### 3.3.2 Δημοσίευση Κώδικα

Όπως προαναφέρθηκε για την δημοσίευση του κώδικα της παρούσας βιβλιοθήκης χρησιμοποιήθηκε το ηλεκτρονικό αποθετήριο GitHub, παράλληλα με το γεγονός ότι οι εκδόσεις του καταγράφονται μέσω του συστήματος git. Απαραίτητα όμως για την δημοσίευση του παρόντος λογισμικού επιστημονικού υπολογισμού python είναι η ύπαρξη Documentation, η συνεχή του ενσωμάτωση (Continuous Integration) καθώς και η άδεια του.

#### 3.3.2.1 Documentation

Ίσως το πιο σημαντικό βήμα τόσο όσον αφορά τον χρήστη, αλλά και τόσο όσον αφορά τον προγραμματιστή που αναπτύσσει την βιβλιοθήκη, είναι η συγγραφή ενός εγχειριδίου γνωστό με τον όρο *documentation*. Από την ίδια την αναλυτική καταγραφή των κλάσεων, την όμορφη παρουσίαση του κώδικα και τις οδηγίες εγκατάστασης, ένα εγχειρίδιο μπορεί να αποτελείται από πολλά περισσότερα μέρη όπως εισαγωγικό κείμενο για την χρήση της βιβλιοθήκης, θεωρητική ανάλυση των απαραίτητων μεθόδων της, πληροφορίες σχετικά με την επέκταση της, και πιο ενδιαφέροντα παραδείγματα χρήσης της. Η πραγμάτωση όλων των παραπάνω αυτοματοποιείται όσον αφορά την παρουσίαση των κλάσεων και του κώδικα (μέσω κατάλληλης σύνταξης στο επίπεδο των σχολίων) και διευκολύνεται όσον αφορά τα υπόλοιπα, παράγοντας ένα ευχάριστο αισθητικό αποτέλεσμα μέσω του πακέτου Sphinx.

#### 3.3.2.2 Συνεχής Ενσωμάτωση

Κάθε νέα έκδοση ενός λογισμικού η οποία προστίθεται στο ηλεκτρονικό αποθετήριο `GitHub` πρέπει να συνδέεται με μία εγγύηση ότι αυτή η έκδοση είναι λειτουργική. Κάτι τέτοιο αποτελεί μία καθιερωμένη πρακτική τα τελευταία χρόνια, μέσω των πλατφορμών συνεχούς ενσωμάτωσης (continuous integration) οι οποίες επιτρέπουν την ολιγόχρονη αρχικοποίηση ενός λειτουργικού συστήματος και τον προγραμματισμό μία σειράς βημάτων, μέσω της οποίας μπορούν να ελέγχονται αυτόματα αν έρχονται εις πέρας, η κατάλληλη παραμετροποίηση του συστήματος, η εγκατάσταση και η δοκιμή της βιβλιοθήκης, κάθε φορά που μία νέα έκδοση προστίθεται στο ηλεκτρονικό αποθετήριο. Συγκεκριμένα ο έλεγχος της βιβλιοθήκης GraKeL γίνεται σε λειτουργικό σύστημα Linux και OSX μέσω της πλατφόρμας συνεχούς ενσωμάτωσης Travis και σε λειτουργικό σύστημα Windows μέσω της πλατφόρμας Appveyor. Μιας και αυτά τα περιβάλλοντα είναι στοιχειώδη (minimal) και η βιβλιοθήκη κάτι αφηρημένο σε σχέση με το ίδιο το λειτουργικό σύστημα η εγγύηση λειτουργίας τους σε όλες τις υποστηριζόμενες εκδόσεις Python είναι συνήθως ανεξάρτητη από την έκδοση του λογισμικού (παρόλο που δίνεται η δυνατότητα ορισμού του). Επιπλέον οι πλατφόρμες αυτές χρησιμοποιήθηκαν για την ανάπτυξη και απόθεση (build and deploy) wheels για ένα εύρος συστημάτων και όλες τις υποστηριζόμενες εκδόσεις python στο PyPi, μέσω της βιβλιοθήκης cibuildwheel. Τέλος, μία τρίτη πλατφόρμα χρησιμοποιήθηκε για την ανάπτυξη και απόθεση του documenation, συγκεκριμένα η Circle-CI. Η λειτουργικότητα τις βιβλιοθήκης όπως επισημαίνεται από τα παραπάνω καθώς και το ποσοστό του coverage όπως αυτό αποτίθεται στην πλατφόρμα codecov μέσω του travis, φαίνονται με την μορφή badges στην κύρια σελίδα στο αποθετήριο του GraKeL.

#### 3.3.2.3 Άδεια

Είναι καθιερωμένο για κάθε επίσημα δημοσιοποιημένο λογισμικό να κατέχει μία άδεια χρήσης. Για την δημοσίευση του GraKeL επιλέχθηκε η ίδια άδεια χρήσης με αυτή του scikit-learn, συγκεκριμένα η άδεια BSD 3 ρητρών. Η άδεια αυτή είναι μία άδεια αποδεκτή από την κοινότητα ελευθέρου λογισμικού (FSF-approved), με τρία στοιχειώδη απαιτούμενα. Συγκεκριμένα, την επανατοποθέτηση αυτής της άδειας σε αναδιανομές του λογισμικού τόσο αν αυτές είναι σε μορφή κώδικα ή εκτελέσιμου, καθώς και την διαφύλαξη των μελών του προσώπου δικαίου που φέρει τα πνευματικά δικαιώματα, από την χρήση των ονομάτων τους για την πρόκριση ή την προώθηση παραγώγων αυτού του λογισμικού, χωρίς να έχει προηγηθεί η γραπτή τους άδεια.

# Κεφάλαιο 4

# Πειραματική Αξιολόγηση

Στο κεφάλαιο αυτό, θα παρουσιάσουμε μία πειραματική αξιολόγηση του λογισμικού grakel. Στο πρώτο μέρος θα περιγράψουμε την πειραματική διάταξη, συγκεκριμένα πιο είναι το πείραμα με το οποίο θα αξιολογήσουμε καθώς και την μετρική μέσω της οποίας θα αξιολογήσουμε συγκριτικά την απόδοση κάθε πυρήνα. Στη συνέχεια θα παρουσιάσουμε τα σύνολα δεδομένων στα οποία θα εκτελέσουμε τα πειράματα, χωρίζοντας τα σε κατηγορίες με βάση το είδος της πληροφορίας των γράφων που παρέχονται. Έπειτα θα παρουσιάσουμε τα αποτελέσματα των πειραμάτων, μαζί με μία σύντομη 'εμπειρική' αξιολόγηση. Στο τέλος, θα παρουσιάσουμε κάποια θεωρητικά και πρακτικά συμπεράσματα, που προκύπτουν από την αξιολόγηση του λογισμικού.

## 4.1 Πειραματική Διάταξη

Για την αξιολόγηση του grakel εκτελέσαμε σε μεμονωμένους πυρήνες ενός cluster[1] αποτελούμενου από 80 'Intel® Xeon® CPU E7- 4860 @ 2.27GHz' και 8 board σύγχρονης μνήμης DDR3 - 1067 MHz συνολικού μεγέθους '1TB', τον υπολογισμό του πλήρους kernel πίνακα μέσω της μεθόδου fit_transform σε ένα εύρος τιμών (βλέπε πίνακα 4.3) και μία σειρά από συνόλων δεδομένων. Ένα όριο τοποθετήθηκε στο μέγιστο χρόνο εκτέλεσης κάθε υπολογισμού καθώς και στην μέγιστη μνήμη RAM που μπορούσε να χρησιμοποιηθεί. Συγκεκριμένα για όλους τους υπολογισμούς τοποθετήθηκε το όριο της μίας μέρας (συμβολίζεται ως `OOT`) και των 64GB (συμβολίζεται ως `OOM`). Ταυτόχρονα για την έγκυρη σύγκριση των πυρήνων ο μέγιστος αριθμός από threads που χρησιμοποίησε η βιβλιοθήκη BLASS ορίστηκε ίσος με 1. Σε όλους τους αλγορίθμους για τους οποίους η επαύξηση μίας παραμέτρου αύξανε ή κρατούσε σταθερή την πολυπλοκότητα μνήμης και υπολογισμού μία τιμή προς δοκιμή αγνοήθηκε στην περίπτωση που για την προηγούμενη υπήρξε `OOT` ή `OOM`. Στη συνέχεια έχοντας κρατήσει τις επισημειώσεις κάθε στοιχείου της μήτρας πυρήνα επιχειρήσαμε 10-fold cross validation σε ένα ταξινομητή SVM με βάση την μετρική της ευστοχίας (βλ. 4.1.1). Συγκεκριμένα χρησιμοποιήσαμε τον ταξινομητή `sklearn.svm.SVC` που μας δίνει την δυνατότητα να λύσουμε

[1] Θα ήθελα να ευχαριστήσω τον καθηγητή Απόστολο Παπαδόπουλο, για την προθυμία, την υποστήριξη και την παραχώρηση πρόσβασης και χρήσης στο μηχάνημα `hyperion` του Αριστοτελείου Πανεπιστημίου Θεσσαλονίκης, για την εκτέλεση των πειραμάτων του grakel.

το πρόβλημα SVM παρέχοντας μία προϋπολογισμένη μήτρα Gram. Κάθε fold, αφορά τον χωρισμό των δεδομένων σε δύο μέρη 90% και 10%, γνωστά ως train set και test set. Τα folds ήταν κοινά για όλους τους πυρήνες που εκτελέστηκαν σε αυτό το dataset. Για καθένα από τα 10 fold, χωρίζουμε το δείγμα εκπαίδευσης σε δύο μέρη με μεγέθη 90%/10% , γνωστά ως (validation train/test set) και στα οποία για ένα εύρος τιμών $C$, συγκεκριμένα το $\{10^{-7}, 10^{-5}, .., 10^{5}, 10^{7}\}$ (για το $C$ βλέπε εξίσωση 2.11) εκπαιδεύουμε το SVM στο validation train set. Για τον συνδυασμό παραμετροποίησης και $C$ που μεγιστοποιεί την μετρική ευστοχίας στο validation test set αποθηκεύουμε την τιμή της μετρικής ευστοχίας που προκύπτει από την εκπαίδευση του SVM στο train-set κατά την πρόβλεψη των τιμών στο test-set. Υπολογίσουμε σαν συνολική τιμή της μετρικής ευστοχίας την μέση τιμή τους για όλα τα fold, υπολογίζουμε την μέση τιμή και την διακύμανση αυτών των τιμών, για 10 επαναλήψεις. Τέλος καταγράφουμε με τον ίδιο τρόπο σε συγκριτικούς πίνακες για κάθε dataset την μνήμη, τον χρόνο, ως μέση τιμή και διακύμανση από τις μέσες τιμές όλων της μνήμης και του χρόνου εκτέλεση για τους πυρήνες εκείνων των παραμετροποιήσεων που μεγιστοποιούν την τιμή της μετρικής ευστοχίας στο validation test set καθενός fold.

### 4.1.1 Μετρική Ευστοχίας

Για την αξιολόγηση των πυρήνων αναφέραμε πως χρησιμοποιούν την μετρική της ευστοχίας. Συγκεκριμένα για ένα πρόβλημα δυαδικής ταξινόμησης με θετικά και αρνητικά δείγματα υπάρχουν τέσσερεις δυνατές προβλέψεις:

1. True Positive (TP) - το σύστημα προβλέπει σωστά μία θετική κλάση για ένα παράδειγμα που είναι θετικό
2. True Negative (TN) - το σύστημα προβλέπει σωστά μία αρνητική κλάση για ένα αρνητικό παράδειγμα
3. False Positive (FP) - το σύστημα προβλέπει σωστά μία λανθασμένη κλάση για ένα λανθασμένο παράδειγμα
4. False Negative (FN) - το σύστημα προβλέπει λανθασμένα μία θετική κλάση για ένα αρνητικό παράδειγμα

Αυτή η πληροφορία συνήθως παρουσιάζεται σε ένα $2 \times 2$ πίνακα *σύγχυσης* (confusion matrix), όπως απεικονίζεται στον πίνακα 2.2. Στη βάση των τεσσάρων παραπάνω προβλέψεων προκύπτουν διάφορες πολύ γνωστές μετρικές αξιολόγησης. Αυτές οι μετρικές μετρούν ποσοτικά την επίδοση ταξινόμησης για μία μόνο μέθοδο σε ένα και μόνο σύνολο δεδομένων. Στην ταξινόμηση γράφων, η πιο γνωστή μετρική είναι αυτή της ευστοχίας (accuracy), που ορίζεται ως:

$$acc = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.1}$$

Η ευστοχία υπολογίζει μία μέθοδο βάσει του τμήματος των προβλέψεων τις που είναι σωστές. Το κύριο μειονέκτημα της ευστοχίας είναι ότι στην περίπτωση μη ισορροπημένων κατανομών κατηγορίας, μπορεί να πάρει τεχνητά υψηλές τιμές. Για παράδειγμα, αν σε ένα πρόβλημα

| | | Predicted (Class) | |
|---|---|---|---|
| | | Positive | Negative |
| Actual | Positive | TP | FN |
| | Negative | FP | TN |

Πίνακας 4.1: Πίνακας σύγχυσης για ένα πρόβλημα δυαδικής ταξινόμησης.

δυαδικής ταξινόμησης το 99% των παραδειγμάτων είναι θετικά, τότε ένας αλγόριθμος μπορεί να πετύχει 99% ευστοχία προβλέποντας μόνο την θετική κατηγορία! Για την επίλυση αυτού του προβλήματος, έχουν προταθεί άλλες μετρικές αξιολόγησης.
Από την άλλη, όπως φαίνεται στον πίνακα 4.1, η πλειοψηφία των συνόλων δεδομένων που χρησιμοποιείται στην ταξινόμηση γράφων είναι σύνολα δεδομένων δυαδικής ταξινόμησης και στις περισσότερες περιπτώσεις, οι κλάσεις είναι ισορροπημένες. Λόγω αυτής της παρατήρησης και προκειμένου τα αποτελέσματα να είναι συγκρίσιμα με προηγούμενες μελέτες, χρησιμοποιήσαμε την ευστοχία ως μέτρο αξιολόγησης.

### 4.1.2 Παραμετροποίηση Πυρήνων

Στο σύνολο τους οι πυρήνες μπορούν να διαχωριστούν με βάση το είδος γράφων που δέχονται, όπως φαίνεται στον πίνακα 4.2. Συνεπώς για κάθε κατηγορία συνόλου δεδομένων εκτελέσαμε ένα διαφορετικό σύνολο πυρήνων. Συγκεκριμένα, όλοι οι πυρήνες μπορούν να εκτελεστούν σε σύνολα δεδομένων χωρίς επισημειώσεις και όλοι οι πυρήνες που δέχονται συνεχείς επισημειώσεις μπορούν να εκτελεστούν σε σύνολα δεδομένα διακριτών (βλέπε υποενότητες 4.3.2, 4.3.3 αντίστοιχα). Προκειμένου να υπολογιστεί η βέλτιστη ευστοχία, με την μέθοδο που παρουσιάστηκε στην πειραματική διάταξη, υπολογίσαμε για κάθε πυρήνα και σύνολα δεδομένων τα αποτελέσματα τους σε ένα εύρος παραμέτρων 4.3. Οι τιμές των παραμέτρων επιλέχθηκαν, τόσο βάσει των τιμών που αναφέρονται στην βιβλιογραφία κατά την πειραματική τους αξιολόγηση, όσο και μέσω εμπειρικών κανόνων και εκτιμήσεων που έχουν προκύψει από την χρήση τους.

## 4.2 Datasets

Η παραπάνω πειραματική διάταξη εφαρμόστηκε σε ένα μεγάλο εύρος συνόλου δεδομένων και πυρήνων γράφων με βάση το είδος τους.

### 4.2.1 Χωρίς Επισημειώσεις

Για την αξιολόγηση των πυρήνων που δέχονται στην είσοδό τους γράφους χωρίς επισημειώσεις, χρησιμοποιήσαμε τα παρακάτω σύνολα δεδομένων. Προκειμένου οι πυρήνες που χρησιμοποιούν τις διακριτές και συνεχείς επισημειώσεις των γράφων της εισόδου τους, να είναι εκτελέσιμοι στα ίδια σύνολα δεδομένων, αποδώσαμε σε κάθε κόμβο ή ακμή μία σταθερή επισημείωση (τον αριθμό 1) και ένα μοναδιαίο διάνυσμα ενός στοιχείου (`numpy.array([1.0])`) αντίστοιχα.

| Πυρήνες | Υποενότητα | Αναγνωριστικό | Διακριτές Επισημειώσεις | | Συνεχείς Επισημειώσεις | |
|---|---|---|---|---|---|---|
| | | | Κόμβοι | Ακμές | Κόμβοι | Ακμές |
| Τυχαίων Περιπάτων | 2.6.1 | RW | ✓* | - | - | - |
| Κοντινότερων Μονοπατιών | 2.6.2 | SP | ✓* | - | ✓* | - |
| Γραφιδίων | 2.6.3 | GR | - | - | - | - |
| Πολυκλιμακωτός Λαπλασιανός | 2.6.8 | ML | - | - | ✓ | - |
| Ταιριάσματος Υπογράφων | 2.6.12 | SM | ✓* | ✓* | ✓* | ✓* |
| Lovasz $\vartheta$ | 2.6.6 | $L_\vartheta$ | - | - | - | - |
| SVM $\vartheta$ | 2.6.7 | $SVM_\vartheta$ | - | - | - | - |
| Κατακερματισμού Γειτονιάς | 2.6.11 | NH | ✓ | - | - | - |
| Αποστάσεων ζευγαριών υπογράφων γειτονιάς | 2.6.10 | NSPDK | ✓ | - | - | - |
| ODD-STh | 2.6.15 | ODD-STh | ✓ | - | - | - |
| Διάδοσης | 2.6.16 | P2K | ✓ | - | ✓ | - |
| Πυραμιδικού ταιριάσματος | 2.6.5 | PM | ✓* | - | - | - |
| Ταιριάσματος Ιστογραμμάτων | 2.6.4 | VH | ✓ | - | - | - |
| Αλμάτων Γράφων | 2.6.14 | GH | - | - | ✓ | - |
| Σκελετός Weisfeiler-Lehman | 2.6.4 | WL | ✓ | - | - | - |
| Σκελετός Core | 2.6.9 | CORE | - | - | - | - |

Πίνακας 4.2: Χωρισμός των πυρήνων γράφων που χρησιμοποιήθηκαν για τα πειράματα με βάση το είδος των επισημειώσεων που περιμένουν στις εισόδους τους. Με ✓συμβολίζουμε την περίπτωση όπου ένα πυρήνας περιμένει οι γράφοι της εισόδου του να έχουν αυτού του τύπου την επισημείωση, ενώ ✓* όταν μπορεί να λειτουργήσει και χωρίς, υλοποιώντας έναν σχετικά διαφορετικό αλγόριθμο. Στην περίπτωση που ένας πυρήνας δέχεται γράφους εισόδου με ✓ και στις συνεχείς και στις διακριτές επισημειώσεις αυτό αφορά δύο σχετικά διαφορετικούς αλγορίθμους. Ακόμα παρατίθενται τα αναγνωριστικά που χρησιμοποιούνται στους πίνακες των πειραμάτων καθώς και η υποενότητα στην οποία περιγράφεται καθένας από τους πυρήνες.

| Πυρήνες | Παραμετροποίηση | |
|---|---|---|
| | Σταθερή | Κινητή |
| RW | $\lambda = 10^{\lceil \log_{10}(\frac{1}{\delta^2_{\max}}) \rceil}$ | $p \in \{2, \dots 10, \inf\}$ |
| SP | - | - |
| GR | $k = 5$ | $n_{\text{samples}} = \{200, 500, 1000, 2000, 5000\}$ |
| ML | $\gamma = 0.01, \eta = 0.01, P = 10$ | $L \in \{0, \dots, 5\}, N \in \{50, 100, 200, 300\}$ |
| SM | $k = 3$ | - |
| $\text{L}_\vartheta$ | $2 \leq \lvert S \rvert \leq 8$ | $n_{\text{samples}} = \{100, 200, 500, 1000\}$ |
| $\text{SVM}_\vartheta$ | $2 \leq \lvert S \rvert \leq 8$ | $n_{\text{samples}} = \{100, 200, 500, 1000\}$ |
| NH | CS-NH | $R \in \{1, \dots, 6\}$ |
| NSPDK | - | $r \in \{1, \dots, 6\}, d \in \{3, \dots, 7\}$ |
| ODD-STh | - | $h \in \{1, \dots, 11\}$ |
| PK | $w = 10^{-5}$ | $t_{\max} \in \{1, \dots, 6\}$ |
| PM | - | $L \in \{2, 4, 6\}, d \in \{4, 6, 8, 10\}$ |
| GH | - | linear/gaussian-kernel |
| VH | - | - |
| WL | - | $n_{\text{iter}} \in \{4, \dots, 8\}$ |
| CORE | - | - |

Πίνακας 4.3: Οι παραμετροποιήσεις των πυρήνων, που επιλέχθηκαν για την πειραματική τους αξιολόγηση. Η σταθερή παραμετροποίηση αφορά τιμές παραμέτρων του πυρήνα που ήταν σταθερές για όλη την εναλλαγή τιμών της κινητής. Οι τιμές της κινητής παραμετροποίησης συνεπάγονται έναν υπολογισμό της μήτρας πυρήνα για όλα τα δυνατά ζευγάρια παραμετροποιήσεων που προκύπτουν από τα ζευγάρια που σημειώνονται, ενώ σε περίπτωση απουσίας '-' η μήτρα πυρήνα υπολογίστηκε μία φορά. Με $\delta_{\max}$ συμβολίζουμε τον μέγιστο βαθμό του αντίστοιχου συνόλου δεδομένων στο οποίο υπολογίζεται η μήτρα πυρήνα.

**COLLAB** Μία συλογή δεδομένων επιστημονικής συνεργασίας που αποτελείται από τα *δίκτυα προσωπικότητας* (ego-networks) αρκετών ερευνητών από τρία υποπεδία της φυσικής (Φυσική Υψηλών Ενεργειών, Φυσική Στερεάς Κατάστασης και της Αστροφυσικής). Ο σκοπός είναι είναι να προσδιοριστεί το υποπεδίο της φυσικής στο οποίο ανήκει το δίκτυο προσωπικότητας του κάθε ερευνητή [91].

**IMDB-BINARY, IMDB-MULTI** Αυτά τα σύνολα δεδομένων δημιουργήθηκαν από το IMDb (`www.imdb.com`), μία online βάση δεδομένων με πληροφορίες που συνδέονται με ταινίες και προγράμματα τηλεόρασης. Οι γράφοι που περιέχονται στα δύο σύνολα δεδομένων αντιστοιχούν σε συνεργασίες εντός ταινιών. Οι κόμβοι κάθε γράφου αναπαριστούν ηθοποιούς και δύο κόμβοι συνδέονται με μία ακμή αν οι αντίστοιχοι ηθοποιοί παίζουν στην ίδια ταινία. Κάθε γράφος είναι ένα δίκτυο προσωπικότητας ηθοποιών και ο στόχος είναι η πρόβλεψη της κατηγορίας ταινιών (genre) στην οποία ανήκει μία ταινία [91].

**REDDIT-BINARY, REDDIT-MULTI-5k, REDDIT-MULTI-12k** Οι γράφοι που περιέχονται σε αυτά τα τρία σύνολα δεδομένων αναπαριστούν κοινωνικές αλληλεπιδράσεις μεταξύ χρηστών του Reddit (`www.reddit.com`), ένα από τα πιο δημοφιλή μέσα κοινωνικής δίκτυωσης. Κάθε γράφος αναπαριστά ένα νήμα συζήτησης στον ιστό. Συγκεκριμένα, κάθε κόμβος αντιστοιχεί σε ένα χρήστη και δύο χρήστες συνδέονται από μία ακμή αν τουλάχιστον ένας από αυτούς αντέδρασε στο σχόλιο του άλλου. Στόχος είναι η ταξινόμηση γράφων είτε σε κοινότητες είτε σε 'υπο-reddit' (subreddits) [91].

### 4.2.2 Με Διακριτές Επισημειώσεις

Για την αξιολόγηση των πυρήνων που δέχονται στην είσοδο τους γράφους με **διακριτές** επισημειώσεις, χρησιμοποιήσαμε τα παρακάτω σύνολα δεδομένων. Προκειμένου οι πυρήνες, που δέχονται στην είσοδο τους γράφους με συνεχείς επισημειώσεις, να είναι εκτελέσιμοι στα ίδια σύνολα δεδομένων, αποδώσαμε σε κάθε κόμβο ένα *One-Hot Vector* με βάση το σύνολο όλων των επισημειώσεων που εμφανίζονται σε κάθε σύνολο δεδομένων.

**MUTAG** Αυτό το σύνολο δεδομένων αποτελείται από 188 μεταλλαξιογόνες αρωματικές-ετεροαρωματικές νιτρικές ενώσεις. Ο στόχος είναι η πρόβλεψη του αν μία χημική ένωση έχει μεταλλαξιογόνα δράση στο αρνητικό κατά Gram βακτήριο Salmonella typhimurium [73].

**ENZYMES** Αποτελείται από 600 τριτογενείς δομές πρωτεϊνών που ανήκουν στην βάση ενζύμων BRENDA. Κάθε ένζυμο είναι ταξινομημένο στην αφηρημένη κατάταξη Enzyme Commission και σκοπός είναι ο ορθός προσδιορισμός της κλάσης στην οποία ανήκει ένα ένζυμο [11].

**DD** Αυτό το σύνολο δεδομένων περιέχει πάνω από χίλιες δομές πρωτεϊνών. Κάθε πρωτεΐνη είναι ένας γράφος που οι κόμβοι του αντιστοιχούν σε αμινοξέα και ένα ζευγάρι αμινοξέων συνδέεται με μία ακμή αν η απόσταση τους είναι λιγότερη από 6 Ångstrom. Στόχος είναι να προβλέψουμε αν μία πρωτεΐνη είναι ένζυμο (ή όχι) [21, 73].

**NCI1** Αυτό το σύνολο δεδομένων περιέχει μερικές χιλιάδες χημικά στοιχεία στα οποία καταγράφεται η δραστηριότητα τους απέναντι σε καρκινικά κύτταρα του πνεύμονα και των ωοθηκών βάση της πορείας κυτταρικής διαίρεσης τους (cell lines) σε ελεγχόμενες συνθήκες εργαστηρίου [86].

**PTC_MR** Αυτό το σύνολο δεδομένων περιέχει 344 οργανικά μόρια που έχουν αναπαρασταθεί ως γράφοι. Στόχος είναι η πρόβλεψη καρκινογένεσης σε αρσενικούς αρουραίους [78].

**AIDS** Αποτελείται από 2000 χημικές ενώσεις που έχουν αναπαρασταθεί ως γράφοι, οι οποίες έχουν δοκιμαστεί για την αποτελεσματικότητα τους απέναντι στον ιό HIV. Σκοπός λοιπόν του προβλήματος ταξινόμησης είναι η πρόβλεψη του κατά πόσον μία χημική ένωση μπορεί να είναι ή όχι αποτελεσματική απέναντι στον ιό [67].

**PROTEINS** Περιέχει πρωτεΐνες που έχουν αναπαρασταθεί ως γράφοι, όπου οι κόμβοι είναι δευτερογενή δομικά στοιχεία και μία ακμή υπάρχει μεταξύ των κόμβων αν οι κόμβοι είναι γείτονες στην ακολουθία αμινοξέων ή στον τρισδιάστατο χώρο. Σκοπός είναι η ταξινόμηση μίας πρωτεΐνης ως ένζυμο (ή όχι) [13].

### 4.2.3 Με Επισημειώσεις Χαρακτηριστικών

Για την αξιολόγηση πυρήνων που δέχονται στην είσοδο τους γράφους με **συνεχείς** επισημειώσεις, χρησιμοποιήσαμε τα παρακάτω σύνολα δεδομένων.

**ENZYMES** Αποτελείται από 600 τριτογενείς δομές πρωτεϊνών που ανήκουν στην βάση ενζύμων BRENDA. Κάθε ένζυμο είναι ταξινομημένο στην αφηρημένη κατάταξη Enzyme Commission και ο σκοπός είναι ο ορθός προσδιορισμός της κλάσης στην οποία ανήκει ένα ένζυμο [11].

**Synthie** Είναι ένα τεχνητό σύνολο δεδομένων που αποτελείται από 400 γράφους. Το σύνολο δεδομένων υποδιαιρείται σε 4 κατηγορίες. Κάθε κόμβος επισημειώνεται με ένα διάνυσμα 15 στοιχείων. Για την κατασκευή του παράγονται δύο σύνολα με διαφορετική παραμετροποίηση 200 γράφων Erdös-Rényi όπου το 25% των ακμών τους αφαιρείται τυχαία, ενώ κατηγοριοποιούνται σε δύο κλάσεις, διαλέγοντας και συνδέοντας τυχαία 10 γράφους με πιθανότητα 0.8 και 0.2 από το πρώτο και το δεύτερο σύνολο αντίστοιχα για την πρώτη κατηγορία και με αντίστροφες πιθανότητες για την δεύτερη κατηγορία. Έπειτα δημιουργώντας δύο σύνολα χαρακτηριστικών δεκαπενταδιάστατων διανυσμάτων δύο κατηγοριών, οι παραπάνω δύο κατηγορίες χωρίζονται σε άλλες δύο, όπου για την πρώτη υποκατηγορία, αν ένα κόμβος προερχόταν από το πρώτο σύνολο γράφων επισημειώνεται τυχαία με ένα διάνυσμα του πρώτου συνόλου χαρακτηριστικών ενώ σε αντίθετη περίπτωση με διάνυσμα του δεύτερου. Για την παραγωγή της δεύτερης κατηγορίας συμβαίνει το αντίθετο. Στόχος του προβλήματος ταξινόμησης είναι βάσει των χαρακτηριστικών, να ανιχνευθεί σε ποια από τις τέσσερεις υποκατηγορίες ανήκει ένας γράφος [58].

**BZR** Αυτό το σύνολο δεδομένων αποτελεί από 684 χημικές ενώσεις κατηγοριοποιημένες ως μεταλλαξιογόνες ή μη, βάσει ενός πειράματος γνωστό ως Salmonella/microsome assay. Αυτό το σύνολο δεδομένων είναι ισοσταθμισμένο, με 341 μεταλλαξιογόνες χημικές ενώσεις και 343 μη-μεταλλαξιογόνες [53, 60].

**PROTEINS_full** Αυτό το σύνολο δεδομένων αποτελείται 1113 από χημικές ενώσεις προερχόμενες από την βάση δεδομένων πρωτεΐνών PDB. Διαχωρισμένες σε ένζυμα (59%) και μη-ένζυμα (41%), οι πρωτεΐνες έχουν διαλεχτεί έτσι ώστε καμία ακολουθία να μην ταιριάζει με μία άλλη. Παρέχουν πλούσια επισημείωση για κάθε κόμβο σε μορφή 29-διάστατων χαρακτηριστικών χρησιμοποιώντας μεταξύ άλλων την κρυσταλλογραφική τους πληροφορία [21, 13, 60]

**SYNTHETICnew** είναι ένα τεχνητό σύνολο δεδομένων 300 τυχαία δειγματοληπτημένων γράφων που αποτελούνται από 100 κόμβους και 196 ακμές, στους κόμβους των οποίων αναπτύθενται μονοδιάστατες συνεχείς επισημειώσεις από το $\mathcal{N}(0,1)$. Έπειτα δημιουργούνται δύο ισοσταθμισμένες κατηγορίες 150 επισημειώσεων, αφαιρώντας και επαναπροσθέτοντας τυχαία 5 ακμές και μεταθέτοντας τυχαία τις επισημειώσεις 10 κόμβων για την πρώτη κατηγορία και 10, 5 για την δεύτερη προσθέτοντας στο τέλος τυχαίο θόρυβο σε όλες τις επισημειώσεις από την $\mathcal{N}(0, 0.452)$ [22].

Όλα τα σύνολα δεδομένων που αναφέρθηκαν παραπάνω προέρχονται από το [46]. Στατιστικά στοιχεία και πληροφορίες σχετικά με την ύπαρξη και τον τύπο των επισημείωσεων τους παρουσιάζονται συνοπτικά στον πίνακα 4.4.

## 4.3 Αποτελέσματα & Αξιολόγηση

Σε αυτό το μέρος θα παρουσιάσουμε τα αποτελέσματα από όλα τα πειράματα όπως περιγράφηκαν παραπάνω. Η αξιολόγηση θα αποτελείται από 3 πίνακες στους οποίους: ο πρώτος θα αποτελείται από τις επιδόσεις ευστοχίας των πυρήνων για όλους τους συνδυασμούς παραμέτρων όπως περιγράφονται στην ενότητα 4.1, για όλες τις αντίστοιχες κατηγορίες από dataset όπως περιγράφονται στην ενότητα 4.2, ενώ ο δεύτερος και ο τρίτος θα αποτελείται από τον χρόνο εκτέλεσης και την μέγιστη μνήμη για τις καταγραφόμενες στην ευστοχία εκτελέσεις. Έτσι για κάθε σύνολο δεδομένων θα σημειώνονται οι αποτελεσματικότεροι πυρήνες, ενώ παράλληλα θα σχολιάζονται οι επιδόσεις τους σε σχέση με τις απαιτήσεις χρόνου και μνήμης των υπόλοιπων πυρήνων.

### 4.3.1 Χωρίς Επισημειώσεις

Στα σύνολα δεδομένων IMDB-BINARY, IMDB-MULTI, COLLAB, REDDIT-MULTI-12K, REDDIT-BINARY και REDDIT-MULTI-5K με επισημειώσεις χαρακτηριστικών, όπως περιγράφονται στην υποενότητα 4.2.1, εκτελέστηκαν οι πυρήνες GH, GR, $L_\vartheta$, ML, NH, NSPDK, ODD-STh, P2K, PM, RW, SM, SP, $SVM_\vartheta$ και VH. Καλύτερη επίδοση τόσο σε ευστοχία

| Dataset Name | Statistics | | | | Node-Labels/Node-Attributes (Dim.) | |
|---|---|---|---|---|---|---|
| | #Graphs | #Classes | Avg. #Nodes | Avg. #Edges | Node-Lab. | Node-Attr. |
| AIDS | 2000 | 2 | 15.69 | 16.20 | + | + (4) |
| BZR | 405 | 2 | 35.75 | 38.36 | + | + (3) |
| COLLAB | 5000 | 3 | 74.49 | 2457.78 | – | – |
| DD | 1178 | 2 | 284.32 | 715.66 | + | – |
| ENZYMES | 600 | 6 | 32.63 | 62.14 | + | + (18) |
| IMDB-BINARY | 1000 | 2 | 19.77 | 96.53 | – | – |
| IMDB-MULTI | 1500 | 3 | 13.00 | 65.94 | – | – |
| MUTAG | 188 | 2 | 17.93 | 19.79 | + | – |
| PTC_MR | 344 | 2 | 14.29 | 14.69 | + | – |
| PROTEINS | 1113 | 2 | 39.06 | 72.82 | + | + (1) |
| PROTEINS_full | 1113 | 2 | 39.06 | 72.82 | + | + (29) |
| REDDIT-BINARY | 2000 | 2 | 429.63 | 497.75 | – | – |
| REDDIT-MULTI-5k | 4999 | 5 | 508.52 | 594.87 | – | – |
| REDDIT-MULTI-12k | 11929 | 11 | 391.41 | 456.89 | – | – |
| SYNTHETICnew | 300 | 2 | 100.00 | 196.25 | – | + (1) |
| Synthie | 400 | 4 | 95.00 | 172.93 | – | + (15) |

Πίνακας 4.4: Στατιστικά στοιχεία για τα σύνολα δεδομένων καθώς και πληροφορίες σχετικά με την ύπαρξη και τον τύπο των επισημειώσεων.

όσο και σε χρόνο και μνήμη, είχαν οι πυρήνες NH και PM με τον πρώτο να παρουσιάζει μεγαλύτερη ευστοχία στα σύνολα δεδομένων IMDB-BINARY, IMDB-MULTI, COLLAB και REDDIT-MULTI-5K, PROTEINS_full και τον δεύτερο στα REDDIT-MULTI-12K και REDDIT-BINARY όπως φαίνεται στον πίνακα 4.5. Ακόμα ο πυρήνας SM αδυνατεί να ανταποκριθεί σχεδόν σε όλα τα σύνολα δεδομένων τόσο λόγω χρόνου όσο και μνήμης, μιας και στην περίπτωση των γράφων χωρίς επισημειώσεις συνυπολογίζει όλες τις κλίκες. Ο πυρήνας $L_\vartheta$ ξεπερνάει το όριο χρόνου όπως είναι αναμενόμενο, καθώς τα σύνολα δεδομένων αυξάνονται, ενώ παράλληλα η προσέγγιση του από τον $S_\vartheta$ φαίνεται να έχει καλύτερη επίδοση, χωρίς από την άλλη να πλησιάζει τις βέλτιστες τιμές. Το πείραμα σε μη επισημειωμένα δεδομένα είναι χρήσιμο για να παρατηρήσει κανείς την επίδοση των πυρήνων σε κλιμακούμενα δεδομένα στο μέγεθος των γράφων, μιάς και το μικρότερο σύνολο δεδομένων είναι το IMDB-BINARY με 1000 γράφους.

### 4.3.2 Με Διακριτές Επισημειώσεις

Στα σύνολα δεδομένων NCI1, PTC_MR, ENZYMES, DD, PROTEINS, MUTAG και AIDS με διακριτές επισημειώσεις, όπως περιγράφονται στην υποενότητα , εκτελέστηκαν οι πυρήνες AIDS, GH, ML, NH, NSPDK ODD-STh, PK, PM, RW, SM, SP, VH, CORE-SP, CORE-WL-VH, WL-PM, WL-SP και WL-VH. Καλύτερη επίδοση τόσο σε ευστοχία όσο και σε χρόνο και μνήμη, είχαν οι πυρήνες CORE-WL-VH, WL-PM και WL-VH, με κοινή καλύτερη επίδοση στο NCI1 και καλύτερες επιδόσεις στο PROTEINS, MUTAG και ENZYMES, και στο PTC_MR αντίστοιχα, όπως φαίνεται στον πίνακα 4.8. Κάτι τέτοιο φαίνεται να επαληθεύει την υπόθεση ότι οι παραπάνω σκελετοί πυρήνα κάνουν πιο εκφραστικούς τους υπάρχοντες πυρήνες γράφων είτε εισάγοντας μία ιεραρχία στη δομή (όπως στην περίπτωση του CORE) είτε εκφράζοντας την ίδια την δομή μέσω των επισημειώσεων (όπως στην περίπτωση του WL). Η μέγιστη δυνατή ευστοχία του GH στο AIDS, δεν τονίζεται καθώς δεν φαίνεται σημαντική σε σχέση με τα σκορ των υπολοίπων πυρήνων.

### 4.3.3 Με Επισημειώσεις Χαρακτηριστικών

Στα σύνολα δεδομένων ENZYMES, SYNTHETICnew, Synthie, BZR και PROTEINS_full με επισημειώσεις χαρακτηριστικών, όπως περιγράφονται στην υποενότητα , εκτελέστηκαν οι πυρήνες GH, ML, PK, SM, SP.

Καλύτερη επίδοση τόσο σε ευστοχία όσο και σε χρόνο και μνήμη, είχαν οι πυρήνες GH και ML με τον πρώτο να παρουσιάζει μεγαλύτερη ευστοχία στα σύνολα δεδομένων ENZYMES, SYNTHETICnew, PROTEINS_full και τον δεύτερο στα Synthie και BZR όπως φαίνεται στον πίνακα 4.11. Ακόμα ο πυρήνας SM (που στην περίπτωση αυτή φέρει την συνάρτηση εσωτερικού γινομένου για να υπολογίσει μία τιμή πυρήνα μεταξύ των χαρακτηριστικών στους κόμβους) εκτελείται φέροντας ικανοποιητικό σκορ μόνο στο BZR ενώ σε όλους τους υπολοίπους διακόπτεται είτε επειδή ξεπερνάει το όριο χρόνου είτε γιατί ξεπερνάει την μνήμη. Τέλος, ο πυρήνας κοντινότερων μονοπατιών στην περίπτωση των χαρακτηριστικών είναι εξαιρετικά αργός και ως αποτέλεσμα δεν μας ξαφνιάζει η ανεπάρκεια τερματισμού του σε ικανοποιητικό

| | IMDB-BINARY | IMDB-MULTI | COLLAB | REDDIT-MULTI-12K | REDDIT-BINARY | REDDIT-MULTI-5K |
|---|---|---|---|---|---|---|
| GH | 0.57 ± 0.01 | 0.4 ± 0.01 | 0.6 | TIMEOUT | TIMEOUT | TIMEOUT |
| GR | 0.65 ± 0.01 | 0.39 ± 0.01 | 0.71 | 0.23 | - | 0.34 ± 0.01 |
| $L_\vartheta$ | 0.49 ± 0.01 | 0.4 ± 0.01 | TIMEOUT | TIMEOUT | TIMEOUT | TIMEOUT |
| ML | 0.47 ± 0.01 | 0.38 ± 0.01 | 0.56 | MEM_RSS | 0.74 | TIMEOUT |
| NH | 0.73 ± 0.01 | 0.5 ± 0.01 | 0.8 | 0.4 | 0.82 ± 0.01 | 0.49 |
| NSPDK | 0.69 | 0.45 ± 0.01 | FINISHED | TIMEOUT | TIMEOUT | TIMEOUT |
| ODD-STh | 0.64 ± 0.02 | 0.47 ± 0.01 | 0.52 | 0.3 | 0.51 ± 0.01 | 0.43 |
| P2K | 0.51 ± 0.01 | 0.33 ± 0.01 | 0.59 | 0.24 | 0.63 ± 0.01 | 0.34 |
| PM | 0.67 ± 0.01 | 0.45 | 0.75 | 0.41 | 0.87 | 0.48 |
| RW | 0.64 ± 0.01 | 0.46 ± 0.01 | MEM_RSS | MEM_RSS | - | FINISHED |
| SM | TIMEOUT | TIMEOUT | TIMEOUT | MEM_RSS | - | MEM_RSS |
| SP | 0.55 ± 0.02 | 0.39 ± 0.01 | 0.59 | TIMEOUT | - | 0.48 |
| $SVM_\vartheta$ | 0.51 ± 0.01 | 0.38 | 0.56 | 0.23 | 0.75 | 0.3 |
| VH | 0.47 ± 0.01 | 0.3 | 0.52 | 0.22 | 0.47 ± 0.01 | 0.18 |

Πίνακας 4.5: Μέσοι όροι και διακυμάνσεις της μετρικής ευστοχίας από 10 επαναλήψεις 10-fold cross validation στα σύνολα δεδομένων **χωρίς** επισημειώσεις. Τα αποτελέσματα που τονίζονται αφορούν τα καλύτερα σκορ για κάθε σύνολο δεδομένων ως προς την μέση τιμή. Σε περίπτωση συμψηφισμού, τονίζονται αυτά με την μεγαλύτερη διακύμανση. Στην περίπτωση που δεν σημειώνεται η διακύμανση, είναι διότι αντιστοιχεί σε μηδέν με ακρίβεια δύο δυαδικών ψηφίων. Με MEM_RSS περιγράφουμε την περίπτωση που ο υπολογισμός του πυρήνα ξεπέρασε το όριο μνήμης που είχαμε θέσει, και με TIMEOUT το όριο χρόνου.

| | IMDB-BINARY | IMDB-MULTI | COLLAB | REDDIT-MULTI-12K | REDDIT-BINARY | REDDIT-MULTI-5K |
|---|---|---|---|---|---|---|
| GH | 2.19 m | 2.06 m | 5.86 h | TIMEOUT | TIMEOUT | TIMEOUT |
| GR | 19.18 m ± 3.52 m | 19.14 m ± 8.04 m | 3.14 h ± 46.53 m | 48.37 m ± 23.57 m | NaN | 49.97 m ± 12.44 m |
| $L_\vartheta$ | 5.85 h ± 27.68 m | 6.09 h ± 49.34 m | TIMEOUT | TIMEOUT | TIMEOUT | TIMEOUT |
| ML | 30.39 s | 1.17 m | 13.71 m | - | 13.83 h | TIMEOUT |
| NH | 19.75 s ± 3.64 s | 27.29 s ± 4.24 s | 37.1 m ± 5.78 m | 9.46 h ± 57.42 m | 22.15 m ± 3.16 m | 2.68 h ± 29.16 m |
| NSPDK | 4.35 m ± 45.5 s | 2.67 m ± 27.05 s | - | TIMEOUT | TIMEOUT | TIMEOUT |
| ODD-STh | 5.01 s ± 1.57 s | 5.14 s ± 0.65 s | 2.02 h | 8.34 m | 1.89 m | 4.82 m |
| P2K | 7.09 s ± 0.49 s | 14.22 s ± 1.12 s | 4.46 m ± 12.2 s | 20.48 m ± 55.03 s | 1.39 m ± 2.68 s | 5.76 m ± 18.94 s |
| PM | 1.51 m ± 4.67 s | 2.25 m ± 18.83 s | 36.44 m ± 2.48 m | 3.84 h | 10.06 m ± 9.1 s | 51.75 m |
| RW | 12.32 m ± 4.27 s | 7.74 m | - | - | - | - |
| SM | TIMEOUT | TIMEOUT | TIMEOUT | - | NaN | - |
| SP | 11.51 s | 7.92 s | 1.15 h | TIMEOUT | NaN | 12.67 h |
| $SVM_\vartheta$ | 42.05 s ± 7.93 s | 1.02 m ± 14.72 s | 6.05 m ± 27.1 s | 51.88 m ± 2.53 m | 19.44 m ± 6.42 s | 23.33 m ± 59.26 s |
| VH | 0.07 s | 0.15 s | 1.12 s | 6.37 s | 0.67 s | 2.2 s |

Πίνακας 4.6: Μέσοι όροι και διακυμάνσεις των χρόνων εκτέλεσης που αντιστοιχούν στις καλύτερες τιμές της μετρικής ευστοχίας για 10 επαναλήψεις 10-fold cross validation στα σύνολα δεδομένων **χωρίς** επισημειώσεις, όπως καταγράφονται στον πίνακα 4.5. Οι χρόνοι που τονίζονται αφορούν αυτούς με τα καλύτερα σκορ ευστοχίας. Τα κελιά που σημειώνονται με '-' αφορούν τιμές που η εκτέλεση δεν ολοκληρώθηκε, καθώς υπερέβη την μέγιστη επιτρεπτή μνήμη και με "TIMEOUT" όταν ξεπέρασε το μέγιστο επιτρεπτό χρόνο.

| | IMDB-BINARY | IMDB-MULTI | COLLAB | REDDIT-MULTI-12K | REDDIT-BINARY | REDDIT-MULTI-5K |
|---|---|---|---|---|---|---|
| GH | 0.25G | 0.28G | 11.98G | - | - | - |
| GR | 0.22G ± 0.13M | 0.22G ± 2.08M | 11.17G ± 13.64M | 8.22G ± 3.87M | NaN | 3.3G ± 8.45M |
| $L_\vartheta$ | 1.32G ± 0.07M | 0.36G ± 0.09M | - | - | - | - |
| ML | 0.25G | 0.27G | 12.39G | MEMORY-OUT | 23.8G | - |
| NH | 0.31G ± 0.66M | 0.34G ± 0.52M | 22.28G ± 13.75M | 15.68G ± 74.26M | 2.42G ± 28.56M | 7.49G ± 98.39M |
| NSPDK | 0.42G ± 11.24M | 0.54G ± 21.1M | - | - | - | - |
| ODD-STh | 0.31G ± 13.7M | 0.35G ± 6.18M | 49.4G | 14.41G | 2.17G | 6.68G |
| P2K | 0.25G ± 0.04M | 0.26G ± 0.35M | 11.99G ± 0.72M | 44.53G ± 3.35M | 10.13G ± 2.09M | 25.28G ± 4.08M |
| PM | 0.21G ± 0.48M | 0.23G ± 1.42M | 11.12G ± 0.03M | 8.6G | 1.17G ± 0.43M | 3.57G |
| RW | 2.83G ± 0.05M | 0.54G | MEMORY-OUT | MEMORY-OUT | - | - |
| SM | - | - | - | MEMORY-OUT | NaN | MEMORY-OUT |
| SP | 0.21G | 0.21G | 11.02G | - | NaN | 3.33G |
| $SVM_\vartheta$ | 0.21G ± 0.13M | 0.22G ± 0.18M | 11.13G ± 0.04M | 8.27G ± 2.66M | 1.35G ± 0.04M | 3.4G ± 0.23M |
| VH | 0.34G | 0.43G | 22.12G | 14.22G | 2.12G | 6.68G |

Πίνακας 4.7: Μέσοι όροι και διακυμάνσεις της μέγιστης τιμής μνήμης από τις εκτελέσεις που αντιστοιχούν στις καλύτερες τιμές της μετρικής ευστοχίας για 10 επαναλήψεις 10-fold cross validation στα σύνολα δεδομένων **χωρίς** επισημειώσεις όπως φαίνονται στον πίνακα 4.5. Οι χρόνοι που τονίζονται αφορούν αυτούς με τα καλύτερα σκορ ευστοχίας. Τα κελιά που σημειώνονται με '-' αφορούν τιμές που η εκτέλεση διακόπηκε καθώς υπερέβη τον μέγιστο επιτρεπτό, ενώ με "MEMORY-OUT" όταν υπερέβη την μέγιστη επιτρεπτή μνήμη.

| | NCI1 | PTC_MR | ENZYMES | DD | PROTEINS | MUTAG | AIDS |
|---|---|---|---|---|---|---|---|
| GH | 0.71 | 0.56 ± 0.01 | 0.38 ± 0.01 | TIMEOUT | 0.72 ± 0.01 | 0.81 ± 0.02 | **1.0** |
| ML | 0.7 | 0.59 ± 0.02 | 0.42 ± 0.01 | **0.8** ± 0.01 | **0.75 ± 0.01** | 0.84 ± 0.01 | 0.99 |
| NH | 0.75 | 0.6 ± 0.02 | 0.44 ± 0.01 | 0.76 ± 0.01 | 0.71 ± 0.01 | 0.86 ± 0.02 | 0.99 |
| NSPDK | 0.74 | 0.57 ± 0.02 | 0.12 ± 0.02 | 0.79 | 0.73 | 0.86 ± 0.02 | 0.98 |
| ODD-STh | 0.73 | 0.57 ± 0.01 | 0.33 ± 0.01 | 0.76 | 0.71 ± 0.01 | 0.77 ± 0.02 | 0.91 |
| PK | 0.81 ± 0.01 | 0.61 ± 0.01 | 0.4 ± 0.01 | 0.79 | 0.71 ± 0.01 | 0.77 ± 0.03 | 0.97 |
| PM | 0.74 | 0.56 ± 0.02 | 0.39 ± 0.02 | 0.77 ± 0.01 | 0.69 ± 0.01 | 0.83 ± 0.02 | 0.99 |
| RW | 0.6 | 0.53 ± 0.02 | 0.13 ± 0.01 | MEM_RSS | 0.66 ± 0.01 | 0.83 ± 0.01 | 0.8 |
| SM | TIMEOUT | 0.58 ± 0.02 | 0.37 ± 0.01 | MEM_RSS | MEM_RSS | 0.84 ± 0.02 | 0.92 |
| SP | 0.72 | 0.6 ± 0.02 | 0.42 ± 0.01 | 0.79 ± 0.01 | 0.76 | 0.82 ± 0.02 | 0.99 |
| VH | 0.56 | 0.56 | 0.11 ± 0.01 | 0.75 | 0.71 | 0.66 ± 0.01 | 0.8 |
| CORE-SP | 0.73 ± 0.01 | 0.58 ± 0.02 | 0.42 ± 0.01 | 0.79 | 0.75 | 0.84 ± 0.02 | 0.99 |
| CORE-WL-VH | **0.85** | 0.6 ± 0.01 | 0.46 ± 0.03 | 0.79 | **0.75 ± 0.01** | 0.87 ± 0.01 | 0.99 |
| WL-PM | **0.85** | 0.61 ± 0.02 | **0.51 ± 0.05** | MEM_RSS | 0.75 | **0.87 ± 0.02** | 0.99 |
| WL-SP | 0.62 | 0.55 ± 0.01 | 0.19 ± 0.03 | 0.76 | 0.72 | 0.77 ± 0.02 | 0.99 |
| WL-VH | **0.85** | **0.63** ± 0.02 | 0.51 ± 0.04 | 0.79 | 0.75 | 0.85 ± 0.02 | 0.98 |

Πίνακας 4.8: Μέσοι όροι και διακυμάνσεις της μετρικής ευστοχίας για 10 επαναλήψεις 10-fold cross validation στα σύνολα δεδομένων με **διακριτές** επισημειώσεις. Τα αποτελέσματα που τονίζονται αφορούν τα καλύτερα σκορ για κάθε σύνολο δεδομένων ως προς την μέση τιμή. Σε περίπτωση συμψηφισμού, τονίζονται αυτά με την μεγαλύτερη διακύμανση. Στην περίπτωση που δεν σημειώνεται η διακύμανση, είναι διότι αντιστοιχεί σε μηδέν σε ακρίβεια δύο δυαδικών ψηφίων. Με MEM_RSS περιγράφουμε την περίπτωση που ο υπολογισμός του πυρήνα ξεπέρασε το όριο μνήμης που είχαμε θέσει, και με TIMEOUT το όριο χρόνου.

| | NCI1 | PTC_MR | ENZYMES | DD | PROTEINS | MUTAG | AIDS |
|---|---|---|---|---|---|---|---|
| GH | 3.73 h ± 1.32 m | 1.63 m ± 0.26 s | 15.71 m ± 2.14 s | TIMEOUT | 3.76 h ± 2.25 m | 26.15 s ± 0.12 s | **39.88 m** |
| ML | 9.57 m ± 4.99 s | 4.81 s ± 0.2 s | **12.12 s ± 0.14 s** | **4.23 m ± 25.7 s** | **44.76 s ± 1.86 s** | 2.04 s ± 0.53 s | 2.39 m ± 0.64 s |
| NH | 7.26 m ± 43.29 s | 1.14 s ± 0.15 s | 7.25 s ± 1.93 s | 5.26 m ± 43.65 s | 52.33 s ± 6.28 s | 0.41 s ± 0.03 s | 32.78 s |
| NSPDK | 1.54 m ± 2.38 s | 4.54 s ± 0.62 s | 19.63 s ± 1.42 s | 4.0 h ± 24.07 m | 4.74 m ± 1.36 m | 2.05 s ± 0.1 s | 26.99 s ± 1.58 s |
| ODD-STh | 35.23 m ± 50.94 s | 0.72 s ± 0.24 s | 1.07 m ± 2.01 s | 43.67 m ± 5.09 m | 2.82 m ± 44.33 s | 1.28 s ± 0.19 s | 1.87 m ± 3.43 s |
| PK | 9.98 m ± 27.77 s | 1.87 s ± 0.08 s | 7.34 s ± 0.78 s | 11.44 m ± 1.15 m | 48.66 s ± 4.09 s | 0.44 s ± 0.03 s | 1.77 m ± 5.19 s |
| PM | 44.77 m ± 43.26 s | 6.38 s ± 0.57 s | 16.32 s ± 0.67 s | 6.82 m ± 39.13 s | 1.64 m ± 4.67 s | 1.69 s ± 0.04 s | 3.72 m ± 16.0 s |
| RW | 15.14 h | 13.32 m | 39.41 m ± 14.03 m | - | 47.67 m | 1.11 m ± 8.26 s | 1.41 h |
| SM | TIMEOUT | 4.33 m | 3.43 h | - | - | 1.95 m | 4.45 h |
| SP | 1.16 m | 1.52 s | 11.03 s | 55.98 m | 1.32 m | 0.92 s | 13.93 s |
| VH | 0.84 s | 0.02 s | 0.04 s | 0.24 s | 0.1 s | 0.01 s | 0.25 s |
| CORE-SP | 3.28 m | 3.97 s | 48.02 s | 5.04 h | 3.53 m | 2.69 s | 40.11 s |
| CORE-WL-VH | **10.92 m ± 33.24 s** | 1.01 s ± 0.05 s | 12.41 s ± 1.31 s | 17.04 m | **1.39 m ± 4.55 s** | 0.55 s ± 0.04 s | 40.4 s ± 2.9 s |
| WL-PM | **12.79 h ± 50.2 m** | 3.6 m ± 1.51 m | **23.48 m ± 4.27 m** | - | 3.42 h ± 50.71 m | **56.06 s ± 8.19 s** | 2.31 h |
| WL-SP | 5.0 m | 7.31 s | 54.68 s | 4.25 h | 4.99 m | 4.39 s | 58.9 s |
| WL-VH | **6.33 m ± 14.26 s** | **0.38 s ± 0.03 s** | 3.12 s ± 0.19 s | 5.3 m ± 12.05 s | 25.12 s ± 2.85 s | 0.16 s ± 0.01 s | 34.06 s ± 1.98 s |

Πίνακας 4.9: Μέσοι όροι και διακυμάνσεις των χρόνων εκτέλεσης που αντιστοιχούν στις καλύτερες τιμές της μετρικής ευστοχίας για 10 επαναλήψεις 10-fold cross validation στα σύνολα δεδομένων με **διακριτές** επισημειώσεις, όπως καταγράφονται στον πίνακα 4.8. Οι χρόνοι που τονίζονται αφορούν αυτούς με τα καλύτερα σκορ ευστοχίας. Τα κελιά που σημειώνονται με '-' αφορούν τιμές που η εκτέλεση δεν ολοκληρώθηκε, καθώς υπερέβη την μέγιστη επιτρεπτή μνήμη και με "TIMEOUT" όταν ξεπέρασε το μέγιστο επιτρεπτό χρόνο.

| | NCI1 | PTC_MR | ENZYMES | DD | PROTEINS | MUTAG | AIDS |
|---|---|---|---|---|---|---|---|
| GH | 3.07G ± 6.28M | 0.15G ± 0.09M | 0.41G ± 0.04M | - | 1.73G ± 0.01M | 0.12G | **0.91G** |
| ML | 1.1G ± 0.78M | 0.12G ± 0.32M | **0.19G ± 0.72M** | **5.45G ± 0.21M** | **0.35G ± 3.21M** | 0.11G ± 0.54M | 0.34G ± 0.5M |
| NH | 0.77G ± 7.07M | 0.11G ± 0.1M | 0.18G ± 0.82M | 1.87G ± 8.74M | 0.3G ± 1.06M | 0.11G ± 0.04M | 0.24G |
| NSPDK | 1.96G ± 20.64M | 0.14G ± 2.62M | 0.29G ± 8.6M | 18.18G ± 1.66G | 1.32G ± 0.13G | 0.11G ± 0.79M | 0.74G ± 18.21M |
| ODD-STh | 25.3G ± 0.51G | 0.11G ± 3.56M | 1.46G ± 41.67M | 39.8G ± 4.53G | 2.83G ± 0.66G | 0.12G ± 2.64M | 1.92G ± 53.45M |
| P2K | 1.34G ± 1.23M | 0.12G ± 0.26M | 0.19G ± 0.47M | 6.88G ± 0.24G | 0.32G ± 0.15M | 0.11G ± 0.59M | 0.4G ± 0.46M |
| PM | 0.97G ± 10.73M | 0.12G ± 1.24M | 0.18G ± 0.42M | 2.09G ± 22.69M | 0.3G ± 1.1M | 0.11G ± 0.58M | 0.3G ± 2.62M |
| RW | 1.18G | 0.65G | 0.28G ± 0.22G | MEMORY-OUT | 0.45G | 0.11G ± 0.66M | 0.34G |
| SM | - | 0.13G | 1.14G | MEMORY-OUT | MEMORY-OUT | 0.11G | 0.4G |
| SP | 0.81G | 0.11G | 0.18G | 2.84G | 0.29G | 0.11G | 0.29G |
| VH | 0.71G | 0.11G | 0.19G | 1.83G | 0.33G | 0.1G | 0.33G |
| CORE-SP | 1.44G | 0.12G | 0.2G | 8.98G | 0.38G | 0.11G | 0.43G |
| CORE-WL-VH | **13.11G ± 0.55G** | 0.16G ± 3.66M | 1.1G ± 0.11G | 50.41G | **5.08G ± 0.28G** | 0.12G ± 2.07M | 1.81G ± 96.69M |
| WL-PM | **15.55G ± 1.17G** | 0.88G ± 0.35G | **1.65G ± 0.46G** | MEMORY-OUT | 15.52G ± 4.07G | **0.41G ± 46.76M** | 3.85G |
| WL-SP | 1.4G | 0.11G | 0.17G | 1.92G | 0.28G | 0.1G | 0.39G |
| WL-VH | **5.24G ± 0.37G** | **0.14G ± 1.33M** | 0.39G ± 17.77M | 17.3G ± 0.58G | 1.46G ± 0.14G | 0.12G ± 0.68M | 1.51G ± 68.83M |

Πίνακας 4.10: Μέσοι όροι και διακυμάνσεις της μέγιστης τιμής μνήμης από τις εκτελέσεις που αντιστοιχούν στις καλύτερες τιμές της μετρικής ευστοχίας για 10 επαναλήψεις 10-fold cross validation στα σύνολα δεδομένων με **διακριτές** επισημειώσεις όπως φαίνονται στον πίνακα 4.8. Οι χρόνοι που τονίζονται αφορούν αυτούς με τα καλύτερα σκορ ευστοχίας. Τα κελιά που σημειώνονται με '-' αφορούν τιμές που η εκτέλεση διακόπηκε καθώς υπερέβη τον μέγιστο επιτρεπτό, ενώ με "MEMORY-OUT" όταν υπερέβη την μέγιστη επιτρεπτή μνήμη.

| | ENZYMES | SYNTHETICnew | Synthie | BZR | PROTEINS_full |
|---|---|---|---|---|---|
| GH | **0.67** ± 0.01 | **0.77** ± 0.03 | 0.72 ± 0.01 | 0.83 ± 0.01 | **0.72** ± 0.01 |
| ML | 0.61 ± 0.01 | 0.53 ± 0.02 | **0.73** ± 0.01 | **0.83 ± 0.02** | 0.6 |
| PK | 0.15 ± 0.01 | 0.47 ± 0.03 | 0.49 ± 0.02 | 0.79 | 0.6 |
| SM | TIMEOUT | TIMEOUT | TIMEOUT | 0.81 ± 0.01 | MEM_RSS |
| SP | TIMEOUT | TIMEOUT | TIMEOUT | TIMEOUT | TIMEOUT |

Πίνακας 4.11: Μέσοι όροι και διακυμάνσεις των χρόνων εκτέλεσης που αντιστοιχούν στις καλύτερες τιμές της μετρικής ευστοχίας για 10 επαναλήψεις 10-fold cross validation στα σύνολα δεδομένων με **συνεχείς** επισημειώσεις, όπως καταγράφονται στον πίνακα 4.11. Οι χρόνοι που τονίζονται αφορούν αυτούς με τα καλύτερα σκορ ευστοχίας. Τα κελιά που σημειώνονται με '-' αφορούν τιμές που η εκτέλεση δεν ολοκληρώθηκε, καθώς υπερέβη την μέγιστη επιτρεπτή μνήμη και με "TIMEOUT" όταν ξεπέρασε το μέγιστο επιτρεπτό χρόνο.

| | ENZYMES | SYNTHETICnew | Synthie | BZR | PROTEINS_full |
|---|---|---|---|---|---|
| GH | **16.61 m** | **13.91 m ± 3.84 s** | 24.37 m | 4.41 m ± 0.37 s | **5.28 h** |
| ML | 20.54 s | 6.64 s | **11.87 s** | **8.76 s** | 1.43 m |
| PK | 15.91 s ± 1.82 s | 14.31 s ± 1.8 s | 30.6 s ± 6.45 s | 10.4 s | 1.74 m ± 3.39 s |
| SM | TIMEOUT | TIMEOUT | TIMEOUT | 8.03 h | - |
| SP | TIMEOUT | TIMEOUT | TIMEOUT | TIMEOUT | TIMEOUT |

Πίνακας 4.12: Μέσοι όροι και διακυμάνσεις της μέγιστης τιμής μνήμης από τις εκτελέσεις που αντιστοιχούν στις καλύτερες τιμές της μετρικής ευστοχίας για 10 επαναλήψεις 10-fold cross validation στα σύνολα δεδομένων με **συνεχείς** επισημειώσεις όπως φαίνονται στον πίνακα 4.11. Οι χρόνοι που τονίζονται αφορούν αυτούς με τα καλύτερα σκορ ευστοχίας. Τα κελιά που σημειώνονται με '-' αφορούν τιμές που η εκτέλεση διακόπηκε καθώς υπερέβη τον μέγιστο επιτρεπτό, ενώ με "MEMORY-OUT" όταν υπερέβη την μέγιστη επιτρεπτή μνήμη.

| | ENZYMES | SYNTHETICnew | Synthie | BZR | PROTEINS_full |
|---|---|---|---|---|---|
| GH | **0.41G ± 0.02M** | **0.22G ± 0.02M** | 0.31G | 0.18G ± 0.41M | **1.74G** |
| ML | 0.19G | 0.24G | **0.29G** | **0.14G** | 0.39G |
| PK | 0.22G ± 4.63M | 0.23G ± 3.87M | 0.3G ± 7.64M | 0.14G | 0.53G ± 6.16M |
| SM | - | - | - | 0.2G | MEMORY-OUT |
| SP | - | - | - | - | - |

Πίνακας 4.13: Μέσοι όροι και διακυμάνσεις της μέγιστης τιμής μνήμης από τις εκτελέσεις που αντιστοιχούν στις καλύτερες τιμές της μετρικής ευστοχίας για 10 επαναλήψεις 10-fold cross validation στα σύνολα δεδομένων με **συνεχείς** επισημειώσεις όπως φαίνονται στον πίνακα 4.11. Οι χρόνοι που τονίζονται αφορούν αυτούς με τα καλύτερα σκορ ευστοχίας. Τα κελιά που σημειώνονται με '-' αφορούν τιμές που η εκτέλεση διακόπηκε καθώς υπερέβη τον μέγιστο επιτρεπτό, ενώ με "MEMORY-OUT" όταν υπερέβη την μέγιστη επιτρεπτή μνήμη.

χρόνο σε όλα τα σύνολα δεδομένων.

Ο χαμηλός χρόνος που φαίνεται να παρουσιάζει ο πυρήνας ML οφείλεται στο γεγονός ότι το καλύτερο αποτέλεσμα επιτυγχάνεται στην περίπτωση όπου $N = 0$. Κάτι τέτοιο φαίνεται να συμβαίνει, λόγω του προβλήματος της υπεροχής της διαγωνίου (diagonal dominance) της μήτρας πυρήνα, που δεν εμφανίζεται στην περίπτωση που τα διανύσματα χαρακτηριστικών προκύπτουν ως *one-hot vectors* των διακριτών επισημειώσεων. Κάτι τέτοιο φαίνεται να συμβαίνει, μιας και η πληροφορία θέσης των μορίων π.χ. στο σύνολο δεδομένων ENZYMES είναι πολύ *συγκεκριμένη*, οδηγώντας στην έντονη ελάττωση της ομοιότητας μεταξύ δύο διαφορετικών γράφων (σε σχέση με τον εαυτό τους) καθώς ο πυρήνας συνυπολογίζει νέες κλίμακες. Όσον αφορά την μνήμη και τους χρόνους, βλέπουμε πως παρότι ο πυρήνας GH είναι ικανοποιητικός παρουσιάζει μεγάλο overhead όσον αφορά το σύνολο δεδομένων PROTEINS_full τόσο σε μνήμη όσο και σε χώρο παρά τα βέλτιστα αποτελέσματά του.

## 4.4 Σύνοψη Αποτελεσμάτων

Η εκτέλεση των παραπάνω πειραμάτων ήταν μία διαδικασία που διήρκεσε περίπου τρεις μήνες. Μέσα σε αυτό το διάστημα διορθώθηκαν διάφορα λάθη στην υλοποίηση των πυρήνων, ενώ έγινε εκτενής σύγκριση τους με τα αποτελέσματα και τις υλοποιήσεις άλλων πακέτων για την ταυτοποίηση αντιστοιχιών και την διάψευση ύπαρξης προβλημάτων, στην αντίθετη περίπτωση. Συνολικά τόσο σε ταχύτητα υλοποίησης όσο και σε χρόνο εκτέλεσης οι πυρήνες CORE-WL-VH, WL-VH, WL-PM, WL-SP, ML, GH φαίνεται να παρουσιάζουν τα καλύτερα αποτελέσματα στις περισσότερες περιπτώσεις. Από την άλλη τόσο το πρόβλημα της *υπεροχής της διαγωνίου* (diagonal dominance) όσο και το *φράγμα ευστοχίας* (accuracy gap) φαίνεται να απασχολούν και αυτήν την κατηγορία προβλημάτων, όπου ακόμα και οι φυσικές αναπαραστάσεις πολλών αντικειμένων σε μορφή γράφων δεν είναι ικανές, προκειμένου αυτά να είναι απόλυτα διαχωρίσιμα ή να πετυχαίνουν την απαραίτητη εκφραστική ευστοχία, για μία μέθοδο πυρήνα. Η καλή επίδοση των σκελετών πυρήνα φαίνεται να συμβαίνει, καθώς η αναπαράσταση των γράφων σε φλοιούς (δομής, συνεκτικότητας, κλπ) ενισχύει την εκφραστικότητα ενός πυρήνα, κλιμακώνοντας (ή υπο-κλιμακώνοντας αντίστοιχα) ταυτόχρονα τις τιμές ομοιότητας, αφού αυτές εξετάζονται σε ένα καλύτερα οργανωμένο πολλαπλάσιο των δεδομένων. Παράλληλα οι αλγόριθμοι PM, ML που έχουν προκύψει από την μεταφορά αλγορίθμων της όρασης υπολογιστών, δίνουν ίσως μία ένδειξη πως η γεωμετρική διάρθρωση μίας εικόνας και η δομή ενός γράφου ίσως να μπορούν να ειδωθούν υπό το ίδιο ερευνητικό πρίσμα, όσον αφορά την αναγκαιότητα μετάβασης από το τοπικό στο γενικό, κατά την ανάπτυξη νέων πυρήνων.

Η αδυναμία των πυρήνων γράφων να ξεπεράσουν το φράγμα ευστοχίας ή αυτό της επικράτησης της διαγωνίου, δεν φαίνεται να είναι αυτό της ανεπάρκειας τους να λύσουν το πρόβλημα του ισομορφισμού. Είναι αυτό της αδυναμίας τους να *διαχωρίσουν* τις δομές εκείνες που φαίνονται καθοριστικές, για το εκάστοτε πρόβλημα ταξινόμησης. Από τη στιγμή που αναπαραστάσεις τους στους χώρους Hilbert δεν είναι πάντα άμεσες, δεν μπορούν σε αρκετές περιπτώσεις να χρησιμοποιηθούν μέθοδοι όπως αυτές των πολύ δημοφιλών νευρωνικών δικτύων βαθιάς εκμάθησης (deep learning), μιας και αυτές χρειάζονται τις αναπαραστάσεις όλων των γράφων

σε ένα διανυσματικό χώρο (graph embeddings). Η μη-επιβλεπόμενη φύση της προσέγγισης τους φαίνεται να είναι και το ίδιο τους το όριο (όσον αφορά τα ίδια τα προβλήματα ταξινόμησης), που ίσως να μην μπορεί να γεφυρωθεί τόσο υπολογιστικά όσο και θεωρητικά. Ως επακόλουθο κρίνουμε πως η έρευνα στον χώρο των **επιβλεπόμενων** πυρήνων γράφων, φαίνεται να έχει ιδιαίτερο ερευνητικό ενδιαφέρον.

# Κεφάλαιο 5

# Συμπεράσματα

Λόγω της αύξησης των δεδομένων που εμφανίζονται με αναπαράσταση γράφων, οι υπολογιστικά εφικτές λύσεις σε προβλήματα μηχανικής που εξετάζουν δεδομένα αυτής της μορφής, φαίνονται ιδιαίτερα δελεαστικές. Λόγω της μεγάλης ανάλυσης που αφορά τις μεθόδους πυρήνα και το εύρος εύρωστων αλγορίθμων που μπορούν να λύσουν προβλήματα μηχανικής μάθησης, αρκετές από τις προσπάθειες επιτέλεσης μηχανικής μάθησης με γράφους εστιάστηκαν στον χώρο των πυρήνων γράφων. Παρόλο που το πρόβλημα των πυρήνων γράφων είναι ένα πολυμελετημένο πρόβλημα στην σύγχρονη βιβλιογραφία με αρκετές πολύ αποτελεσματικές προσεγγίσεις, δεν είχε επιχειρηθεί στο παρελθόν η συλλογή του σε ένα ώριμο υπολογιστικό πακέτο ευρείας χρήσης σε μία εύκολη στην χρήση γλώσσα προγραμματισμού. Ως επακόλουθο, επιλέξαμε να σχεδιάσουμε το πακέτο GraKeL το οποίο περιέχει τους πιο πρωτοποριακούς πυρήνες γράφων που εμφανίζονται στην βιβλιογραφία των τελευταίων χρόνων σε γλώσσα Python. Περιέχει συμπληρωματικό εγχειρίδιο χρήσης και διανέμεται σε ανοιχτό κώδικα με συνεχή ενσωμάτωση σε ανοιχτό αποθετήριο στον ιστό. Ταυτόχρονα η συμβατότητα του με την δημοφιλή βιβλιοθήκη επιστημονικού υπολογισμού scikit-learn, παρέχει την δυνατότητα εύκολης ενσωμάτωσης υπάρχοντων έργων μηχανικής μάθησης (όπως η ταξινόμηση ή ή συσταδοποίηση), σε δεδομένα εισόδου που φέρουν την μορφή γράφων. Η πειραματική αξιολόγηση του `GraKeL` φαίνεται ικανοποιητική τόσο σε χρόνους όσο και στα αποτελέσματα εν συγκρίσει με πλήθος εργασιών της βιβλιογραφίας που μελετήσαμε. Από την άλλη το `GraKeL` είναι ένα καινούργιο πακέτο που ίσως αλλάξει μέσα στο μέλλον. Σημαντικό είναι λοιπόν να προτείνουμε μελλοντικές επεκτάσεις.

## 5.1 Μελλοντικές Επεκτάσεις

Αναφέρουμε στην συνέχεια ορισμένα θέματα, τα οποία θα μπορούσαν να είναι αντικείμενο πιθανών επεκτάσεων.

**Η Κλάση `Graph`** Η κλάση `Graph` σχεδιάστηκε προκειμένου να στηρίζει αποκλειστικά τους γράφους του πακέτου προσφέροντας ένα μικρό εύρος λειτουργικών συναρτήσεων που είναι κοινοί μεταξύ όλων των πυρήνων. Λόγω της αρκετά διαφορετικής τους φύσης και της σύνδεσης

κάθε πυρήνα με ένα είδος αναπαράστασης γράφου (π.χ. ως λίστα ακμών, ως πίνακας γειτνίασης κλπ) η κλάση αυτή συνδέθηκε με ένα μικρό πλήθος λειτουργιών, που καλύπτουν την ανάγκη ενός γενικού τύπου αναπαράστασης. Η ευρωστία, η ελαχιστοποίηση του χρόνου και της μνήμης και τέλος η αφαίρεση των λαθών (στο βαθμό που δεν έχει ήδη γίνει) σε αυτήν την κλάση φαίνεται ένα πολύ σημαντικό πρόβλημα. Έτσι η περαιτέρω μελέτη, ελαχιστοποίηση και υπολογιστική βελτιστοποίηση του συνολικού κώδικα αυτού του αντικειμένου φαίνεται να είναι στη βάση μελλοντικών επεκτάσεων τόσο για την βελτίωση της απόδοσης υπάρχοντων πυρήνων όσο και για την δυνατότητα υποστήριξης νέων.

**Πυρήνες και με Συνεχείς και Διακριτές Επισημειώσεις** Αυτήν την στιγμή η είσοδος κάθε αντικειμένου φαίνεται να υποστηρίζει είτε συνεχείς είτε διακριτές επισημειώσεις στους κόμβους και στις ακμές. Παρόλο που κάθε διακριτή αναπαράσταση επισημειώσεων μπορεί να μετασχηματιστεί σε μία συνεχή, κάτι τέτοιο δεν φαίνεται αποδοτικό για πυρήνες με μεγάλο πλήθος διαφορετικών επισημειώσεων. Συνεπώς τόσο η μορφή της εισόδου και κατ΄ επέκταση η κλάση `Graph` όσο και οι αντίστοιχοι πυρήνες πρέπει να προσαρμοστούν για να συμπεριλαμβάνουν αυτήν την κοινή πληροφορία, σε υπάρχοντες και μελλοντικούς πυρήνες.

**Πιο Σύνθετοι Σκελετοί Πυρήνα** Όσον αφορά τους σκελετούς πυρήνα της παρούσας διπλωματικής θεωρήσαμε πως όλοι χρησιμοποιούν στη βάση τους έναν άλλο πυρήνα γράφων. Κάτι τέτοιο δεν φαίνεται να ισχύει για ένα άλλο πλήθος σκελετών πυρήνα στη βιβλιογραφία, όπως π.χ. οι *deep graph kernels* [91] ή οι *optimal assignment kernels* [48], οι οποίοι χρησιμοποιούν δεδομένα που οι υπάρχοντες πυρήνες παράγουν σε κάποιο ενδιάμεσο στάδιο του υπολογισμού της μήτρας πυρήνα, με βάση το θεωρητικό μοντέλο που ακολουθούν. Κάτι τέτοιο φυσικά απαιτεί μία πιο αντικειμενοστρεφή οργάνωση των πυρήνων.

**Πιο Αντικειμενοστρεφής Οργάνωση** Το παρόν λογισμικό δεν έχει οργανωθεί εκτενώς σε μία θεωρητική βάση. Για παράδειγμα οι πυρήνες δεν διαχωρίζονται με βάση το αν είναι π.χ. R-Convolutional [84] ή Optimal-Assignment [24]. Στην περίπτωση αυτή ένα πλήθος συναρτήσεων που αφορούν λειτουργίες των R-Convolutional πυρήνων όπως ο υπολογισμός πινάκων χαρακτηριστικών για κάθε σύνολο γράφων (αντί για τιμών πυρήνα), ενώ συμπληρωματικά, στην περίπτωση των Optimal-Assignment, άλλες λειτουργίες όπως ο υπολογισμός μίας ιεραρχίας ομοιοτήτων μεταξύ όλων των γράφων, μπορούν να οριστούν. Κατ΄ επέκταση οι πυρήνες, κληρονομώντας κλάσεις όπως η `RConvolutional` ή η `OptimalAssignment` κ.ο.κ., θα πρέπει να υλοποιούν αποτελεσματικά ένα εύρος επιπρόσθετων λειτουργιών. Ερευνητικό ενδιαφέρον παρουσιάζει σε αυτό το σημείο η συνέχιση συμβατότητας με το scikit-learn και η σχεδίαση όλων των παραπάνω λειτουργιών σε σχέση με τα `fit`, `fit_transform` και `transform`, γεγονός που φαίνεται ιδιαίτερα σημαντικό.

**Περισσότερη Συμβατότητα με τους Πρωτότυπους Κώδικες** Από την εκτενή μελέτη ενός μεγάλου εύρους της βιβλιογραφίας, διαπιστώσαμε ότι η αποτύπωση της θεωρητικής περιγραφής του υπολογισμού ενός πυρήνα με την ουσιαστική, αυτής δηλαδή που εμφανίζεται σε επίπεδο υλοποίησης, ήταν θολή. Από μικρές λεπτομέρειες 'που κάνουν την

διαφορά' μέχρι υπολογιστικά μοτίβα που μειώνουν σημαντικά την πολυπλοκότητα, καθώς και τεχνάσματα που αφορούν την ίδια την γλώσσα προγραμματισμού που χρησιμοποιούν για την υλοποίηση τους ικανά να κάνουν τον πυρήνα να φαίνεται πιο αποτελεσματικός στην πράξη από υπάρχοντες. Η χρήση και η προσαρμογή καίριων σημείων του κώδικα άλλων γλωσσών και η συμβατότητα με τα υπάρχοντα (στο βαθμό που δεν έχει ήδη μελετηθεί) παρουσιάζει ένα ερευνητικό ενδιαφέρον και αποτελεί μία πρόκληση που πρέπει να λύσει κάθε μοντέρνα βιβλιοθήκη επιστημονικού υπολογισμού.

Εν κατακλείδι, το GraKeL είναι ένα έργο που η εξέλιξη του φαίνεται να προκαλεί ιδιαίτερο ερευνητικό ενδιαφέρον, ενώ έχει μόλις διανύσει τα πρώτα του στέρεα βήματα.

# Βιβλιογραφία


[1] F. Aiolli et al. "Fast On-line Kernel Learning for Trees". In: *Sixth International Conference on Data Mining (ICDM'06)*. Dec. 2006, pp. 787–791. DOI: 10.1109/ICDM.2006.69.

[2] M. A. Aizerman, E. A. Braverman, and L. Rozonoer. "Theoretical foundations of the potential function method in pattern recognition learning." In: *Automation and Remote Control,* Automation and Remote Control, 25. 1964, pp. 821–837.

[3] I. Alvarez-Hamelin et al. "Large scale networks fingerprinting and visualization using the k-core decomposition". In: *Advances in Neural Information Processing Systems* 18 (2006), pp. 41–50.

[4] N. Aronszajn. "Theory of Reproducing Kernels". In: *Transactions of the American Mathematical Society* 68.3 (1950), pp. 337–404. ISSN: 00029947. URL: http://www.jstor.org/stable/1990404.

[5] Francis R. Bach. "Graph Kernels Between Point Clouds". In: *Proceedings of the 25th International Conference on Machine Learning*. ICML '08. Helsinki, Finland: ACM, 2008, pp. 25–32. ISBN: 978-1-60558-205-4. DOI: 10.1145/1390156.1390160. URL: http://doi.acm.org/10.1145/1390156.1390160.

[6] L. Bai, E. R. Hancock, and P. Ren. "Jensen-Shannon graph kernel using information functionals". In: *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*. Nov. 2012, pp. 2877–2880.

[7] V. Batagelj and M. Zaveršnik. "Fast algorithms for determining (generalized) core groups in social networks". In: *Advances in Data Analysis and Classification* 5.2 (2011), pp. 129–145.

[8] Kristin Bennett and O.L. Mangasarian. "Robust Linear Programming Discrimination Of Two Linearly Inseparable Sets". In: 1 (Jan. 2002).

[9] "Book Reviews". In: *Mathematical Methods of Operations Research* 53.2 (June 2001), pp. 349–352. ISSN: 1432-5217. DOI: 10.1007/s001860000083. URL: https://doi.org/10.1007/s001860000083.

[10] K. M. Borgwardt and H. Kriegel. "Shortest-path kernels on graphs". In: *Proceedings of the 5th International Conference on Data Mining*. 2005, pp. 74–81.

[11] Karsten M. Borgwardt and Hans-Peter Kriegel. "Shortest-Path Kernels on Graphs". In: *Proceedings of the Fifth IEEE International Conference on Data Mining*. ICDM '05. Washington, DC, USA: IEEE Computer Society, 2005, pp. 74–81. ISBN: 0-7695-2278-5. DOI: 10.1109/ICDM.2005.132. URL: http://dx.doi.org/10.1109/ICDM.2005.132.

[12] Karsten Michael Borgwardt. "Graph Kernels". July 2007. URL: http://nbn-resolving.de/urn:nbn:de:bvb:19-71691.

[13] Karsten M Borgwardt et al. "Protein function prediction via graph kernels". In: *Bioinformatics* 21.suppl 1 (2005), pp. i47–i56.

[14] Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. "A Training Algorithm for Optimal Margin Classifiers". In: *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*. COLT '92. Pittsburgh, Pennsylvania, USA: ACM, 1992, pp. 144–152. ISBN: 0-89791-497-X. DOI: 10.1145/130385.130401. URL: http://doi.acm.org/10.1145/130385.130401.

[15] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. New York, NY, USA: Cambridge University Press, 2004. ISBN: 0521833787.

[16] Stephen Cass. *Top 10 Programming Languages for 2017*. https://spectrum.ieee.org/computing/software/the-2017-top-programming-languages. Accessed: 2018-07-07.

[17] Corinna Cortes and Vladimir Vapnik. "Support-Vector Networks". In: *Machine Learning* 20.3 (Sept. 1995), pp. 273–297. ISSN: 1573-0565. DOI: 10.1023/A:1022627411411. URL: https://doi.org/10.1023/A:1022627411411.

[18] Fabrizio Costa and Kurt De Grave. "Fast Neighborhood Subgraph Pairwise Distance Kernel". In: *Proceedings of the 26th International Conference on Machine Learning*. 2010, pp. 255–262.

[19] Fabrizio Costa and Kurt De Grave. "Fast Neighborhood Subgraph Pairwise Distance Kernel". In: *Proceedings of the 27th International Conference on International Conference on Machine Learning*. ICML'10. Haifa, Israel: Omnipress, 2010, pp. 255–262. ISBN: 978-1-60558-907-7. URL: http://dl.acm.org/citation.cfm?id=3104322.3104356.

[20] Eric H. Davidson et al. "A Genomic Regulatory Network for Development". In: *Science* 295.5560 (2002), pp. 1669–1678. ISSN: 0036-8075. DOI: 10.1126/science.1069883. eprint: http://science.sciencemag.org/content/295/5560/1669.full.pdf. URL: http://science.sciencemag.org/content/295/5560/1669.

[21] Paul D. Dobson and Andrew J. Doig. "Distinguishing Enzyme Structures from Non-enzymes Without Alignments". In: *Journal of Molecular Biology* 330.4 (2003), pp. 771–783. ISSN: 0022-2836. DOI: https://doi.org/10.1016/S0022-2836(03)00628-4. URL: http://www.sciencedirect.com/science/article/pii/S0022283603006284.

[22] Aasa Feragen et al. "Scalable kernels for graphs with continuous attributes". In: *Advances in Neural Information Processing Systems 26*. Ed. by C. J. C. Burges et al. Curran Associates, Inc., 2013, pp. 216–224. URL: http://papers.nips.cc/paper/5155-scalable-kernels-for-graphs-with-continuous-attributes.pdf.

[23] Aasa Feragen et al. "Scalable kernels for graphs with continuous attributes". In: *Advances in Neural Information Processing Systems*. 2013, pp. 216–224.

[24] Holger Fröhlich et al. "Optimal Assignment Kernels for Attributed Molecular Graphs". In: *Proceedings of the 22Nd International Conference on Machine Learning*. ICML '05. Bonn, Germany: ACM, 2005, pp. 225–232. ISBN: 1-59593-180-5. DOI: 10.1145/1102351.1102380. URL: http://doi.acm.org/10.1145/1102351.1102380.

[25] T. Gärtner, P. Flach, and S. Wrobel. "On Graph Kernels: Hardness Results and Efficient Alternatives". In: *Learning Theory and Kernel Machines*. 2003, pp. 129–143.

[26] Thomas Gärtner. "A Survey of Kernels for Structured Data". In: *SIGKDD Explor. Newsl.* 5.1 (July 2003), pp. 49–58. ISSN: 1931-0145. DOI: 10.1145/959242.959248. URL: http://doi.acm.org/10.1145/959242.959248.

[27] Thomas Gärtner, Peter Flach, and Stefan Wrobel. "On graph kernels: Hardness results and efficient alternatives". In: *IN: CONFERENCE ON LEARNING THEORY*. 2003, pp. 129–143.

[28] C. Giatsidis et al. "CORECLUSTER: A Degeneracy Based Graph Clustering Framework". In: *Proceedings of the 28th AAAI Conference on Artificial Intelligence*. 2014, pp. 44–50.

[29] Pierre-Louis Giscard and Richard C. Wilson. "The All-Paths and Cycles Graph Kernel". In: *CoRR* abs/1708.01410 (2017). arXiv: 1708.01410. URL: http://arxiv.org/abs/1708.01410.

[30] David Gitchell and Nicholas Tran. "Sim: A Utility for Detecting Similarity in Computer Programs". In: *The Proceedings of the Thirtieth SIGCSE Technical Symposium on Computer Science Education*. SIGCSE '99. New Orleans, Louisiana, USA: ACM, 1999, pp. 266–270. ISBN: 1-58113-085-6. DOI: 10.1145/299649.299783. URL: http://doi.acm.org/10.1145/299649.299783.

[31] Goran Glavaš and Jan Šnajder. "Recognizing identical events with graph kernels". In: 2 (Jan. 2013), pp. 797–803.

[32] John C Gower. "A general coefficient of similarity and some of its properties". In: *Biometrics* (1971), pp. 857–871.

[33] Kristen Grauman and Trevor Darrell. "The Pyramid Match Kernel: Efficient Learning with Sets of Features". In: *The Journal of Machine Learning Research* 8 (2007), pp. 725–760.

[34] Nick Coghlan Guido van Rossum Barry Warsaw. *PEP 8 Style Guide for Python Code*. 2001. URL: https://www.python.org/dev/peps/pep-0008/#overriding-principle (visited on 06/05/2001).

[35] Masahiro Hattori et al. "Development of a Chemical Structure Comparison Method for Integrated Analysis of Chemical and Genomic Information in the Metabolic Pathways". In: *Journal of the American Chemical Society* 125.39 (2003). PMID: 14505407, pp. 11853–11865. DOI: 10.1021/ja036030u. eprint: https://doi.org/10.1021/ja036030u. URL: https://doi.org/10.1021/ja036030u.

[36] David Haussler. "Convolution Kernels on Discrete Structures". In: 1999.

[37] David Haussler. *Convolution Kernels on Discrete Structures*. 1999.

[38] Linus Hermansson et al. "Entity disambiguation in anonymized graphs using graph kernels". In: *CIKM*. 2013.

[39] S. Hido and H. Kashima. "A Linear-Time Graph Kernel". In: *2009 Ninth IEEE International Conference on Data Mining*. Dec. 2009, pp. 179–188. DOI: 10.1109/ICDM.2009.30.

[40] Tamás Horváth, Thomas Gärtner, and Stefan Wrobel. "Cyclic Pattern Kernels for Predictive Graph Mining". In: *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '04. Seattle, WA, USA: ACM, 2004, pp. 158–167. ISBN: 1-58113-888-1. DOI: 10.1145/1014052.1014072. URL: http://doi.acm.org/10.1145/1014052.1014072.

[41] Vinay Jethava et al. "Lovász $\vartheta$ function, SVMs and Finding Dense Subgraphs". In: *The Journal of Machine Learning Research* 14.1 (2013), pp. 3495–3536.

[42] Fredrik Johansson et al. "Global graph kernels using geometric embeddings". In: *Proceedings of the 31st International Conference on Machine Learning*. 2014, pp. 694–702.

[43] Minoru Kanehisa and Susumu Goto. "KEGG: Kyoto Encyclopedia of Genes and Genomes". In: *Nucleic Acids Research* 28.1 (2000), pp. 27–30. DOI: 10.1093/nar/28.1.27. eprint: /oup/backfile/content_public/journal/nar/28/1/10.1093_nar_28.1.27/1/280027.pdf. URL: http://dx.doi.org/10.1093/nar/28.1.27.

[44] H. Kashima, K. Tsuda, and A. Inokuchi. "Marginalized Kernels Between Labeled Graphs". In: *Proceedings of the 20th Conference in Machine Learning*. 2003, pp. 321–328.

[45] Tetsuya Kataoka and Akihiro Inokuchi. "Hadamard Code Graph Kernels for Classifying Graphs". In: *Proceedings of the 5th International Conference on Pattern Recognition Applications and Methods*. ICPRAM 2016. Rome, Italy: SCITEPRESS - Science and Technology Publications, Lda, 2016, pp. 24–32. ISBN: 978-989-758-173-1. DOI: 10.5220/0005634700240032. URL: http://dx.doi.org/10.5220/0005634700240032.

[46] Kristian Kersting et al. *Benchmark Data Sets for Graph Kernels*. 2016. URL: http://graphkernels.cs.tu-dortmund.de.

[47] Risi Kondor and Horace Pan. "The Multiscale Laplacian Graph Kernel". In: *Advances in Neural Information Processing Systems*. 2016, pp. 2990–2998.

[48] Nils M. Kriege, Pierre-Louis Giscard, and Richard Wilson. "On Valid Optimal Assignment Kernels and Applications to Graph Classification". In: *Advances in Neural Information Processing Systems 29*. Ed. by D. D. Lee et al. Curran Associates, Inc., 2016, pp. 1623–1631. URL: http://papers.nips.cc/paper/6166-on-valid-optimal-assignment-kernels-and-applications-to-graph-classification.pdf.

[49] Nils Kriege and Petra Mutzel. "Subgraph Matching Kernels for Attributed Graphs". In: *ICML*. 2012.

[50] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. "Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories". In: *Proceedings of the 2006 Conference on Computer Vision and Pattern Recognition*. Vol. 2. 2006, pp. 2169–2178.

[51] Giorgio Levi. "A note on the derivation of maximal common subgraphs of two directed or undirected graphs". In: *Calcolo* 9.4 (1973), p. 341.

[52] László Lovász. "On the Shannon capacity of a graph". In: *IEEE Transactions on Information Theory* 25.1 (1979), pp. 1–7.

[53] P. Mahé and J. Vert. "Graph kernels based on tree patterns for molecules". In: *Machine learning* 75.1 (2009), pp. 3–35.

[54] Pierre Mahe and Jean-Philippe Vert. "Graph kernels based on tree patterns for molecules". In: *Machine Learning* 75.1 (Apr. 2009), pp. 3–35. ISSN: 1573-0565. DOI: 10.1007/s10994-008-5086-2. URL: https://doi.org/10.1007/s10994-008-5086-2.

[55] Pierre Mahé et al. "Extensions of marginalized graph kernels". In: *Proceedings of the 21st International Conference on Machine Learning*. 2004, p. 70.

[56] Giovanni Da San Martino, Nicolò Navarin, and Alessandro Sperduti. "A Tree-Based Kernel for Graphs". In: *SDM*. 2012.

[57] D. Matula and L. Beck. "Smallest-last Ordering and Clustering and Graph Coloring Algorithms". In: *Journal of the ACM* 30.3 (1983), pp. 417–427.

[58] Christopher Morris et al. "Faster Kernels for Graphs with Continuous Attributes via Hashing". In: *CoRR* abs/1610.00064 (2016). arXiv: 1610.00064. URL: http://arxiv.org/abs/1610.00064.

[59] Alessandro Moschitti. "Efficient Convolution Kernels for Dependency and Constituent Syntactic Trees". In: *Machine Learning: ECML 2006*. Ed. by Johannes Fürnkranz, Tobias Scheffer, and Myra Spiliopoulou. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 318–329. ISBN: 978-3-540-46056-5.

[60] Marion Neumann et al. "Propagation Kernels: Efficient Graph Kernels from Propagated Information". In: *Mach. Learn.* 102.2 (Feb. 2016), pp. 209–245. ISSN: 0885-6125. DOI: 10.1007/s10994-015-5517-9. URL: http://dx.doi.org/10.1007/s10994-015-5517-9.

[61] G. Nikolentzos et al. "A Degeneracy Framework for Graph Similarity". In: *Proceedings of the 27th International Joint Conference on Artificial Intelligence*. 2018.

[62] Giannis Nikolentzos, Polykarpos Meladianos, and Michalis Vazirgiannis. "Matching Node Embeddings for Graph Similarity." In: *Proceedings of the 31st AAAI Conference on Artificial Intelligence*. 2017, pp. 2429–2435.

[63] Francesco Orsini, Paolo Frasconi, and Luc De Raedt. "Graph Invariant Kernels". In: *Proceedings of the 24th International Conference on Artificial Intelligence*. IJCAI'15. Buenos Aires, Argentina: AAAI Press, 2015, pp. 3756–3762. ISBN: 978-1-57735-738-4. URL: http://dl.acm.org/citation.cfm?id=2832747.2832773.

[64] Fabian Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12.Oct (2011), pp. 2825–2830.

[65] Nataša Pržulj. "Biological network comparison using graphlet degree distribution". In: *Bioinformatics* 23.2 (2007), e177–e183.

[66] Jan Ramon and Thomas Gärtner. "Expressivity versus efficiency of graph kernels". In: *Proceedings of the First International Workshop on Mining Graphs, Trees and Sequences*. 2003, pp. 65–74.

[67] Kaspar Riesen and Horst Bunke. "IAM Graph Database Repository for Graph Based Pattern Recognition and Machine Learning". In: *Structural, Syntactic, and Statistical Pattern Recognition*. Ed. by Niels da Vitoria Lobo et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 287–297. ISBN: 978-3-540-89689-0.

[68] François Rousseau, Emmanouil Kiagias, and Michalis Vazirgiannis. "Text Categorization as a Graph Classification Problem". In: *ACL*. 2015.

[69] Bernhard Schölkopf, Ralf Herbrich, and Alex J. Smola. "A Generalized Representer Theorem". In: *Proceedings of the 14th Annual Conference on Computational Learning Theory and and 5th European Conference on Computational Learning Theory*. COLT '01/EuroCOLT '01. London, UK, UK: Springer-Verlag, 2001, pp. 416–426. ISBN: 3-540-42343-5. URL: http://dl.acm.org/citation.cfm?id=648300.755324.

[70] Bernhard Scholkopf and Alexander J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge, MA, USA: MIT Press, 2001. ISBN: 0262194759.

[71] S. Seidman. "Network Structure and Minimum Degree". In: *Social networks* 5.3 (1983), pp. 269–287.

[72] N. Shervashidze et al. "Efficient Graphlet Kernels for Large Graph Comparison". In: *Proceedings of the International Conference on Artificial Intelligence and Statistics*. 2009, pp. 488–495.

[73] N. Shervashidze et al. "Weisfeiler-Lehman Graph Kernels". In: *The Journal of Machine Learning Research* 12 (2011), pp. 2539–2561.

[74] Mahito Sugiyama and Karsten Borgwardt. "Halting in Random Walk Kernels". In: *Advances in Neural Information Processing Systems*. 2015, pp. 1639–1647.

[75] Mahito Sugiyama et al. "graphkernels: R and Python packages for graph comparison". In: *Bioinformatics* 34.3 (2017), pp. 530–532.

[76] Sanjay Joshua Swamidass et al. "Kernels for small molecules and the prediction of mutagenicity, toxicity and anti-cancer activity". In: *Bioinformatics* 21 Suppl 1 (2005), pp. i359–68.

[77] M. Takashima et al. "A circuit comparison system with rule-based functional isomorphism checking". In: *25th ACM/IEEE, Design Automation Conference.Proceedings 1988.* June 1988, pp. 512–516. DOI: 10.1109/DAC.1988.14808.

[78] Hannu Toivonen et al. "Statistical evaluation of the Predictive Toxicology Challenge 2000–2001". In: *Bioinformatics* 19.10 (2003), pp. 1183–1193. DOI: 10.1093/bioinformatics/btg130. eprint: /oup/backfile/content_public/journal/bioinformatics/19/10/10.1093/bioinformatics/btg130/2/btg130.pdf. URL: http://dx.doi.org/10.1093/bioinformatics/btg130.

[79] Evgeni Tsivtsivadze et al. "Semantic Graph Kernels for Automated Reasoning". In: *Proceedings of the 2011 SIAM International Conference on Data Mining*, pp. 795–803. DOI: 10.1137/1.9781611972818.68. eprint: https://epubs.siam.org/doi/pdf/10.1137/1.9781611972818.68. URL: https://epubs.siam.org/doi/abs/10.1137/1.9781611972818.68.

[80] Nick M. Vandewiele et al. "Genesys: Kinetic model construction using chemo-informatics". In: *Chemical Engineering Journal* 207-208 (2012). 22nd International Symposium on Chemical Reaction Engineering (ISCRE 22), pp. 526–538. ISSN: 1385-8947. DOI: https://doi.org/10.1016/j.cej.2012.07.014. URL: http://www.sciencedirect.com/science/article/pii/S1385894712009059.

[81] V Vapnik and A Lerner. "Pattern Recognition using Generalized Portrait Method". In: *Automation and Remote Control* 24 (1963).

[82] S. V. N. Vishwanathan and Alexander J. Smola. "Fast Kernels for String and Tree Matching". In: *Proceedings of the 15th International Conference on Neural Information Processing Systems*. NIPS'02. Cambridge, MA, USA: MIT Press, 2002, pp. 585–592. URL: http://dl.acm.org/citation.cfm?id=2968618.2968691.

[83] S. V. N. Vishwanathan and Alexander J. Smola. "Fast Kernels for String and Tree Matching". In: *Proceedings of the 15th International Conference on Neural Information Processing Systems*. NIPS'02. Cambridge, MA, USA: MIT Press, 2002, pp. 585–592. URL: http://dl.acm.org/citation.cfm?id=2968618.2968691.

[84] S. V. N. Vishwanathan et al. "Graph Kernels". In: *The Journal of Machine Learning Research* 11 (2010), pp. 1201–1242.

[85] C. Wagner et al. "Malware analysis with graph kernels and support vector machines". In: *2009 4th International Conference on Malicious and Unwanted Software (MALWARE)*. Oct. 2009, pp. 63–68. DOI: 10.1109/MALWARE.2009.5403018.

[86] Nikil Wale, Ian A. Watson, and George Karypis. "Comparison of descriptor spaces for chemical compound retrieval and classification". In: *Knowledge and Information Systems* 14.3 (Mar. 2008), pp. 347–375. ISSN: 0219-3116. DOI: 10.1007/s10115-007-0103-5. URL: https://doi.org/10.1007/s10115-007-0103-5.

[87] Boris Weisfeiler and AA Lehman. "A reduction of a graph to a canonical form and an algebra arising during this reduction". In: *Nauchno-Technicheskaya Informatsia* 2.9 (1968), pp. 12–16.

[88] Tsachy Weissman et al. "Inequalities for the $L_1$ deviation of the empirical distribution". In: *Hewlett-Packard Labs, Tech. Rep* (2003).

[89] Christopher KI Williams and Matthias Seeger. "Using the Nyström Method to Speed Up Kernel Machines". In: *Advances in Neural Information Processing Systems*. 2001, pp. 682–688.

[90] S. Wuchty and E. Almaas. "Peeling the yeast protein network". In: *Proteomics* 5.2 (2005), pp. 444–449.

[91] Pinar Yanardag and S.V.N. Vishwanathan. "Deep Graph Kernels". In: *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '15. Sydney, NSW, Australia: ACM, 2015, pp. 1365–1374. ISBN: 978-1-4503-3664-2. DOI: 10.1145/2783258.2783417. URL: http://doi.acm.org/10.1145/2783258.2783417.

# Γλωσσάριο

| Ελληνικός όρος | Αγγλικός όρος |
|---|---|
| αποθετήριο | repository |
| σταθμίζεται | is weighted |
| σταθμισμένος | weighted |
| συμβολοσειρά | string |
| γράφος | graph |
| γραφίδιο | graphlet |
| διανύσματα χαρακτηριστικών | feature vectors |
| διεπαφή | interface |
| μηχανική μάθηση | machine learning |
| εγχειρίδιο ανάγνωσης | documentation |
| εκφυλισμός | degeneracy |
| εκτελέσιμο | binary |
| εξόρυξη γνώσης από δεδομένα | data mining |
| επιβλεπόμενη | supervised |
| επιστημονικός υπολογισμός | scientific computing |
| επισημείωση | label |
| ισοσταθμισμένες | equally-weighted |
| ισομορφισμός (γράφων) | graph isomorphism |
| κανονική διάταξη | canonical ordering |
| κατακερματισμός | hashing |
| σύγκρουση κατακερματισμών | hash collisions |
| κλίκα | clique |
| κληρονομία | inheritance |
| κόμβος | vertex, node |
| μήτρα | matrix |
| οκνηρό | lazy |
| πολλαπλότητα | manifold |
| πολυσύνολο | multiset |
| πίνακας αντιστοίχησης | associative array |
| πυρήνες γράφων | graph kernels |
| ριζωμένοι | rooted |

| | |
|---|---|
| σύνολα δεδομένων | dataset |
| συνδυαστικό | combinatorial |
| συρραφή (κώδικα) | (code) wrapping |
| συσκευασία (κώδικα) | (code) packaging |
| ταξινομητής μηχανών διανυσμάτων υποστήριξης | support vector machine classifier |
| υψηλό επίπεδο (προγραμματισμού) | high level (programming) |
| χημιοπληροφορική | chemoinformatics |