Εθνικο Μετσοβιο Πολυτεχνειο

Σχολη Ηλεκτρολογων Μηχανικων και Μηχανικων Υπολογιστων

Τομεας Επικοινωνιων, Ηλεκτρονικης και Συστηματων Πληροφορικης

GraKeL: μία βιβλιοθήκη για Πυρήνες Γράφων

Διπλωματικη Εργασια

του

ΣΙΓΛΙΔΗ ΓΙΑΝΝΗ

Εξωτερικός Επιβλέπων: Μιχάλης Βαζιργιάννης
Καθηγητής Ecole Polytechnique, Καθηγητής AUEB

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

Αθήνα, Οκτόμβριος 2018

Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Τομέας Επικοινωνιών, Ηλεκτρονικής και Συστημάτων Πληροφορικής

GraKeL: μία βιβλιοθήκη για Πυρήνες Γράφων

Διπλωματικη Εργασια

του

ΣΙΓΛΙΔΗ ΓΙΑΝΝΗ

Εξωτερικός Επιβλέπων: Μιχάλης Βαζιργιάννης
Καθηγητής Ecole Polytechnique, Καθηγητής AUEB

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

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 29η Ιουλίου 2018.

(Υπογραφή)	*(Υπογραφή)*	*(Υπογραφή)*
.....	.....	.....
Ανδρέας Σταφυλοπάτης	Θεοδώρα Βαρβαρίγου	Ιωάννα Ρουσσάκη
Καθηγητής Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.

Αθήνα, Φεβρουάριος 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 υπολογιστίκης 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 να 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 αντικειμενοστραφή δομή. Η βιβλιοθήκη είναι υλοποιημένη σε γλώσσα προγραμματισμού Python και είναι κατασκευασμένη βάση του πρότυπο της βιβλιοθήκης scikit-learn. Είναι εύκολη στη χρήση και μπορεί να συνδιαστεί 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 όλο το διάστημα που εργαζόμουν μαζί του στο Εργαστήριο LiX, στο Παρίσι. Επίσης 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 .

Περιεχόμενα

Κατάλογος Σχημάτων

Κατάλογος Πινάκων

Κεφάλαιο 1

Εισαγωγή

Η αναπάραστηση γράφων αποτελεί έναν ευέλικτο και περιεκτικό τροπο αναπαράστασης μεταξύ οντοτήτων και των σχέσεων τους σε διάφορα πεδία της επιστήμης και της τεχνολογίας. Τα τελευταία χρόνια η αναπαράσταση δεδομένων με την μορφή γράφων έχει βιώση μία τρομερή άνθιση σε πολλά πεδία, από τα κοινωνικά δίκτυα μλεχρι την βιοπληροφορική. Διάφορα προβλήματα που εγείρουν όλο και πιο πολύ το ενδιαφέρον, εγκαλούν την χρήση τεχνικών μηχανική μάθησης σε δεδομένα που έχουν αναπαρασταθεί ως γράφοι. Η δυσκολία και συχνά η αναπαοτελεσματικότητα των πειραματικών μεθόδων, σε επιστήμες όπως η χημία και η βιολογία, οδήγησε στην διερεύνηση τεχνικών στο χώρο της μηχανικής μάθησης, σαν μία πιο αποδοτική εναλλακτική λύση. Για παράδειγμα, η κατανόηση της λειτουργίας μία πρωτείνης με γνωστή ακολουθία, μέσα από αναλυτικές πειραματικές μεθόδους είναι μία ιδιαίτερα επίπονη και χρονοβόρα διαδικασία. Αντί γιάυτό μπορούμε να δοκιμάσουμε υπολογιστικές προσεγγίσεις προκειμένου να προβλέψουμε την πρωτεινική λειτουργία. Αναπαριστώντας τις πρωτείνες σαν γράφους, το πρόβλημα μπορεί να οριστεί σαν πρόβλημα ταξινόμησης γράφων (graph classification problem) όπου η λειτουργία μία νεοανακαλυφθήσας πρωτείνης προβλέπεται βάση ενός συνόλου πρωτεινών με γνωστή λειτουργία. Πάραλληλα με την ανάγκη για μεθόδους με μικρότερο υπολογιστικό κόστος, ένα μεγάλο πλήθος από εργασίες δεν μπορούν έρθουν εις πέρας από ανθρούπους, λόγω του πολύ μεγάλου όγκου δεδομένων που είναι απαιτούν επεξεργασία. Για παράδειγμα, ο αριθμός των κακόβουλων (malicious) εφαρμογών έχει αυξηθεί αρκετά τα τελευταία χρόνια. Η χειρονακτική επιθεώρηση δειγμάτων κώδικα με στόχο την ανίχνευση κακόβουλης λειτουργίας δεν είναι εφικτή καθώς ο αριθμός των δειγμάτων αυξάνεται. Χάρης το 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, εγκαθιδρύεται ως μία πρόβλημα κλειδί σε ένα μεγάλο εύρος εφαρμογών. Η ταξινόμηση γράφων είναι πολύ στενά συνδεδεμένη με το πρόβλημα της σύγκρισης γράφων, ένα κομβικό πρόβλημα στην θεωρία γράφων. Παρόλο που το πρόβλημα αυτό μελετάτε έντονα

για πολλά, 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 κάλιστα να συμπεράνουμε ότι μία τέτοια λύση μπορεί να μην υπάρχει, πράγμα που μας οδήγει στην αποδοχή της διαφορετικότητας στην απόδοση και στην προσέγγιση των υπάρχοντων μεθόδων, σε όλη τους την ολότητα. Παράλληλα η επισήμανση και η επανανοηματοδότηση της έννοιας του υπολογισμού στην σύγχρονη επιστημονικοτεχνολογική ανάπτυξη, οδηγεί στην αναγκη ύπαρξης υπολογιστικών προτύπων τα οποία θα έχουν την ίδια θέση, που είχε το πρότυπο χιλιόγραμμο στην επιστημονικοτεχνολογική ανάπτυξη της εποχής του. Η δημιουργία μίας βιβλιοθήκης για την επίλυση ενός προβλήματος δεν απότελει κατάυτον 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 η 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). Περιοδικά αναθεωρήσης (journals), στα οποία οι ερευνήτριες/ητές συνοψίζουν τα αποτελέσματα της ερευνητικής πρακτικής σε σχέση με τα πεδία τους συμμετέχοντας στην διαδικασία ταξινόμησης της γνώσης, έχουν αρχίσει να δίνουν αντίστοιχο χώρο δημοσίευσης στο ίδιο το λογισμικό τοποθετώντας κατά την επιλογή του κριτήρια εγγυρότητας, διάυγειας, νομικής άδειας χρήσης, δυνατότητα συμμετοχής και προσβασιμότητας στο επίπεδο του κώδικα (βλ. jmlr).

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 γραφοπυρήνες (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 αυτών των 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 ο συλλογή τους σε ένα ελεύθερο λογισμικό, αντικειμενοστραφούς 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. [72]) είναι ληψές τόσο ως προς το περιεχόμενο και το εύρος απεύθυνσης τους αλλά και για τον ίδιο τον τρόπο συσκευασίας (packaging) του κώδικα. Στο πλαίσιο της ανάπτυξης του παραπάνω λογισμικού έλαβε χώρα εκτενή μελέτη της βιβλιογραφίας των πυρήνων γράφων και επιλογή των πιο σημαντικών, υλοποίηση αιχμής (state-of-the-art implementations) σε επίπεδο κώδικα τόσο στο κομμάτι της πολυπλοκότητας

όσο και σε αυτό των εργαλείων, καθώς και έργο παιδαγωγικού χαρακτήρα αναγκαίο για την απεύθυνση του σε ευρύ κοινό μέσω της συγγραφής ενός συστηματικού εγχειριδίου ανάγνωσης (documentation) τόσο για την χρήση αλλά και για την συμμετοχή (contribution) στην συγγραφή κώδικα. Η γλώσσα προγραμματισμού που επιλέχθηκε ήταν η python, μία γλώσσα δημοφιλής τόσο σε επιστημονικούς και τεχνολογικούς κύκλους. Ακολουθεί το πρότυπο του αντικειμενοστραφούς προγραμματισμού (OOP - όπως περιγράφεται από τους κατασκευαστές της), γνωστή ως γλώσσα σεναρίων (script language - από την προγραμματιστική κοινότητα) ακολουθεί εκτέλεση με διερμηνέα και οκνηρό (lazy) σύστημα τύπων που δίνουν την ευκολία στον προγραμματιστή να αναπτύσει γρήγορα αδ-ηος εφαρμογές, ευδιάκριτα γραμμένες σε υψηλό επίπεδο (high level). Παράλληλα το οικοσύστημα της python (python ecosystem), το σύνολο δηλαδή των βιβλιοθηκών που η γλώσσα περιλαμβάνει τόσο από τους ίδιους τους κατασκεβαστές τις όσο και μέσω τρίτων στο επισήμο αποθετηρίο 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 συρραφής κώδικα (code wrapping) άλλων γλωσσών, σε περιβάλλον προγραμματισμού python.

1.2 Οργάνωση του τόμου

Η παρούσα διπλωματική εργασία είναι οργανωμένη σε πέντε κεφάλαια. Μία θεωρητική εισαγωγή στους γραφοπηρύνες και τα δίνεται στα Κεφάλαιο 2. Εισαγωγή στο λογισμικό, την οργάνωση και τις σχεδιαστικές λύσεις γίνεται στο Κεφάλαιο 3. Στο κεφάλαιο 4 παρέχεται μία εισαγωγή στον τρόπο χρήσης του λογισμικού καθως και παρουσιάζεται η εφαρμογή του σε κοινά παραδείγματα. Τέλος στο κεφάλαιο ;; παρουσιάζονται τα συμπεράσματα της διπλωματικής και παρέχονται ιδέες και κατευθύνσεις για την προοπτική μελλοντικής εξέλιξής της.

Κεφάλαιο 2

Πυρήνες Γράφων

Στο κεφάλαιο αυτό θα περιγράψουμε τα κίνητρα που μας οδήγησαν να σχεδιάσουμε μία βιβλιοθήκη για του πυρήνες γράφων ενώ ταυτόχρονα θα παρέχουμε μία εισαγωγή στις βαδσικές έννοιες, εργαλεία και στην ορολογία που θα χρησιμοποιηθεί κατά την διάρκεια της διπλωματικής. Αρχικά θα παρουσιάσουμε τα προλήματα της σύγκρισης γράφων (graph comparison) και της ταξινόμησης γράφων (graph classification) δύο θεμελιώδη προβλήματα στην μηχανική μάθηση μέσω γράφων. Έπειτα δίνουμε όρισμους θεμελιωδών έννοιων της θεωρίας γράφων. Εν συνεχεία, θα δώσουμε 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. Βασική αναφορά γίνεται στους γραφοπυρήνες ενώ περιγράφονται οι πιο βασικοί που υλοποιήθηκαν στην παρούσα εργασία. Τέλος θα περιγράψουμε τις συλλογές δεδομένων (dataset) στις οποίες έγιναν πειράματα καθώς και την πειραματική διάταξη που χρησιμοποιήθηκε σε αυτήν την διπλωματική.

2.1 Κίνητρο

Τα τελευταία χρόνια πλήθηναν ραγδαία τα διαθέσιμα δεδομένα που μπορούσαν να αποδωθούν φυσικά ως γραφοι. Τέτοιοι τύποι δεδομένων έχουν προκύψει σε πολλά πεδία εφαρμογών από τα κοινωνικά δίκτυα μέχρι την βιολογία και την χημία. Τα κοινωνικά δίκτυα μπορούν να μοντελοποιήσουν μία μεγάλυ πληθώρα καταστάσεων, και χρησιμοποιούνται για να αναπραστήσουν τις σχέσεις μεταξύ ένα σύνολο ατόμων. Τέτοιες σχέσεις μπορεί να αποτελούν την φιλία σε ένα κοινωνική ιστιοσελίδα, ή ακόμα της συνεργασίες ενός συνόλου εργαζομένου στο χώρο του σινεμά σε μία ιστοσελίδα που καταγράφει ταινίες. Στην 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 αναπαραστίσουν το ΔΝΑ ακολουθίες, δίκτυα πρωτεϊνικών αλληλεπιδράσεων, μεταβολικά δίκτυα, γενετικά ρυθμιστικά δίκτυα, και φυλογενετικά δίκτυα. Αναπαράστασεις

με τη μορφή γράφων εντοπίζουμε και σε αρκετά τεχνολογικά δίκτυα. Το πιο άμεσο παράδειγμα είναι το διαδίκτυο, στο οποίο οι κόμβοι ανταποκρίνονται σε ιστιοσελίδες και οι ακμές σε υπερσυνδέσμους (hyperlinks) μεταξύ ιστιοσελίδων. Από το παραπάνω γίνεται εύλογο, ότι οι αναπαραστάση γράφων προτιμάται για δεδομένα με δομή που συμπίπτει με αυτή γράφων. Ωστόσο παρόλο που συμπίπτει, θα μπορούσαν να αναπαρασταθούν από διανύσματα χαρακτηριστικών (feature vectors). Μία ερώτηση που φάινεται δόκιμη σε αυτό το σημείο είναι: ¨ποιά 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 που είναι μία πολύ κοινή αναπαράσταση στις κοινότητες της εξόρυξης δεδομένων (data mining) και της μηχανικής μάθησης (machine learning).

2.1.1 Αναπαράσταση Γράφου

Δεν υπάρχει αμφιβολία ότι οι γράφοι χαρακτηρίζονται από πολύ δυνατή αναπαραστατική δυνατότητα. Ένας γράφος δεν αναπαραστιτά μονάχα οντότητες αλλά και τις σχέσεις τους. Με άλλα λόγια, οι γράφοι περιγράφουν αναλυτικά της σχέσης μεταξύ διάφορα μέρη ενός αντικειμένου. Μερικές ευρέως διαδεδομένες δομές δεδομένων μπορούν να αναπαρασταθούν με τη μορφή γράφων [11]. 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 αντιστοίχησης (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 ακολουθίες χαρακτήρων (strings) μπορούν επίσης να αναπαρασταθούν σαν γράφοι, όπου κάθε κόμβος είναι ένας χαρακτήρας και οι ακμές συμβολίζουν την σειρά. Λόγω της εκφραστικότητας των γράφων ακόμα και δεδομένα που δεν έχουν εγγενώς μία αναπαράσταση γράφου, είναι χρήσιμο να απεικονίζονται κατά αυτόν τον τρόπο. Ένα πολύ κοινό παράδειγμα είναι δεδομένα κειμένου, στα οποία οι γράφοι χρησιμοποιουνται για να αναπαραστήσουν σχέσεις μεταξύ γλωσσολογικών τμημάτων.

2.1.2 Γράφοι ή Διανύσματα Χαρακτηριστικών

Στον χώρο της εξόρυξης δεδομένων και στη μηχανική μάθηση τα δεδομένα αναπαριστόνται συνήθως μέσω διανύσματων. 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 οντότητες και τις σχέσεις τους και ως επακόλουθο δεν περιγράφουν απλώς τιμές (διαφόρων οντοτήτων) σε ένα χώρο, αλλά δομή (τις σχέσεις μεταξύ αυτών των οντοτήτων). 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 ατνιστοιχεί επακριβώς στο μέγεθος και την πολυπλοκότητα κάθε τμήματος των δεδομένων (που αποτελεί ξεχωριστή οντότητα). Με βάση τα παραπάνω, οι γράφοι μοιάζουν

με την ιδανική αναπράσταση δεδομένων σε διάφορους επιστημονικούς τομείς. Παρόλαυτα η αναπαράσταση μέσω γράφων έχει τα μειονεκτήματα της: οι γράφοι δεν μπορούν να ενταχθούν στον πολύ πλούσιο φορμαλισμό των διανυσματικών χώρων, ενώ ακόμα πολύ τελεστές (operators) μεταξύ γράφων είναι υπολογιστικά ακριβείς, παρόλη την απλή τους μαθηματική διατύπωση. Το πιο χαρακτηριστικό παράδειγμα αυτών των τελεστών είναι αυτός 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), δεν έχουν βρεθεί μέχρι στιγμής αλγόριθμοι πολυωνυμικού χρόνου. Γενικότερα το πρόβλημα της σύγκρισης δύο αντικειμένων, έχει πολύ πιο ασθενή φορμαλισμό στους γράφους σε σχέση με τα διανύσματα. Στους κοινούς (ευκλείδιους) διανυσματικούς χώρους που χρησιμοποιούνται στην πράξη, η απόσταση μπορεί να υπολογιστεί αποδοτικά χρησιμοποιώντας την γενικά αποδεκτή μετρική της ευκλείδιας απόστασης. Δυστυχώς τέτοια δεν υπάρχει μετρική μεταξύ γράφων. Σάυτό οφείλεται το γεγονός ότι δεν υπάρχει κανονική διάταξη (canonical ordering) μεταξύ των κόμβων του γράφου και άρα απουσιάζει μία ¨ένα προς ένα' αντιστοίχηση μεταξύ των κόμβων δύο γράφων. Επιπροσθέτως, η αναγνώση κοινών μερών μεταξύ δύο γράφων, δεν είναι εξίσου υπολογιστικά προσβάσιμη, μιάς και δεδομένου ότι 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 είναι ότι για κάποιες ακόμα απλές πράξεις σε διανυσματικούς χώρους δεν υπάρχει αντίστοιχη πράξη στους γράφους, όπως για παράδειμα το άθροισμα και η διαφορά. Παρόλο λοιπόν που οι γράφοι αποτελούν έναν πολύ άμεσο τρόπο αναπράστασης 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 αναπαράστασης των δεδομένων είναι κομβικός στους χώρους της εξόρυξης δεδομένων και της μηχανικής μάθησης. Κάθε αλγόριθμος είναι σχεδιασμένος για δεδομένα μία συγκεκριμένης αναπαράστασης. Λόγω της ευελιξίας των γράφων, κάποιος θα περίμενε ότι θα υπάρχει μεγάλη πρόοδος στην ανάπτυξη αλγορίθμων που μπορούν δέχονται ως είσοδο δεδομένα αναπαραστημένα ως γράφους. Αντίθετα κάτι τέτοιο δεν συμβαίνει, μιας και λόγω της συνδιαστικής (combinatorial) φύσης των γράφων η αντιμετόπιση αυτού του προβλήματος δεν λήφθηκε ποτέ σοβαρά υπόψη. Ως επακόλουθο η έρευνα σε αυτές τις περιοχές εστιάστηκε κυρίως σε αλγορίθμους μεταξύ διανυσμάτων, μιάς και τα διανύσματα παρουσιάζουν πολλές ενδιαφέροντες μαθηματικές ιδιότητες και μπορούν και οι πράξεις και ο χειρισμός παρουσιάζει μικρότερη υπολογιστική πολυπλοκότητα. Συνεπώς, δεν μας κάνει να παραξενευόμαστε το γεγονός ότι οι ποιο δημοφιλείς αλγόριθμοι μάθησης είναι σχεδιασμένοι να λαμβάνουν ως είσοδο διανύσματικές αναπραστάσεις των δεδομένων. Ακόμα και σε εφαρμογές που οι γράφοι είναι η φυσική αναπαράσταση των δεδομένων, γίνονται απόπειρες να αναπαρασταθούν σαν διανσματα χαρκατηριστικών και να χρησιμοποιούθουν υπάρχουσες τεχνικές, αντί να σχεδιαστούν αλγόριθμοι που δέχονται στην εισοδό τους γράφους. Ιδανικά, θα θέλαμε να υπάρχει τρόπος

να μετασχηματίσουμε τους γράφους σε διανύσματα χαρακτηριστικών, χωρίς να χάνουμε το αναπαραστατικό τους πλεονέκτημα. Αυτή η ανώριμη επιθυμία, μπορεί εύκολα να καταριφθεί 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 είναι αληθές το γεγονός ότι ο σχεδιασμός ενός αλγορίθμου με γράφους αυξάνει είτε την σημασιολογική είτε την υπολογιστική πολλυπλοκότητα επίλυσης ιδιαιτέρος όταν αυτό συμβαίνει σε δεδομένα με μεγάλο μέγεθος. Από την άλλη αυτοί οι λγόριθμοι αντιστοιχούν έχουν μεγαλύτερη αποτελεσματικότητα συμπεριλαμβανομένης και της ικανότητας τους να γενικεύουν σε σχέση με ανταγωνιστικούς αλγορίθμους (για προβλήματα που δεχονται πολλές προσεγγίσεις - βλ. text classification). Συνεπώς η έξυπνη σχεδίαση, αποδοτική υλοποίηση και η ευκατανόητη διάδοση τεχνικών για γράφους, μπορούν άμεσα να ωφελήσουν εφαρμογές εφαρμογές στις οποίες εμφανίζονται αναπαραστάσεις 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. Ώσον 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 γράφους.

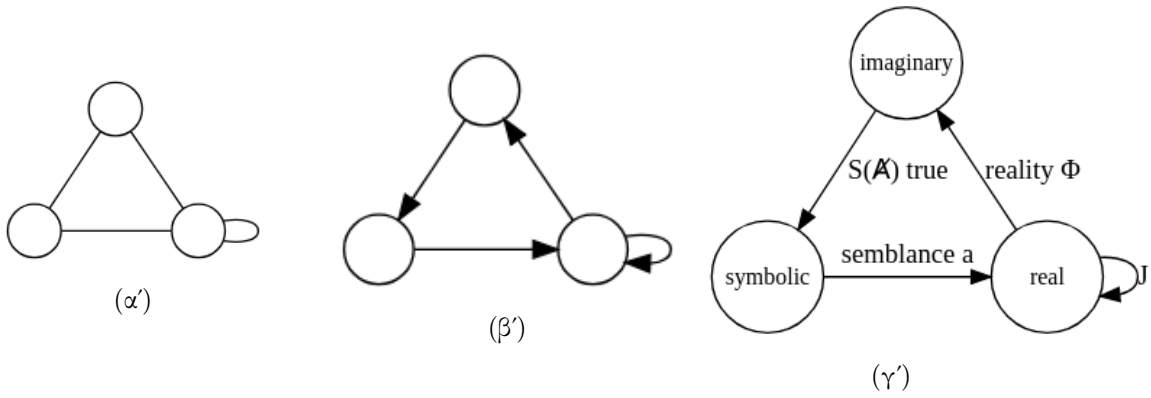
Ορισμός 2.2 (Κατευθυνόμενος και μη Κατευθυνόμενος Γράφος). *Ένας γράφος $G_d = (V_d, E_d)$ είναι κατευθυνόμενος αν οι ακμές του έχουν μία κατεύθυνση, ότι το E_d είναι ένα σύνολο από διατεταγμένα ζευγάρια κόμβων. Οι ακμές ενός κατευθυνόμενου γράφου λέγονται τόξα (arcs). Ένας μη κατευθυνόμενος γράφος είναι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

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

Καθόλο την θεωρητική ανάλυση αυτής της διπλωματικής θα θεωρούμε ότι ασχολούμαστε με μη-κατευθυνόμενους γράφους. Θέλοντας να αυξήσουμε την πυκνότητα της πληροφορίας που αναπαριστά ένας γράφος φτανουμε 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.*

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



(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

ακμές λέγεται πλήρως επισημειωμένος (βλ. Σχήμα 2.1γ′). Στην παρούσα διπλωματική και τα τρία παραπάνω ήδη θα μας απασχολήσουν. Οι επισημειώσεις μπορούν να είναι είτε διακριτές (discrete) είτε συνεχής (continuous). Οι συνεχείς επισημειώσεις ονομάζονται και χαρακτηριστικά (attributes).
Μία εναλλακτική 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:*

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

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 αντιστοιχέι 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, εξέρχον-βαθμός (in-degree) και εισέρχον-βαθμός (out-degree) για τις ακμές που εισέρχονται και εξέρχονται αντίστοιχα.

Ορισμός 2.7 (Υπογράφημα). *Δεδομένου ενός γράφου $G = (V, E)$, ένας γράφος $G' = (V', E')$ θα λέγεται υπογράφημα (subgraph) του G και θα συμβολίζετα με $G' \subseteq G$, αν ισχύει $V' \subseteq V$ και $E' \subseteq E$.*

Ορισμός 2.8 (Επαγώμενο 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:*

$$E(U) = \{\{v_i, v_j\} \in E : v_i, v_j \in U\}\} \tag{2.1}$$

Ο βαθμός ενός κόμβου $v_i \in U, d_G(U)(v_i)$, ισούτε με τον αριθμό κόμβο που είναι γειτονικοί του v_i στον $G(U)$. Τηε δενσιτψ οφ α γραπη Γ ις δ (Γ) = μ/

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.

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

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 $\{v_{k+1}, v_1\} \in E$.

Ορισμός 2.12 (Κοντινότερο Μονοπάτι). *Κοντινότερο μονοπάτι (shortest path) μεταξύ ενός κόμβου 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.

2.3 Προβλήματα μηχανικής μάθησης

Το σύνολο των προβλημάτων στον χώρο της μηχανικής μάθησης είνα τριχοτομημένο στις εξής τρεις μεγάλες κατηγορίες (1) επιβλεπόμενη (supervised) μάθηση, (2) μη-επιβλεπόμενη (unsupervised) μάθηση και (3) ενισχυτική (reinforcement) μάθηση. Στην επιτηρούμενη μάθηση,

ο στόχος είναι η μάθηση μία αντιστοίχηση μεταξύ εισόδου εξόδου, δεδομένου ενός συνόλου ζευγαριών τιμών εισόδου εξόδου (που λέγεται σύνολο εκπαίδευσης). Στην περίπτωση που έχουμε διακριτή έξοδο, ένα 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 (αλλιώς οι επισημειώσεις των κατηγοριών) όσον αφορά τα ψηφεία που αναπαριστούν. Δεδομένου ενός εκπαιδευτικού συνόλου εικόνων και των κατηγορικών επισημειώσεων τους, ο στόχος είναι η μάθηση μία αντιστοίχησης από εικόνες σε κατηγορικές επισημειώσεις, που μπορούν να χρησιμοποιηθούν για να αναγνωρίσουμε την κατηγορία που 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 τρομακτική με συνεχείς μετρήσεις από το 0 έως το 5, πρόβλεψη της ¨τρομακτικότητας' μίας μελωδίας, με βάση την εξαγωγή κάποιων χαρακτηριστικά της στην είσοδο. Στην επιτηρούμενη μάθηση, μας δίνονται μόνο είσοδοι και ο στόχος μας είναι να εξάγουμε χρήσιμα προτύπα (patterns) από αυτές. Ένα παράδειγμα μη επιτηρούμενης 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 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 εισόδουν. Τέλος όσον αφορά την ενισχυτική μάθηση, στόχος είναι να προσδιορίσουμε ποιές δράσεις πρέπει να λάβουμε σε μία δεδομένη κατάσταση προκειμένου να μεγιστοποιήσουμε μία 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 αλληλεπίδρα με ένα άλλο σύστημα-περιβάλλον, το οποίο το ανταμοίβει η 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 οι γράφοι χρησιμοποιούνται για να αναπαραστήσουν δεδομένα από πολύ διαφορετικές πηγές. Ως επακόλουθο δεν μας εκπλήσει το γεγονός ότι μία πληθώρα προβλημάτων μάθησης είναι ορισμένα για γράφους. Τα ακόλουθα τέσσερα προβλήματα είναι ίσως τα πιο πολύ μελετημένα 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 υπόψης 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 συχνοεμφανιζόμενο πρόβλημα στο χώρο της μηχανικής μάθησης. Σε ένα πρόβλημα ταξινόμησης, στόχος είναι η μάθηση μίας αντιστοίχησης 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 το 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 $\mathcal{Q}$ είναι ένα σύνολο αντικειμένων που θα θέλαμε να ταξινομήσουμε. Ως επακόλουθο $q \in \mathcal{Q}$. Οι έξοδοι παίρνουν τις τιμές τους από ένα πεπερασμένο σύνολ $y \in \mathcal{Y} = C_1, \ldots, C_{|Y|}$, όπου το C είναι ένα πλήθος από κατηγορίες. Αν το $|\mathcal{U}| = 2$ τότε το προκύπτον πρόβλημα λέγεται δυαδική ταξινόμηση. Εναλλακτικά, αν $|\mathcal{Y}| > 2$, λέγεται *πολυ-κατηγορική ταξινόμηση*. 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 που ανήκουν. Θα υποθέσουμε οτι τα δεδομένα εκπαίδευσης $\mathcal{D}$ προέκυψαν από μία άγνωστη κατανομή. Με βάση αυτά το σύνολο εκπαίδευσης, ο στόχος είναι να μάθουμε μία συνάρτηση $f : \mathcal{X} \leftarrow \mathcal{Y}$ 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 υπάρχει 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 ποίοτητας 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. Σε αυτήν την περίπτωση, 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 γράφων που μπορούν να δωθούν ως είσοδοι σε αυτήν την συνάρτηση και $\mathcal{Y}$ είναι το σύνολο των δυνατών επισημειώσεις που μπορούν να αποδωθούν στους γράφους. Η συνάρτηση αυτή μπορεί να χρησιμοποιηθεί ύστερα για την ταξινόμηση γράφων που δεν εμφανίστηκαν μέχρι τώρα (όπως αυτοί στο πειραματικό σύνολο) σε κατηγορίες. Το πρόβλημα της ταξινόμησης γράφων έχει αποτελέσει μία δημοφιλή περιοχή έρευνας τα τελευταία χρόνια, μιάς και έχει συγκεντρώσει πολλές εφαρμογές τα τελευταία χρόνια σε ένα μεγάλο εύρος περιοχών. Τέτοιες συνοπτικά εμφανίζονται σε ένα

εύρος από τον χαρακτηρισμό ενός χημικού μορίου ως μεταλλαξιογόνου [73] και την πρόβλεψη της λειτουργίας μίας πρωτεΐνης δεδομένου της νουκλεοτιδικής του ακολουθίας [10], μέχρι τον εντοπισμό του αν ένα λογισμικό είναι κακόβουλο [82].

2.3.3 Σύγκριση Γράφων

Το πρόβλημα της ταξινόμησης γράφων συνδέεται άμεσα με αυτό της σύγκρισης. Πολλοί αλγόριθμοι στο χώρο της μηχανικής μάθησης λαμβανουν αποφάσεις, βάση ενός μέτρου ομοίτητας (similarity) ή ενός μέτρου απόστασης (distance) μεταξύ δύο στιγμιοτύπων εισόδου. Για παράδειγμα δημοφιλείς ταξινομητές, όπως ο ταξινομητής κ-κοντινότερων γειτόνων ή ο ταξινομητής μηχανών διανυσμάτων υποστήριξης, μπορούν να επιλύσουν προβλήματα μάθησης πολύ διαφορετικών δεδομένων, προσδιορίζοντας μονάχα μία μετρική ομοιότητας μεταξύ τους. Καταυτόν τον τρόπο ο προσδιορισμο μία συνάρτησης απόστασης $d : \mathcal{G} \times \mathcal{G} \rightarrow \mathbb{R}^+$ μεταξύ γράφων, , μπορούμε αμέσως να χρησιμοποιήσουμε έναν από τους παραπάνω αλγορίθμους, για να ταξινομήσουμε δεδομένα που έχουν αναπαρασταθεί ως γράφοι. Από την άλλη, η σύγκριση γράφων είναι από μόνο του ένα πολύ δύσκολο πρόβλημα. Πιο τυπικά, δεδομένου δύο γράφων G_i, G_j σε ένα χώρο γράφων $\mathcal{G}$, το πρόβλημα της σύγκρισης γράφων αφορά τον προσδιορισμό μίας συνάρτησης αντιστοίχησης: $f : \mathcal{G} \times \mathcal{G} \rightarrow \mathbb{R}$, τέτοια ώστε να ποσοτικόποιεί μία έννοια ομοιότητας μεταξύ G_i, G_j ή την απόσταση τους, που να συνδέεται σημασιολογικά με αυτό την ουσία του προβλήματος μάθησης. Η σύγκριση γράφων είτε από μόνη της είτε στον χώρο της ταξινόμησης γράφων βρίσκει εφαρμογή σε πολλά πεδία. Συγκεκριμένα στην χημιο-πληροφορική, χρησιμοποιήθηκε για την πρόβλεψη κινητικών μοντέλων μεταξύ χημικών αντιδράσεων [77] και για την ταξινόμηση χημικών μορίων με στόχο π.χ. την ανίχνευση φαρμακευτικής δράσης [83]. Στην βιοπληροφορική, χρησιμοποίηθηκε για την κατάταξη γονιδίων και γονιδιωμάτων σε μία βάση γνώσης[41, 33] και στην κατανόηση μηχανισμών γονιδιακής ρύθμισης [19]. Πέρα από τους δύο αυτούς τομείς, οι ομοιότητα γράφων έχει χρησιμοποιηθεί για την σύγκριση ηλεκτρικών κυκλωμάτων [74], προγραμμάτων γλώσσας C [28], κειμένων [65], ειδησεογραφικών γεγονότων [29] αλλά και για την αυτοματοποιημένη συλογιστική [76] και για την αποσαφήνηση σχεσιακών οντοτήτων [36].

2.4 Πυρήνες Γράφων

Οι πυρήνες γράφων, αποτελούν μία από τις πιο μοντέρνες τεχνικές στην ταξινόμηση γράφων. Τεχνικές σαν αυτή παρουσιάζουν μερικές πολύ ελκυστικές στατιστικές ιδιότητες. Παράλληλα δύνανται να συνδιάσουν την αναπαραστατική δύναμη των γράφων και τις δυνατότητες διαχωρισμού της πληροφορίας που επιτυγχάνουν οι μεθόδοι που βασίζονται σε πηρύνες. Συνεπώς, αποτελούν πολύ ισχυρά εργαλεία τόσο για την αντιμετόπιση 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 προσβάσιμους αυτές της τεχνικές σε ένα σημείς της ιέραρχιας επίλυσης ενός προβλημάτος μηχανικής μάθησης. Εν συνεχεία θα δώσουμε μία λεπτομερή περιγραφή των πυρήνων, δίνοντας μία αναλυτική περιγραφή των πηρύνων που υλοποιήθηκαν στο συγκεκριμένο λογισμικό (μέχρι την παρούσα έκδοση του).

2.4.1 Συναρτήσεις πηρύνα

Αρχικά θα προσπαθήσουμε να δώσουμε μία εισαγωγή στις συναρτήσεις πηρύνα.

Ορισμός 2.13 (Γκραμ Μήτρα). *Δεδομένου ενός συνόλου εισόδων $x_1, \dots x_N \in \mathcal{X}$ και μία συνάρτησης $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, η $N \times N$ μήτρα K που ορίζεται σαν:*

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

ονομάζεται πίνακας Γκραμ (Gram Matrix) ή μήτρα πηρύνα του k με βάση της εισόδους $x_1, \dots x_N$

Σε όλη την συνέχεια θα αναφερόμαστε στις μήτρες Γκραμ σαν μήτρες πηρύνα.

Ορισμός 2.14 (Θετικά Ημιορισμένος Πηρύνας). *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 $x_1, \dots, x_N \in \mathcal{X}$, παράγει μία θετικά ημιορισμένη μήτρα, θα λέγεται θετικά ημιορισμένος πηρύνας, ή απλώς πηρύνας.*

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

$$\forall x_i, x_j \in \mathcal{X} : k(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle \tag{2.2}$$

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 παίρνεται 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):

$$\forall f \in \mathcal{H}, \forall x \in \mathcal{X} : f(x) = \langle f, k(x, \dot{}) \rangle \tag{2.3}$$

Λόγω αυτής της ιδιότητας η μήτρα $\mathcal{H}$ 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 μέ έναν RKHS και αντίστροφα [4].

2.4.2 Μέθοδοι Πυρήνα

Οι μέθοδοι πηρύνα (Kernel methods) είναι ένα σύνολο από αλγορίθμους μηχανικής μάθησης, που λειτουργούν σε ένα σύνολο δεδομένων εισόδου τα οποία έχουν αναπαρασταθεί σε έναν ιμπλιςιτ χώρο χαρακτηριστικών χρησιμοποιώντας μία συνάρτηση πυρήνα. Ένα από τα σημαντικότερα πλεονεκτήματα αυτών των μεθόδων είναι ότι λειτουργούν σε πολλά και διαφορετικά είδη δεδομένων [67]. Ο χώρος εισόδων $\mathcal{X}$, δεν χρειάζεται να είναι ένας διανυσματικός χώρος, αλλά μπορεί να αναπαριστά όποιαδήποτε κατηγορία δεδομένων, όπως των χώρο των συμβολοσειρών (string) ή τον χώρο των γράφων [24]. 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 υπόρητα αναπαριστούν δεδομένα σε ένα χώρο χαρακτηριστικών και υπολογίζουν το εσωτερικό γινόμενο μεταξύ τους σε αυτόν τον χώρο χρησιμοποιώντας μία συνάρτηση πυρήνα. Τέτοια εσωτερικά γινόμενα μπορούν να ερμηνευτούν ως μέτρα ομοιότητας μεταξύ των αντίστοιχων αντικειμένων, που συγκρίνουν. και συσταδοποίηση, μπορούν να έρθουν εις πέρας χρησιμοποιώντας μόνο εσωτερικά γινόμενα που έχουν υπολογιστεί σε αυτόν τον χώρο χαρακτηριστικών. Οι μέθοδοι πυρήνα είναι πολύ δημοφιλείς και έχουν φανεί ιδιαίτερα επιτυχημένες σε ένα μεγάλο σύνολο εφαρμογών. Για να διασαφηνήσουμε 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 χαμηλότερο δυνατόν. Συνεπώς αυτό που μας ενδιαφέρει είναι να ελαχιστοποιήσουμε το σφάλμα εκπαίδευσης, αλλά ταυτόχρονα μας ενδιαφέρει να έχουμε κάλη απόδοση σε νέα δεδομένα (που δεν εμφανίστηκαν στο σύνολο εκπαίδευσης).

Οι μέθοδοι μεγάλου περιθορίου (large margin methods), όπως οι μηχανές διανυσμάτων υποστήριξης αναζητούν ένα υπερεπίπεδο που διαχωρίζει τα μέρος των δειγμάτων εκπαίδευσης που ανήκουν στην κατηγορία -1 από αυτά που ανήκουν στην κατιγορία 1. Κατά συνέπεια, η f μπορεί να πάρει την μορφή $f(x) = \text{sign}(\langle w, x\rangle + b)$, όπου η συνάρτηση sign() αντιστοιχή στην συνάρτηση προσήμου η οποία παίρνει την τιμή 1 αν το ορίσμα της είναι θετικό και -1 αλλιώς. Έπειτα 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, υπάρχουν άπειρα τέτοτα υπερεπίπεδα. Οι ταξινομητές μεγάλου περιθορίου (large margin classifiers), επιλέγουν μεταξύ αυτών των υπερεπιπέδων αυτό που μεγιστοποιεί το περιθώριο μεταξύ των δύο κατηγοριών των σημειακών δεδομένα εκπαίδευσης, αυτό δηλαδή που η απόσταση από τα κοντινότερα δεδομένα των δύο κατηγοριών είναι μέγιστο.

Για την εύρεση του βέλτιστου υπερεπιπέδου, οι ταξινομητές λύνουν ένα πρόβλημα κυρτής τετραγωνικής βελτιστοποίησης. Το διάνυσμα w που προκύπτει ως λύση σε αυτό το πρόβλημα είναι ένας γραμμικός συνδιασμός των παραδειγμάτων εκπαίδευσης:

$$w = \sum_{i=1}^{N} a_i y_i x_i \tag{2.4}$$

όπου το $\alpha_i \in \mathbb{R}^+$.

Χρησιμοποιώντας το παραπάνω αποτέλεσμα, που είναι γνωστό ως το θεόρημα αναπαράστασης

(representer theorem) [66], ο γραμμικός ταξινομητής f μπορεί να γραφεί ως:

$$f(x) = \text{sign}(\sum_{i=1}^{N} a_i y_i \langle x_i, x \rangle + b) \tag{2.5}$$

Στην περίπτωση που τα δεδομένα εκπαίδευσης δεν είναι γραμμικά διαχωρίσιμα, αναζητούμε ένα υπερεπίπεδο που μεγιστοποιεί το περιθώριο και την ίδια στιγμη ελαχιστοποιεί μία ποσότητα αντιστρόφος ανάλογη στο πλήθος των λαθών ταξινόμησης. Ο υπολογισμός αυτού του υπερεπιπέδω μπορεί ξανά να διατυπωθεί ως ένα πρόβλημα κυρτής τετραγωνικής βελτιστοποιήσης, καθώς και το διάνυσμα λύσης w παραμένει ένας γραμμικός συνδιασμός των δεδομένων εισόδου. Σε σύνθετα προβλήματα ταξινόμησης, μπορεί να μην υπάρχουν υπερεπίπεδα τέτοια ώστε να διαχωρίζουν θετικά επισημειωμένα παραδείγματα από θετικά και να παρέχουν ένα καλό αποτέλεσμα ταξινόμησης. Η απάντηση σε αυτό το πρόβλημα, που συχνά είναι γνωστή ως τέχνασμα πηρύνα (kernel trick: [2, 13]), 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 $\phi : \phi : \mathcal{X} \to \mathcal{H}$ 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 $\langle x_i, x \rangle$ με $\langle \phi(x_i), f(x) \rangle$. Έστω $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ μία συνάρτηση πηρύνα με την ακόλουθη ιδιότητα:

$$k(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle \tag{2.6}$$

Η συνάρτηση απόφασης f μπορεί τώρα να γραφτεί ως:

$$f(x) = \text{sign}(\sum_{i=1}^{N} \alpha_i y_i \langle \phi(x_i), \phi(x) \rangle + b) = \text{sign}(\sum_{i=1}^{N} \alpha_i y_i k(x_i, x) + b) \tag{2.7}$$

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

2.5 Μηχανή Διανυσμάτων Υποστήριξης

Εν συνεχεία θα εστιάσουμε στον ταξινομητή μηχανής διανυσμάτων υποστήριξης (support vector machine - SVM), τον πιο δημοφιλή αλγόριθμο που έχει ως βάση του πηρύνες. Ο κλασσικός ταξινομητής SVM, θέτει το ακόλουθο πρόβλημα: δεδομένου ενός συνόλου N αντικειμένων εκπαίδευση, μαζί με τις κατηγορίες τους $D = \{(x_i, y_i)\}_{i=1}^{N}, x_i \in \mathcal{X} = \mathbb{R}^d, y_i \in \mathcal{Y} = \{-1, +1\}$, βρές ένα ταξινομητή $f : \mathcal{X} \to \mathcal{Y}$ που προβλέπει τις κατηγορίες νέων δεδομένων. Στον φορμαλισμό προβλημάτων *διακριτού ορίου* (hard margin), θεωρούμε ένα γραμμικά διαχωρίσιμο πρόβλημα. Όπως αναφέρθηκε παραπάνω, ο SVM ανήκει στην οικογένει των ταξινομητών μεγάλου περιθωρίου και ως επακόλουθο αναζητά ένα υπερεπίπεδο που διαχωρίζει στιγμιότυπα

της κατηγορίας -1 από την κατηγορία $+1$ [78]. Έστω (w, b) οι παράμετροι του υπερεπιπέδου. Τότε, η απόσταση μεταξύ ενός σημείου $x \in \mathbb{R}^d$ και του υπερεπιπέδου υπολογίζεται ως:

$$\frac{|w^T x + b|}{||w||} \tag{2.8}$$

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

$$\frac{1}{||w||} \tag{2.9}$$

Ως επακόλυθο η μεγιστοποίηση του περιθωρίου συνεπάγεται ελαχιστοποίηση του $||w||$, πράγμα που είναι ισοδύναμο με την ελαχιστοποίηση του $\frac{1}{2}||w||^2$. Συνεπώς σε μία περίπτωση που τα δεδομένα διαχωρίζονται 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 [14]. Αν τώρα στραφούμε στο δυϊκό (κατά την Λαγκραντζιανή) αυτού του προβλήματος:

$$\begin{aligned} \underset{\alpha}{\text{maximize}} \quad & \sum_{i=1}^{N} \alpha_i - \frac{1}{4} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha \\ \text{subject to} \quad & \sum_{i=1}^{N} \alpha_i y_i = 0 \\ & \alpha_i \geq 0\ \forall i \in \{1, \ldots, N\} \end{aligned} \tag{2.10}$$

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

$$w = \sum_{i=1}^{N} a_i y_i x_i \tag{2.11}$$

Η παράμετρος w του υπερεπιπέδου (δηλ. η λύση του SVM) είναι ένας γραμμικός συνδιασμός των σημείων εκπαίδευσης $x_1 \ldots x_N$. Ένα σημείο x_i συμβάλει στην διαμόρφωση του w ανν $a_i > 0$. 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 κατάυτόν τον τρόπο ϋποστηρίζουν' το υπερεπίπεδο. Τα σημεία αυτά ονομάζονται διανύσματα υποστήριξης 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 ¨ελαστικού ορίου' (soft margin) επιτρέπει κάποια σημεία εκπαίδευσης να μην ταξινομηθούν σωστά [8, 16]. Συγκεκριμένα στην συνάρτηση που πρόκειται να ελαχιστοποιηθεί προστίθεται ένας όρος ποινής (penalty term) για κάθε σημείο που έχει ταξινομηθεί λάθος. Η μεγαλύτερη η απόσταση ενός σημείου από το όριο, ο μεγαλύτερος ο όρος ποινής. Επιτρέποντας λανθασμένη ταξινόμηση στα δεδομένα εισόδου είναι προφανές ότι δεν μπορούμε να χρησιμοποιήσουμε την διατύπωση του SVM όπως δώθηκε στην εξίσωση 2.10, μιάς και ο περιορισμός δεν ικανοποιείται για τα δεδομένα που κατηγοριοποιούνται λανθασμένα. Ως επακόλουθο, εισάγουμε μία βοηθητική μεταβλητή ζ_i για κάθε δεδομένο εκπαίδευσης x_i. Η βοηθητική μεταβλητή αυτή υπολογίζει το βαθμό που ένα σημειακό δεδομένο παραβιάζει το όριο του περιθωρίου. Για σημειακά δεδομένα που βρίσκονται πάνω ή μέσα στο περιθώριο, ισχύει ότι $\zeta = 0$, ενώ για τα άλλα σημεία ισχύει:

$$\zeta_i = |y_i - (\langle w, x_i \rangle + b)| \tag{2.12}$$

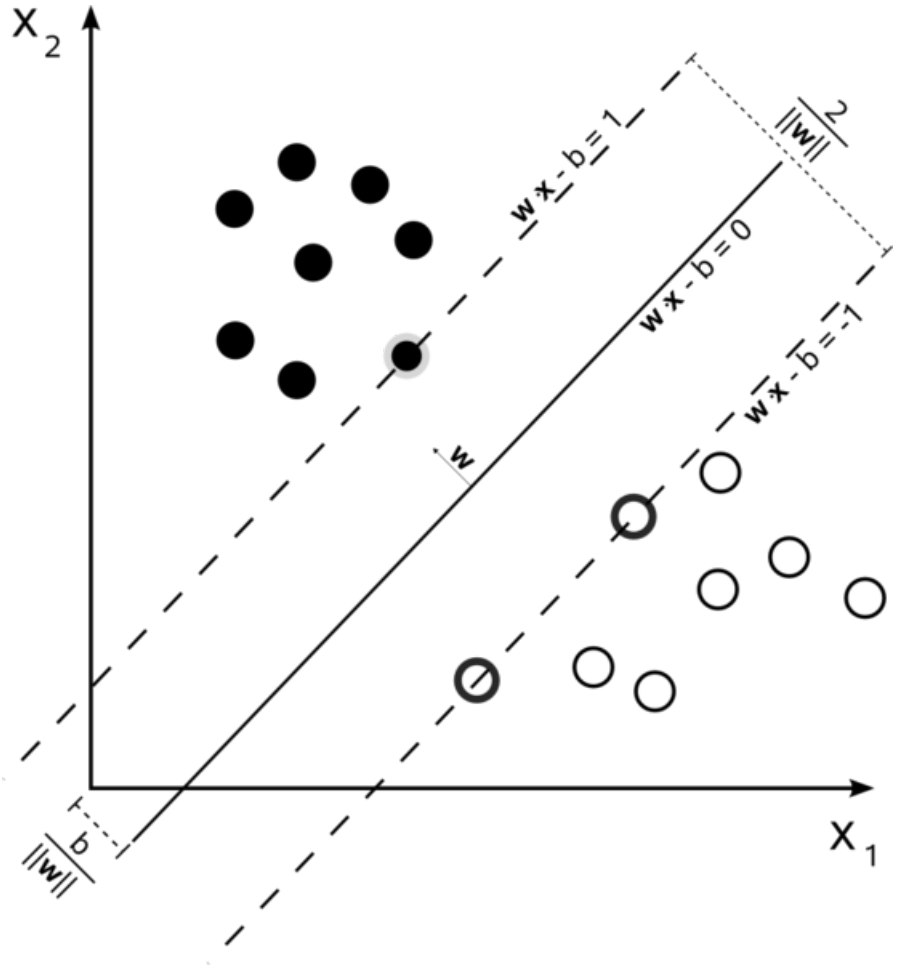
Συνεπώς, ένα σημειακό δεδομένο x_i που βρίσκεται πάνω στο διαχωριστικό υπερεπίπεδο θα έχει $\zeta_i = 1$ και τα σημεία που είναι λανθασμένα ταξινομημένα θα έχουν $\zeta_i > 1$ (πρβλ. 2.2). Σε τούτη την διατύπωση, στόχος μας είναι να μεγιστοποιήσουμε το περιθώριο, ενώ ταυτόχρονα μας ενδιαφέρει να ελέγξουμε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

$$\begin{aligned} \underset{w,b,\zeta}{\text{minimize}} \quad & \frac{1}{2}||w||^2 + C\sum_{i=1}^{N}\zeta_i \\ \text{subject to} \quad & y_i(\langle w, x_i \rangle + b) \geq 1 - \zeta_i \;\; \forall i \in \{1, \ldots, N\} \\ & \zeta_i \geq 0 \;\; \forall i \in \{1, \ldots, N\} \end{aligned} \tag{2.13}$$

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

$$\begin{aligned} \underset{\alpha}{\text{maximize}} \quad & \sum_{i=1}^{N}\alpha_i - \frac{1}{4}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha \\ \text{subject to} \quad & \sum_{i=1}^{N}\alpha_i y_i = 0 \\ & C \geq \alpha_i \geq 0 \;\; \forall i \in \{1, \ldots, N\} \end{aligned} \tag{2.14}$$

Το παραπάνω πρόβλημα βελτιστοποίησης είναι κατά ενδιαφέρον τρόπο το ίδιο με το 2.10, προσθέτοντας το C ως άνω φράγμα στις παραμέτρους α_i.



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

2.6 Πυρήνες Γράφων

Οι πυρήνες γράφων έχουν πρόσφατα προκύψει σαν μία πολλά υποσχόμενη προσέγγιση στην μηχανική μάθηση σε δεδομένα αναπαραστημένα με γράφους. Αυτές οι μέθοδοι επεκτείνουν την εφαρμοσιμότητα των μεθόδων πυρήνα στους γράφους. Οι πυρήνες γράφων μπορούν να χωριστούν σε δύο κατηγορίες: (1) αυτού που συγκρίνουν κόμβους του ίδιου γράφου και (2) αυτούς που συγκρίνουν γράφους. Οι πυρήνες που αναπτύχθηκαν στο λογισμικό της παρούσας διπλωματική βρίσκονται αποκλειστικά στην δεύτερη κατηγορία, δηλαδή τους πυρήνες μεταξύ γράφων. Οπουδήποτε ο όρος αυτός συναντιέται παρακάτω θα περιγράφει στυναρτήσεις πυρήνα που βρίσκονται στην δεύτερη κατηγορία. Από τα προηγούμενα πρέπει να είναι πλέον φανερό ότι η εφαρμογή των συναρτήσεων πυρήνα αποτελείται από δύο στάδια. Πρώτα, σχεδιάζετια μία συνάρτηση πυρήνα, και βάση αυτής κατασκευάζεται ο πίνακας πυρήνα. Έπειτα, ένας αλγόριθμός μάθησης χρησιμοποιείται για τον υπολογισμό μίας βέλτιστης αφηρημένης επιφάνειας (manifold) στον χώρο χαρακτηριστικών (π.χ. ένα υπερεπίπεδο σε ένα πρόβλημα δυαδικού προγραμματισμού). Από την στιγμή που υπαχουν διάφοροι ώριμοι και διαθέσιμοι στη βιβλιογραφία ταξινομητές, που χρησιμοποιούν στην βάση τους πυρήνες είναι, η έρευνα στον χώρο των πυρήνων γράφων έχει εστιάσει στο πρώτο στάδιο. Σαν επακόλουθο, η κύρια προσπάθεια εστίασε στην ανάπτυξη εκφραστικών και αποδοτικών πυρήνων γράφων, ικανών να μετρήσουν με ακρίβεια την ομοιότητα μεταξύ γράφων της εισόδου. Τέτοιοι πυρήνες έμμεσα προβάλουν γράφους σε ένα χώρο χαρακτηριστικών $\mathcal{H}$. Σε σχέση με το δέυτερο βήμα, για το σώμα της παρούσας διπλωματικής και για την πειραματική αξιολόγηση του λογισμικού (ρεφ), μας χρειάζεται μόνο ο ταξινομητής SVM που αναφέρθηκε παραπάνω. Ο κύριος στόχος στην εφαρμογή μεθόδων πυρήνα σε γράφους είναι ο προσδιορισμός κατάλληλων θετικά ημι-ορισμένων συναρτήσεων πηρύνα σε ένα σύνολο δεδομένων εισόδου που δύνονται να υπολογίσουν την ομοιότητα μεταξύ τους. Πόσο εκφραστικός μπορεί όμως να είναι ένας πυρήνας στην πράξη· Ας υποθέσουμε αρχικά ότι ένας πηρύνας δύναται να ξεχωρήσει μεταξύ όλων των (μη-ισομορφικών) γράφων στον χώρο χαρακτηριστικών. Ένας τέτοιος πυρήνας λέγεται *πλήρης*.

Ορισμός 2.15 (Πλήρης Πυρήνας Γράφων)**.** *Ένας πυρήνας γράφων* $k(G_i, G_j) = \langle \phi(G_i), \phi(G_j) \rangle$ *είναι πλήρης αν η* ϕ *είναι "1-1".*

Οι Gärtner, Flach και Wrobel [25] έδειξαν ότι υπολογίζοντας οποιονδήποτε πλήρη πυρήνα γράφων είναι τουλάχιστον τόσο δύσκολο όσο να αποφασίσουμε αν δύο γράφοι είναι ισομορφικοί. Ως αποτέλεσμα, είναι αδύνατη η χρήση πλήρων πυρήνων γράφων 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 που τα αποτελούν και συγκρίνοντας αυτά τα μέρη με γνωστούς πυρήνες και συλλέγοντας τα αποτελέσματα με ένα συγκεκριμένο τρόπο, να φτιάξουμε τελικά έναν πυρήνα. Οι πυρήνες που προκύπτουν, χρησιμοποιώντας τον παραπάνω σκελετό λέγονται R-συνελικτικοί πυρήνες (R-convolutional kernels) [34]. Οι περισσότεροι

πυρήνες στην βιβλιογραφία ακολουθούν αυτόν τον σκελετό. Αυτοί οι πυρήνες αποσυνθέτουν τους γράφους σε ένα σύνολο υπό-δομών και προσθέτουν της ομοιότητες κάθε ζευγαριού αυτών. Η πιο προφανής τέτοια υποδομή, που μπορούμε να σκεφτούμε είναι ένας τέτοιος υπογράφος. Οι Gärtner, Flach και Wrobel έδειξαν επίσης ότι το πρόβλημα υπολογισμού ενός πυρήνα που συγκρίνει όλα τα υπογραφήματα είναι NP-δύσκολο. Κατά συνέπεια, γίνεται σαφές ότι χρειάζεται να σκεφτούμε μία εναλλακτική: λιγότερο δυνατούς πυρήνες γράφων, που είναι υπολογίσιμοι σε πολυωννυμικό χρόνο. Από την άλλη, είναι απαραίτητο ότι αυτοί οι πυρήνες θα αποτελούν μία εκφραστική μετρική της *ομοιότητας* μεταξύ των γράφων. Είναι κοινό στην βιβλιογραφεία οι πυρήνες να οργανόνονται σε μεγάλες κατηγορίες, που η κάθε μία μελετάει διαφορετικά δομικά χαρκατηριστικά ενός γράφου. Συγκεκριμνα, υπάρχουν πυρήνες που συγκρίνουν γράφους, βάση τυχαίων περιπάτων [23, 12, 81], υποδέντρων [63, 5, 51], κύκλων [38], μονοπατιών [10, 27] και μικρούς υπογράφους [18, 37, 46, 69]. 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]. Ο σκελετός πυρήνα (kernel framework) Weisfeiler-Lehman λειτουργεί πάνω από υπάρχοντες πυρήνες και βελτιώνει την απόδοση τους χρησιμοποιώντας μία επαναληπτική διαδικασία επισημείωσης των κόμβων που βασίζεται στο τεστ ισομορφισμού Weisfeiler-Lehman [70]. Ο σκελετός πυρήνα k-core αποσύνθετει τον γράφο σε ιεραρχίες ένθετων υπογραφημάτων, καθένα από τα οποία παρουσιάζει ισχυρότερη συνδεσιμότητα σε σχέση με το προηγούμενο και έπειτα συνθέτει τις ομοιότητες μεταξύ όλων αυτών συνδιάζοντας τα αποτελέσματα που προκύπτουν [58]. Οι περισσότεροι από τους προαναφερθέντες πυρήνες γράφων συγκρίνουν συγκεκριμένες υποδομές των γράφων (π.χ. γραφίδια (graphlets), κύκλους, υποδέντρα κλπ). Αυτές οι υποδομές αντιστοιχούν σε είτε μικρούς υπογράφους ή σε σχέσεις μεταξύ πολύ μικρών υποσυνόλων από κόμβους. Κατά συνέπεια οι αλγόριθμοι αυτοί εστιάζουν σε τοπικές ιδιότητες ή χαρακτηριστικά των γράφων. Κάποιοι πυρήνες γράφων που δεν περιορίζονται στην σύγκριση υπογραφημάτων, αλλά 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 [40] και πυρήνες που χρησιμοποιούν μετρικές από την θεωρία πληροφοριών όπως την απόκλιση (divergence) Jensen Shannon [6]. Άλλοι πυρήνες όπως η πολυκλιμακωτός Λαπλασιανός (Multiscale Laplacian) [45]. Οι περισσότεροι πυρήνες γράφων έχουν σχεδιαστεί για να λειτουργούν ταυτόχρονα σε γράφους με και χωρίς επισημειώσεις. Αυτό επιτυγχάνεται εύκολα στις περιπτώσεις που ένας γράφος δεν έχει επισημείωσεις στου κόμβους πέρνοντας ως επισημείωση ένα τοπικό χαρκατηριστικό π.χ. τον βαθμό του κόμβου και στις περιπτώσεις των ακμών αντιστοιχίζοντας μία τούπλα με τις επισημειώσεις των κόμβων που ενώνουν την ακμή. Παρόλαυτα, οι περισσότεροι πυρήνες θεωρούν πως οι ακμές έχουν διακριτές (discrete) επισημειώσεις (νούμερο, συμβολοσειρά κλπ). Παρόλο που έχουν μελετηθεί λιγότερο συνεχίζουν να υπάρχουν πυρήνες για γράφους με συνεχείς (continuous) επισημειώσεις (π.χ. διανυσματα). Υπάρχοντες πυρήνες όπως ο *κοντινότερου-μονοπατιού* μπορεί να επεκταθεί για να υποστηρίζει συνεχείς επισημειώσεις, έχοντας σαν αντίβαρο την σημαντική αύξηση υπολογιστικής πολλυπλοκότητας. Πρόσφατα, έγιναν κάποιες ερευνητικές απόπειρες προκειμένου να αναπτυχθούν

πυρήνες γράφων, που συνεχίζουν να λειτουργούν αποδοτικά όσο το μέγεθος του γράφου αυξάνεται [21, 60, 55], χωρίς βέβαια να είναι υπολογιστικά συγκρίσιμοι με πυρήνες βαθμωτών επισημειώσεων. Στη συνέχεια θα δωθεί μία εισαγωγή και θεμελιώδης αναλυση των πυρήνων γράφων που αναπτύχθηκαν στο GraKeL (έκδοση 0.1α3), πληθώρα εκ των οποίων αναφέρθηκε παραπάνω.

2.6.1 Πυρήνες Τυχαίων Περιπάτων

Η πιο πολυμελετημένη οικογένεια πυρήνων γράφων είναι οι *πυρήνες τυχαίων περιπάτων* που ποσοτικοποιούν την ομοιότητα μεταξύ ενός ζευγαριού γράφων βάση του αριθμού των κοινών περιπάτων σε αυτούς τους γράφους [42, 23, 52, 12, 81, 71]. Πυρήνες που ανήκουν σε αυτή την οικογένει έχουν εστιάση 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 εύχρηστος πυρήνας από αυτήν την οικογένεια είναι ο πυρήνας τυχαίων περιπάτων γεωμετρικής προόδου [23] που συγκρίνει άπειρους περιπάτους μέχρι αναθέτωντας ένα βάρος λ^k ($\lambda < 1$) σε περιπάτους με μήκος k προκειμένου να διασφαλήσει σύγκλιση της αντίστοιχης γεωμετρικής σειρά. 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 έανς γράφος με σύνολο κόμβων:

$$V_\times = \{(v_i, v_j) : v_i \in V_i \wedge v_j \in V_j \wedge \ell(v_i) = \ell(v_j)\} \tag{2.15}$$

και σύνολο ακμών:

$$E_\times = \{\{(v_i, v_j), (u_i, u_j)\} : \{v_i, u_i\} \in E_i \wedge \{v_j, u_j\} \in E_j\} \tag{2.16}$$

Η εκτέλεση ενός τυχαίου περιπάτου στο $G_\times$ είναι ισοδύναμο με την εκτέλεση ταυτόχρονων τυχαίων περιπάτων στο G_i και G_j. Ο γεωμετρικός πυρήνας τυχαίων μονοπατιών μετράει κοινούς περιπάτους (εν δυνάμει απείρου μήκους) σε δύο γράφους και ορίζεται ως εξής.

Ορισμός 2.16 (Γεωμετρικός 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 γεωμετρικός πυρήνας τυχαίων περιπάτων ορίζεται ως*

$$K_\times^\infty(G_i, G_j) = \sum_{p,q=1}^{|V_\times|} \Big[\sum_{l=0}^{\infty} \lambda^l A_\times^l\Big]_{pq} = e^T(I - \lambda A_\times)^{-1} e \tag{2.17}$$

όπου I είναι ο ταυτοτικός πίνακας, e είναι ένα διάνυσμα που περιέχει μόνο άσσους και λ 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 ο υπολογισμος 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 εφαρμοσιμότητα 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 $\mathcal{O}(n^3)$ [81]. 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 της ιδιοτιμές του πίνακα και ο P, είναι ο πίνακας που στις στήλες του φέρει τα αντίστοιχα ιδιοδιανυσμάτα που αντιστοιχούν στο αντίστοιχο διαγώνιο στοιχείο του D. Επειδή ο πίνακας A θεωρείτε συμμετρικός $P^{-1} = P^T$. Σαν συνέπεια έχουμε:

$$\begin{aligned} e^T(I - \lambda A_\times)^{-1}e &= e^T(I - \lambda A_i \otimes A_j)^{-1}e \\ &= e^T(I - \lambda(P_i D_i P_i^{-1}) \otimes (P_j D_j P_j^{-1}))^{-1}e \\ &= e^T(I - \lambda(P_i \otimes P_j)(D_i \otimes D_j)(P_i^{-1} \otimes P_j^{-1}))^{-1}e \\ &= (e^T P_i^{-1}) \otimes (e^T P_j^{-1})(I - \lambda D_i \otimes D_j)^{-1}(P_i e) \otimes (P_j e) \end{aligned} \tag{2.18}$$

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

Ορισμός 2.17 (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 γεωμετρικός πυρήνας τυχαίων περιπάτων ορίζεται ως*

$$K_\times^\infty(G_i, G_j) = \sum_{p,q=1}^{|V_\times|} \Big[\sum_{l=0}^{\infty} \frac{\lambda^l}{l!} A_\times^l \Big]_{pq} = e^T \exp(\lambda A_\times) e \tag{2.19}$$

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

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

$$e^T \exp(\lambda A_\times) e = (e^T P_i) \otimes (e^T P_j) \exp(\lambda D_i \otimes D_j)(P_i^{-1} e) \otimes (P_j^{-1} e) \tag{2.20}$$

Ο Mahé κ.α. πρότειναν περαιτέρω επεκτάσεις των πυρήνων τυχαίων μονοπατιών [52]. Συγκεκριμένα, πρότειναν μία μέθοδο εμπλουτισμού (enrichment) των επισημειώσεων που ως προσέγγιση αυξάνει την διακριτική ακρίβεια (specificity) του πυρήνα και στις περισσότερες φορές ελλατώνει την υπολογιστική πολλυπλοκότητα. 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 αντιμετοπίσουν ο Mahé κ.α. πρότειναν ένα δευτέρας τάξης τυχαίο περίπατο Markov. Οι Sugiyama και Borgwardt έδειξαν ότι στην περίπτωση του γεωμετρικού πυρήνα, ο υποβαρισμός 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 ομοίτητας να κυριαρχείται από περιπάτους μήκους 1, ένα φαινόμενο γνωστό ως "halting" [71].

2.6.2 Πυρήνας Κοντινότερων Μονοπατιών

Ο πυρήνας κοντινότερων μονοπατιών shortest-path kernel αποσυνθέτει τον γράφο σε συντομότερα μονοπάτια και συγκρίνεις ζευγάρια από συντομότερα μονοπάτια σε σχέση με τα μήκη τους και τις επισημειώσεις που έχουν οι κόμβοι στα ακρά τους. Σε πρώτη φάση ο πυρήνας μετατρέπει τους γράφους σε γράφους κοντινότερων μονοπατιών. 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 ελάχιστου μονοπάτιου μεταξύ των δύο αυτών κόμβων. Στη συνέχεια δεδομένου του γράφου κοντινότερων μονπατιών ο πυρήνας κοντινότερων μονοπατιών ορίζεται ως εξής:

Ορισμός 2.18 (Πυρήνας Κοντινότερων Μονοπατιών)**.** *Έστω G_i, G_j δύο γράφοι και S_i, S_j οι αντίστοιχοι γράφοι κοντινότερων μονοπατιών. Ο πυρήνας κοντινότερω μονοπατιών ορίζεται στα $S_i = (V_i, E_i)$ και $S_j = (V_j, E_j)$ ως*

$$k(S_i, S_j) = \sum_{e_i \in E_i} \sum_{e_j \in E_j} k^{(1)}_{walk}(e_i, e_j) \tag{2.21}$$

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

Σε γράφους με επισημειωσεις 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:

$$\begin{aligned} k^{(1)}_{walk}(e_i, e_j) = {} & k_v(\ell(v_i), \ell(v_j))\ k_e(\ell(e_i), \ell(e_j))\ k_v(\ell(u_i), \ell(u_j)) \\ & + k_v(\ell(v_i), \ell(u_j))\ k_e(\ell(e_i), \ell(e_j))\ k_v(\ell(u_i), \ell(v_j)) \end{aligned} \tag{2.22}$$

όπου k_v είναι έανς πυρήνας που συγκρίνει επισημειώσεις κόμβων και k_e ένας πυρήνας που συγκρίνει μήκη κοντινότερων μονοπατιών. Οι επισημειώσεις κόμβων συχνά συγκρίνονται μέσω ενος πυρήνα Dirac, ενώ τα μήκη των συντομότερων μονοπατιών συχνά συγκρίνονται με ένα πυρήνα Dirac και πιο σπάνια με ένα πυρήνα brownian bridge [9]. Ο πυρήνας Dirac ορίζεται ως:

Ορισμός 2.19 (Πυρήνας Dirac)**.** *Έστω δύο αντικείμενα $o_i, o_j \in \mathcal{O}$ και μία πράξη ισότητας $'='$ στον χώρο $\mathcal{O}$. Τότε ο πυρήνας Dirac ορίζεται ως:*

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

Ακόμα ο πυρήνας ορίζεται ως:

Ορισμός 2.20 (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 πολλυπλοκότητα 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 Σ το σύνολο όλων των δυνατών συνδιασμών μήκους κοντινότερων μονοπατιών και επισημειώσεων, πράγμα που αποτελεί και τον κυριάρχο όρο στην υπολογιστική πολλυπλοκότητα.

2.6.3 Πυρήνας Γραφιδίων

Ο πυρήνας γραφιδίων αποσυνθέτει ένα γράφο σε γραφίδια (δλδ. μικρού μεγέθους υπογράφος με k κόμβους, όπου συνήθως $k \in \{3, 4, 5\}$) [62] και μετράει το πλήθος των γραφιδίων που ταιριάζονται στους γράφους 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 2.21 (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 συχότητα εμφάνισης κάθε γραφιδίου μεγέθους k σε δύο γράφους. Τότε ο πυρήνας γραφιδίων ορίζεται ως*

$$k(G_i, G_j) = f_{G_i}^\top \, f_{G_j} \tag{2.23}$$

Όπως φαίνεται από τον παραπάνω ορισμός, ο πυρήνας γραγιδίων υπολογίζεται με άμεσα ορισμένες αναπαραστάσεις χαρακτηριστικών. Αρχικά, υπολογίζουμε την αναπαράσταση κάθε γράφου στο χώρο χαρακτηριστικών και έπειτα η τιμή του πυρήνα υποογίζεται 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 [69]. Χρησιμοποιώντας τις ανισότητες του Weissman κ.α. [85], έδειξαν ότι δειγματολειπτώντας 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 απαρίθμιση 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 ισομορπησμ γράφων [84]. Η κύρια ιδέα του αλγορίθμου Weisfeiler-Lehman είναι η αντικατάσταση των επισημειώσεων κάθε κόμβου με ένα πολυσύνολο επισημειώσεων που αποτελούνται από την επισημείωση του αρχικού κόμβου και τών διατεταγμένων επισημειώσεων των γειτόνων. Το προκύπτον πολυσύνολο συμπιέζεται σε μία καινούργια 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 ανν ταυτίζονται οι επισημειώσεις πολυσυνόλων τους. Πιο τυπικά, δεδομένου ενός γράφου $G = (V, E)$ 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 ℓ_i η οποία έχει προκύψει μετά από i επαναλήψεις της διαδικασίας επανεπισημείωσεις 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 εως το h, $\{G_0, G_1, \ldots, G_h\}$.

Ορισμός 2.22 (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*

$$k_{WL}(G, G') = k(G_0, G'_0) + k(G_1, G'_1) + \ldots + k(G_h, G'_h) \tag{2.24}$$

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 υπόψην διακριτές επισημειώσεις κόμβων μπορεί να ενταχθεί στον σκελετό πυρήνα Weisfeiler-Lehman και να συγκρίνει τους γράφους βάση ολόκληρης της ακολουθίας Weisfeiler-Lehman.
Όταν ο πυρήνας βάσης συγκρίνει υποδέντα που εχξήχθησαν από τους δύο γράφους, ο υπολογισμός αφορά την αρίθμηση κοινών αρχικών και συμπιεσμένων επισημειώσεων στους δύο γράφους. Ο πυρήνας Weisfeiler-Lehman-υποδέντρων είναι ένα πολύ δημοφιλής αλγόριθμος που θεωρείτε από τους πιο καλούς αυτή τη στιγμή όσον αφορά την ταξινόμηση γράφων.

Ορισμός 2.23 (Πυρήνας Weisfeiler-Lehman υποδέντρων)**.** *Έστω G, G' δύο γράφοι. Ορίζουμε ως $\Sigma_i \subseteq \Sigma$ το σύνολο των γραμμάτων που προκύπτουν αν σαν επισημειώσεις κόμβων τουλάχιστον μία φορά στα G και G' στο τέλος της i^{th} του αλγορίθμου Weisfeiler-Lehman. Έστω Σ_0 το σύνολο των αρχικών επισημειώσεων των γράφων G και G'. Θα υποθέσουμε ότι όλα τα Σ_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 το νούμερο των εμφανίσεων του γράμματος σ_{ij} στον γράφο G.*

Ο πυρήνας υποδέντρων Weisfeiler-Lehman (Weisfeiler-Lehman subtree kernel)μεταξύ δύο γράφων G και G' με h επαναλήψεις ορίζεται ως

$$k(G, G') = \langle \phi(G), \phi(G') \rangle \tag{2.25}$$

όπου

$$\phi(G) = (c_0(G, \sigma_{01}), \ldots, c_0(G, \sigma_{0|\Sigma_0|}), \ldots, c_h(G, \sigma_{h1}), \ldots, c_h(G, \sigma_{h|\Sigma_h|})) \tag{2.26}$$

και

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

Μπορούμε να δείξουμε ότι παραπάνω ορισμός είναι ισοδύναμος με την σύγκριση των αριθμών κοινών υποδέντρων μεταξύ των δύο γράφων [70]. Ο πυρήνας υποδέντρων ονομάζεται σε άλλες περιπτώσεις πυρήνας ιστογραμμάτος 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 Πύρηνας Πυραμιδικού Ταιριάσματος

Ο πυρήνας πυραμιδικού τεριάσματος (pyramid match είναι πολύ δημοφιλής στον χώρο της όρασης υπολογιστών και έχει αποδειχθεί ιδιαίτερα χρήσιμως σε πολλές εφαρμογε από την αναγνώρισης αντικειμένων μέχρι την ανάκτηση εικόνας ρετριεαλ [31, 47]. Ο πυρήνας πυραμιδικού τεριάσματος επεκτείνει την εφαρμοσιμότητα τους σε δεδομένα δοομημένα σαν γράφους [59]. Αυτός ο πηρύνας μπορεί να διαχειριστεί γράφους με διακριτές επισημειώσεις αλλά και γράφους χωρίς επισημειώσεις.

Ο πυρήνας πυραμιδικού τεριάσματος πρώτα παριστά όλα 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 της 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 ιστογράματα με μάι βεβαρισμένη συνάρτηση τομής ιστογραμμάτων.

Αρχικά, ο πυρήνας διαμερίζει των χώρο χαρακτηριστικών σε περιοχές με αυξάνων μεγαλύτερο μέγεθος και πέρνει το βεβαρισμένο άθροισμα όλων των αντιστοιχίσεων που προκύπτουν σε κάθε επίπεδο. Δύο σημεία ταιριάζουν αν πέφτουν στην ίδια περιοχή, ενώ ταιριάσματα μεταξύ περιοχών μεγαλύτερου βαρίζονται λιγότερο από αυτά μικρότερων περιοχών. Ο πυρήνας επαναληπτικά προσαρμόζει ένα πλέγμα κελιών αυξανόμενου μεγέθους στον d-διάστατο μοναδιαίο υπερκύβο. Κάθε κελί συνδέται με μία συγκεκριμένη διάσταση και το μέγεθος του σε κάθε διασταση 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 μοναδιάιος υπερκύβος έχει 2^l κελιά σε κάθε διάσταση και $D = 2^l d$ κελιά στο σύνολο. Δεδομένου ενός ζευγαριού γράφων G, G', έστω H^l_G και $H^l_{G'}$ που συμβολίζουν τα ιστογράμματα των G και G' στα επίπεδα l, και $H^l_G(i)$, $H^l_{G'}(i)$, ο αριθμός των κόμβων στα G, G' που βρίσκονται στο i-οστό κελί. Ο αριθμός των σημείων μεταξύ δύο σύνολα, τα οποία ταυτίζονται στο επίπεδο l υπολογίζεται έπειτα χρησιμοποιώντας την συνάρτηση τομής ιστογραμμάτων

$$I(H^l_G, H^l_{G'}) = \sum_{i=1}^{D} \min\left(H^l_G(i), H^l_{G'}(i)\right) \tag{2.28}$$

Τα ταιριάσματα που προκύπτουν στο επίπεδο l συμβαίνπυν ακόμα στα επίπεδα $0, \ldots, l-1$. Μας ενδιαφέρουν μόνο εκείνα τα τεριάσματα που είναι καινούργια σε κάθε νέο ταίριασμα που δίνονται από την διαφορά $I(H^l_{G_1}, H^l_{G_2}) - I(H^{l+1}_{G_1}, H^{l+1}_{G_2})$ για $l = 0, \ldots, L-1$. Ο αριθμός των νέων ταιριασμάτων που προκύπτει σε κάθε επίπεδο στην πυραμίδα βαρίζεται με βάση το μέγεθος των κελιών αυτού του επιπέδου. Ταιρίασματα που προκύπτουν μεταξύ μικρότερων κελιών βαρίζονται περισσότερο από αυτά που που γίνονται 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

$$k(G, G') = I(H^L_G, H^L_{G'}) + \sum_{l=0}^{L-1} \frac{1}{2^{L-l}} \left(I(H^l_G, H^l_{G'}) - I(H^{l+1}_G, H^{l+1}_{G'})\right) \tag{2.29}$$

Η πολλυπλοκότητα του είναι ίση με $\mathcal{O}(dnL)$ όπου n ο αριθμός των κόμβων στους γράφους υπό σύγκριση.

Στην περίπτωση γράφων με επισημειώσεις, ο πυρήνας περιορίζει τα ταιριάσματα να 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 προκύπτον πυρήνας για επισημειωμένους γράφους αντιστοιχεί στο άθροισμα των ξεχωριστών πυρήνων

$$k(G, G') = \sum_{i=1}^{c} k^i(G, G') \tag{2.30}$$

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 δωθεί η επισημείωση i.

2.6.6 Πηρύνας Lovász ϑ

Ο αριθμός Lovász $\vartheta(G)$ ενός γράφου $G = (V, E)$ είναι ένας πραγματικός αριθμός που αποτελεί το άνω φράγμα στην χωρητικότητα Shannon ενός γράφου. Εισήχθει από τον László Lovász το 1979 [49]. Ο αριθμός Lovász είναι πολύ στενά συνδεδεμένος με την έννοια της ορθοκανονικής αναπαραστάσης γράφων. Η ορθοκανονική αναπαράσταση ενός γράφου G αποτελείται από ένα σύνολο μοναδιαίω διανυσμάτων $U_G = \{\mathbf{u}_i \in \mathbb{R}^d : ||\mathbf{u}_i|| = 1\}_{i \in V}$ όπου σε κάθε κόμβος 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

$$\vartheta(G) = \min_{\mathbf{c}, U_G} \max_{i \in V} \frac{1}{(\mathbf{c}^\top \mathbf{u}_i)^2} \tag{2.31}$$

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 κόνο που εσωκλύει 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 βελτιστοποίηση ημιορισμένου προγραμματισμού.

Ο πυρήνας Lovász ϑ χρησιμοποιεί τις ορθοκανονικές αναπαραστάσεις που συνδέονται με τον αριθμό Lovász για να συγκρίνει δύο γράφους [40]. Αυτός ο πυρήνας αφορά γράφους χωρίς επισημειώσεις. Δεδομένου μίας συλλογής γράφων, πρώτα αναπαριστά τις ορθοκανονικές αναπαραστάσεις των κόμβων κάθε γράφου, υπολογίζοντας τον αριθμό Lovász ϑ. 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

$$\vartheta_S(G) = \min_{\mathbf{c}} \max_{i \in S} \frac{1}{(\mathbf{c}^\top \mathbf{u}_i)^2} \tag{2.32}$$

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 $\vartheta(G)$ οφ G. Η τιμή Lovász ενός συνόλου κόμβων S αναπαριστά την γωνία του μικρότερου κόνου 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:

$$k_{Lo}(G, G') = \sum_{S \subseteq V} \sum_{S' \subseteq V'} \delta(|S|, |S'|) \frac{1}{Z_{|S|}} k(\vartheta_S(G), \vartheta_{S'}(G')) \tag{2.33}$$

όπου $Z_{|S|} = \binom{|V|}{|S|}\binom{|V'|}{|S|}$, $\delta(|S|, |S'|)$ είναι ένας πυρήνας dirac 2.19 και k είναι ένας θετικά ημιορισμένος πυρήνας μεταξύ πραγματικών αριθμών (π.χ. γραμμικός ή γκαουσιανός).

Ο πυρήνας Lovász ϑ αποτελείται από δύο κύρια βήματα: (1) υπολογισμός του αριθμού Lovász ϑ 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 ϑ, δεν είναι κάτι εφικτό στις περισσότερες πραγματικές εφαρμογές από την στιγμή που απαιτεί τον υπολογισμό των ελάχιστων κόνων που εσωκλείουν 2^n σύνολα κόμβων.

Όταν λοιπόν ασχολούμαστε με μεγάλους γράφους, είναι σημαντικό να να καταφύγουμε στην δειγματοληψία. Δεδομένου ενός γράφου G, αν αντί να υπολογίσουμε την τιμή Lovász σε όλα τα 2^n σύνολα κόμβων, ο αλγόριθμος φέρει εις πέρας αυτόν τον υπολογισμό σε ένα μικρότερο πλήθος από υπογράφους $\mathfrak{S} \in 2^V$. Τότε, ο πυρήνας Lovász ϑ ορίζεται ως εξής

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

όπου $\hat{Z}_{|S|} = |\mathfrak{S}_{|S|}||\mathfrak{S}'_{|S|}|$ και $\mathfrak{S}_{|S|}$ είναι το υποσύνολο του $\mathfrak{S}$ που αποτελείται από όλα τα σύνολα πληθηκοτήτων $|S|$, που είναι $\mathfrak{S}_{|S|} = \{B \in \mathfrak{S} : |B| = |S|\}$.
Η χρονική πολλυπλοκότητα του υπολογισμού $\hat{k}_{Lo}(G, G')$ είναι $\mathcal{O}(n^2 m\epsilon^{-1} + s^2 T(k) + sn)$ όπου $T(k)$ είναι η πολλυπλοκότητα 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 βλετιστοποίησης ημιορισμένου προγραμματισμού που υπολογίζει τον αριθμό Lovász ϑ. 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-ϑ

Ο πυρήνας SVM-ϑ συνδέεται άμεσα με τον πυρήνα Lovász ϑ [40]. Ο πυρήνας Lovász ϑ υποφέρει από μεγάλη μεγάλη υπολογιστική πολλυπλοκότητα και ο πυρήνας SVM-ϑ σχεδιάστηκε σαν μία πιο αποδοτική εναλλακτική. Όπως και ο πυρήνας Lovász ϑ, αφορά γράφους χωρίς επισημειώσεις.
Δωθέντος ενός γράφου $G = (V, E)$ τέτοιου ώστε $|V| = n$, ο αριθμός Lovász του G μπορεί να οριστεί ως

$$\vartheta(G) = \min_{\mathbf{K} \in L} \omega(\mathbf{K}) \tag{2.34}$$

όπου $\omega(\mathbf{K})$ είναι το Σ"Μ μίας κατηγορίας που δίνεται από

$$\omega(\mathbf{K}) = \max_{\alpha_i > 0} 2 \sum_{i=1}^{n} \alpha_i - \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j \mathbf{K}_{ij} \tag{2.35}$$

και L είναι ένα σύνολοα πό θετικά ημιορισμένους πίνακες που ορίζεται ως

$$L = \{\mathbf{K} \in S_n^+ : \mathbf{K}_{ii} = 1, \mathbf{K}_{ij} = 0 \, \forall (i, j) \notin E\} \tag{2.36}$$

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

$$\mathbf{K}_{LS} = \frac{\mathbf{A}}{\rho} + \mathbf{I} \tag{2.37}$$

όπου $\mathbf{A}$ είναι ο πίνακας γειτνίασης του G, $\mathbf{I}$ είναι ο $n \times n$ ταυτοτικός πίνακς και $\rho \geq -\lambda_n$ με λ_n να είναι η μικρότερη ιδιοτιμή του $\mathbf{A}$. Ο πίνακας $\mathbf{K}_{LS}$ είναι θετικά ημιορισμένος από κατασκευή και έχει αποδεικνύεται στο [39] ότι

$$\omega(\mathbf{K}_{LS}) = \sum_{i=1}^{n} \alpha_i \tag{2.38}$$

όπου α_i είναι οι όροι που μεγιστοποιούν την εξίσωση 2.35. Ακόμα, έχει αποδειχθεί ότι σε συγκεκριμένς οικογένειες γράφων (π.χ. τυχαίοι γράφοι Erdös Rényi), το $\omega(\mathbf{K}_{LS})$ είναι με μεγάλη πιθανότητα ένας προσεγγίζει κατά ένα σταθερό παράγοντα το $\vartheta(G)$.

Τότε, ο πυρήνας SVM-ϑ ορίζεται ως εξής

$$k_{SVM}(G,G') = \sum_{S\subseteq V}\sum_{S'\subseteq V'} \delta(|S|,|S'|)\frac{1}{Z_{|S|}} k\Big(\sum_{i\in S}\alpha_i, \sum_{j\in S'}\alpha_j\Big) \tag{2.39}$$

όπου $Z_{|S|} = \binom{|V|}{|S|}\binom{|V'|}{|S|}$, $\delta(|S|,|S'|)$ είναι ένας πυρήνας Dirac 2.19 και k ένας θετικά ημιορισμένος πυρήνας μεταξύ πραγματικών τιμών (π.χ. γραμμικός, γκαουσιανός).

Ο πυρήνας SVM-ϑ απαρτίζεται από τρία κύρια βήματα: (1) κατασκευή του πίνακα $\mathbf{K}_{LS}$ του G που παίρνει χρόνο $\mathcal{O}(n^3)$ (2) λύση το προβλήματος SVM μίας κλάσης σε χρόνο $\mathcal{O}(n^2)$, προκειμένου να χρησιμοποιήσουμε τις τιμές των α_i και (3) υπολογισμό του αθροίσματος των τιμών α_i για όλους τους υπογράφους (δηλαδή τα υποσύνολα κόμβων $S \subseteq V$) του κάθε γράφου. Όπως και στον πυρήνα Lovász ϑ ο υπολογισμός της παραπάνω ποσότητας για όλα τα σύνολα 2^n των κόμβων δεν είναι υπολογιστικά εφικτή σε πραγματικά δεδομένα.

Το αδδρεσς της αβοε ισσυε, της SVM-ϑ κερνελ εμπλοψς σαμπλινγ σχημες. Γιεν α γραπη G, της κερνελ σαμπλες α σπεςιφις νυμβερ οφ συβγραπης $\mathfrak{S} \in 2^V$. Τηεν, της SVM-ϑ κερνελ ις δεφινεδ ας φολλοως

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

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 $\hat{k}_{SVM}(G,G')$ είναι $\mathcal{O}(n^3 + s^2T(k) + sn)$ όπου $T(k)$ είναι η πολλυπλοκότητα 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 α_i. Ο τρίτος και τελευταίος όρος αφορά το κόστος υπολογισμού του αθροίσματος των τιμών α_i για τα δειγματοληπτημένα υποσύνολα κόμβων.

2.6.8 Πολυκλιμακωτός Λαπλασιανός Πυρήνας

Ο πολυκλιμακωτός Λαπλασιανός πυρήνας μπορεί να χειριστεί επισημειωμένους γράφους, με είτε διακριτές επισημειώσεις είτε χαρακτηριστικά [45]. Λαμβάνει υπόψη την δομή των γράφων σε ένα εύρος διαφορετικών κλιμάκων χτίζοντας μία ιεραρχία από εμφωλιασμένους υπογράφους. Αυτοί οι υπογράφοι συγκρίνονται ο ένας με τον άλλον κάνοντας χρήση ενός άλλου πυρήνα, που αποκαλείται Λαπλασιανός πυρήνας στον χώρο χαρακτηρηστικών. Αυτός ο πυρήνας δύναται να καταστεήσει έναν πυρήνα που ορίζεται μεταξύ κόμβους γράφων σε ένα πυρήνα μεταξύ των ίδιων των γράφων. Από τη στιγμή που ο ακριβής υπολογισμός του πολυκλιμακωτού Λαπλασιανού είναι μία πολύ υπολογιστικά ακριβή διαδικασία, ο πυρήνας χρησιμοποιείε μία πιθανοτική διαδικασία παρόμοια με την πολύ γνωστή προσσέγγιση μέθοδο Nyström για τον υπολογισμό μητρών πυρήνα [86].

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 γ 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. Παρόλαυτα, 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 μετάθεσεις χαρακτηριστικά των κόμβων. Τέτοια χαρακτηριστικά μπορούν για παράσδειγμα 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 υπ'όψην 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 γράφφων, 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 κ μεταξύ δύο κόμβων v_1 και v_2 αντιστοιχεί στο εσωτερικό γινόμενο των διανυσμάτων χαρακτηριστικών τους

$$\kappa(v_1, v_2) = \phi(v_1)^\top \phi(v_2) \tag{2.40}$$

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

$$\mathbf{K}_{ij} = \kappa(\bar{v}_i, \bar{v}_j) = \phi(\bar{v}_i)^\top \phi(\bar{v}_j) \tag{2.41}$$

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

$$\xi_i = \frac{1}{\sqrt{\lambda_i}} \sum_{l=1}^{n_1+n_2} [\mathbf{u}_i]_l \phi(\bar{v}_l) \tag{2.42}$$

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 χαρακτηρηστικών του Λαπλασιανού πυρήνα γράφων που δημιουργείται από τον πυρήνα βάσης κ ορίζεται ως

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

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 εμφωλιασμένων υπογράφων, όπου κάθε υπογράφος έχει κέντρο γύρω από έναν κόμβο και υπολογίζει τον γενικευμένο χώρο χαρακτηρηστικών του Λαπλασιανού πυρήνα γράφων μεταξύ κάθε ζευγαριού υπογράφων. Έστω G ο πυρήνας με σύνολο κόμβων V και κ ένας θετικά ημιορισμένος πίνακς στο V. Ας υποθέσουμε ότι για κάθε $v \in V$, έχουμε μία εμφωλιασμένη ακολουθία από L γειτονιές

$$v \in N_1(v) \subseteq N_2(v) \subseteq \ldots \subseteq N_L(v)$$

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 χαρακτηρηστικών k^{κ}_{FLG} που επάγεται από τον πυρήνα βάσης κ μεταξύ των υπογράφων του χαμηλότερου επιπέδου (δηλ. των κόμβων)

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

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 $G_l(u)$

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

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 $\mathcal{O}(Ln^2)$ υπολογισμού 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 χαρακτηρηστικών μπορεί να χρειαστεί $\mathcal{O}(n^3)$ χρόνο. Αυτό σημαίνει ότι στη χειρότερη περίπτωση, το χειρότερο κόστος υπολογισμού είναι 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 ιδιοσιανυσμάτων 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 γενικευμενος Λαπλασιανός πυρήνας χαρακτηρηστικών γράφων που προκύπτει από το κ 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 χαρακτηρηστικών. 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 προκύπτον υποδειγματολειπτημένος χώρος χαρακτηρηστικών 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 προσσεγγιστικός 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 προσσέγγισης περιορίζοντας το $\tilde{W}$ να είναι ο χώρος που προκύπτει από τα πρώτα $\hat{p} < \tilde{p}$ διανύσματα βάσης (ταξινομημένα κατά φθίνουσα ιδιοτιμή), εφαρμόζοντας αποτελεσμετικά την τεχνική του PCA πυρήνα (kernel PCA) στο $\{\phi(\tilde{v})\}_{\tilde{v} \in \tilde{V}}$. Ο συνδιασμός 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 [58]. 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 *συνδετικότητας* (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 [68]. Ιν ρεςεντ ψεαρς, η αποσύνθεση k-core έχει καθιερωθεί σαν ένα βασικό εργαλείο σε πολλές εφαρμογές, όπως η αναπαράσταση δικτύων [3], στην πρόβλεψη πρωτεϊνικής λειτουργίας [87] και στη συσταδοποίηση γράφων [26].

2.6.9.1 Core Αποσύνθεση

Έστω $G = (V, E)$ ένας μη κατευθυνόμενος και μη βεβαρισμένος γράφος. 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 $G' = (S, E(S))$ (βλ. 2.8) των κόμβων S. Έστω G ένας γράφος και G' ο υπογράφος του G (βλ. 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, οποίος δεν κατάναγκη συνδεδεμένος. Ο αριθμός κόρας $c(v)$ ενός κόμβου v ισούται με την μεγαλύτερη τάξη κόρας στην οποία ο v

ανήκει. 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

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

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 $\mathcal{O}(n+m)$ [54, 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 να είναι οποιοσδήποτε πυρήνας για γράφους. Τότε, η παραλλαγή core του πυρήνα βάσης k ορίζεται ως

$$k_c(G, G') = k(C_0, C'_0) + k(C_1, C'_1) + \ldots + k(C_{\delta^*_{min}}, C'_{\delta^*_{min}})$$

όπου δ^*_{min} είναι οι ελάχιστος βαθμός εκφαλισμού 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,..., δ^*_{min}οστής-κόρας των G και G', αντίστοιχα. Αποσυνθέτοντας τους γράφους σε υπογράφους αυξανόμενης βαρύτητας ο αλγόριθμος είναι ικανός να αποδόσει με μεγαλύτερη ακρίβεια υφέρπουσα ομοίοτητα στην δομή δύο γράφων.

Η υπολογιστική πολλυπλοκότητα του σκελετού πυρήνα core εξαρτάται από την πολλυπλοκότητα του πυρήνα βάσης και τον εκφυλισμό των υπο σύγκριση γράφων. Δεδομένου ενός ζευγαριού γράφων G, G' και ενός αλγορίθμου A για την σύγκριση των δύο γράφων, έστω $\mathcal{O}_A$ η χρονική πολλυπλοκότητα του αλγορίθμου A. Έστω ακόμα $\delta^*_{min} = \min\left(\delta^*(G), \delta^*(G')\right)$ οι ελάχιστοι βαθμοί εκφυλισμού των δύο γράφων. Τότε, η πολλυπλοκότητα υπολογισμού της παραλλαγής core ενός αλγορίθμου A 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 κομβων του και την μέγιστη ιδιοτιμή του πίνακα γειτνίασης λ_1. Από την στιγμή που για τους περισσότερους *πραγματικούς* γράφους ισχύει ότι $\lambda_1 \ll n$ και $\delta^*_{max} \ll n$, η επαύξηση της χρονική πολλυπλοκότητα του πυρήνα βάσης δεν είναι ιδιαίτερα μεγάλη.

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 εώς μία συγκεκριμένη απόσταση από την ρίζα [17]. Έπειτα συγκρίνει τους γράφους βάση αυτών των ζευγαριών ριζωμένων υπογράφων. Για να αποφευχθεί ο έλεγχος ισομορφισμού χρησιμοποιούνται σταθερά στοιχεία των γράφων προκειμένου να παράγουν μία αντιπροσωπευτική κωδικοποίηση καθενός 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 $R_{r,d}(A_v, B_u, G)$ ε, B_u και ενός γράφου $G = (V, E)$ που είναι αληθές ανν και τα δύο A_v, B_u βρίσκονται στο $\{N_r^v : v \in V\}$, όπου απαιτούμε τα A_v, B_u να είναι ισομορφικά σε μία N_r^v προκειμένου να επαληθεύσουμε ότι ανήκουν στο σύνολο, καθώς και ότι $D(u, v) = d$. Θα συμβολήσουμε 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 χρησιμοποεί τον παρακάτω πυρήνα

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

όπου η συνάρτηση δ είναι 1 αν οι υπογράφοι της εισόδου είναι ισομορφικοί 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

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

όπου $\hat{k}_{r,d}$ είναι μία κανονικοποιημένη έκδοση 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 από πάνω έκδοση εξασφαλίζει ότι σε όλες οι 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 ταίριασματος μεταξύ δύο γράφων (δλδ. τον δ πυρήνα) που είναι ισοδύναμος με την επίλυση του προβλήματος ισομορφισμού. Η λύση του προβλήματος ισομορφισμού γράφων δεν 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 Σ^* είναι το σύνολο συμβολοσειρών σε ένα

πεπερασμένοαλφάβητο Σ. 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 τριπλέτες αποστάσης-απόστασης 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. Συνεπώ, 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 βάση των νέων επισημειώσεων των τερματικών της κόμβων και την αρχική 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 $\mathcal{L}^e(u,v)$ φορ αλλ $(u,v) \in E(S)$. 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, ο πυρήνας ελέγχει απλώς αν το αναγνωριστικό όλων των ζευγαριών ταυτίζεται.

Η υπολογιστική πολλυπλοκότητα του πυρήνα νειγηβορηοοδ συβγραπη παιρωισε διστανσε κερνελ είναι $\mathcal{O}(|V||S||E(S)|\log|E(S)|)$ και κυριαρχείται από της επαναλαμβανόμενες επαναλήψεις υπολογισμού της αναλοίωτης κάθε γράφου για κάθε κόμβο του. Συνεπώς για μικρές τιμές των d^* και r^* είναι μία διαδικασία σταθερού χρόνου και συνεπώς η συνολική πολλυπλοκότητα του αλγορίθμου στην πράξη είναι 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 [37]. Συγκρίνει γράφους ανανεώνοντας τις επισημειώσεις των κόμβων τους και μετρώντας τον κοινό αριθμό τους. Ο πυρήνας αντικαθιστά τις διακριτές επισημειώσεις των κόμβων με δυαδικούς πίνακες δεδομένου μήκους και έπειτα χρησιμοποιεί λογικές πράξεις με τις οποίες τις ανανεώνει προκειμένου να περιέχουν πληροφορία που αφορά την δομή της γειτονικής δομής κάθε κόμβου.

Έστω $\ell : \mathcal{V} \to \Sigma$ μία συνάρτηση που απεικονίζει τους κόμβους του γράφου σε ένα αλφάβητο Σ, το οποίο αποτελεί το σύνολο των δυνατών διακριτών κόμβων 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 συμπεριλαμβάει 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

κύλιση (ROT). Έστω ότι με $XOR(s_i, s_j) = s_i \oplus s_j$ συμβολίζουμε την πράξη XOR μεταξύ δύο επισημειώσεις 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

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

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 και η διαδικασία ανενέωσης 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 κατακερματισμό γειτονιάς ως

$$NH(v) = ROT_1\big(\ell(v)\big) \oplus \big(\ell(u_1) \oplus \ldots \oplus \ell(u_d)\big)$$

Ο προκύπτων κατακερματισμός $NH(v)$ είναι και πάλι ένας διαδικός 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 που συμπτωματικά ταυτίζονται οι κατακερματισμένες τιμές (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 (πρβλ. την διαδικασίασια ανανέωσης επισημειώσεων του σκελετού πυρήνα Weisfeiler Lehman ;;).

2.6.11.2 Ευαίσθητο στο μέτρημα κατακερματισμό γειτονιών

Ο απλός κατακερματισμός γειτονιών όπως περιγράφτηκε παρπάνω υποφέρει από ένα πλήθος συμπτωματικών ταυτίσεων των κατακερματισμένων τιμών. Συγκεκριμένα, οι τιμές κατακερματισμού των γειτονιών για δύο ανεξάρτητους κόμβους έχουν μικρότερη πιθανότητα να ταυτίζονται ακόμα και ακόμα και στην περίπτωση που δεν υπάρχουν συμπτωματικές συγρούσεις κατακερματισμών. Τέτοιες προβληματικές συγκρούσεις κατακερματισμών, μπορεί να επηρεάσουν την το γεγονός ότι η τελική μήτρα πυρήνα πίνακας 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 (συκεκριμένα radix sort με πολυπλοκότητα $\mathcal{O}(dn)$ γραμμική ως προς n, δεδομένου ενός σταθερού d) για να ευθυγραμμίσει τις bit επισημειώσεις 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

$$\ell'_i = ROT_o\big(\ell_i \oplus o\big) \tag{2.43}$$

όπου ℓ_i, ℓ'_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 εξαρτώμενη του αριθμού επαναλήψεων κάθε επισημείωσης. Τελικά, ο ευαίσθητος στο μέτρη κατακερματισμός γειτονιών υπολογίζεται ως

$$CSNH(v) = ROT_1\big(\ell(v)\big) \oplus \big(\ell'_1 \oplus \ldots \oplus \ell'_l\big) \tag{2.44}$$

Τόσο ο απλό και ο ευαίσθητος στο μέτρημα καταμερισμός γειτονιών μπορούν να ειδωθούν ως γενικές προσεγγίσεις για τον εμπλουτισμό των επισημειώσεων των κόμβων βάση της κατανομής επισημειώσεων των γειτονικών κόμβων.

2.6.11.3 Υπολογισμός Πυρήνα

Η διαδικασίες ανανέωσης επισημειώσεων κατακερματισμού γειτονιάς που παρουσιάζονται παραπάνω συνοψίζουν την πληροφορία γειτονιάς των γειτόνων για κάθε κόμβο. Τότε, δεδομένου δύο γράφων G και G', οι ανανεωμένες επισημειώσεις των κόμβων τους συγκρόνονται βάση της επόμενης συνάρτησης

$$\kappa(G, G') = \frac{c}{|V| + |V'| - c} \tag{2.45}$$

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 ημιορισμένο [30].

Οι διαδικασίες ανανέωσης επισημειώσεων δεν εφαρμόζονται κατάνάγκη μία και μοναδική φορά, αλλά μπορούν να εφαρμοστούν επαναληπτικά. Ανανεόνοντας 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 τους ανανεώνονται $1, \ldots, h$ φορές αντόστοιχα. Τότε, δεδομένου δύο γράφων G και G',

ο πυρήνας κατακερματισμού των γειτονιών ορίζεται ως

$$k(G, G') = \frac{1}{h} \sum_{i=1}^{h} \kappa(G_i, G'_i) \tag{2.46}$$

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 [46]. Ο πυρήνας είναι πολύ γενικός από τη στιγμή που μπορεί να εφαρμοστεί σε γράφους που περιέχουν επισημειώσεις κόμβων, επισημειώσεις ακμών, χαρακτηριστικά κόμβων και χαρακτηριστικά ακμών.

Έστω $\mathcal{G}$ ένα σύνολο γράφων. Θα υποθέσουμε ότι οι γράφοι που περιέχονται στο σύνολο έχουν επισημειώσεις ή χαρακτηρηστικά. Συγκεκριμένα, έστω $\ell : \mathcal{V} \cup \mathcal{E} \to \mathcal{L}$ μία συνάρτηση επισημείωσης που αντιστοιχίζει είτε διακριτές επισημειώσεις είτε συνεχείς στους κόμβους και τις ακμές. Ένας ισομορφισμό γράφων 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, δηλ. αν $\psi \in V \times V \to V' \times V'$ είναι η αντιστοίχιση των ζευγαριών κόμβων όπως προκύπτει από την ¨1-1' απεικόνιση ϕ 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. 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. Ο πυρήνας ταιριάσματος υπογράφων ορίζετια ως

$$k(G, G') = \sum_{\phi \in \mathcal{B}(G,G')} \lambda(\phi) \prod_{v \in S} \kappa_V(v, \phi(v)) \prod_{e \in S \times S} \kappa_E(e, \psi(e)) \tag{2.47}$$

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

Μία στιγμιότυπο του πυρήνα ταιριάσματος υπογράφων προκύπτει άμα θέσουμε τις συναρτήσεις κ_V, κ_E φυνςτιονς ως εξής

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

$$kai \tag{2.48}$$

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

Που είναι γνωστός σαν ο πυρήνας ταιριάσματος υπογράφων. Αυτό ο πυρήνας μετράει τον αριθμό των ισομορφικών υπογράφων που περιέχονται μεταξύ των δύο γράφων.

Για να μετρήσει τον αριθμό των ισομορφισμών μεταξύ υπογράφων, ο πυρήνας στηρίζεται πάνω στο αποτέλεσμα του Levi [48] 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 το κοινό πυρήνα ισομορφισμού υπογράφων απλώς μετρώντας τις κλίκες στον γινόμενο γράφο.

Ο γενικός πυρήνας ταιριάσματος υπογράφων επεκτείνει την θεορία του Levi και κατασκευάζει ένα βεβαρισμένο γινόμενο γράφο προκειμένου να επιτρέψει πιο ευέλικτο φλεξιβλε σχορινγ οφ βιθεστιονς. Δεδομένου δύο γράφων $G = (V, E)$, $G' = (V', E')$, και πυρήνες κόμβων και ακμών κ_V και κ_E, ο βεβαρισμένος γινόμενος γράφος $G_P = (V_P, E_P)$ των G και G' ορίζεται ως

$$\begin{aligned} V_P &= \{(v, v') \in V \times V' : \kappa_V(v, v') > 0\} \\ E_P &= \{((v, v'), (u, u')) \in V_P \times V_P : v \neq u \wedge v' \neq u' \wedge \kappa_E((v, v'), (u, u')) > 0\} \\ c(u) &= \kappa_V(v, v') \quad \forall u = (v, v') \in V_P \\ c(e) &= \kappa_E((v, u), (v', u')) \quad \forall e \in E_P, \\ \text{όπου } e &= ((v, v'), (u, u')) \end{aligned} \tag{2.49}$$

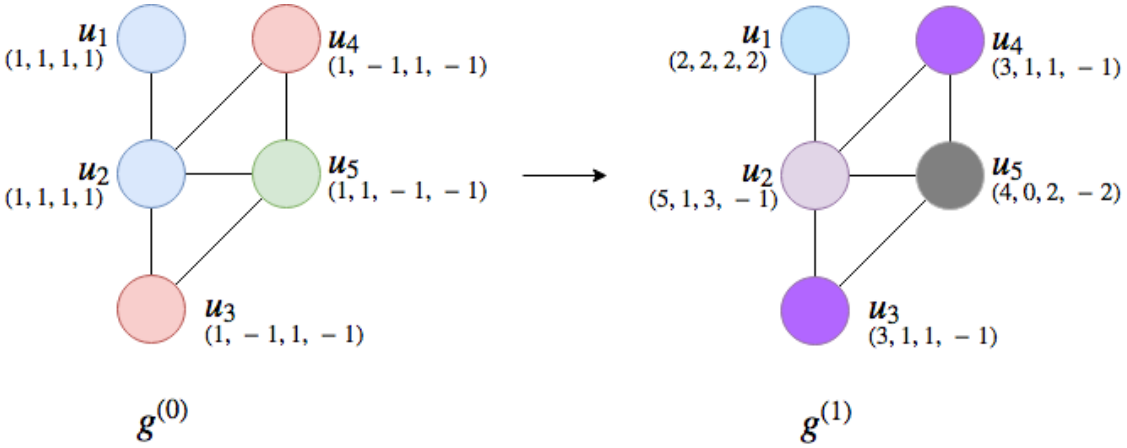
Αφού δημιουργήσει τον βεβαρισμένο γινόμενο γράφο, ο πυρήνας απαριθμεί τις κλίκες. Συγκεκριμένα ξεκινάει με μία κενή κλίκα και την επεκτείνει βήμα βήμα μέσω όλων των κόμβων, διατηρώντας την ιδιότητα της κλίκας. Έστω w το βάρος μίας κλίκας C. Όποτε μία κλίκα C επεκτείνεται σε ένα νέο κόμβο v, το βάρος τις 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 εμπλοτισμού των επισημειώσεων σαν αυτή του πυρήνα κατακερματισμού γειτόνων και του σκελετού πυρήνα Weisfeiler-Lehman, εισήχθει από τους Tetsuya Kataoka και Akihito Inokuchi στο [43], γνωστός ως πυρας κώδικα Hadamard.

Δεδομένου ενός συνόλου διακριτά επισημειωμένων γράφων $\mathbf{G} = [G]_{i=1}^{N}$, συλλέγουμε το σύνολο Σ όλων των διαφορετικών επισημειώσεων του $\mathbf{G}$. Η 2^k-οστή μήτρα κώδικα Hadamard H_{2^k}, ορίζεται ως εξής:



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

$$H_{2^{k+1}} = \begin{cases} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \text{ ιφ } k = 0 \\ \\ \begin{pmatrix} H_{2^k} & H_{2^k} \\ H_{2^k} & -H_{2^k} \end{pmatrix}, \text{ιφ } k > 0 \end{cases} \tag{2.50}$$

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

$$l^{(0)}(v) = \texttt{row}_i \mathbb{H}, \text{ } \textbf{ανν} \text{ } label(v) = \Sigma_i \tag{2.51}$$

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

$$l^{(k+1)}(v) = l^{(k)}(v) + \sum_{u \in N(v)} l^{(k)}(u) \tag{2.52}$$

όπου με $N(v)$ συμβολίζουμε το σύνολο των κόμβων που είναι γείτονες του κόμβου v.
Ακολουθώντας τον παραπάνω κανόνα επανεπισημείωσης (βλ. 2.3), επαναληπτικά για έναν δεδομένο αριθμό επαναλήψεων, μπορούμε να χρησιμοποιήσουμε ένα κοινό πυρήνα βάσης για διακριτές επισημειώσεις όπως στην περίπτωση του σκελετού πυρήνα Weisfeiler-Lehman, αθροίζοντας του από όλες τις επαναλήψεις.
Η πολυπλοκότητα επανεπισημείωσης του πυρήνα Hadamard είναι επίσης γραμμική (οπως και του πυρήνα κατακερματισμού γειτονιών) αλλά έχει αποδειχθεί πειραματικά πως έχει την δυνατότητα να μπορεί να ανταποκριθεί αποτελεσματικά σε κλιμακούμενο μέγεθος δεδομένων.

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 [22]. Ο πυρήνας λαμβάνει υπόψιν τόσο

τα μήκη τον μονοπατιών όσο και τους κόμβους που συναντά, ενώ κάνει 'άλματα' μεταξύ συντομότερων μονοπατιών. Ο πυρήνας είναι ισοδύναμος με ένα βεβαρισμένο 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 (βλ. **;;**) π από το v στο u, θα συμβολίζουμε με $\pi(i) = v_i$ τον iοστό κόμβο που συναντάμε ενώ κάνουμε άλματα από τον έναν κόμβο στον άλλο μέσα στο μονοπάτι. Αν συμβολίσουμε με $l(\pi)$ το βεβαρισμένο μήκος του π και με $|\pi|$ το μήκος του όσον αφορά τον αριθμό των κόμβων που περιέχει π. Το ελάχιστο μονοπάτι π_{ij} από v_i στο v_j ορίζεται με βάση το βεβαρισμένο μήκος. Η διάμετρος $\delta(G)$ του G είναι ο μεγιστος δυνατός αριθμός κόμβων σε ένα ελάχιστο μονοπάτι του G, όσον αφορά το βεβαρισμένο μήκος.

Ο πυρήνας αλμάτων γράφων ορίζεται ως το άθροισμα των πυρήνων μονοπατιών k_p στις οικογένειες ελαχίστων μονοπατιών P, P' των G, G'

$$k(G, G') = \sum_{\pi \in P} \sum_{\pi' \in P'} k_p(\pi, \pi') \tag{2.53}$$

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

$$k_p(\pi, \pi') = \begin{cases} \sum_{j=1}^{|\pi|} k_n(\pi(j), \pi'(j)), & \text{ιφ } |\pi| = |\pi'|, \\ 0, & \text{οτηερωισε.} \end{cases} \tag{2.54}$$

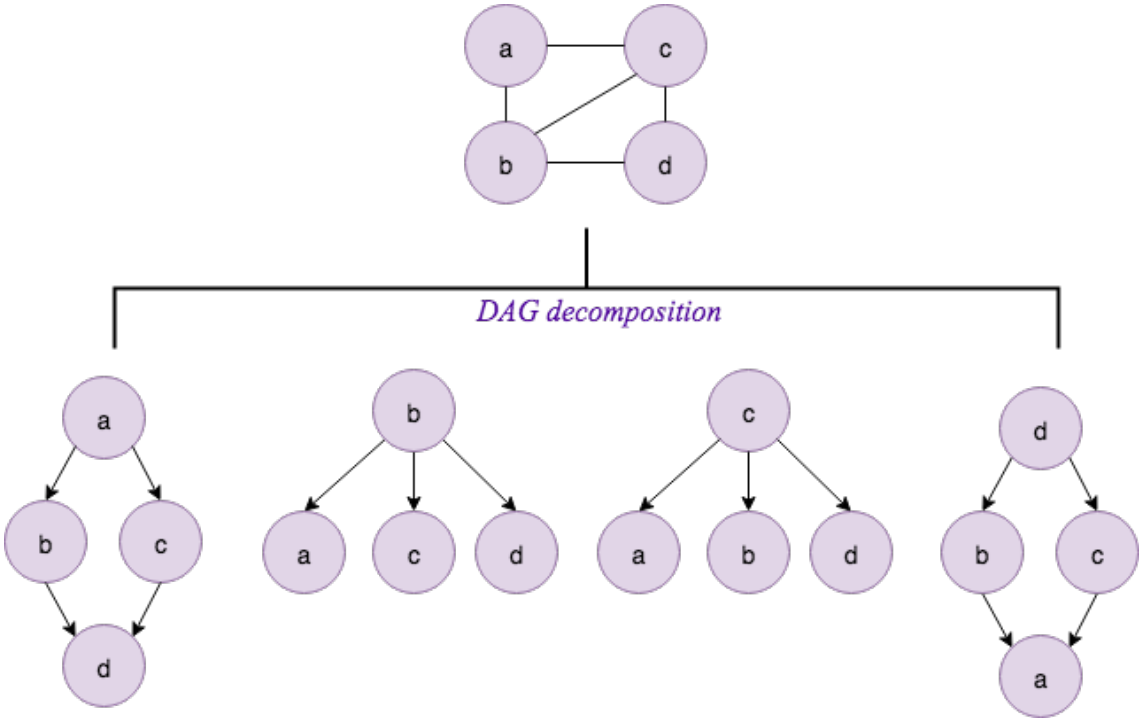
Ο πυρήνας $k(G, G')$ μπορεί να γραφέι ως ένα άθροισμα από πυρήνες κόμβων

$$k(G, G') = \sum_{v \in V} \sum_{v' \in V'} w(v, v') k_n(v, v') \tag{2.55}$$

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

$$w(v, v') = \sum_{j=1}^{\delta} \sum_{i=1}^{\delta} |\{(\pi, \pi') : \pi(i) = v, \pi'(i) = v', |\pi| = |\pi'| = j\}| = \sum_{j=1}^{\delta} \sum_{i=1}^{\delta} [\mathbf{M_v}]_{ij} [\mathbf{M_{v'}}]_{ij} \tag{2.56}$$

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 iοστή 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 πολλυπλοκότητα 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).



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

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 λόγο της μεγάλης εκφραστικότητας τους και τις χαμηλής υπολογιστικής τους πολλυπλοκότητας [35, 56, 79].

Η ιδέα πίσω από τον πυρήνα ODD-STh που προτείνεται στο [53], εκκινά από την αποσύνθεση δύο γράφων σε ένα σύνολο ταξινομήμενων ακυκλικών γραφημάτων και την πρόσθεση όλων των τιμών ενός πυρήνα K_{DAG} για ακυκλικά γραφήματα μεταξύ όλων των ζευγαριών τους ως εξής:

$$K_{K_{DAG}}(G_1, G_2) = \sum_{\substack{D_1 \in DD(G_1) \\ D_2 \in DD(G_2)}} K_{DAG}(D1, D2) \tag{2.57}$$

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 (βλ. 2.4)

Εν συνεχεία, ο πυρήνας K_{DAG} ανάγεται στο αθροισμα ενός πυρήνα ριζωμένων δέντρων με

διάταξη $C()$ ως εξής

$$K_{DAG} = \sum_{\substack{v_1 \in V(D_1) \\ v_2 \in V(D_2)}} C(root(T(v_1)), root(T(v_2))) \tag{2.58}$$

Ως πυρήνα $C()$ θα θεωρούμε από εδώ και στο εξής τον πυρήνα υποδέντρων (**S**ub-**T**ree kernel), όπως ορίζεται στο [80]. Το $T()$ αντιστοιχεί στο σύνολο *επισκέψεων δέντρων* (βλ. 2.5) σε ένα κατευθυνόμενο ακυκλικό γράφημα οι οποίες ακολουθούν τον ακόλουθο κανόνα *διάταξης* μεταξύ των κόμβων τους.

Ορισμός 2.24 (Αυστηρή σχέση μερικής διάταξης $\dot{>}$ μεταξύ των κόμβων ενός ΚΑΓ)**.** *Ας υποθέσουμε ότι μεταξύ των κόμβων ενός γράφου έχουμε ολική διάταξη (π.χ. λεξικογραφική). Τότε η σχέση $\dot{>}$ που 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

   $$ch_1[v_j], ch_2[v_j], \ldots, ch_m[v_j] ch_1[v_j], ch_2[v_j], \ldots, ch_m[v_j]$$

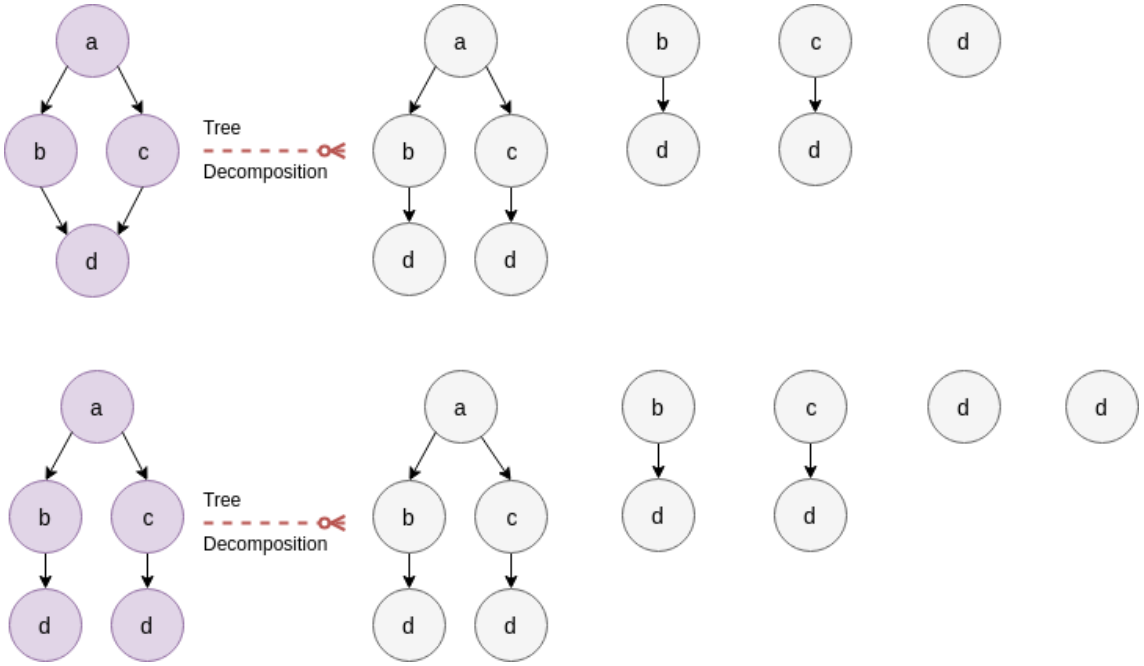
   όπου $m = \delta^+(v_i) = \delta^+(v_j)$ και κάθε $ch_k[v_i]$ είναι ένα (όχι απαραίτητα μοναδικό) ελάχιστο στοιχείο στο σύνολο που ορίζεται ως $\{ch_l[v_i] | l \in \{k, \ldots, m\}\}$. Τότε η σχέση αληθεύει αν:

   $$(\exists l. ch_l[v_i] \dot{<} ch_l[v_j]) \wedge (\neg \exists k < l. (ch_k[v_i] \dot{<} ch_k[v_j] \vee ch_k[v_j] \dot{<} ch_k[v_i]))$$

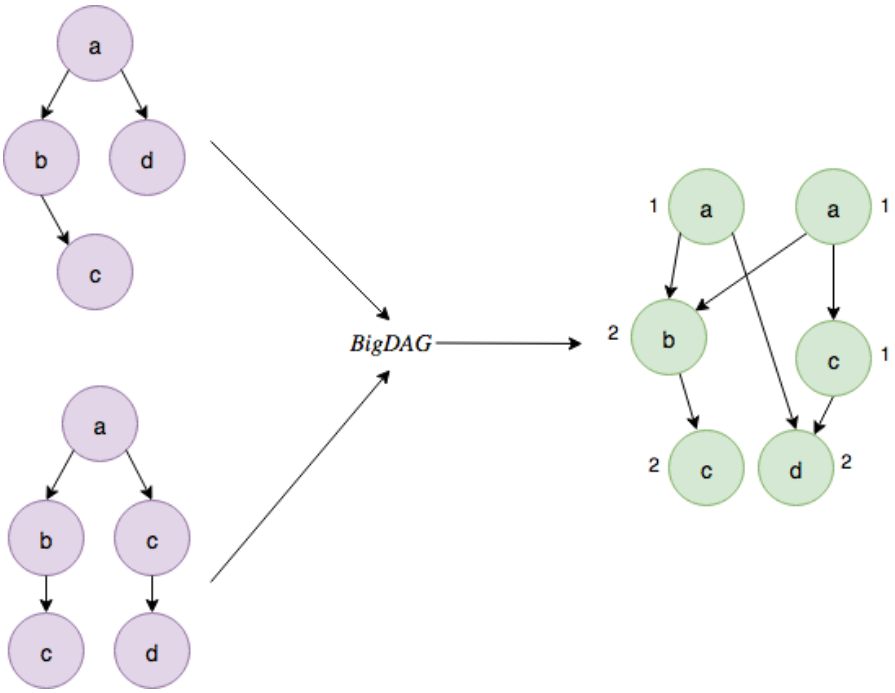
ορίζει μία μερική διάταξη (βλ. [53, Θεώρημα 5.1]).

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), βάση 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 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 (βλ. 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:

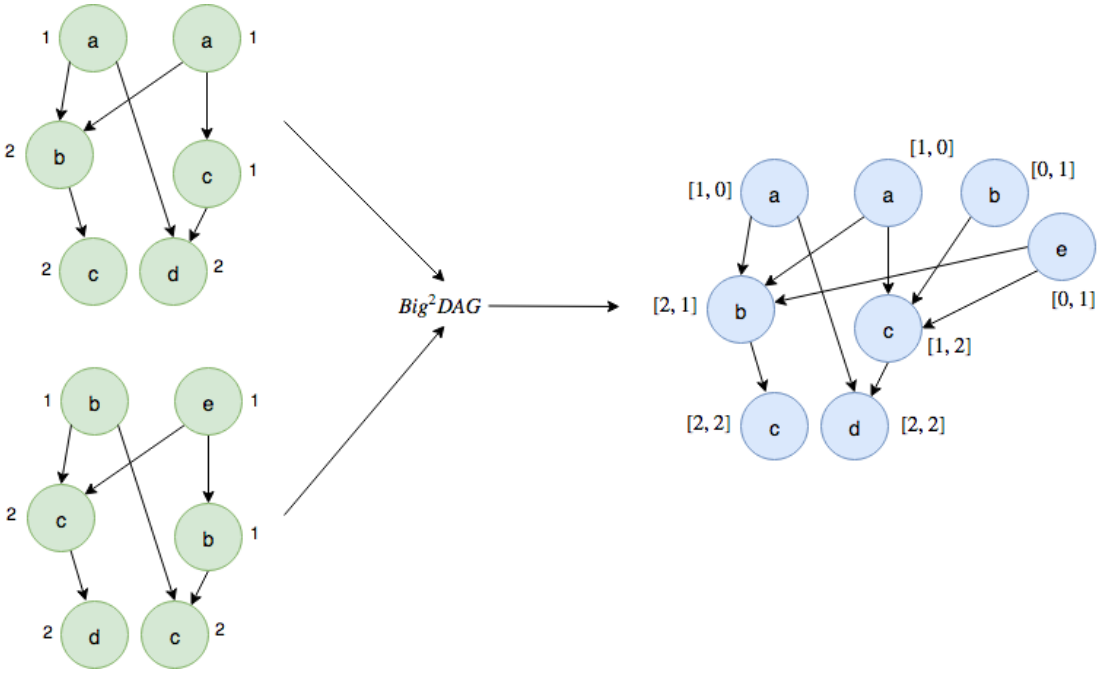
$$K_{K_{DAG}}(G_1, G_2) = K_{BigDAG}(G_1, G_2) = \sum_{\substack{u_1 \in V(BigDAG(G_1)) \\ u_2 \in V(BigDAG(G_2)}} f_{u_1} f_{u_2} C(u_1, u_2) \tag{2.59}$$



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.



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



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.

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 ακολουθήσουμε να προσθέτουμε τις συχνότητες εμφάνισης των κόμβων όταν αυτή τατίζονται τις αντικαταστήσουμε με ένα διάνυσμα συχνοτήτων το οποίο έχει στην i-οστή θέση την τιμή 0 αν ο κόμβος δεν ανήκει στο i-οστό και όσο η συχνότητα του σε αυτό το γράφημα αν ανήκε (βλ. 2.7).
Στο τελικό Big^2DAG γράφημα, ο υπολογισμός της μήτρας πυρήνα ανάγεται στον παρακάτω υπολογισμό

$$K_{Big^2DAG}(G_i, G_j) = \sum_{u_1,u_2 \in V(Big^2DAG)} F_{u_1}[i] \star F_{u_2}[j] C(u_1, u_2) \tag{2.60}$$

που είναι ισοδύναμο με

$$K_{Big^2DAG}(G_i, G_j) = \sum_{u \in V(Big^2DAG)} F_u[i] \star F_u[j] C(u, u) \tag{2.61}$$

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

$$C(u_1, u_2) \neq 0 \leftrightarrow T(u_1) = T(u_2) \tag{2.62}$$

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

$$ID_u = \begin{cases} L_u, \text{ αν } \delta^+(u) = 0 \\ L_u(ID_{ch_1[u]}, \cdots, ID_{ch_{\delta^+(u)}[u]}), \text{ αλλιώς} \end{cases} \tag{2.63}$$

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 πολλυπλοκότητα του αλγορίθμου θεωρώντας το h είναι σταθερό και παίρνει μικρές τιμές είναι της τάξης του $\mathcal{O}(n \log n)$ [53, Υποενότητα 5.5]

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 τον βασικό σκελετό οργάνωσης της βιβλιοθήκης, το μοντέλο προγραμματισμού την οργάνωση και το Έπειτα θα ασχοληθούμε με συγκεκριμένες υπομονάδες του ίδιου του πακέτου προκειμένου να περιγράψουμε των τρόπο με τον οποίο λειτουργούν καθώς και τις δυνατότητες τους. 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 στις εφαρμογές [15]. Διαθέτει εκτέλεση με διερμηνέα (interpreter), που διευκολύνει τον προγραμματιστή να αναπτύσσει εφαρμογές πολύ γρήγορα, καθώς λόγω του οκνηρού συστήματος τύπων που διαθέτει, τα σημασιολογικά λάθη προκύπτουν μόνο την στιγμή που θα αποτελέσουν πρόβλημα. Κάτι τέτοιο κάνει την διαδικασία της διόρθωσης του προγράμματος (debugging) συγχρονική της ίδια της εκτέλεσης. Ταυτόχρονα υποστηρίζει το μοντέλο του αντικειμενοστραφούς προγραμματισμού που επιτρέπει την σύγχρονη σχεδίαση μία 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 εσωτερικές μεταβλητές (attributes) και μεθόδους. Π.χ. μία κλάση μπορεί να είναι ο γράφος που 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 αντικειμένων αυτής της κλάσης γράφων όπως το γινόμενο. Το χαρακτηριστικό αυτό μίας γλώσσας δίνει την ευελιξία στην/ον προγραμματίστρια/τιστή τόσο να επεκτείνει τρομερά αποτελεσματικά τις υπάρχουσες δομές

δεδομένων που υπάρχουν από την δημιουργία της, όσο και να εκμεταλλεύεται την ανεξάρτητη χρονική οντότητα κάθε αντικειμένου, ενσωματώνοντας το αυτόματα με τις παραμετροποιημένες εκδόσεις ενός σύνολο συναρτήσεων που το αφορούν. Ακόμα οι συντακτικοί κανόνες της γλώσσας είναι διαμορφωμένοι με τέτοιο τρόπο που η στοίχιση κώδικα χρησιμοποιείται προκειμένου να μειωθεί την χρήση συνταχτικών συμβόλων, κάνοντας πολύ ευκολότερη την ανάγνωση του κώδικα και κατέπέκταση την περαιτέρω ανάπτυξη ή ενσωμάτωση υπάρχοντος κώδικα σε εφαρμογές.
Ο σημαντικότερος βέβαια 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 τους κι 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) [61]. Προκειμένου διάφοροι χρήστες της να μπορούν να προτείνουν δυνατές επεκτάσεις της καθώς και για να καθιερώσει ένα πρότυπο κλάσεων μηχανικής μάθησης για τον αντικειμενοστραφή προγραμματισμό, ένα πλαίσιο λογισμικού sklearn-template δημιουργήθηκε στο GitHub. Λόγω των παραπάνω και με βάση το γεγονός ότι δεν υπήρχε υποστήριξη για το είδος των τεχνικών πυρήνα, συγκεκριμένα των πυρήνων γράφων το παραπάνω σχεδιαστικό πρότυπο επιλέχθηκε για την ανάπτυξη του GraKeL. Σαν σχέση υιοθέτησης δύο θεμελιωδών κλάσεων του σκικιτ-λεαρν 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 να αποτελεί μία κλάση που υιοθετεί 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 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*, βάση 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 αρχικοποιημένο αντικείμενο μιας κλάσης προκειμένου να ξαναχρησιμοποιηθεί αρχικοποιημένο με διαφορετικές παραμετροποιήσεις, χωρίς την χρήση της ίδιας της κλάσης. Κάθε κλάση πρέπει περαιτέρω να διατυπώνει όλες τις παραμέτρους ορισμού της, ρητά. Τα παραπάνω είναι σημαντικά προκειμένου ένας Transformer που σχεδιάζει ο προγραμματιστής να μπορεί να εισαχθεί στο λεγόμενο scikit-learn `Pipeline`. Ως τέτοιο μπορεί να εισαχθεί, να αντικατασταθεί 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. Για να τηρούνται συνεπώς τα παραπάνω σημαντική ήταν η σχεδίαση της κλάσης πυρήνα.

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 απεικόνηση:

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

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. 2.13). Οποιαδήποτε μήτρα που προκύπτει από ένα μέτρο ομοιότητας είναι μήτρα πυρήνα αν για κάθε συλλογή εισόδων είναι θετικά ημιορισμένη (βλ. 2.14), 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.
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 αρχικών, είτε εξάγοντας και συγκρίνοντας όμοια χαρακτηριστικά με αυτά του `fit` είτε υπολογίζοντας τιμές βάση των χαρακτηριστικών του `fit` είτε τέλος επεκτείνοντας τα και υπολογίζοντας μία γενικής μετρικής (χωρίς βέβαια την αποθήκευση της επέκτασης).

Τέλος για `fit_transform` έχουμε ένα συνδυασμό των παραπάνω πράγμα που συνήθως αποτελεί την μόνη λειτουργία εκτελούν όλοι οι υπάρχοντες υλοποιημένοι 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 πυρήνας κλάση 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 εκτέλεση του κερνελ 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 η ο πίνακας πυρήνας θα είναι κανονικοποιημένος τόσο στα αποτελέσματα του `fit_transform` όσο και του `transform` Η κανονικοποίηση είναι μία πολύ απλή πράξη διαίρεσης δεδομένων των τιμών της διαγωνίου της μήτρας πυρήνα ως εξής:

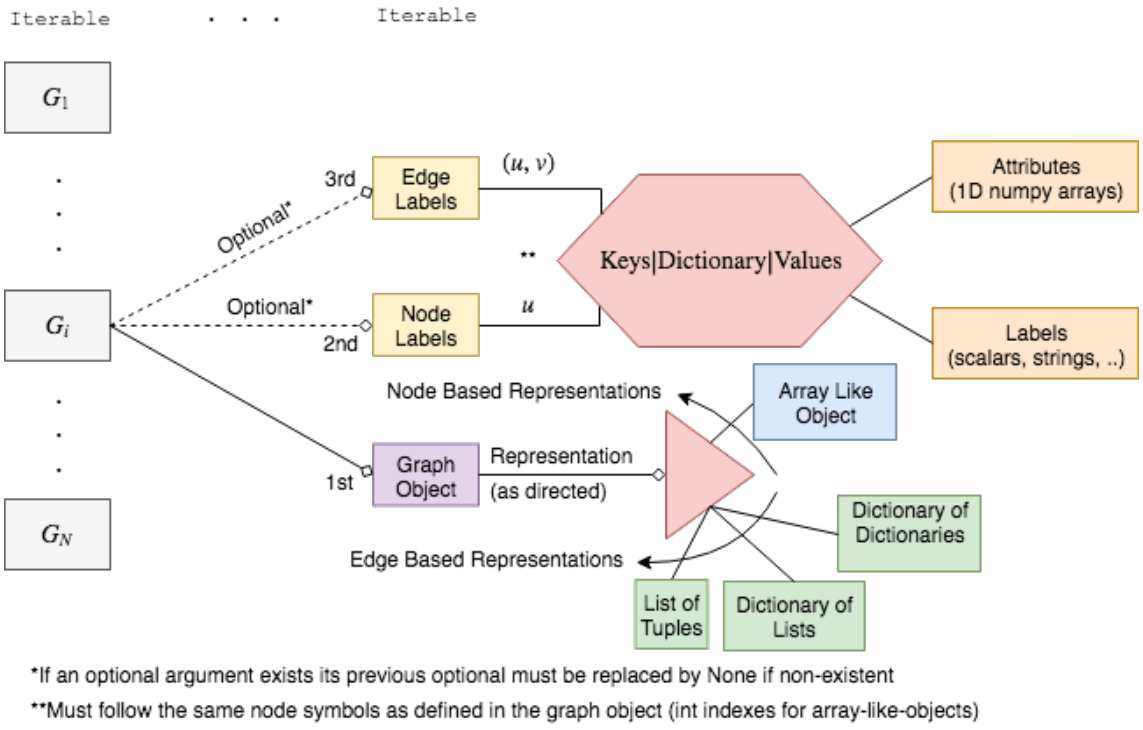
$$[\hat{\mathcal{K}}]_{ij} = \frac{[\mathcal{K}]_{ij}}{\sqrt{[\mathcal{K}]_{ii} * [\mathcal{K}]_{jj}}} \tag{3.2}$$

- `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 framework δεν ορίστηκε μία ξεχωριστή κλάση. Παρόλαυτα 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. 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 δεν δέχεται ορίσματα και πρέπει να έχει πάντοτε την ακόλουθη συμπεριφορά. Αν ένας πυρήνας έχει γίνει `fit` αλλά όχι `transform`, τότε επιστρέφει την διαγώνιο $[\mathcal{K}^{n\times n}]_{ii}$ (πρβλ. 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` (δλδ. 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 δημιουργία ενός προτύπου αναπαράστασης της εισόδου των μεθόδους `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 (πρβλ. 3.1). Κάθε γράφος μπορεί να αναπαρασταθεί από ένα `Iterable` τουλάχιστον ενός και το πολύ τριών στοιχείων.
Πρώτο στοιχείο κάθε γράφου πρέπει να είναι ένα αντικείμενο που αναπαριστά το 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 αναπαραστάσεις 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 βαθμωτές, σύμβολα είτε διανύσματα πραγματικών τιμών (χαρακτηριστικά, πρβλ 2.3). Τρίτο και τελευταία στοιχείο είναι ένα λεξικό μεταξύ ζευγαριών κόμβων για όλες τις υπάρχουσες ακμές και είτε βαθμωτών, συμβολικών επισημειώσεων είτε διανυσμάτων πραγματικών τιμών. Τα δύο στοιχεία μπορούν να παραληφθούν ή να αντικατασταθούν από κενά

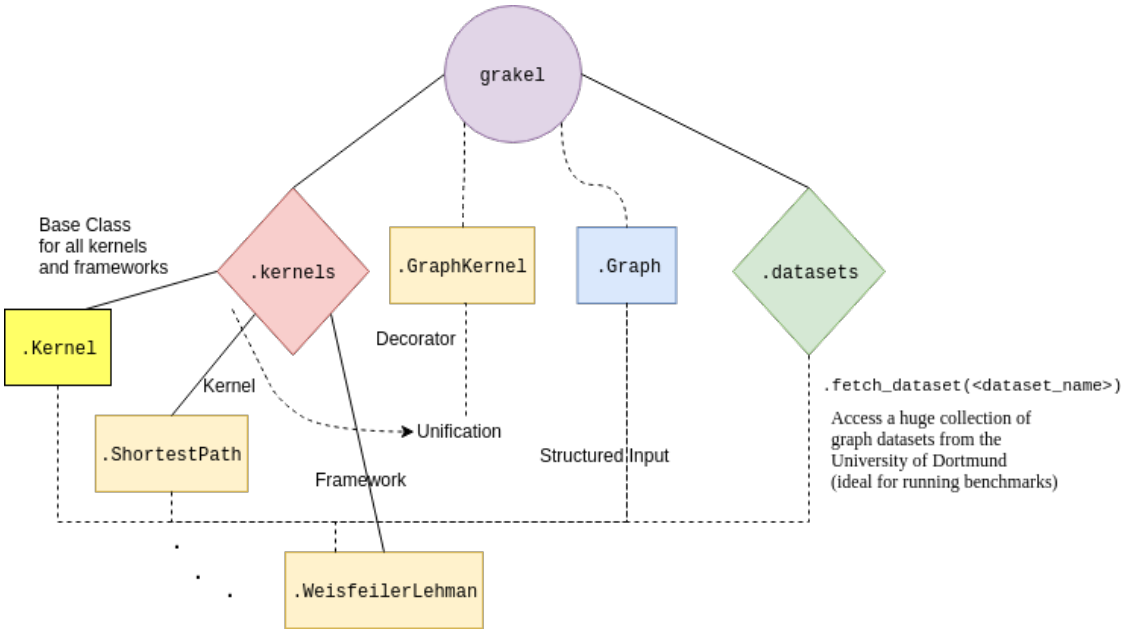


Σχήμα 3.1: Σχηματική απεικόνιση του τρόπου αναπαράστασης της εισόδου για της μεθόδους `fit`, `fit_transform` και `transform` κάθε αντικειμένου τύπου `Kernel`

ορίσματα (τύπου `None`) στην περίπτωση που δεν υπάρχουν.

Η αναπαράσταση των γράφων ήταν πολύ σημαντική για τους ίδιους τους πυρήνες. Πολλοί πυρήνες όπως ο *πυρήνας τυχαίων περιπάτων* χρησιμοποιούν άμεσα τον πίνακα γειτνίασης, πυρήνες όπως ο *πυρήνας κοντινότερων μονοπατιών* χρειάζονται μόνο τον πίνακα κοντινότερων μονοπατιών (που υπολογίζεται ταχύτερα αν έχουμε αναπαράσταση ακμών, λόγω του αλγορίθμου Διθκστρα) ενώ 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 χρειάζονταν ένας τρόπος προκειμένου οι γράφοι να μην αναπαριστούν πολύ μνήμη, ενώ ταυτόχρονα να μπορούμε να ελέγχουμε το είδος της εσωτερικής εσωτερική αναπαράστασης ανά περίπτωση, έχοντας ταυτόχρονα την δυναότητα να καθορίσουμε την μορφή του αποτελέσματος συναρτήσεων πάνω στους γράφους, χρήσιμων για τους υπολογισμούς των πυρήνων, χωρίς να χρειάζεται συνεχώς να ασχολούμαστε με την μορφή της πρώτης. 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 μετέπειτα κατά τον προγραμματισμού του με την μορφή των δεδομένων εισόδου ή της ίδια της εσωτερικής αναπαράστασης. Παρόλο που μία τέτοια σχεδιαστική προσέγγιση φαίνεται πολύ απλοϊκή προτιμήθηκε σε σχέση με την χρήση μίας υπάρχουσας βιβλιοθήκης όπως η `networkx`, μίας και χρειάζεται 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 ενόνόματι `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 τεράστεια συλλογή όπως αυτή συντηρείται και ελέγχεται από την ερευνητική ομάδα για τους πυρήνες γράφων στο πανεπιστήμιο του Dortmund [44]. 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 αναφέρθηκε, συνοψίζεται παρακάτω:



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 (πρβλ. 3.3). Από την μελέτη της σχετικής βιβλιογραφίας ο υπολογισμός ενός γράφου πυρήνα μπόρεσε να αποδομηθεί στα εξής δύο αφηρημένα βήματα: ανάγνωση της εισόδου και εξαγωγή χαρακτηριστικών (1) και υπολογισμός πίνακα πυρήνα (2).

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 ιδιότη-

τα της κλάσης. Παράλληλα προκειμένου η κλάση `Kernel` να υιοθετεί 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 δευτερεύοντων χαρακτηριστικών ανατέθηκε στην συνάρτηση `initialize_` η οποία καλείται στην πρώτη γραμμή της συνάρτησης μεθόδου `fit`.

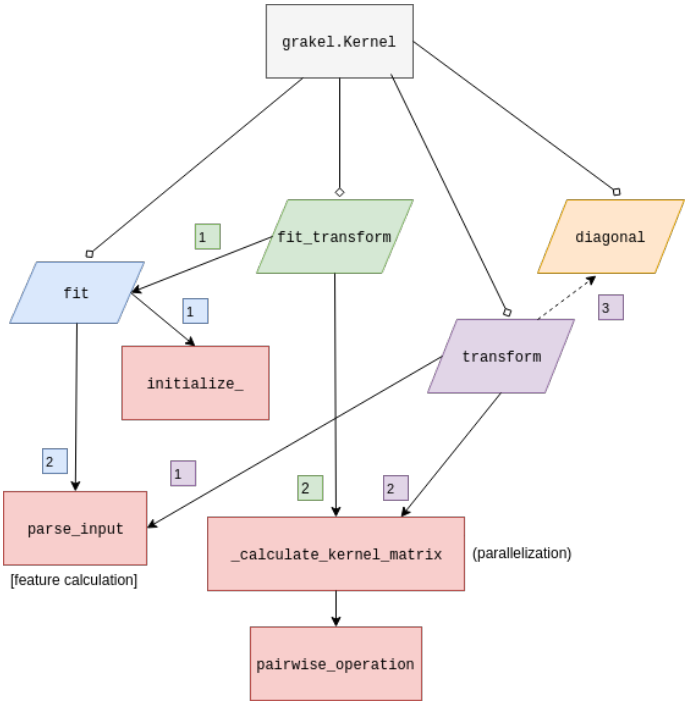
3.2.2 Η μέθοδος `fit_transform`

Για τον υπολογισμό της μήτρας πυρήνα είναι υπεύθυνες δύο μέθοδοι:
η `_calculate_kernel_matrix` (1) και η `pairwise_operation` (2). 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 αναμένονται να είναι στην μορφή ενός `Iterable` το οποίο σε κάθε στοιχείο 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 διαγώνιου (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 επανεγραφεί 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 πλήθος παράλληλων εργασιών `n_jobs` χωρίζουμε 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 αντιστοιχούν 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`, 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



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

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 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 φυσικά επιλύεται με την χρήση μία ιδιωτικής self._method_calling ιδιότητας της κλάσης, που είναι $1, 2, 3$ 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, υπολογίζουμε pairwise_operation. Σε αυτήν την περίπτωση, δεδομένου ενός πλήθος παράλληλων εργασιών n_jobs η παραλληλοποίηση επιτυγχάνεται αν χωρίζουμε ομοιόμορφα την λίστα δεικτών $K = [0, \dots, NM - 1]$ (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 κανονικοποημένος 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 μία 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 ανιχνεύσει τις αλλαγές τις στο παρελθόν και να γνωρίζει τον παρούσα μορφή της για να δύναται 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 αυτές οι 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 όπως το μέγιστο δυνατό μήκος μίας γραμμής κώδικα και την στοίχηση των ορισμάτων κατά την κλήση ή τον ορισμό μία 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) [32]. 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 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 της, γνωστό ως pip. Η python ως γλώσσα με διερμηνέα δεν διαθέτει την έννοια των εκτελέσιμων 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 η ανάγκη τα λεγόμενα wheels εισήχθησαν από την κοινότητα που αναπτύσσει την γλώσσα python. Κάθε σύγχρονη βιβλιοθήκη επιστημονικού υπολογισμού (scientific-computing) καλείται να μπορεί συνταιριάξει, το ευχάριστο και μή-περιοριστικό προγραμματιστικός της περιβάλλον, ενώ ταυτόχρονα να μπορεί να συμπεριλάβει τα άγρια και αδέξια πλάσματα των γλωσσών C, C++, Fortran λόγω της αποδοτικότητας και της αμεσότητας τους. Τόσο στο επίπεδο του setup.py 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 ενό 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 αυτά να παρουσιάζεται συνοπτικά, ενώ ενσωματώνουν άλλα όπως το coverage που είναι υπεύθυνο για να παρουσίαζει 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 κώδικας 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 η κατάλληλη παραμετροποίηση του συστήματος, η εγκατάσταση και η δοκιμή της βιβλιοθήκης μπορούν να ελέγχονται αυτόματα αν έρχονται εις πέρας, κάθε φορά που μία νέα έκδοση προτίθεται στο ηλεκτρονικό αποθετήριο. Συγκεκριμένα ο έλεγχος της βιβλιοθήκης GraKeL γίνεται σε λειτουργικά συστήμα Linux και OSX μέσω της πλατφόρμας συνεχούς ενσωμάτωσης Travis και σε λειτουργικό σύστημα Windows μέσω της πλατφόρμας Appveyor. Μιάς 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. 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 coveralls μέσω του travis, φαίνονται με την μορφή badges στην κύρια σελίδα στο αποθετήριο του ΓραΚεΛ.

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 την 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

4.1 Πειραματική Διάταξη

Για την αξιολόγηση του πακέτου grakel τρέξαμε στο ίδιο μηχάνημα (βάλε τα στοιχεία του) τον υπολογισμό του πλήρους kernel πίνακα μέσω της μεθόδου fit_transform σε ένα εύρος τιμών (βλ πίνακα ταδε) 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 χρησιμοποιήσε η βιβλιοθήκη BLASS ορίστηκε ίση με 1. Σε όλους τους αλγορίθμους για τους οποίους η επάυξηση 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 το πρόβλημα SVM παρέχοντας μία προϋπολογισμένη μήτρα Γκραμ. Λύνοντας αυτό το πρόβλημα υπολογίζουμε την μέγιστη μετρική ευστοχίας για ένα εύρος τιμών C (βλ. 2.10) που εξαρτόνται από την είσοδο και διαλέγουμε την μέγιστη. Υπολογίζοντας την μέση τιμή των μέγιστων μετρικών ευστοχίας για όλα τα fold υπολογίζουμε την μέση τιμή και την διακυμανσή τους για 10 επαναλήψεις. Τα folds ήταν κοινά για όλες τους πυρήνες που εκτελέστηκαν σε αυτό το dataset. Τέλος καταγράφουμε σε συγκριτικούς πίνακε για κάθε dataset την μνήμη, τον χρόνο και την παραμετροποίηση για την οποία πέτυχε ο κάθε πυρήνα πέτυχε την μέγιστη ευστοχία, καθώς και την τιμή της.

4.1.1 Μετρική Ευστοχίας

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

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

Πίνακας 4.1: Πίνακας σύγχησης για ένα πρόβλημα δυαδικής ταξινόμησης.

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 λανθεσμένη 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 2×2 πίνακα *σύγχησης* (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 βάση 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 δυαδική ταξινόμησης το 99% των παραδειγμάτω είναι θετικά, τότε ένας αλγόριθμος μπορεί να πετύχει 99% ευστοχία προβλέποντας μόνο την θετική κατηγορία! Ως επίλυση σε αυτό το πρόλημα έχουν προταθεί άλλες μετρικές αξιολόγησης.
Από την άλλη, όπως μπορεί να ειδωθεί στον πίνακα 4.1, η πλειοψηφία των συνόλο 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.

4.2 Datasets

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

4.2.1 Χωρίς επισημειώσεις

Για την αξιολόγηση πυρήνων που δέχονται ως είσοδο γράφους χωρίς επισημειώσεις, εκτελέσαμε τους πυρήνες ... στα παρακάτω σύνολα δεδομένων. Προκειμένου οι πυρήνες με διακριτές και συνεχείς, να είναι εκτελέσιμοι στα ίδια σύνολα δεδομένων, αποδώσαμε σε κάθε κόμβο ή ακμή μία σταθερή έπισημείωση (τον αριθμό 1) και ένα μοναδιαίο διάνυσμα ενός στοιχείου (`numpy.array([1.0])`) αντίστοιχα.

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 δίκτυα προσωπικότητας του κάθε ερευνητή [88].

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 ςορρεσπονδ 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 [88].

REDDIT-BINARY REDDIT-MULTI-5k REDDIT-MULTI-12k Οι γράφοι που περιέχονται σε αυτά τα τρία δαταςετ 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 ένα από αυτούς αντίδρασε στο σχόλιο του άλλου. Στόχος είναι η ταξινόμηση γράφων είτε σε κοινότητες είτε σε ϋπο-ρεδδιτ' (subreddits) [88].

4.2.2 Με διακριτές επισημειώσεις

Για την αξιολόγηση πυρήνων που δέχονται ως είσοδο γράφους με **διακριτές** επισημειώσεις, εκτελέσαμε τους πυρήνες ... στα παρακάτω σύνολα δεδομένων. Προκειμένου οι πυρήνες με συνεχείς, να είναι εκτελέσιμοι στα ίδια σύνολα δεδομένων, αποδώσαμε σε κάθε κόμβο ένα *Ονε-Ηοτ ἔςτορ* 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 μεταλλαξιγόνα δράση στο αρνητικό κατά Γκραμ βακτήριο Salmonella typhimurium [70].

ENZYMES Αποτελείται από 600 τριτογενείς δομές προτεϊνών που ανήκουν στην βάση ενζύμων BRENDA. Κάθε ένζυμο ειναί ταξινομημένο στην αφηρημένη κατάταξης Enzyme Commission και ο σκοπός είναι ο ορθός προσδιορισμός της κλάσης στην οποία ανήκει ένα ένζυμο [10].

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) [20, 70].

NCI1 Αυτό το σύνολο δεδομένων περιέχει μερικές χιλιάδες χημικά στοιχεία στα οποία καταγράφεται η δραστηριότητα τους απέναντι σε καρκινικά κύταρα του πνεύμονα και των ωοθηκών βάση της πορείας κυτταρικής διαίρεσης τους (cell lines) σε ελεγχόμενες συνθήκες εργαστηρίου [83].

PTC_MR Αυτό το σύνολο δεδομένων περιέχει 344 οργανικά μόρια που έχουν αναπαραστηθεί ως γράφοι. Στόχος είναι η πρόβλεψη καρκινογένεσης σε αρσινικούς αρουραίους [75].

AIDS Αποτελείται από 2000 χημικές ενώσεις αναπαραστημένως ως γράφους οι οποίες έχουν δοκιμαστεί για την αποτελεσματικότητα απέναντι στον ιό HIV. Σκοπός λοιπόν του προβλήματος ταξινόμησης είναι η πρόβλεψη του καταπόσο μία χημική ένωση μπορεί να είναι ή όχι αποτελεσματική απέναντι στον ιό [64].

PROTEINS Περιέχει πρωτεΐνες αναπαραστιμένες ως γράφους όπου οι κόμβοι είναι δευτερογενή δομικά στοιχεία και μία ακμή υπάρχει μεταξύ των κόμβων αν οι κόμβοι είναι γείτονες στην ακολουθία αμινοξέων ή στον 3σδιάστατο χώρο. Σκοπός είναι η ταξινόμηση μίας πρωτεΐνης ως ένζυμο (ή όχι) [12].

4.2.3 Με επισημειώσεις χαρακτηρηστικών

Για την αξιολόγηση πυρήνων που δέχονται ως είσοδο γράφους με **συνεχείς** επισημειώσεις, εκτελέσαμε τους πυρήνες ... στα παρακάτω σύνολα δεδομένων.

ENZYMES Αποτελείται από 600 τριτογενείς δομές προτεϊνών που ανήκουν στην βάση ενζύμων BRENDA. Κάθε ένζυμο ειναί ταξινομημένο στην αφηρημένη κατάταξης Enzyme Commission και ο σκοπός είναι ο ορθός προσδιορισμός της κλάσης στην οποία ανήκει ένα ένζυμο [10].

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 χαρακτηρηστικών 15-διάστατων διανυσμάτων δύο κατηγοριών οι παραπάνω δύο κατηγορίες χωρίζονται σε άλλες δύο όπου στο καθένα για την πρώτη αν ένα κόμβος προερχόταν από το πρώτο σύνολο γράφων επισημειώνεται τυχαία με ένα διάνυσμα του πρώτου συνόλου χαρακτηρηστικών ενώ σε αντίθετη περίπτωση με διάνυσμα του δεύτερου. Για την παραγωγή της δεύτερης κατηγορίας συμβαίνει το αντίθετο. Στόχος το προβλήματος ταξινόμησης είναι βάση των χαρακτηρηστικών, να ανιχνευθεί σε ποιά από τις τέσσερεις υποκατηγορίες ανήκει ένας γράφος [55].

BZR Αυτό το σύνολο δεδομένων αποτελεί από 684 χημικές ενώσεις κατηγοριοποιημένες ως μεταλαξιογόνες ή μη βάση ενός πειράματος γνωστό ως Salmonella/microsome assay. Αυτό το σύνολο δεδομένων είναι ισοβαρισμένο με 341 μεταλαξιογόνες χημικές ενώσεις και 343 μη-μεταλαξιογόνες [50, 57].

PROTEINS_full Αυτό το σύνολο δεδομένων αποτελείται 1113 από χημικές ενώσεις προερχόμενες από την βάση δεδομένων πρωτεΐνών ΠΔΒ. Διαχωρισμένες σε ενζυμα (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 χαρακτηρηστικών χρησιμοποιώντας μεταξύ άλλων την κρυσταλλογραφική τους πληροφορία [20, 12, 57]

SYNTHETICnew είναι ένα τεχνητό σύνολο δεδομένων 300 τυχαία δειγματοληπτημένων γράφων που αποτλούνται από 100 κόμβους και 196 ακμές, που στους κόμβους των οποίων ανατίθενται μονοδιάστατες συνεχείς επισημειώσεις από το $\mathcal{N}(0,1)$. Έπειτα δύο ισοβαριμένες κατηγορίες 150 επισημειώνων δημιουργούνται 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 $\mathcal{N}(0, 0.452)$ [21].

Όλα τα σύνολα δεδομένων που αναφέρθηκαν παραπάνω προέρχονται από το [44]. Στατιστικά στοιχεία και πληροφορίες σχετικά με την ύπαρξη και τον τύπο των επισημείωσεων τους παρουσιάζονται συνοπτικά στον πίνακα 4.2.

4.3 Αποτελεσμάτα & Αξιολόγηση

4.3.1 Χωρίς επισημειώσεις

4.3.2 Με διακριτές επισημειώσεις

4.3.3 Με επισημειώσεις χαρακτηριστικών

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

4.4 Συμπεράσματα και μελλοντικές επεκτάσεις

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] 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.

[10] 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.

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

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

[13] 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.

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

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

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

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

[23] 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.

[24] 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.

[25] 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.

[26] 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.

[27] 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.

[28] 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.

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

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

[31] 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.

[32] 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).

[33] 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.

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

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

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

[37] 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.

[38] 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.

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

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

[41] 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.

[42] 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.

[43] 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.

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

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

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

[47] 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.

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

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

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

[51] 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.

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

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

[54] 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.

[55] 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.

[56] 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.

[57] 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.

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

[59] 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.

[60] 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.

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

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

[63] 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.

[64] 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.

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

[66] 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.

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

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

[69] 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.

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

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

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

[73] 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.

[74] 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.

[75] 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.

[76] 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.

[77] 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.

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

[79] 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.

[80] 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.

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

[82] 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.

[83] 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.

[84] 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.

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

[86] 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.

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

[88] 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
γράφος	graph
διανύσματα χαρακτηρηστικών	feature vectors
διεπαφή	interface
μηχανική μάθηση	machine learning
εγχειρίδιο ανάγνωσης	documentation
εξόρυξης δεδομένων	data mining
επιστημονικού υπολογισμού	scientific computing
ισομορφισμός (γράφων)	graph isomorphism
κανονική διάταξη	canonical ordering
κόμβος	vertex, node
μήτρα	matrix
πηρύνες γράφων	graph kernels
συλλογές δεδομένων	dataset
συνδιαστικό	combinatorial
συρραφή (κώδικα)	(code) wrapping
συσκευασία (κώδικα)	(code) packaging
ταξινομητής μηχανών διανυσμάτων υποστήριξης	support vector machine classifier
υψηλό επίπεδο (προγραμματισμού)	high level (programming)