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

Label Ranking: Προσέγγιση μέσω Συγκρίσεων ανά Ζεύγη και Τεχνικών Συνάθροισης

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΓΕΩΡΓΙΟΣ ΓΙΑΝΝΑΚΟΥΛΙΑΣ

Επιβλέπων : Δημήτριος Φωτάκης
Αναπληρωτής Καθηγητής Ε.Μ.Π.

Αθήνα, Ιούλιος 2021

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

Label Ranking: Προσέγγιση μέσω Συγκρίσεων ανά Ζεύγη και Τεχνικών Συνάθροισης

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΓΕΩΡΓΙΟΣ ΓΙΑΝΝΑΚΟΥΛΙΑΣ

Επιβλέπων : Δημήτριος Φωτάκης
Αναπληρωτής Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 05η Ιουλίου 2021.

.....	.....	.....
Δημήτριος Φωτάκης	Στρατής Ιωαννίδης	Αριστείδης Παγουρτζής
Αναπληρωτής Καθηγητής, Ε.Μ.Π.	Associate Professor, Northeastern Un.	Αναπληρωτής Καθηγητής, Ε.Μ.Π.

Αθήνα, Ιούλιος 2021

Γεώργιος Γιαννακούλιας

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Περίληψη

Το label ranking (κατάταξη ετικετών) είναι ένα πρόβλημα που έχει προσελκύσει πρόσφατα μεγάλο ερευνητικό ενδιαφέρον λόγω της γενικότητας του και την πληθώρα εφαρμογών που το συναντάται. Διάφορες προσεγγίσεις για την επίλυση του προβλήματος label ranking έχουν προταθεί. Σε αυτήν την εργασία, ωστόσο, θα επικεντρωθήκαμε στην αρθρωτή προσέγγιση RPC που λύνει την label ranking χρησιμοποιώντας συγκρίσεις ανά ζεύγη και τεχνικές συνάθροισης. Παρουσιάστηκε για πρώτη φορά από τους Hullermeier και Fürnkranz το 2008, αλλά παραμένει μια αποτελεσματική προσέγγιση, επομένως απαιτείται περαιτέρω έρευνα. Σε αυτήν την εργασία, πραγματοποιήσαμε συγκριτική πειραματική αξιολόγηση του μοντέλου RPC, που αποτελείται από δύο στάδια. Πειραματιστήκαμε με διάφορους αλγόριθμους μάθησης στο πρώτο στάδιο και καταλήξαμε στο συμπέρασμα ότι επηρεάζουν σημαντικά τις προβλέψεις του μοντέλου RPC. Επιθεωρήσαμε επίσης δημοφιλείς και καινοτόμες τεχνικές συνάθροισης στο δεύτερο στάδιο και καταλήξαμε στο συμπέρασμα ότι τα αποτελέσματα δεν επηρεάζονται σοβαρά, δεδομένου ότι οι τεχνικές συνάθροισης ακολουθούν μια απλή και μαθηματικά ορθή λογική διαδικασία. Το πιο σημαντικό, ερμηνεύουμε τις παραμέτρους και τα χαρακτηριστικά του μοντέλου για να βγάλουμε συμπεράσματα σχετικά με το γιατί ορισμένοι αλγόριθμοι μάθησης και τεχνικές συνάθροισης λειτουργούν καλύτερα από τους υπόλοιπους. Τέλος, κάνουμε εμπεριστατωμένη ανάλυση των συνόλων δεδομένων, που χρησιμοποιούνται για τη συγκριτική αξιολόγηση. Με τη χρήση διαφορετικών μετρήσεων και οπτικοποιήσεων, εξηγούμε τους λόγους για τους οποίους η κατάταξη ετικετών είναι ένα περίπλοκο πρόβλημα. Καταλήγουμε στο συμπέρασμα ότι η συνάρτηση αντιστοίχησης μεταξύ του χώρου των γνωρισμάτων και του χώρου των μεταθέσεων επηρεάζει σοβαρά την απόδοση όλων των μοντέλων και εμποδίζει την ανάπτυξη ενός μοντέλου που λύνει αποτελεσματικά όλα τα προβλήματα μάθησης.

Abstract

Label ranking is a problem that has recently attracted research attention due to its generality and number of applications. Several approaches for solving the label ranking problem have been proposed. In this work however, we focused on the modular RPC approach that solves label ranking using pairwise comparisons and aggregation techniques. It was first presented by Hullermeier and Fürnkranz in 2008 but still remains a state-of-the-art approach, thus further investigation is required. In this work, we conducted experimental evaluation of the two-stage RPC model. We experimented with several learning algorithms at the first stage and concluded that they significantly impact the predictions of the RPC model. We also inspected popular and innovative aggregation techniques at the second stage and concluded that the results are not seriously affected, given that the aggregation techniques follow a simple and solid logical procedure. Most importantly, we interpret the scores of the model to make conclusions on why some learning algorithms and aggregation techniques work better than the rest. Lastly, we make thorough analysis on the datasets, used for benchmarking. With the use of different metrics and visualisations, we explain the reasons why label ranking is a highly complex problem. We conclude that the mapping function between the instances' space and the rankings' space severely affects the performance of all models and prohibits the development of a model that efficiently solves all learning problems.

Keywords Preference learning, Label Ranking, Pairwise classification, Rank aggregation

Acknowledgments

As my thesis is finished, a big cycle of my life and career comes to end. I could not feel more grateful for the experiences I have lived and the lessons I have learned during this period! Looking back to all the opportunities I was given and the achievements and failures I have been through, I feel blessed and fulfilled in both an academic and a personal level.

Firstly, I would like to thank to my supervisor and teacher Prof. Fotakis, who was really supportive throughout all this period. He gave significant guidance whenever it was needed and showed interest in me as a student and a person as well.

Next, I would like to express my gratitude to my supervisor, Eleni Psaroudaki, who was present in every step of the process. I feel ultimate respect for her and I am convinced that she will make great achievements in the near future. This thesis would not have been completed without her help!

maiLastly, I would like to thank my family and close friends, the unseen heroes that supported me along the way.

With a bit of nostalgia about the past and full of excitement about the future, I hope to live and create in the present and reach the best version of myself!

Giannakoulias Georgios

Contents

List of Tables

List of Figures

Κεφάλαιο 1

Εκτεταμένη Ελληνική Περίληψη

1.1 Εισαγωγή στο πρόβλημα

Τα τελευταία χρόνια παρουσιάζεται μεγάλη άνθηση στην μελέτη της τεχνητής νοημοσύνης, Κυρίως λόγω της πληθώρας πρακτικών προβλημάτων τα οποία αντιμετωπίζονται με χρήση τεχνικών που προτείνονται στο χώρο της τεχνητής νοημοσύνης και ιδιαίτερα της μηχανικής μάθησης. Πιο συγκεκριμένα, υπάρχει μεγάλο ερευνητικό ενδιαφέρον Σε προβλήματα επιβλεπόμενη μηχανικής μάθησης.

Στο κλασικοτερο πρόβλημα της επιβλεπόμενη μηχανικής μάθησης έχουμε ένα σύνολο ετικετών ή κατηγοριών έχουμε κάποιες παραδειγματικές εισόδους και κάθε εισοδος αντιστοιχίζεται με μια επιθυμητή έξοδο από ένα σύνολο εξόδων. Στόχος είναι να δημιουργηθεί ένα μοντέλο που μπορεί να προβλέπει για μια νέα άγνωστη είσοδο την σωστή έξοδο. Μπορούμε να θεωρήσουμε ότι οι έξοδοι αποτελούν ένα σύνολο από κατηγορίες, που περιγράφουν κατά μια έννοια την είσοδο που τους αντιστοιχίζουμε.

Σε πρακτικά προβλήματα όμως, η αντιστοίχιση μίας εισόδου με μία μόνο έξοδο είναι περιοριστική. Υπάρχουν προβλήματα στα οποία θα θέλαμε μια είσοδος να αντιστοιχίζεται σε περισσότερες εξόδους. Στο Label Ranking αναφερόμαστε στις εξόδους ως ετικέτες και μας ενδιαφέρει περισσότερο η σχετική θέση προτίμησης μεταξύ των ετικετών δεδομένης μιας εισόδου. Στόχος είναι η δημιουργία ενός μοντέλου που αντιστοιχίζει κάθε είσοδο σε μία σειρά προτίμησης όλων των ετικετών.

Για την επίλυση του προβλήματος έχουν προταθεί διαφορετικών ειδών προσεγγίσεις. Ανάμεσα στις πιο γνωστές συγκαταλέγονται οι πιθανοτικές προσεγγίσεις, που χρησιμοποιούν στατιστικά μοντέλα και τεχνικές βελτιστοποίησης για να υπολογίσουν την καλύτερη πιθανή μετάθεση ως έξοδο, και οι προσεγγίσεις που χρησιμοποιούν ειδικά προσαρμοσμένα δέντρα για την επίλυση του προβλήματος Label Ranking. Εμείς θα επικεντρωθούμε στην προσέγγιση RPC, η οποία ανήκει στην ευρύτερη κατηγορία προσεγγίσεων που κάνουν χρήση αναγωγής. Οι προσεγγίσεις αυτές ανάγουν το αρχικό πολύπλοκο πρόβλημα σε απλούστερα υποπροβλήματα, επιλύουν τα υποπροβλήματα αυτά και στη συνέχεια συνδυάζουν τις απαντήσεις τους για να παράγουν μία λύση για το αρχικό πρόβλημα.

Λόγω της γενικότητας του, το πρόβλημα Label Ranking βρίσκει εφαρμογή σε πολλούς επιστημονικούς τομείς. Στον τομέα της βιοϊατρικής, συνήθης στόχος είναι να κατατάσσουμε γονίδια ανάλογα με το βαθμό εκφράσης τους βασιζόμενοι σε κάποια χαρακτηριστικά τους. Επίσης βασική χρήση είναι τα συστήματα προτάσεων. Για παράδειγμα, μία ιστοσελίδα ειδησεογραφίας επιθυμεί, βασιζόμενη στα χαρακτηριστικά ενός ανθρώπου, να κατάταξη τα άρθρα της με τέτοιο τρόπο ώστε να εμφανίσει πρώτα αυτά που θα είναι στο χρήστη πιο ενδιαφέροντα.

Σε πολλές περιπτώσεις, όπως και στα προηγούμενα προβλήματα, είναι δύσκολο για τους χρήστες να ποσοτικοποιήσουν τις προτιμήσεις τους. Είναι πιο εύκολο για έναν άνθρωπο να δώσει μια προτίμηση της μορφής προτιμώ το Α περισσότερο από το Β χωρίς όμως να δηλώνει το πόσο περισσότερο. Στο Label Ranking, όμως αυτό είναι μόνο οι ποιοτικές συγκρίσεις μεταξύ των ετικετών. Αξίζει να αναφέρουμε ότι μέθοδος RPC χρησιμοποιεί μοντέλα δυαδικής ταξινόμησης που απαντούν στην ερώτηση προτιμώ το Α ή το Β για όλα τα ζευγαράκια ετικετών, κατηγοριών, πράγμα που σημαίνει ότι είναι μία φυσική προσέγγιση του προβλήματος

Στην συγκεκριμένη διπλωματική στόχος μας είναι να μελετήσουμε το πρόβλημα PLabel Ranking και συγκεκριμένα μέσω της προσέγγισης RPC. Η προσέγγιση RPC παρότι έχει προταθεί εδώ και 10 χρόνια συνεχίζει να έχει πολύ ανταγωνιστικά αποτελέσματα. Μέσω συγκριτικής αξιολόγησης πειραματιζόμαστε με τις διάφορες παραμέτρους του μοντέλου ώστε να δούμε τις δυνατότητες του. Επιπλέον, κάνοντας σε βάθος ανάλυση και μελέτη των χαρακτηριστικών και των ιδιοτήτων του προβλήματος, αναλύουμε την πολυπλοκότητα του Και εξάγουμε συμπεράσματα Σχετικά με τα επίπεδα προβλέψεις που αναμένουμε από τα μοντέλα μας για κάθε πρόβλημα μάθησης ξεχωριστά. Ελπίζουμε ότι η δουλειά αυτή θα συμπληρώσει το έργο που ήδη έχει γίνει και ταυτόχρονα θα δεν ισχύει στο ερευνητικό ενδιαφέρον πάνω στο συγκεκριμένο πρόβλημα.

Σε αυτή την ενότητα ξεκινήσαμε με μια γρήγορη εισαγωγή στο πρόβλημα Label Ranking 1.1. Στην επόμενη ενότητα 1.2 περιγράφουμε το θεωρητικό υπόβαθρο, όπου θα αναφερθούμε στα μαθηματικά εργαλεία που είναι απαραίτητα για την κατανόηση του προβλήματος. Στην ενότητα 1.3 θα δώσουμε έναν ορισμό του προβλήματος και θα εξηγήσουμε πως λειτουργεί η μέθοδος RPC. Στην ενότητα 1.4 θα κάνουμε μία συγκριτική ανάλυση των διαφόρων μοντέλων που κατασκευάζουμε, πειραματιζόμενοι με τα διάφορα στάδια της μεθόδου RPC. Θα εξετάσουμε ποια μοντέλα πετυχαίνουν καλύτερα αποτελέσματα και θα εξηγήσουμε γιατί. Στην επόμενη ενότητα 1.5, κάνουμε εντατική ανάλυση των χαρακτηριστικών των διαφόρων προβλημάτων μάθησης και εξετάζουμε σε βάθος τους παράγοντες που κάνουν πολύπλοκο το πρόβλημα Label Ranking. Στην τελευταία ενότητα 1.6, αναφέρουμε τα συμπεράσματα και τα μελλοντικά σχέδια.

1.2 Θεωρητικό Υπόβαθρο

Για την μελέτη του προβλήματος πρέπει να ορίσουμε πρώτα τις απαραίτητες μαθηματικές έννοιες. Τα βασικά εργαλεία τα οποία θα μας απασχολήσουν είναι οι μεταθέσεις, οι αποστάσεις μεταξύ μεταθέσεων και οι τεχνικές συναθροισης.

1.2.1 Μεταθέσεις

Υπάρχουν πολλοί τρόποι να εκφράσουμε μια μετάθεση ή ταξινόμηση στοιχείων. Μπορούμε να θεωρήσουμε ότι η μετάθεση είναι μια αμφιμονοσήμαντη συνάρτηση.

Definition. *Έστω $\pi : [n] \to [n]$ μια αμφιμονοσήμαντη συνάρτηση. Θα λέμε ότι η π είναι μια μετάθεση των στοιχείων του συνόλου $[n] = \{1, 2, 3, \ldots, n\}$.*

Για παράδειγμα, ας θεωρήσουμε την συνάρτηση $\pi : [5] \rightarrow [5]$, που έχει τιμές $\pi(1) = 3, \pi(2) = 4, \pi(3) = 5, \pi(4) = 2$ και $\pi(5) = 1$. Η συνάρτηση ορίζει μια μετάθεση στα στοιχεία του συνόλου $[5]$, και σηματοδοτεί ότι το στοιχείο 3 προτιμάται περισσότερο, έπειτα το στοιχείο 4, το στοιχείο 5, το στοιχείο 2 και τέλος το στοιχείο 1. Υπάρχει επίσης η αντίστροφη συνάρτση π^{-1}, με τιμές $\pi^{-1}(1) = 5, \pi^{-1}(2) = 4, \pi^{-1}(3) = 1, \pi^{-1}(4) = 2$ και $\pi^{-1}(5) = 3$. Η συνάρτηση $\pi(i) = j$ απαντάει στην ερώτηση "ποιός αριθμός βρίσκεται στην θέση i;", ενώ η αντίστροφη συνάρτηση $\pi^{-1}(j) = i$ απαντάει στην ερώτηση "σε ποια θέση τοποθετείται ο αριθμός j;".

Μια μετάθεση ενός συνόλου αντικειμένων είναι μια τοποθέτηση των αντικειμένων αυτών με μια συγκεκριμένη σειρά. Η μετάθεση είναι ένα μαθηματικό εργαλείο το οποίο χρησιμοποιείται για να εκφράσει την σειρά προτίμησης ανάμεσα σε κάποια αντικείμενα. Από ένα σύνολο n αντικειμένων μπορούν να δημιουργηθούν n! πλήρεις μεταθέσεις.

Για κάθε $i, j \in [n]$, συμβολίζουμε την σχέση προτεραιότητας ανάμεσα στα στοιχεία με $i \succ_\pi j$. Ο συμβολισμός αυτό αντιστοιχεί σε $\pi(i) < \pi(j)$, δηλαδή ότι το στοιχείο i προτιμάται έναντι του j.

Σε περίπτωση που τοποθετούμε σειρά προτίμησης μόνο σε ένα υποσύνολο των αντικειμένων του αρχικού συνόλου $[n]$ αντικειμένων, λέμε ότι έχουμε μια μερική μετάθεση.

1.2.2 Αποστάσεις μεταξύ μεταθέσεων

Στην επιτηρούμενη μάθηση το υπολογιστικό πρόγραμμα δέχεται τις παραδειγματικές εισόδους και τα επιθυμητά αποτελέσματα για αυτές και στόχος είναι να δημιουργηθεί ένα μοντέλο που θα μάθει έναν γενικό κανόνα αντιστοίχισης των εισόδων σε εξόδους, δηλαδή μεταθέσεις του συνόλου των ετικετών. Σ' αυτή διαδικασία μάθησης, το μοντέλο είναι απαραίτητο να καταλαβαίνει πόσο κοντά βρίσκονται οι προβλέψεις του από τις πραγματικές αναμενόμενες απαντήσεις. Απαιτείται δηλαδή η ποσοτικοποίηση της διαφοράς μεταξύ προβλεπόμενων και αναμενόμενων εξόδων. Για τον σκοπό αυτό χρησιμοποιούμε τις αποστάσεις μεταξύ μεταθέσεων. Υπάρχουν πολλές μετρικές για αποστάσεις μεταξύ μεταθέσεων, όπως Hamming Distance, Spearman footrule ή Spearman distance, αλλά η πιο γνωστή μετρική για το πρόβλημα Label Ranking είναι η Kendall tau Distance. Ο ορισμός της απόστασης μεταξύ δύο μεταθέσεων π και σ έχει ως εξής:

$$D_K(\pi, \sigma) = \#\{(i,j)|\pi(i) > \pi(j) \wedge \sigma(i) < \sigma(j)\} + \#\{(i,j)|\pi(i) < \pi(j) \wedge \sigma(i) > \sigma(j)\}\ ,$$

όπου $1 \leq i < j \leq m$ και m είναι το πλήθος των ετικετών που περιέχει το σύνολο L ($|L| = m$).

Ουσιαστικά η μετρική αυτή υπολογίζει το πλήθος των ζευγών από ετικέτες που βρίσκονται σε αντίστροφη σειρά ανάμεσα στις δύο μεταθέσεις.

Η κανονικοποίηση της μετρικής αυτής ονομάζεται kendall tau coefficient και υπολογίζεται από το ακόλουθο τύπο:

$$\tau(\pi, \sigma) = 1 - \frac{4D_K(\pi, \sigma)}{m(m-1)}$$

Η μετρική kendall tau coefficient παίρνει τιμές στο διάστημα $[-1, 1]$. Στην συνέχεια θα χρησιμοποιηθεί ως φυσική και απλή μετρική για την μέτρηση της επίδοσης των μοντέλων θα υλοποιήσουμε.

1.2.3 Τεχνικές συνάθροισης

Οι τεχνικές συναθροίσεις σκοπεύουν στο να δημιουργήσουν μια αντιπροσωπευτική μετάθεση - έξοδο, δεδομένων κάποιον μικρότερων κομματιών πληροφορίας, στη συγκεκριμένη περίπτωση δυαδικές προτιμήσεις μεταξύ ζευγών από ετικέτες.

1.3 Ορισμός Προβλήματος και μοντέλο RPC

Στην ενότητα αυτή δίνουμε τον μαθηματικό ορισμό του προβλήματος Label Ranking. Στην συνέχεια, περιγράφουμε μια προσέγγιση επίλυσης του προβλήματος, που ονομάζουμε RPC και θα μας απασχολήσει στα πλαίσια της εργασίας αυτής.

1.3.1 Ορισμός Προβλήματος

Υπάρχουν οι εξής χώροι που μας ενδιαφέρουν στο συγκεκριμένο πρόβλημα.

- ο χώρος δειγμάτων ή εισόδων X: Η είσοδος μπορεί να θεωρηθεί ως ένα διάνυσμα x από τον πεδίο ορισμού X. Κάθε είσοδος έχει d γνωρίσματα οπότε συνολικά ο χώρος εισόδων X είναι d-διάστατος

- ο χώρο ετικετών L: ο χώρος $L = \{\lambda_1, \ldots, \lambda_m\}$ περιέχει τις ετικέτες (κατηγορίες) που αφορούν το πρόβλημα.

- ο χώρος μεταθέσεων ή εξόδων Ω ή Y: Το σύνολο Ω ή Y είναι το πεδίο τιμών της συνάρτησης που προσπαθεί να προσεγγίσει το μοντέλο που επιλύει το πρόβλημα Label Ranking. Ο χώρος αυτό αποτελείται από πλήρεις μεταθέσεις των στοιχείων του συνόλου L και έχει μέγεθος $m!$.

Το $m!$ έχει τεράστιο ρυθμο αύξησης καθώς αυξάνεται το m, και αυτό αποτελεί σημαντικό παράγοντα δυσκολίας του προβλήματος.

Δεδομένου ενός δείγματος εισόδου x από το πεδίο ορισμού X και του συνόλου ετικετών L στο label ranking στόχος είναι η εκπαίδευση ενός μοντέλου που αντιστοιχίζει το x με μια πλήρη μετάθεση των L. Αυτό σημαίνει ότι υπάρχει η μια πλήρης, μεταβατική και ασύμμετρη σχέση $\succ_{\boldsymbol{x}}$ στο L, όπου $\lambda_i \succ_{\boldsymbol{x}} \lambda_j$ υποδηλώνει ότι το λ_i προηγείται του λ_j στην μετάθεση που αφορά το δείγμα x, με $1 \leq i, j \leq m$. Αφού, κάθε μετάθεση δηλώνει και μια σχέση προτίμησης μπορούμε να θεωρήσουμε ότι $\lambda_i \succ_{\boldsymbol{x}} \lambda_j$ υποδηλώνει ότι το λ_i προτιμάται έναντι του λ_j για το συγκεκριμένο δείγμα $\boldsymbol{x}$. Η μετάθεση μπορεί να εκφραστεί χρησιμοποιώντας μια συνάρτηση $\pi_{\boldsymbol{x}}$ όπου $\pi_{\boldsymbol{x}}(i) = \pi_{\boldsymbol{x}}(\lambda_i)$ μας δείχνει την θέση που λαμβάνει η ετικέτα στην αντίστοιχη μετάθεση. Η μετάθεση συμβολίζεται ως:

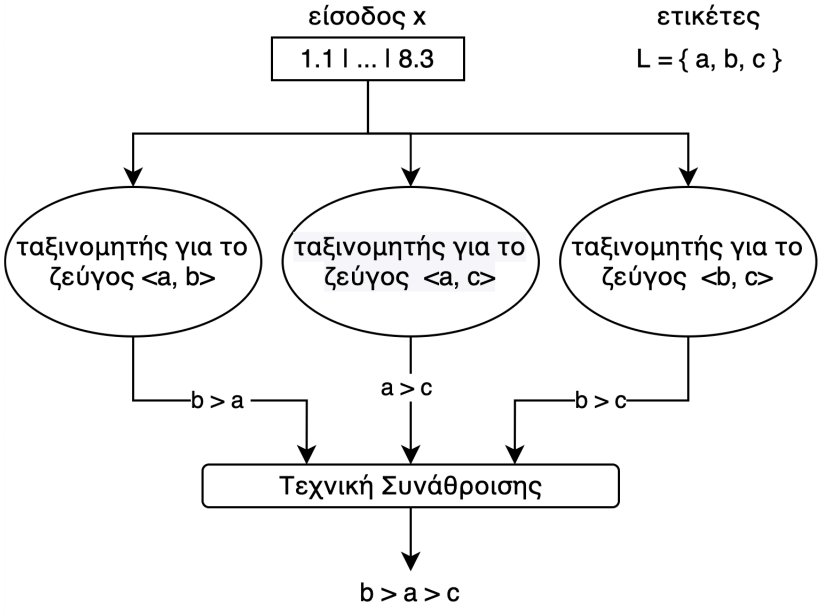
$$\lambda_{\pi_{\boldsymbol{x}}^{-1}(1)} \succ_{\boldsymbol{x}} \lambda_{\pi_{\boldsymbol{x}}^{-1}(2)} \succ_{\boldsymbol{x}} \cdots \succ_{\boldsymbol{x}} \lambda_{\pi_{\boldsymbol{x}}^{-1}(m)}$$

Για να μετρήσουμε την απόσταση μεταξύ ενός μοντέλου επίλυσης του Label Ranking, χρησιμοποιούμε την μετρική Kendall's tau correlation coefficient, που είναι γνωστή στην στατιστική και η πιο διαδεδομένη στην βιβλιογραφία του συγκεκριμένου προβλήματος.

1.3.2 Προσέγγιση RPC

Η μέθοδος RPC χωρίζει σε δύο στάδια. Το πρώτο στάδιο μπλαμπλά. Το δεύτερο στάδιο μπλα. Τα θετικά και τα αρνητικά.

Η μέθοδος RPC εντάσσεται στις προσεγγίσεις αναγωγής. Αποσυνθέτει το πρόβλημα label ranking σε απλούστερα δυαδικά προβλήματα ταξινόμησης και στην συνέχεια οι λύσεις αυτών των προβλημάτων συνδυάζονται κατάλληλα για την πρόβλεψη μιας μετάθεσης. Είναι μια μέθοδος που προτάθηκε από τον Hullermeier et al. το 2010 αλλά ακόμα παραμένει από τις πιο ανταγωνιστικές προσεγγίσεις. Η κύρια ιδέα βασίζεται στην απευθείας μοντελοποίηση των προτιμήσεων μεταξύ ζευγών ετικετών. Για κάθε ζεύγος ετικετών, υλοποιείται ένα μοντέλο που προβλέπει ποια ετικέτα προηγείται έναντι της άλλης δεδομένου ενός δείγματος x. Στην συνέχεια, οι προβλέψεις αυτές συνδυάζονται με μια μέθοδο συνάθροισης ώστε να σχηματιστεί η τελική πρόβλεψη. Μια οπτικοποίηση της μεθόδου φαίνεται στον επόμενο πίνακα 1.1.



Σχήμα 1.1: Οπτικοποίηση της προσέγγισης RPC

Δυαδικοί ταξινομητές Στο πρώτο στάδιο του αλγορίθμου, για κάθε ζεύγος ετικετών, υλοποιείται ένα δυαδικό μοντέλο που προβλέπει δεδομένου ενός δείγματος x ποια ετικέτα προηγείται ανάμεσα στις δύο ετικέτες του ζεύγους. Χρησιμοποιούμε τον συμβολισμό M_{ij}, με $1 \leq i < j \leq m$ για να δηλώσουμε ότι το δυαδικό μοντέλο είναι υπεύθυνο για τις ετικέτες (λ_i, λ_j) από το σύνολο L. Συνολικά δημιουργούνται $\frac{m(m-1)}{2}$ μοντέλα, όσα και τα ζεύγη των ετικετών. Από κάθε δείγμα εκπαίδευσης, εφόσον έχουμε πλήρεις μετάθεσεις, προκύπτουν μέσω της μεταβατικής ιδιότητας $\frac{m(m-1)}{2}$ δείγματα εκπαίδευσης, ένα για κάθε δυαδικό μοντέλο. Οποιοσδήποτε γνωστός αλγόριθμος ταξινόμησης μπορεί να χρησιμοποιηθεί για τα δυαδικά μοντέλα. Τα αλγόριθμοι μπορεί να είναι είτε classifiers είτε regressors. Στην περίπτωση των classifiers, οι προβλέψεις του δυαδικού μοντέλου M_{ij} ανήκουν στο διακριτό σύνολο $\{0, 1\}$ όπου $y = 0$ αν το μοντέλο θεωρεί ότι το $\lambda_i \succ_x \lambda_j$ και $y = 1$ διαφορετικά. Στην περίπτωση των regressors, οι προβλέψεις του δυαδικού μοντέλο M_{ij} γίνονται στο συνεχές διάστημα $[0, 1]$ και μπορούν να θεωρηθούν ως ένα είδος ποσοτικοποιημένης αυτοπεποίθησης του μοντέλου για το ποια ετικέτα προηγείται. Όταν $y \to 0$ το μοντέλο προβλέπει με μεγάλη σιγουριά ότι το $\lambda_i \succ \lambda_j$ ενώ όταν $y \to 1$ το μοντέλο προβλέπει με μεγάλη αυτοπεποίθηση ότι $\lambda_j \succ \lambda_i$. Όταν $y = 0.5$, το μοντέλο θεωρεί δεν θεωρεί ότι υπάρχει κάποια ισχυρή προτίμηση ανάμεσα στα δύο.

Τεχνικές Συνάθροισης Στο δεύτερο στάδιο απαιτείται ο συνδυασμός των προβλέψεων του πρώτου σταδίου για τον σχεδιασμό μιας μετάθεσης που θα χρησιμοποιείται ως πρόβλεψη του μοντέλου. Ως τεχνική συνάθροισης μπορεί να χρησιμοποιηθεί οποιοσδήποτε γνωστός αλγόριθμος. Η αρχική τεχνική συνάθροισης που προτάθηκε από τον Hullermeier χρησιμοποιεί μια απλή λογική αθροίσματος, που ακολουθεί παρόμοια λογική με την μέθοδο Borda count. Κάθε ετικέτα λαμβάνει ψήφους ίσου βάρους με τις προβλέψεις που είναι υπέρ της από τα δυαδικά μοντέλα που την αφορούν. Κάθε δυαδικό μοντέλο έχει συνολικά ένα ψήφο βάρους 1 που χωρίζεται ανάμεσα στις δύο σχετικές ετικέτες. Για παράδιεγμα, εάν το μοντέλο M_{ij} έχει έξοδο 0.3 τότε δίνουμε 0.3 ψήφους στο τι λ_i και 0.7 ψήφους στο λ_j. Ο μαθηματικός τύπος έχει ως εξής:

$$S(\lambda_i) = \sum_{k=1}^{i-1} M_{ki} + \sum_{k=i+1}^{n} (1 - M_{ik})$$

Αφού ολοκληρωθεί αυτή η διαδικασία, κάθε ετικέτα έχει συλλέξει κάποιος ψήφους. Στην συνέχεια, οι ετικέτες κατατάσσονται σε φθίνουσα σειρά ανάλογα με το πλήθος των ψήφων που έχουν συλλέξει και η στοίχιση που προκύπτει αποτελεί την σειρά προτίμησης των ετικετών στην πρόβλεψη του Θα χρησιμοποιούμε την ονομασία SumBP (Summing Binary Predictions) για να αναφερθούμε σε αυτή την τεχνική συνάθροισης με συντομία.

Θετικά και Αρνητικά Ένα κύριο θετικό της συγκεκριμένης μεθόδου είναι η απλότητα και η κομψότητα. Ταυτόχρονα, η αναγωγή στο πρόβλημα της δυαδικής ταξινόμησης επιτρέπει τη πρόσβαση σε μια μεγάλη πληθώρα τεχνικών και μελετημένων μεθόδων. Τέλος, το σχήμα της προσέγγισης αυτής είναι αρθρωτό, δηλαδή έχει δύο εντελώς ανεξάρτητα στάδια. Πρακτικά, μπορούμε έχοντας εκπαιδεύσει τα δυαδικά μοντέλα του πρώτου σταδίου και αλλάζοντας μόνο την τεχνική συνάθροισης να υλοποιήσουμε ένα καινούργιο μοντέλο που θα πετυχαίνει νέα αποτελέσματα. Το αρνητικό της συγκεκριμένης προσέγγισης είναι ότι έχει υψηλές υπολογιστικές απαιτήσεις, καθώς απαιτείται εκπαίδευση μεγάλου αριθμού δυαδικών ταξινομητών. Ανάλογα το πόσο περίπλοκος και απαιτητικός είναι ο αλγόριθμος ταξινόμησης επηρεάζονται πρακτικά οι χρονικές απαιτήσεις εκπαίδευσης.

Είναι προσέγγιση, ενώ έχει προταθεί εδώ και πολύ καιρό, εξακολουθεί να πετυχαίνει πολύ ανταγωνιστικά αποτελέσματα συγκριτικά με την με άλλες μεθόδους. Αυτό φαίνεται και στον επόμενο πίνακα

dataset	SVC	DTC	RFC100	SVR	DTR	RFR100
authorship	0.943±0.017	0.873±0.022	0.935±0.018	**0.949±0.016**	0.873±0.022	0.924±0.021
bodyfat	0.278±0.059	0.126±0.063	0.204±0.06	**0.287±0.055**	0.127±0.06	0.208±0.061
calhousing	0.298±0.012	0.354±0.011	0.484±0.009	0.299±0.012	0.354±0.011	**0.487±0.01**
cpu-small	0.501±0.014	0.397±0.015	**0.519±0.014**	0.504±0.014	0.398±0.016	0.515±0.013
fried	0.983±0.001	0.987±0.001	0.991±0.001	0.97±0.001	0.987±0.001	**0.993±0.001**
glass	0.862±0.049	0.875±0.04	**0.901±0.035**	0.871±0.046	0.875±0.04	0.894±0.034
housing	0.689±0.027	0.782±0.027	**0.822±0.026**	0.692±0.026	0.782±0.027	0.817±0.023
iris	0.968±0.034	0.949±0.042	0.965±0.037	**0.983±0.025**	0.949±0.042	0.958±0.037
pendigits	0.957±0.002	0.961±0.001	0.975±0.001	0.95±0.002	0.961±0.001	**0.977±0.001**
segment	0.957±0.005	0.969±0.005	**0.977±0.005**	0.951±0.006	0.969±0.005	**0.977±0.004**
stock	0.892±0.013	0.901±0.015	**0.926±0.014**	0.897±0.014	0.901±0.015	0.92±0.013
vehicle	0.866±0.026	0.831±0.032	0.882±0.025	0.861±0.028	0.831±0.032	**0.883±0.027**
vowel	0.897±0.015	0.861±0.015	**0.913±0.015**	0.909±0.014	0.861±0.015	0.902±0.014
wine	0.959±0.043	0.892±0.081	0.941±0.051	**0.965±0.034**	0.892±0.081	0.923±0.061
wisconsin	0.555±0.034	0.512±0.031	0.549±0.034	**0.584±0.032**	0.512±0.032	0.572±0.033

Πίνακας 1.1: Πίνακας αποτελεσμάτων για διαφορετικούς ταξινομητές

1.4 Συγκριτική Αξιολόγηση

Κατά την συγκριτική αξιολόγηση, πειραματιστήκαμε και με τα δύο στάδια της RPC προσέγγισης. Για την υλοποίηση χρησιμοποιήσαμε την γλώσσα προγραμματισμού Python, καθώς είναι γλώσσα γενικού σκοπού με απλή σύνταξη και μεγάλο πλήθος βιβλιοθηκών με έτοιμες συναρτήσεις και μοντέλα τεχνητής νοημοσύνης. Για τον υπολογισμό των Kendall tau scores έχουμε ένα σχήμα 5 επαναλήψεων 10 fold cross validation.

Για τα δυαδικά μοντέλα του πρώτου σταδίου χρησιμοποιήσαμε 3 γνωστούς αλγορίθμους ταξινόμησης. Κάθε ένας υλοποιήθηκε τόσο σε classifier εκδοχή όσο και σε regressor εκδοχή, συνεπώς δημιουργήσαμε 6 διαφορετικά μοντέλα. Για την δυνατότητα σύγκρισης μεταξύ τους, χρησιμοποιήσαμε ως τεχνική συνάθροισης την SumBP. Οι αλγόριθμοι είναι οι εξής:

- Support Vector Machines: Συμβολίζουμε με SVC και SVR τα μοντέλο που χρησιμοποιούν το συγκεκριμένο αλγόριθμο ταξινόμησης στην classifier και regressor μορφή αντίστοιχα.
- Δέντρα Αποφάσεων (Decision Trees) : Συμβολίζουμε με DTC και DTR τα μοντέλα που χρησιμοποιούν το συγκεκριμένο αλγόριθμο ταξινόμησης στην classifier και regressor μορφή αντίστοιχα.
- Τυχαία Δάση (Random Forests) : Συμβολίζουμε με RFC και RFR τα μοντέλα που χρησιμοποιούν το συγκεκριμένο αλγόριθμο ταξινόμησης στην classifier και regressor μορφή αντίστοιχα.

Τα αποτελέσματα για τους δυαδικούς αλγορίθμους ταξινόμησης φαίνονται στον πίνακα 1.1.

Παρατηρούμε ότι RFC και SVR πετυχαίνουν τα καλύτερα αποτελέσματα ως προς kendall tau. Ο SVR αλγόριθμος είναι ο καλύτερος σε datasets τα οποία είναι σε μεγάλο βαθμό γραμμικά διαχωρίσιμα ή είναι εύκολο να γίνουν γραμμικά διαχωρίσιμα μέσω κατάλληλης συνάρτησης. Όπως ήταν αναμενόμενο, οι ταξινομητές DTC και DTR πετυχαίνουν υποδεέστερα αποτελέσματα συγκριτικά με RFC και RFR, αφού οι πρώτοι χρησιμοποιούν ένα μοναδικό δέντρο αποφάσεων ενώ οι δεύτεροι χρησιμοποιούν ένα σύνολο από δέντρα αποφάσεων που λαμβάνουν συνδυαστικά μια τελική πρόβλεψη. Τέλος, σημαντικό είναι να τονίσουμε ότι ο πειραματισμός με τους ταξινομητές προκαλεί αρκετή διαφοροποίηση στα αποτελέσματα πρόβλεψης της προσέγγισης RPC. Συνεπώς, συμπεραίνουμε ότι το πρώτο στάδιο της προσέγγισης είναι σημαντικό.

Όσον αφορά τις τεχνικές συνάθροισης πειραματιστήκαμε με τις εξής:

	SumBP		MaxVTI		MaxVTI-LF		KWIK	
dataset	RFC100	SVR	RFC100	SVR	RFC100	SVR	RFC100	SVR
authorship	0.935±0.018	0.949±0.016	0.936±0.018	0.947±0.015	0.929±0.017	0.925±0.016	0.938±0.017	0.947±0.015
bodyfat	0.204±0.06	0.287±0.055	0.193±0.061	0.268±0.059	0.194±0.061	0.265±0.06	0.198±0.067	0.276±0.058
calhousing	0.484±0.009	0.296±0.011	0.488±0.009	0.299±0.012	0.488±0.01	0.294±0.013	0.481±0.009	0.298±0.012
cpu-small	0.519±0.014	0.507±0.014	0.522±0.014	0.503±0.014	0.523±0.015	0.497±0.014	0.519±0.014	0.503±0.013
glass	0.901±0.035	0.871±0.046	0.897±0.035	0.879±0.047	0.84±0.056	0.807±0.062	0.898±0.037	0.875±0.044
housing	0.822±0.026	0.692±0.026	0.823±0.025	0.701±0.029	0.822±0.023	0.695±0.031	0.82±0.026	0.695±0.028
iris	0.965±0.037	0.983±0.025	0.965±0.037	0.961±0.035	0.965±0.037	0.953±0.043	0.965±0.037	0.961±0.035
segment	0.977±0.005	0.951±0.005	0.912±0.016	0.905±0.017	0.97±0.005	0.938±0.006	0.977±0.005	0.95±0.006
stock	0.926±0.014	0.897±0.014	0.926±0.013	0.899±0.013	0.926±0.013	0.893±0.013	0.925±0.014	0.899±0.013
vehicle	0.882±0.025	0.861±0.028	0.883±0.026	0.872±0.028	0.881±0.026	0.853±0.034	0.882±0.026	0.873±0.027
vowel	0.914±0.016	0.895±0.011	0.912±0.016	0.905±0.017	0.91±0.017	0.903±0.017	0.903±0.017	0.898±0.016
wine	0.941±0.051	0.965±0.034	0.941±0.051	0.96±0.036	0.941±0.051	0.955±0.031	0.955±0.036	0.954±0.046
wisconsin	0.549±0.034	0.584±0.032	0.478±0.034	0.496±0.034	0.478±0.034	0.496±0.034	0.528±0.036	0.554±0.031

Πίνακας 1.2: Πίνακας αποτελεσμάτων για διαφορετικές τεχνικές συνάθροισης

- SumBP: η αθροιστική μέθοδος που προτάθηκε στην αρχική δημοσίευση και περιγράφηκε προηγουμένως

- Max-Votes: στη μέθοδο αυτή, θεωρούμε τις μεταθέσεις που συναντάμε στο σύνολο εκπαίδευσης ως πιθανές απαντήσεις. Ανάλογα με την πρόβλεψη του ταξινομητή και την σειρά των ετικετών του ζεύγους, κάθε μετάθεση λαμβάνει τους αντίστοιχους ψήφους. Στο τέλος της διαδικασίας, η μετάθεση με τους περισσότερους ψήφους επιλέγεται ως πρόβλεψη. Λειτουργεί μόνο για σύνολα εκπαίδευσης που περιέχουν πλήρης ταξινομήσεις.

- Max-Votes-F: η μέθοδος αυτή είναι παρόμοια με την Max-Votes, με την διαφορά ότι λαμβάνει σε μικρό βαθμό υπόψιν της και τον παράγοντα της συχνότητας εμφάνισης της μετάθεσης

- Kwiksort: η μέθοδος αυτή ακολουθεί μια λογική παρόμοια με αυτή τον αλγόριθμο ταξινόμησης quicksort και συνδυάζει τις δυαδικές προβλέψεις των ταξινομητών με αλγόριθμο χαμηλότερης πολυπλοκότητας.

Ο πίνακας αποτελεσμάτων για τις τεχνικές συνάθροισης φαίνεται εδώ 1.2.

Άξιο αναφοράς είναι ότι ο πειραματισμός με τις μεθόδους συνάθροισης δεν επηρεάζει σημαντικά τα αποτελέσματα. Σε αντίθεση με το πρώτο στάδιο της προσέγγισης RPC, το δεύτερο στάδιο αρκεί να είναι απλό και να ακολουθεί μια μαθηματικά σωστή λογική. Επιπλέον, η χρήση περισσότερης πληροφορίας δεν καθιστά απαραίτητα μια μέθοδο καλύτερη. Η πληροφορία πρέπει να είναι ουσιαστική αλλιώς μπορεί να δράσει ως θόρυβος στα δεδομένα μας. Παράλληλα, ο τρόπος με τον οποίο αξιοποιείται η επιπλέον πληροφορία πρέπει να είναι ορθός. Στην περίπτωση της MaxVTI-LF, η επιπλέον πληροφορία για τις συχνότητες των μεταθέσεων πετυχαίνει ελάχιστα υποδεέστερα αποτελέσματα σε σύγκριση με την απλή εκδοχή της τεχνικής συνάθροισης MaxVTI.

Όλες οι προσεγγίσεις δυσκολεύονται σε αντίστοιχα learning προβλήματα. Γιατί κάποια προβλήματα είναι δυσκολότερα από κάποια άλλα. Που έγκειται η πολυπλοκότητα του προβλήματος Label Ranking.

1.5 Ανάλυση της δομής του προβλήματος

Από τα αποτελέσματα των προσεγγίσεων της βιβλιογραφίας προκύπτει το γενικό συμπέρασμα ότι όλες οι προσεγγίσεις πετυχαίνουν κάποια standards ανάλογα το πρόβλημα μάθησης. Για παράδειγμα, στα bodyfat και calhousing όλες οι προσεγγίσεις πετυχαίνουν χαμηλές τιμές ως προς την μετρική kendall tau coefficient ενώ ενώ στα iris και fried πετυχαίνουν υψηλές τιμές. Σε αυτό το κεφάλαιο κάνουμε μια διεξοδική ανάλυση των προβλημάτων μάθησης που χρησιμοποιούνται

κατά κόρον στα ερευνητικά έργα που αφορούν το Label Ranking, εξετάζοντας σε βάθος τους λόγους που κάνουν ένα πρόβλημα μάθησης δύσκολο και χρησιμοποιώντας τεχνικές και μετρικές που μας δίνουν αίσθηση για την δυσκολία του προβλήματος.

Χρησιμοποιούμε διάφορες μετρικές για να αναλύσουμε τα σετ, και να εξετάσουμε σε βάθος τους παράγοντες που κάνουν ένα πρόβλημα μάθησης δύσκολο. Στον επόμενο πίνακα 1.3, βλέπουμε τις τιμές αυτών των μετρικών:

dataset	instances	features	labels	unique rankings	roc auc	kendall tau
bodyfat (B)	252	7	7	236	0.602±0.1	0.204±0.06
calhousing (B)	20640	4	4	24	0.736±0.051	0.484±0.009
cpu-small (B)	8192	6	5	119	0.73±0.038	0.519±0.014
wisconsin (B)	194	16	16	194	0.767±0.118	0.549±0.034
housing (B)	506	6	6	112	0.909±0.069	0.822±0.026
vehicle (A)	846	18	4	18	0.932±0.051	0.882±0.025
glass (A)	214	9	6	30	0.86±0.164	0.901±0.035
vowel (A)	528	10	11	294	0.945±0.04	0.914±0.016
stock (B)	950	5	5	51	0.96±0.027	0.926±0.014
authorship (A)	841	70	4	17	0.927±0.069	0.935±0.018
wine (A)	178	13	3	5	0.975±0.043	0.941±0.051
iris (A)	150	4	3	5	0.981±0.036	0.965±0.037
pendigits (A)	10992	16	10	2081	0.981±0.011	0.975±0.001
segment (A)	2310	18	7	135	0.985±0.013	0.977±0.005
fried (B)	40768	9	5	120	0.996±0.001	0.991±0.001

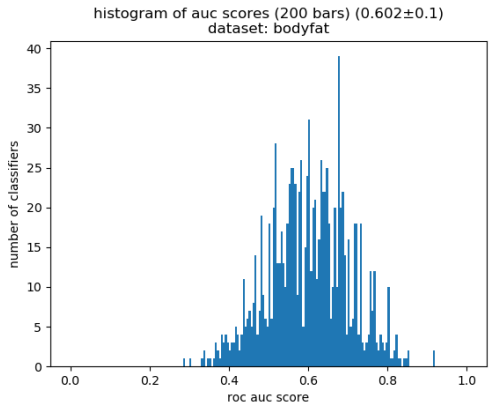
Πίνακας 1.3: Χαρακτηριστικά των προβλημάτων μάθησης (με στοίχηση ως προς kendall tau coefficient)

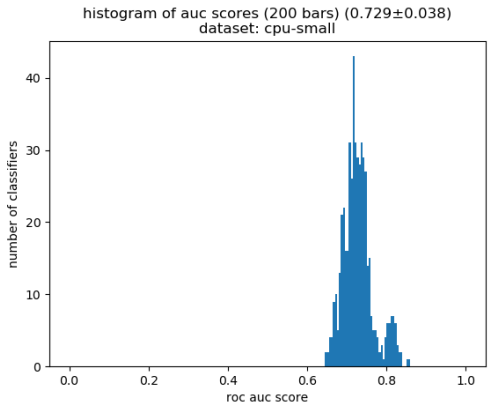
Τα βασικά χαρακτηριστικά των datasets είναι το πλήθος των δειγμάτων, το πλήθος των γνωρισμάτων που έχουν τα δείγματα και το πλήθος των ετικετών. Εισάγουμε κάποιες επιπλέον μετρικές για την ανάλυση μας. Το πλήθος των unique rankings αφορά το πλήθος των διαφορετικών μεταθέσεων που συναντάμε στο σύνολο του προβλήματος μάθησης, χωρίς να μετράμε τις επαναλήψεις τους. Προφανώς το άνω φράγμα της τιμής αυτής είναι το n!, όπου n το πλήθος των labels. Η μετρική ROC AUC αποτελεί γνωστή μετρική μέτρησης της ποιότητας των ταξινομητών. Λαμβάνει τιμές μεταξύ 0 και 1, με τιμή 0.5 να είναι το score που πετυχαίνει ο τυχαίος ταξινομητής και τιμή 1 το score που πετυχαίνει ο βέλτιστος ταξινομητής. Τέλος, μετράμε την ποιότητα του μοντέλου που χρησιμοποιεί ταξινομητή RFC και μέθοδο συνάθροισης SumBP ως προς την μετρική kendall tau coefficient.

Παρατηρούμε πως υπάρχει συσχέτιση ανάμεσα στο ROC AUC και Kendall tau coefficient, γεγονός που επιβεβαιώνει το συμπέρασμα ότι η ποιότητα των ταξινομητών του πρώτου σταδίου επηρεάζει σε μεγάλο βαθμό τα αποτελέσματα της μεθόδου RPC. Όσο το πρωτογενές πρόβλημα ταξινόμηση είναι εύκολο, τόσο ανεβαίνουν τα αποτελέσματα της προσέγγισης RPC. Παρατηρούμε ακόμη ότι το μικρό πλήθος unique rankings και μεγάλο πλήθος instances βοηθάει στο να έχουμε καλύτερα αποτελέσματα, χωρίς αυτό όμως να ισχύει αυστηρά.

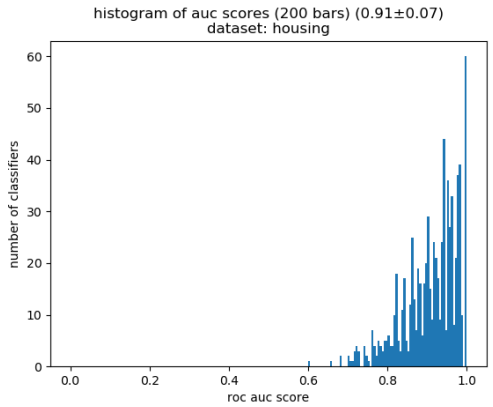
Για διευκόλυνση της ανάλυσης χωρίζουμε τα προβλήματα σε 3 ομάδες με βάση τα kendall tau scores. Για κάθε κατηγορία παρουσιάζουμε τα ιστογράμματα των ROC AUC scores και τα ιστογράμματα των Unique Rankings.

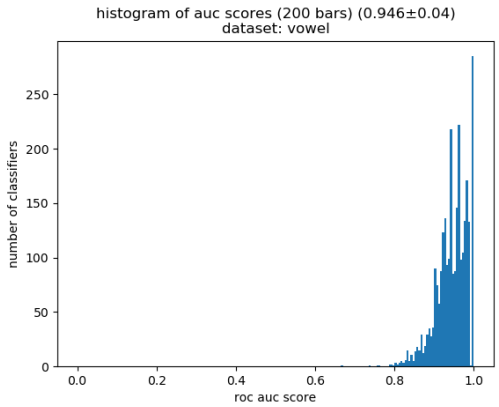
Παρατηρούμε ότι όταν η καμπύλη των unique rankings προσεγγίζει το σχήμα οριζόντιας γραμμής τα scores είναι kendall tau coefficient είναι χαμηλά, ενώ όταν ακολουθεί εκθετική μορφή τα scores είναι συνήθως υψηλά. Αυτό συμβαίνει διότι τα περισσότερα δείγματα στο σύνολο εκπαίδευση αντιστοιχίζονται σε ένα πολύ μικρό μέρος του συνόλου μεταθέσεων. Πρακτικά είναι σαν να μικραίνει το μέγεθος του συνόλου μεταθέσεων που είναι σημαντικός παράγοντας που κάνει το πρόβλημα δύσκολο και συνεπώς τα μοντέλα πετυχαίνουν καλύτερα αποτελέσματα.



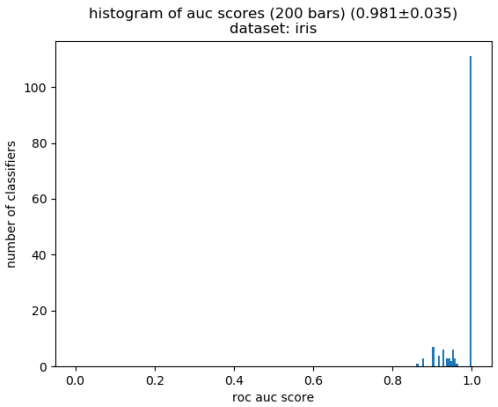


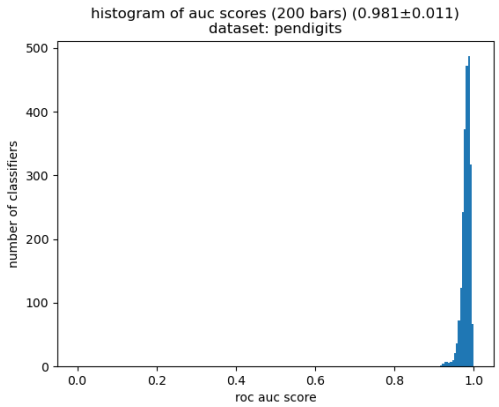
(a) Προβλήματα Μάθησης με χαμηλές αποδόσεις (bodyfat, cpu-small)





(b) Προβλήματα Μάθησης με μέτριες αποδόσεις (housing, vowel)





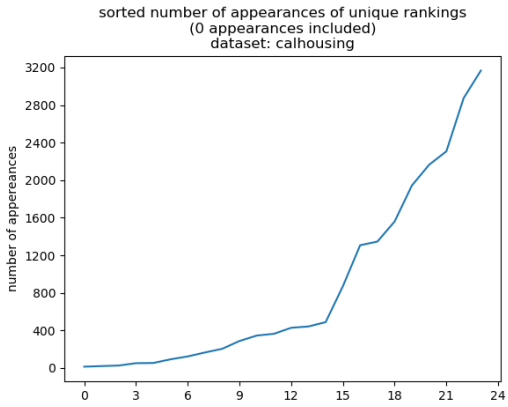
(c) Προβλήματα Μάθησης με υψηλές αποδόσεις (iris, pendigits)

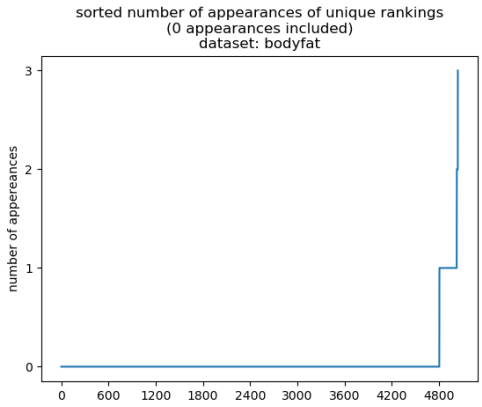
Σχήμα 1.2: Ιστογράμματα ROC AUC scores

Πρακτικά, σημαίνει ότι υπάρχουν λίγες μεταθέσεις που καταλήγουν τα περισσότερα διανύσματα εισόδου συνεπώς το πεδίο τιμών μικραίνει και το πρόβλημα μάθησης γίνεται ευκολότερο.

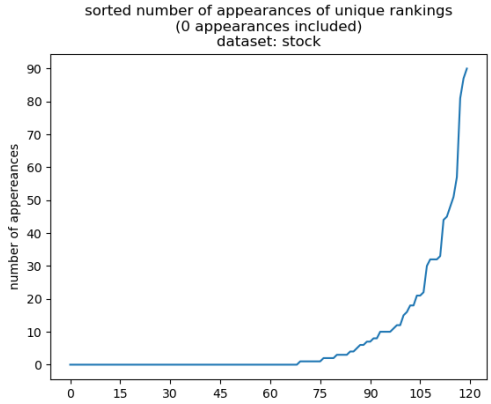
Όσων αφορά τα ιστογράμματα των ROC AUC τιμών των ταξινομητών παρατηρούμε ότι καθώς η ποιότητα των μοντέλα μας πετυχαίνουν καλύτερα αποτελέσματα, η διασπορά των τιμών μειώνεται ενώ ο μέσος όρος της κατανομής των τιμών προσεγγίζει το 1.

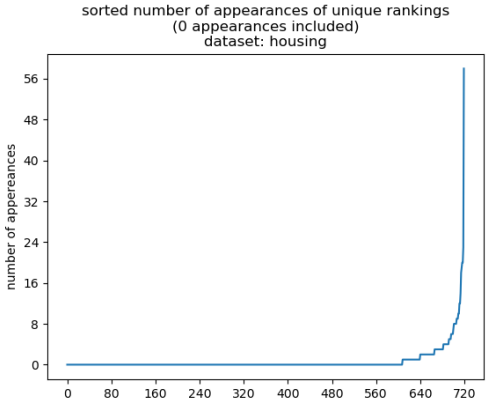
Τέλος χρησιμοποιούμε scatter plots για να δούμε την απεικόνιση των προβλημάτων στον δισδιάστατο χώρο και να κατανοήσουμε διαισθητικά τις διαφορές στην δυσκολία των προβλημάτων. Στην συνέχεια παρουσιάζουμε τις μορφές των scatter plots για τέσσερα προβλήματα μάθησης, με αύξουσα σειρά ως προς kendall tau coefficient. Παρατηρούμε πως με αντίστοιχη σειρά τα πρω-



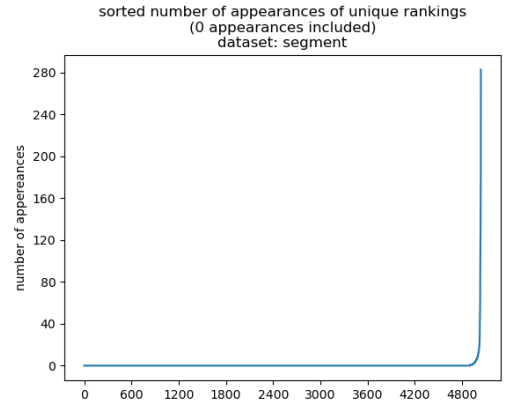


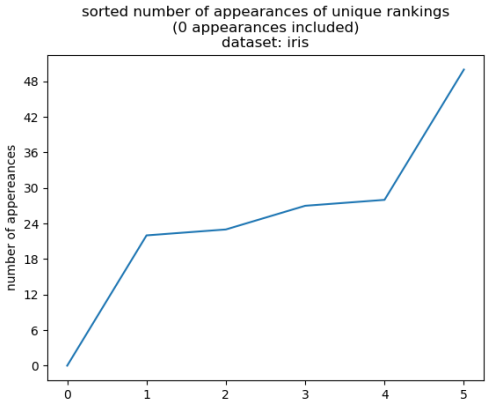
(a) Προβλήματα Μάθησης με χαμηλές αποδόσεις (calhousing, bodyfat)





(b) Προβλήματα Μάθησης με μέτριες αποδόσεις (stock, housing)

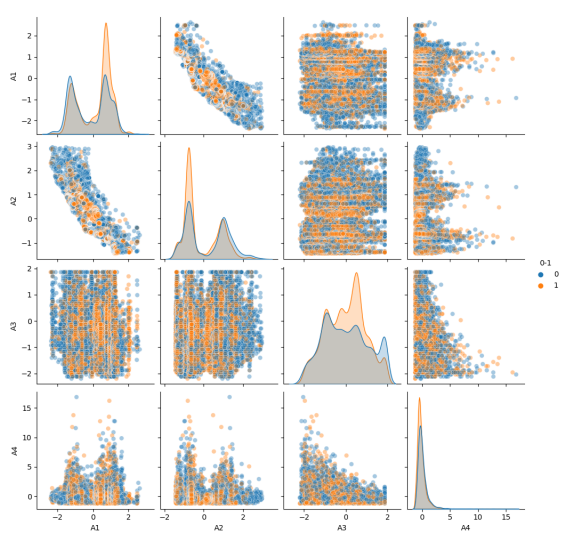




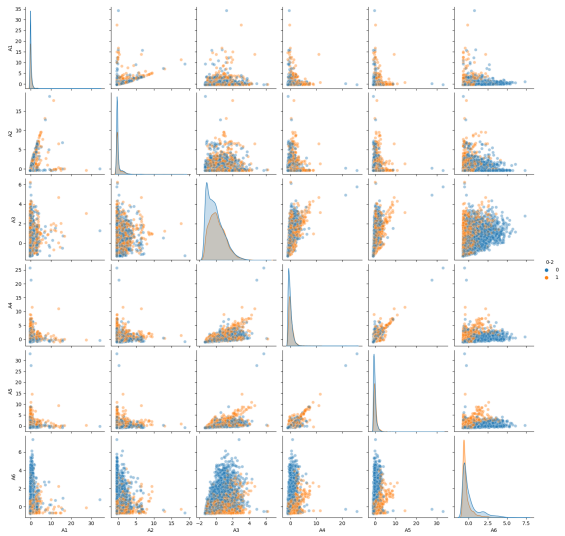
(c) Προβλήματα Μάθησης με υψηλές αποδόσεις (segment, iris)

Σχήμα 1.3: Καμπύλες Ιστογραμμάτων Unique Rankings

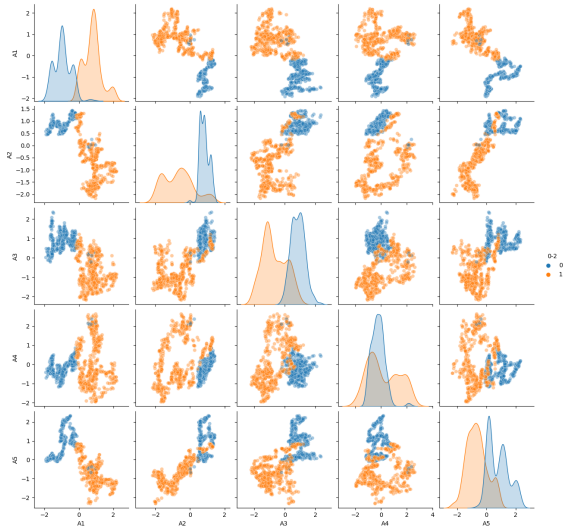
τογενή προβλήματα ταξινόμησης γίνονται πιο εύκολα και κατανοούμε οπτικά την δυσκολία που υπάρχει για ομαδοποίηση για διαχωρισμό ομάδων σε μερικά προβλήματα μάθησης. Συμπεραίνουμε λοιπόν ότι οι μετρικές δίνουν στοιχεία για την συμπεριφορά που αναμένουμε σε ένα πρόβλημα μάθησης. Ωστόσο, ο πιο σημαντικός παράγοντας είναι η δομή που έχει κάθε προβλήματος μάθησης, δηλαδή πως κατανέμονται οι είσοδοι στον χώρο των γνωρισμάτων και πως αντιστοιχίζονται στον χώρο των μεταθέσεων Ω ή .



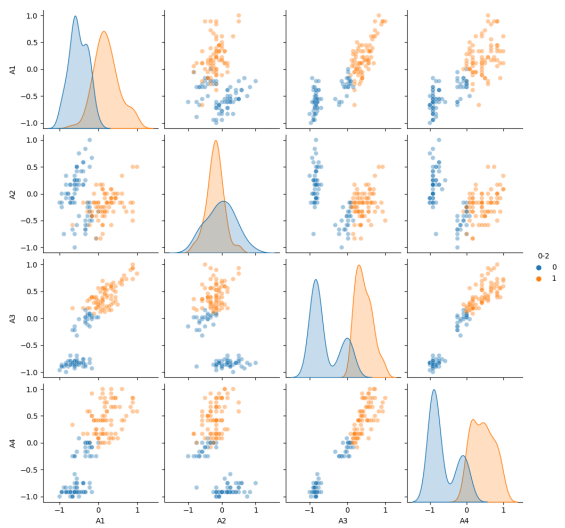
(a) calhousing



(b) cpu-small



(c) stock



(d) iris

Σχήμα 1.4: Scatter Plots για τέσσερα προβλήματα μάθησης

1.6 Συμπεράσματα και Μελλοντικά Σχέδια

Στα πλαίσια αυτής της διπλωματικής, κάναμε μια εισαγωγή στο πρόβλημα του label ranking και υπογραμμίσαμε την σημαντικότητα που έχει λόγω της γενικότητας του και των πολλών πρακτικών εφαρμογών του. Κάναμε μια πιο στοχευμένη αναφορά στην προσέγγιση RPC, που χρησιμοποιεί δυαδικές συγκρίσεις για να ανάγει το πρόβλημα και τεχνικές συνάθροισης για να συνδυάσει τις δυαδικές προτιμήσεις σε μια τελική πρόβλεψη.

Μέσω συγκριτικής ανάλυσης, πειραματιστήκαμε με τις παραμέτρους της προσέγγισης και αποδείξαμε πως ακόμα αποτελεί ένα από τα πιο ανταγωνιστικά μοντέλα. Η μέθοδος RPC αποτελείται από δύο στάδια. Το πρώτο στάδιο, που αποτελείται από τους δυαδικούς ταξινομητές, επηρεάζει σε μεγάλο βαθμό την απόδοση ολόκληρου του μοντέλου. Επίσης, ο αλγόριθμος που χρησιμοποιούμε, είτε δέντρα αποφάσεων, είτε τυχαία δάση, μπορεί να βελτιώσει ή να υποβαθμίσει τα αποτελέσματα της προσέγγισης ανάλογα με την καταλληλότητα του ως προς την επίλυση του εκάστοτε προβλήματος μάθησης. Μια στοχευμένη επιλογή ανάλογα το πρόβλημα μάθησης και παραμετροποίηση του δυαδικού ταξινομητή είναι εφικτή και μπορεί να βελτιώσει τα αποτελέ-

σματα. Το δύσκολο ωστόσο είναι να κατασκευαστεί ένα μοντέλο που μπορεί καθολικά να πετυχαίνει υψηλές επιδόσεις. Αντίθετα, το δεύτερο στάδιο, που χρησιμοποιεί τις τεχνικές συναθροίσεις για να συνδυάσει τις προβλέψεις των διαδίκων ταξινομικών του πρώτου σταδίου, δεν επηρεάζει σε μεγάλο βαθμό τα αποτελέσματα. Αρκεί λοιπόν η μέθοδο συναθροίσεις να είναι απλή και να ακολουθει σωστή λογική πορεία

Παράλληλα, κάναμε διεξοδική ανάλυση ώστε να κατανοήσουμε που έγκειται η δυσκολία του προβλήματος. Σίγουρα, οι υψηλές διαστάσεις του προβλήματος, ειδικά στον χώρο των μεταθέσεων αποτελεί παράγοντα δυσκολίας και όπως εντοπίσαμε στα προβλήματα μάθησης που το μεγαλύτερο μέρος των εισόδων αντιστοιχίζεται σε ένα μικρό υποσύνολο του χώρου Ω, η προσέγγιση RPC πετυχαίνει υψηλές αποδόσεις ωρ προς την μετρική Kendall tau distance. Ο σημαντικότερος παράγοντας δυσκολίας αποδείχθηκε όμως η εσωτερική δομή που έχει το πρόβλημα μάθησης. Πιο συγκεκριμένα ποια είναι η κατανομή των εισόδων στο χώρο των γνωρισμάτων και πως γίνεται η αντιστοίχιση στον χώρο των μεταθέσεων Ω. Τα χαρακτηριστικά του προβλήματος μάθησης μας δίνουν μία εικόνα για την δυσκολία του και υπάρχουν μετρικές πάνω στις οποίες μπορούμε να βασιστούμε για να κάνουμε εκτίμηση των επιδόσεων που αναμένουμε αλλά τον μεγαλύτερο ρόλο παίζει η εσωτερική δομή του εκάστοτε προβλήματος.

Chapter 2

Introduction

2.1 Problem statement

In the classic classification problem of Artificial Intelligence, we are given a set of training instances with their corresponding labels and the goal is to produce a model that will be able to predict the correct label given a new unknown input instance. We can think of instances as data points in the features' space. The classification model implements an approximating mapping function from input variables/instances (X) to discrete output variables/labels (Y).

Although classification is a well-defined and significant problem, there are also numerous real-life scenarios where an instance has more than one corresponding label. There are many applications where we have a finite set of labels and we are interested in knowing the relation of each instance with every output variable/label. We want to learn which labels are preferred by a specific instance the most and which labels are the least interesting for that particular instance. What we want to predict is, in a sense, the whole order of preference on the labels given an instance. The order of preference over the labels is called ranking of the labels. Similar to the conventional classification problem, the instances can be thought of as data points in the features' space. However, the model now needs to implement an approximating mapping function from input variables/instances (X) to output variables/rankings (y) over the finite set of labels.

Label ranking is the well-defined problem that models the aforementioned requirements. In label ranking, we have a finite set of labels that we are interested in. The goal in label ranking is to produce a model that, given a set of instances and their corresponding rankings, will be able to receive one new unknown input and predict a corresponding ranking that will be optimal or at least close to optimal. Label ranking extends the limits of the conventional classification and multi-label classification problems. Instead of searching one or several labels as possible answers, in label ranking, we search for a complete ordering of all the available labels.

Due to the practical significance of the problem, a lot of research has been made and is still receiving increasing attention from the machine learning and data mining community. Many different approaches have been used so far. An overview of label ranking algorithms can be found in Vembu and Gärtner in 2010 [1] and Zhou, Liu, Yang, He, and Liu in 2014 [2], where multiple approaches to address the problem are summarised. The state-of-the-art approaches can be categorized into three main groups, the probabilistic approaches, the tree-based approaches and the reduction approaches.

Probabilistic methods express distributions in the features' space using popular mathematical models and maximize the probability of predicting the correct ranking of labels [3, 4].

Tree-based techniques develop variations of famous tree-based algorithms in the conventional classification such as Decision Trees [3] and Random Forests [5] to predict the correct rankings.

Reduction techniques, as the name suggests, decompose complex prediction problems into several simpler subproblems, and then combine the solutions of these subproblems to solve the original problem [6, 7]. In this thesis, we focus on a specific decomposition approach that uses pairwise comparisons and aggregation techniques to solve the label ranking problem.

Fürnkranz and Hüllermeier, in their paper in 2010 [8], proposed a method of using pairwise comparisons to model preferences between pairs of labels directly and then aggregating the predictions of these binary models to predict an output ranking.

By using the term "pairwise comparisons", we refer to the approach of reducing the complex problem of finding a ranking of all the labels into creating models that predict preferences between pairs of labels directly. So, for each pair of labels, we train one model that will be able to predict the order of preference between the two for a given instance. One problem that arises is the possibility of training models that make contrary predictions. Since each model is trained independently of the others, a cycle of pairs may be created where each label is better than the other. Practically, that means that the transitivity property is not guaranteed to be true.

Apart from that problem though, reducing the problem to binary classifiers is not enough. Our goal is to provide a complete ranking given an instance. So, the need for combining the binary models' predictions in the most effective way arises. The term "aggregation techniques" refers exactly to that problem, the problem of trying to combine different sources of related information in order to provide one concrete output. This reduction approach consists of two distinct steps that are independent of one another. This modularity is an interesting property of this specific reduction approach and is considered an important advantage compared to other approaches.

This approach is simple, intuitive and has shown good performance in experimental studies. However, the decomposition of the complex label ranking problem to binary classification problems does not come for free, since the number of ensemble binary models required for the task is quadratic to the number of labels.

2.2 Examples - Applications

Label ranking has found many practical applications in numerous scientific fields due to its generality. Popular applications, in which the target is to learn an exact label preference of an instance in the form of a complete ranking, are found in the fields of bioinformatics and meta-learning, recommender systems and natural language processing.

In the bioinformatics field, one common task is to rank a set of genes according to their expression level (measured by microarray analysis) based on features of their phylogenetic profile. In meta-learning, the goal is to produce a total ranking of the available algorithms according to how suitable each one is for a specific dataset given the characteristics of the dataset [2].

One particular scenario with much entrepreneurial interest is the creation of recommender systems. In such systems, the goal is to recommend products to customers with a specific order. Personalization is the art of offering tailored content to make engagement with customers more relevant. Ideally, in an electronic newspaper, the articles should be presented to each customer in his order of preference. The most interesting ones should be on top and easily clickable. This improves the customers' experience, prolongs the time spent on the platform, and thus boosts the revenue of a company. To do that kind of prediction we make the safe assumption that people of similar characteristics will like or dislike similar products.

One concrete example would be the recommender system of Netflix. The system would create a small list of movies and ask users to rank them from favorite to least favorite. Given that they already have some data about the users' characteristics, such as gender, age, country of residence, and other relevant information, they can use label ranking to their advantage.

Interestingly, in many of the above cases, it is hard to quantify preferences. For example, for collecting training data in recommender systems, users are better at making qualitative comparisons, i.e. a statement of the form "I prefer X over Y", rather than quantitative comparisons, i.e. a statement of the form "I give X a preference score U and Y a preference score W" where U, W are real numbers. The fact that label ranking predicts rankings of qualitative comparisons makes the problem more general.

Furthermore, given that rankings are qualitative seems to suggest that qualitative approaches like pairwise comparisons are more natural and thus better suited for this problem. In contrast, some approaches use utility functions, which means functions that assign a real-number score to each label and then sort the labels based on this score. Constraint classification belongs to this

approach [6]. However, this approach is not equivalent to probabilistic approaches, random forests techniques and pairwise comparisons as experimental studies suggest [8].

2.3 Motivation - Contribution

Due to the applications of this particular problem and the importance of preference learning in numerous scenarios, we are motivated to study the label ranking problem in depth. Inspired by the work of Fürnkranz and Hüllermeier on solving label ranking by learning pairwise preferences and the extraordinary results that he achieved almost 10 years ago, we decided to revisit this methodology. Through an extensive literature review of the approaches used on label ranking and the artificial intelligence field in general, we proposed ways that improve this approach's results.

In the current work, we aspire not only to achieve state-of-the-art results but also to gain a good understanding of the problem, i.e. to get a grasp of what works best, what does not work as well, and, most importantly, why some approaches or ideas work better than others.

We aim to achieve good comprehension of the complexity of the problem. In the current research, different classification models are used to make predictions based on pairwise comparisons. We compare their performances and analyze the reasons for which some models are superior to others. We also experiment with a range of alternative aggregation techniques based on the binary preferences predictions of the models. The aggregation techniques are different in terms of complexity and strictness. Diversity of strategies is being used, e.g. some aggregation techniques take advantage of the distribution of the training data to improve their results whereas others keep simple voting schemas to predict a solution. To test the performance of our models we use the most popular datasets in this field and to quantify the loss between our prediction and the optimal solution, we use the Kendall tau coefficient, a popular metric for rankings. Every characteristic of a dataset, e.g. the number of features, instances, labels, and the distribution of the instances in the input space, are parameters that affect the performance of our models. We provide an in-depth analysis of the characteristics for each dataset and find patterns between the values of the characteristics and the performance levels of our models. Our goal is to obtain intuition on the problem.

Hopefully, this thesis will spark interest in the label ranking problem and suggest a way of thinking. We want to inspire readers to become aware of the possibilities of label ranking, get informed on what makes the problem hard, and find innovative ways to tackle these difficulties.

2.4 Chapters Outline

In this chapter, we made a brief introduction, gave a quick description of the problem, emphasized its importance and the consequences of it. The outline for the rest of the thesis is given below.

Theoretical Background In chapter 3, we introduce the theoretical framework of our work. We present the mathematical of permutations, which is of high importance for label ranking. We explain the concept of distances between the rankings. We also discuss the popular rank aggregation methods that are used in the following chapters and are closely related to the aggregation by pairwise preferences techniques.

Label Ranking Problem and Related Work In chapter 4, we give a formal mathematical definition for the label ranking problem. We also describe the most popular approaches in recent bibliography. We start with probabilistic approaches. We continue with tree-based approaches, which lately have showed tremendous potential. Lastly, we focus on reduction approaches and most specifically the ranking by pairwise comparison (RPC) approach. RPC approach will serve as the basic inspiration for the models that follow.

Datasets' Analysis In chapter 5, we conduct experimental evaluation of the pairwise comparisons approach. We evaluate our models using measures and observe patterns that explain why some approaches are better than others. We also make in depth analysis on the most popular datasets in the bibliography.

Implementation - Comparisons - Results In chapter 6, we conduct a comparative analysis for the different stages of the ranking by pairwise comparison approach. We experiment with popular learning algorithms for the first stage of the RPC approach. We use existing aggregation techniques and propose innovative aggregation techniques. We derive to conclusions about the different approaches and describe under which circumstances and ther reasons why in these circumstances particular models outperform others.

Closing remarks In chapter 7, we refer to the most important conclusion of this thesis. We also mention ideas for future work.

Chapter 3

Theoretical Background

In this chapter, we present the theoretical background necessary for understanding the label ranking problem and the analysis that the following chapters provide. We start with a brief introduction to the conventional classification problem and its variations. We continue with the ranking data structure which is the mathematical tool for expressing preference and has a critical role in the label ranking problem. We also introduce similarity metrics for rankings, tools that will be useful for evaluating the predictions of our models in the following chapters. Lastly, we describe the fundamentals of rank aggregation, which is a key concept in this work.

3.1 Classification

In statistics and computer engineering, classification is the problem of identifying to which of a set of categories or labels an observation belongs to. Classic example is identifying whether an email should go to spam or not. An algorithm that implements classification, especially in a concrete implementation, is known as a classifier. The term ''classifier" sometimes also refers to the mathematical function, implemented by a classification algorithm, that maps input data to a category. The spam filter is an example of classifier.

Classification falls under the category of Supervised Learning (SL). Supervised Learning is the machine learning task of learning a function that maps an input to an output based on example input-output pairs. The goal of the supervised learning algorithm is to infer a function from a set of training instances. Each training instance typically consists of an input object and a desired output value. Typically, the input object is a vector of numerical values that express different features of the training instance.

Classification is a very important problem in the field of Artificial Intelligence. The core of the artificial intelligence problems has to do with learning distinguishing different classes of data. Classification problems come in many different variations, but the main purpose is to match learning instances (vectors with numeric values usually) to different classes or labels.

There are many variation of the classification problem that belong to the supervised learning problems category. Multiclass classification, multilabel classification and multilabel ranking are among the most interesting in the context of this thesis, since label ranking shares similarities with all of them.

- In binary classification, each training instance is associated with one label between a group of exactly two labels. This is the simplest classification problem and we will use it at the first stage of the ranking by pairwise preferences approach (RPC) in the next sections.

- In multiclass classification, each training instance is associated with a single label from a set of disjoint labels. More information on this problem can be found in [9, 10, 11].

- In multilabel classification, each training instance is associated with a subset of labels, usually more than one. This splits the disjoint set of labels into groups of relevant and irrelevant labels with respect to a query instance. Multiclass classification can be considered special case of

multilabel classification, where each subset contains only one element. More information on this problem can be found in [12, 13, 14].

- In label ranking, each training instance is associated with a complete permutation from the set of disjoint labels. A permutation is used as a tool to express priority or preference relation between two labels. Also note that depending on whether the training data are associated with complete or incomplete rankings, we distinguish two versions of label ranking, complete and incomplete accordingly.

The different types of classification problems can be visually represented with 3.1 [2].

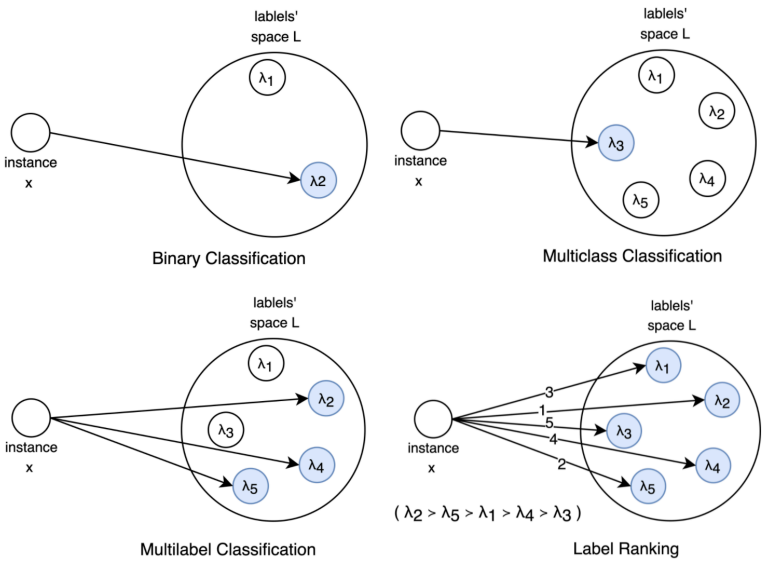


Figure 3.1: Classification variations

Label ranking can be viewed as an extension of multiclass classification and mutlilabel classification can be formalised in terms of label ranking. Mutliclass classification specifically can thought of as predicting the top preferred label of the ranking as the prediction of our model. [7, 15]

The label ranking problem is slightly different than most classification problems. Instead of matching each training instance with the desired output label, each instance is matched with a permutation of labels. This permutation is a basic tool that helps us express preference between labels. Thus, we shall introduce the tool of permutation in the next section.

3.2 Permutations - Rankings

In this chapter, we present the basic principles of permutations. By abuse of notation, we shall sometimes employ the terms "ranking" and "permutation" synonymously. For extensive analysis on rankings, readers can refer to [16] and to [17].

Theorem. *A linear ordering of the elements of the set* $[n] = \{1, 2, 3, \ldots, n\}$ *is called a permutation.*

If we want to stress the fact that it consists of n entries, the ordering is called an n-permutation. In other words, permutations are ways of listing n objects so that each object gets listed exactly once.

In other words, permutations are ways of listing n objects so that each object gets listed exactly once.

Example. *If* $n = 3$*, then the n-permutations are* $123, 132, 213, 231, 312, 321$.

The triads can be represented as a vector of lenght n.

Proposition. *The number of n-permutations is n!.*

The aforementioned definition is not the only possible one. An alternate and not contradicting definition follows.

Definition. *Let $\pi : [n] \to [n]$ be a bijection. We say that π is a permutation of the set $[n]$.*

For any set A of objects, we denote with S_A or $Sym(A)$ the set of all permutations of A. In the particular case that $A = [n] : S_A = S_n$. Formerly, we said that 34521 was a permutation of length five. Now we can reformulate that sentence by saying that the function $\pi : [5] \to [5]$ defined by $\pi(1) = 3, \pi(2) = 4, \pi(3) = 5, \pi(4) = 2$ and $\pi(5) = 1$ is a permutation of $[5]$. We also make use of the inverse π function as well, defined by $\pi^{-1}(1) = 5, \pi^{-1}(2) = 4, \pi^{-1}(3) = 1, \pi^{-1}(4) = 2$ and $\pi^{-1}(5) = 3$. Practically, function $\pi(i) = j$ answers the question "which number is placed in position i?", whereas $\pi^{-1}(j) = i$ answers the question "in which position is number j placed?"

In the context of this work, the objects used to construct the permutations will be labels, i.e we define a set $L = \{\lambda_1, \lambda_2, \lambda_3, \ldots, \lambda_{n-1}, \lambda_n\}$ of labels we are interested in ordering. Each label is paired with a distinct integer from the set $[n]$, that will serve as identification for the labels. To simplify the notation, we will use the identification to refer to a label. Hence, $\pi(i) = \pi(\lambda_i)$ is the position of λ_i in the ranking.

For any $i, j \in [n]$, we can denote preference concisely with $i \succ_\pi j$. This expression is equivalent to $\pi(i) < \pi(j)$. In both cases, we express the fact that in permutation π

In general, permutations are used to express an ordering over a set of objects. In our case, the set of objects is a set of labels that we aim to rank. These labels can be anything and are related in some way. Usually, they are possible values of a high-level variable of interest. In the example of the recommender system for Netflix, the variable of interest may be the type of movie we aim to recommend and the possible labels may be 'Thriller', 'Sci-fi', 'Comedy' and so on.

A full ranking of the objects assigns a complete ordering to the objects. Permutations can be expressed using vectors. We can distinguish the representations of a ranking in two categories: the rank vectors and the order vectors.

Notation. *A rank vector lists the ranks given to the labels, where "1" denotes the best object and "n" denotes the worst object. It presumes the objects are listed in a prespecified order.*

Notation. *An order vector lists the labels themselves, in order from best to worst.*

Abstractly, the objects themselves will be identified with integers. There needs to be an a priori identification of the objects with the integers for both rank vectors and order vectors so that both rankings and orderings will be permutations of the first m integers. It is easy to confuse rank vectors and order vectors. To alleviate some of the confusion we provide the following example.

Example. *Suppose we have a set of types of movies $L = \{$Thriller, Sci-fi, Comedy, Adventure$\}$, with $n = 4$. An individual can rank these labels based on personal preferences. His preferences are expressed using a permutation π. We make an a priori identification of the objects with the integers. The basic idea is to match each integer from $1, \ldots, n$ with a label.*

Let's assume that the listed order is: $[Thriller, Sci - fi, Comedy, Adventure]$. In this case, we actually assign number 1 to Thriller, 2 to Sci-fi, 3 to Comedy and 4 to Adventure.

Let's assume that individual A has the following ranking for set L:

$$Comedy \succ_\pi Sci - fi \succ_\pi Adventure \succ_\pi Thriller$$

so the individual prefers Comedy films the most, then 2nd best are Sci-fi films, followed by Adventure films and, least preferable are Thriller films.

The rank vector would denote the position each label has on our ranking. In this example, given the aforementioned listed order, the rank vector is [4, 2, 1, 3].

The order vector would denote the assigned number for each position. In this example, the order vector is [3, 2, 4, 1].

The two vectors have the following interesting property:

Proposition. *The rank vector is equal to the vector of indices that would sort the order vector. Similarly, the order vector is equal to the vector of indices that would sort the rank vector. We make use of this property in the experimental - programming part of our work.*

So far, we assumed that the permutation function π orders all the objects. The rankings that contain the whole list of inputs are called complete rankings. Nevertheless, there are numerous examples of applications where it may not be possible to rank the whole set of labels. This could happen either because it is impossible to obtain the whole ranking or because it is financially inefficient or impractical to do so. In this cases, we provide a complete ranking of a well-defined subset of the label set L. We refer to these mathematical tools as incomplete rankings. The definition is the following:

Definition. *An incomplete ranking ν is an element of W(Sm), the set of all subsets of Sm.*

For an incomplete ranking to exist, the subset of choice should at least contain two elements. We should remove from consideration the empty set as well as the singleton sets, i.e the set that contain only one element.

3.3 Distance Measures

To train our classification models, there is the need of defining a loss function that can quantify the difference between the real ranking and the predicted ranking, i.e. the distance between them. In binary classification, a 0/1 loss function is sufficient. When it comes to permutations though, a more descriptive measure is required.

The concept of distance between permutations is not as intuitive as it is between numerical values. Many metrics can be used to quantify the difference between two permutations. We will present the most common distance measures between permutations. More information can be found in [18]. A standard reference is the "Rank Correlation Methods" by Kendall [19].

3.3.1 Hamming Distance

Let π and σ be permutations in S_n, with the interpretation that $\pi(i)$ and $\sigma(i)$ are the ranks, i.e. the positions in the ranking, assigned to label i by π and σ respectively. The hamming distance is defined as follows:

$$H(\pi, \sigma) = \#i : \pi(i)! = \sigma(i)$$

.

Put simply, this measure counts the number of labels which have different rank between the two permutations. Despite its simplicity, this metric has the disadvantage of ignoring information about the quality of changes. For example, a swap between the last two objects of a permutation would be equivalent in terms of hamming distance to a swap between the first and the last object, even though intuitively the first swap is less important than the second swap.

3.3.2 Spearman Footrule, Distance and Rank Coefficient

Spearman introduced several intuitive and important distance metrics for measuring similarity between rankings. The Spearman's distance measures were introduced by Charles Spearman in 1904 [20]. The spearman was initially proposed as a measure of disarray is in this paper by Diaconis and Graham [21]. One frequent use is the calculation of a measure of nonparametric association between two permutations [22]. We can distinguish two Spearman distance metrics.

The Spearman's Footrule is defined as follows:

Definition. *Let S_n be the set of all ($n!$) permutations of the first n integers $\{1, \ldots, n\}$. The Spearman's footrule D_F is*

$$D_F(\pi, \sigma) = \sum_{i=1}^{n} |\pi(i) - \sigma(i)|$$

where σ and π are elements of S_n.

The Spearman's footrule measure of disarray is the sum of the absolute values of the difference between the ranks of all the labels.

The Spearman Distance is defined as follows:

Definition. *Let S_n be the set of all ($n!$) permutations of the first n integers $\{1, \ldots, n\}$. The Spearman's distance D_S is*

$$D_S(\pi, \sigma) = \sum_{i=1}^{n} (\pi(i) - \sigma(i))^2$$

where σ and π are elements of S_n.

The Spearman's distance measure is the sum of the squared values of the differences between the ranks. Thus, for Spearman's distance measure, a long length of displacement of a label between the two permutations would have a greater cost in comparison to Spearman's footrule metric. As for the algorithmic complexity of the problem, both Spearman distance measures can be computed in linear time O(m). The efficiency of calculation is an important benefit for using these metrics.

Spearman's rank correlation coefficient is a widely accepted similarity metric for rankings. It is a normalization in the range $[-1, 1]$ of the Spearman's distance and is symbolised by the greek letter τ. It is defined as follows:

$$\rho(\pi, \sigma) = 1 - \frac{6 D_S(\pi, \sigma)}{n(n^2 - 1)}$$

The aforementioned distance measures practically aim to quantify the total element-wise displacement between two permutations. The formulas can be generalised among a center ranking π and multiple rankings $\sigma_1, \ldots, \sigma_m$. For example, the generalised Spearman's distance is defined as follows:

$$D_S(\pi, \sigma_1, \ldots, \sigma_m) = \sum_{i=1}^{m} D_S(\pi_n - \sigma_i) = \sum_{i=1}^{m} \sum_{j=1}^{n} (\pi(j) - \sigma_i(j))^2$$

3.3.3 Kendall tau distance and Kendall tau correlation coefficient

To evaluate the predictive performance of a label ranking algorithm, a suitable evaluation function is necessary. In the statistical literature, several distance measures for rankings have been proposed. To understand the Kendall tau coefficient, we first need to define the concepts of concordant and discordant pairs.

One simple and commonly used measure between two complete permutations π and σ over a set of labels L is the number of discordant label pairs. This metric is also known as the Kendall tau distance [19]. It is defined as follows:

$$D_K(\pi, \sigma) = \#\{(i, j) | \pi(i) > \pi(j) \wedge \sigma(i) < \sigma(j)\} \ ,$$

where $1 \leq i < j \leq m$ and m is the number of labels that L contains ($|L| = m$).

This metric is also known as the Kendall tau distance [19]. It essentially measures the total number of label pairs that are ranked in the opposite order in two rankings. It is an intuitive and easily interpretable measure. The time complexity of computing the Kendall tau distance between two rankings is $O(m \log m)$.

Similarly, we define the number of concordant label pairs:

$$C_K(\pi, \sigma) = \#\{(i, j)|\pi(i) < \pi(j) \wedge \sigma(i) < \sigma(j)\}$$

Given that a pair of labels can either be discordant or concordant in the two permutations the following formula is true:

$$D_K(\pi, \sigma) + C_K(\pi, \sigma) = \frac{\dot{m(m-1)}}{2}$$

The Kendall rank correlation coefficient [19] is one of the most widely used distance measures for rankings. It is also called Kendall tau coefficient and is symbolised by the greek letter τ. The Kendall rank correlation coefficient is a non-parametric test that measures the strength of dependence between two rankings and is defined as follows:

$$\tau(\pi, \sigma) = \frac{C_K(\pi, \sigma) - D_K(\pi, \sigma)}{m(m-1)}$$

or as follows:

$$\tau(\pi, \sigma) = 1 - \frac{4D_K(\pi, \sigma)}{m(m-1)}$$

Kendall's tau coefficient is a normalization of Kendall tau rank distance to the interval [−1, 1]. The Kendall tau coefficient is not a loss function but a correlation measure. It assumes the value 1 if $\pi = \sigma$ and the value −1 if σ is the reversal of π. We shall focus on Kendall's tau as a natural, intuitive, and easily interpretable measure [23] throughout this work, even though other distance measures could of course be used.

Kendall's tau coefficient measures the proportion of the concordant pairs of labels in two rankings. Therefore, this measure can still work with partial rankings, as long as there is at least one pair of labels per instance. When $\tau = 1$, it means that the labels in ranking π and σ are sorted in the same order, while $\tau = -1$ indicates that the labels in these two rankings are sorted in the opposite order. In label ranking, performance comparisons among label ranking algorithms are often based on Kendall's tau coefficient.

As mentioned before, the aforementioned formulas can be generalised among a center ranking π and multiple rankings $\sigma_1, \ldots, \sigma_m$. The generalised Kendall tau distance is defined as follows:

$$D_K(\pi, \sigma_1, \ldots, \sigma_m) = \sum_{i=1}^{m} D_K(\pi_n - \sigma_i)$$

3.3.4 Relation between Kendall tau distance and Spearman's footrule

In a celebrated result, Diaconis and Graham [21] showed that these metrics differ by at most a constant factor (such pair of metrics are said to be equivalent).

Proposition (Diaconis–Graham (DG) inequality). *Let S_n be the set of all $(n!)$ permutations of the first n integers $\{1, \ldots, n\}$. For any two complete permutations $\pi, \sigma \in S_n$ the following inequality is true:*

$$D_K(\pi, \sigma) \leq D_F(\pi, \sigma) \leq 2D_K(\pi, \sigma)$$

This inequality is tight and Spearman footrule is bounded by the value of Kendall tau distance. Therefore, the Spearman footrule practically serves as a good approximation of the Kendall tau distance. We make use of this approximation to solve the rank aggregation problem, which will be analyzed next, efficiently.

Due to their inherent simplicity, the Spearman's rank coefficient and the Kendall tau correlation coefficient measures are major metrics of evaluating the new label ranking algorithm. Nevertheless, these measures fail to take into account some facts in information retrieval or recommender systems,

e.g. we expect errors at the top of the rank should be costlier than errors at the tail of the rank. Additionally, an error on a highly-relevant label should result in a higher penalty than an error on a low- relevant label. For more information about extensions of conventional distance measures that take into account, both label relevance and positional information readers can refer to the paper "Generalized distances between rankings" by R. Kumar and S. Vassilvitskii [24].

3.4 Aggregation of pairwise preferences and Rank Aggregation

The problem of aggregating small pieces of information in order to produce a permutation that comes in agreement with the majority of these pieces is a significant problem with numerous applications. In the context of this work, solving the preference aggregation problem arises at the second stage of the RPC approach, where we have to predict a permutation based on the binary preferences between pairs of labels, that are produced at the first stage.

The aforementioned problem is closely related to finding a permutation that serves as a "mean" between different rankings. The problem of finding one ranking that will serve as a representative given a set of rankings is called rank aggregation and arises in many different scenarios. The notion of optimal ranking does not have a unique interpretation. Instead, this median is measured in terms of a distance metric and the goal is to find a ranking which minimises the total distance among all rankings of the set. Thus, rank aggregation is a minimisation problem. For example, the rank aggregation in terms of the Kendall tau distance aims to find a ranking π that will minimise the generalised distance between the ranking π and a set of rankings $\sigma_1, \ldots, \sigma_m$ using the formula we defined earlier.

We can distinguish different types of rank aggregations. However, in this work, we are going to focus on Kemeny Optimal Aggregation.

Definition (Kemeny Optimal Aggregation). *Suppose we wish to optimise the Kendall tau distance. Given n labels and m full or partial rankings of the candidates, $\{\sigma_1, \ldots, \sigma_m\}$, find a ranking π such that π is a full ranking of the candidates and also minimizes*

$$D_K(\pi, \sigma_1, \ldots, \sigma_m) = \sum_{i=1}^{m} D_K(\pi - \sigma_i)$$

The aggregation obtained by optimising the Kendall tau distance is called Kemeny Optimal Aggregation [25, 26] and in a precise sense corresponds to the geometric median of the input rankings. Kemeny Optimal Aggregation has been studied from a computational perspective. Finding a Kemeny optimal ranking is NP-hard [27] and remains NP-hard even when there are only four input lists to aggregate [28]. This has motivated further studies on finding a ranking that approximately minimizes the Kendall distance with the given input rankings.

3.4.1 Approximation Algorithms for Kendall tau distance

Several approximation algorithms and heuristics have been proposed in [21, 28], as well as [28]. We are going to describe the most important ones.

3.4.2 Pick a permutation

One really simple approximation algorithm, which is known a pick a permutation, is the following:

Definition (Pick-a-permutation). *Given an set of complete rankings $L = \{\pi_1, \ldots, \pi_k\}$, output a permutations $\pi_i \in L$ chosen uniformly at random*

This simple algorithm, given an input set of full rankings, takes one of the input rankings and provides it as a solution. In practice, we can further optimise the pick-a-permutation algorithm, by selecting a permutation that minimizes the cost. However, the randomized version for the analysis is simple and still provides a 2-approximation for the problem.

3.4.3 Solving for Footrule optimal aggregation

In an aforementioned proposition, we showed Kendall distance can be approximated with a factor of 2 from the Spearman distance. An interesting problem to be solved therefore is the footrule optimal aggregation, i.e. the rank aggregation in terms of the Spearman footrule metric. It is defined as follows:

Definition. *Suppose we wish to optimise the Kendall tau distance. Given n labels and m full or partial rankings of the candidates, $\{\sigma_1, \ldots, \sigma_m\}$, find a ranking π such that π is a full ranking of the candidates and also minimizes*

$$D_F(\pi, \sigma_1, \ldots, \sigma_m) = \sum_{i=1}^{m} D_F(\pi - \sigma_i)$$

We make the proposition therefore:

Proposition. *If π is the Kemeny optimal aggregation of the set of full rankings $L = \{\sigma_1, \ldots, \sigma_m\}$ of full rankings and π' is the Spearman footrule optimal aggregation of L, then*

$$D_K(\pi', \sigma_1, \ldots, \sigma_m) < 2D_K(\pi, \sigma_1, \ldots, \sigma_m)$$

That means that the ranking that solves for the footrule optimal aggregation serves also as a 2-approximation solution for Kemeny optimal aggregation problem. Since the optimal aggregation is a good approximation of Kemeny optimal aggregation, we are interested in further investigation.

Footrule optimal aggregation is related to the median of the values in a position vector. For a set of full rankings, the following propositions hold true:

Proposition. *Given the set of full rankings $L = \{\sigma_1, \ldots, \sigma_m\}$, if the median positions of the candidates in the lists form a permutation, then this permutation is a footrule optimal aggregation.*

Footrule optimal aggregation of full rankings can be computed in polynomial time, specifically , the time to find a minimum cost perfect matching in a bipartite graph. The fact that we can compute it efficiently is a significant benefit of this method.

3.4.4 Borda's Method

Borda's method, also known as Borda count, is a famous voting rule in the field of social choice theory [29].

The logic of the algorithm is simple, as described below:

Definition (Borda count). *Given a (complete) ranking σ_i of m labels, the top-label receives m votes, the second-ranked m–1 votes, and so on. Given n rankings σ_1, …, σ_m, the sum of the n votes are computed for each label, and the labels are then ranked according to their total votes.*

Borda's method is a "positional" method, in the sense that it assigns a score corresponding to the positions in which a label appears within each voter's ranking. A primary benefit of positional methods is that they are computationally very easy, since they can be implemented in linear time. They also enjoy the properties called anonymity, neutrality, and consistency in the social choice literature [30].

Nevertheless, it is important to note that Borda's method does not satisfy the so-called Condorcet criterion.

The Condorcet criterion states the following:

Definition. *An electoral system satisfies the Condorcet criterion if it always chooses the Condorcet winner when one exists. The Condorcet winner is the person who would win a two-candidate election against each of the other candidates in a plurality vote.*

For a set of candidates, the Condorcet winner is always the same regardless of the voting system in question, and can be discovered by using pairwise counting on voters' ranked preferences. The terms are named after the 18th-century mathematician and philosopher Marie Jean Antoine Nicolas Caritat, the Marquis de Condorcet.

It is possible to show that no method that assigns a weights to each position and then sorts the results by applying a function to the weights associated with each candidate satisfies Condorcet criterion. Since Borda's method is also one such method, it does not satisfy the Condorcet criterion. Practically, it is possible that a label λi is preferred in most (>50%) rankings comparing to every other label λj (j != i) without being the overall winner of the election (top-label in the ranking).

3.4.5 Extensions of Borda's Method

Borda's method has been proposed since 1770 and is still used widely. Due to a large variety of use cases for this method, many extensions of Borda's method have been proposed since then.

Another similar alternative was proposed in the paper of Cheng. The method proposed is called generalised borda count and has the following definition:

Definition. *If σ_i is an incomplete ranking of $m \leq n$ labels, then the label on rank $i \in \{1, \ldots, m\}$ receives $\frac{(m-i+1)(n+1)}{(m+1)}$ votes, while each missing label receives a vote of $\frac{n+1}{2}$.*

This method is Borda count with a generalized distribution of votes from incomplete rankings.

In some cases, it is useful to have weights on different rankings, that denote a kind of relative importance of a specific ranking. To solve the rank aggregation, we have to take into account these weights. To make our inference procedure amenable to weighted instances, the Borda count principle is replaced by the weighted Borda count, a method proposed in [31].

3.4.6 Feedback arc set on tournaments

The feedback arc set problem on tournaments is closely related to the Rank Aggregation problem.

The definition of the tournament and the feedback arc set problem are the following:

Definition. *A tournament is a directed graph $G = (V, A)$ such that for each pair of vertices i, $j \in V$, either $(i, j) \in A$ or $(j, i) \in A$.*

We fix a ground set $V = 1, \ldots, n$. The definition of the Minimum Feedback Arc Set problem in Tournaments is the following:

Definition. *We are given a tournament $G = (V, A)$. We want to find a permutation π on V minimizing the number of pairs ordered pairs (i, j) such that $i < j$ and $(j, i) \in A$ (backward edges with respect to π).*

We also define the weighted version of the aforementioned problem:

Definition. *We are given weights $w_{ij} \geq 0$ for all ordered $i, j \in V$. We want to find a permutation π on V minimizing $\sum_{i,j:i<_\pi j} w_{ji}$.*

Clearly, the unweighted case can be encoded as a 0/1 weighted case.

In other words, the minimum feedback arc set is the smallest set $A' \subseteq A$ such that $(V, A–A')$ is acyclic. The size of this set is exactly the minimum number of backward edges induced by a linear ordering of V .

A practical example of this problem minimum feedback arc set is the following: Imagine a sports tournament where each player plays against every other player once. How should we rank the players based on these possibly non-transitive (inconsistent) outcomes? The complementary problem to finding a minimum feedback arc set is the maximum acyclic subgraph problem, also known as the linear ordering problem.

3.4.7 Kwiksort

Kwiksort is an algorithm that efficiently solves the feedback arc set problem on unweighted tournaments.

Rank-Aggregation can be cast as a special case of weighted Fas-Tournament, where the objective is to minimize the total weight of backward edges in a linear order of the vertices.

Additionally, it is also possible to transform a weighted tournament to the corresponding unweighted majority tournament $G_w = (V, A_w)$. The definition for the construction of the unweighted majority tournament follows:

Definition. *Given an instance (V, w) of weighted Fas-Tournament, we define the unweighted majority tournament $G_w = (V, A_w)$ as follows: $(i, j) \in A_w$ if $w_i j > w_j i$. If $w_i j = w_j i$, then we decide $(i, j) \in A_w$ or $(j, i) \in A_w$ arbitrarily.*

By reducing the Rank Aggregation problem to the corresponding weighted feedback arc set problem on a tournament and then transforming the weighted tournament to the unweighted majority tournament, we are able to use the Kwiksort algorithm. Thus, we can also provide a 2-approximation solution the Kemeny Optimal Rank Aggregation problem using Kwiksort algorithm.

We will briefly describe the Kwiksort algorithm:

KwikSort$(G = (V, A))$

```
If V = ∅ then return empty-list
Set V_L → ∅, V_R → ∅.
Pick random pivot i ∈ V.

For all vertices j ∈ V \ {i}:
    If (j, i) ∈ A then
        Add j to V_L (place j on left side).
    Else (If (i, j) ∈ A)
        Add j to V_R (place j on right side).

Let G_L = (V_L, A_L) be tournament induced by V_L.
Let G_R = (V_R, A_R) be tournament induced by V_R.

Return order KwikSort(G_L), i, KwikSort(G_R).
  (Concatenation of left recursion, i, and right recursion.)
```

Figure 3.2: Kwiksort

KwikSort is in fact a variation of the classic, well-known quicksort algorithm for ordered data with transitivity violations.

3.4.8 Improved Approximation Ratio for Rank Aggregation

For more information about the Kwiksort algorithm, the mathematical proofs and the related problems can be found in the paper "Aggregating Inconsistent Information" by Nir Ailon, Moses Charikar and Alantha Newman [32].

Notably, they also describe a mixed two step problem algorithm, using the aforementioned Pick a permutation and Kwiksort algorithms, to achieve an improved approximation ratio for Rank Aggregation.

Chapter 4

Label Ranking Problem and Related Work

4.1 Introduction - Overview of Methods

Now that we have the fundamentals of Label Ranking covered, it is necessary to provide a clear description of the problem. In this chapter, we firstly give the mathematical definition of label ranking.

Furthermore, we discuss in more detail, relevant work that has been published on the label ranking problem. For better comprehension, we shall split them in three categories: the probabilistic methods, the tree-based methods and the reduction methods.

More specifically, we are going to focus on the two probabilistic approaches that Cheng et al. introduced. Then, we are going to focus on the decision tree approach by Cheng et al. and on the work of Zhou et al. who used random forest, a tree-based approach that has powerful results both in machine learning in general as well as for label ranking. Finally, as for the reduction methods, we are going to cover the basics of Constraint Classification and Log-Linear models and give an extensive presentation of Hullermeier's work, which was the inspiration for this thesis.

4.2 Mathematical Definition

Given an instance $\boldsymbol{x}$ from an instance space $\mathbb{X}$ and a labels set $L = \{\lambda_1, \ldots, \lambda_m\}$, instead of predicting one or several possible class labels, label ranking tries to associate $\boldsymbol{x}$ with a total order of all class labels. This means that there exists a complete, transitive and asymmetric relation $\succ_{\boldsymbol{x}}$ on L, where $\lambda_i \succ_{\boldsymbol{x}} \lambda_j$ shows that λ_i precedes λ_j in the ranking assigned to $\boldsymbol{x}$, where $1 \leq i, j \leq m$. Since a ranking can be considered as a special type of preference relation, we shall also say that $\lambda_i \succ_{\boldsymbol{x}} \lambda_j$ indicates that λ_i is preferred to λ_j given the instance $\boldsymbol{x}$.

We can identify a ranking $\succ_{\boldsymbol{x}}$ with a permutation $\pi_{\boldsymbol{x}}$ on $\{1, 2, \ldots, m\}$ such that $\pi_{\boldsymbol{x}}(i) = \pi_{\boldsymbol{x}}(\lambda_i)$ is the position of λ_i in the ranking. This permutation encodes the ranking given by

$$\lambda_{\pi_{\boldsymbol{x}}^{-1}(1)} \succ_{\boldsymbol{x}} \lambda_{\pi_{\boldsymbol{x}}^{-1}(2)} \succ_{\boldsymbol{x}} \cdots \succ_{\boldsymbol{x}} \lambda_{\pi_{\boldsymbol{x}}^{-1}(m)}$$

where $\pi_{\boldsymbol{x}}^{-1}(i)$ is the index of the class label at position i in the ranking. The class of permutations is denoted by S_m or Ω and we refer to elements $\pi \in S_m$ as both permutations and rankings.

Our goal in label ranking is to train a model ("label ranker") in the of form of an $\mathbb{X} \to \Omega$. As training data, we use a set of instances $\boldsymbol{x}_i$, where $1 \leq i \leq n$ and n is the total number of instances, together with their associated ranking $\pi_{\boldsymbol{x}_i}$. In the classic version of the problem we assume that the rankings are complete and we shall focus on complete rankings in this work. Nevertheless, the rankings could also be incomplete.

To evaluate the predictive performance of our label ranker, a suitable loss function on Ω is needed. In the statistical literature, several distance measures for rankings have been proposed. We are going to use the most popular one, the Kendall's tau correlation coefficient.

4.3 Probabilistic Methods

By using probabilistic methods, we refer to approaches that are based on statistical models for ranking data. These approaches make assumptions about the conditional probability measure on the class of permutations Ω. Given an input, they predict as a solution a ranking that maximises (or minimises) a formula based on this probability distribution.

4.3.1 Mallows model (IB-M)

One significant probabilistic approach was proposed by Cheng [3] using the Mallows model. The Mallows model is a commonly used distance-based probability model introduced by Mallows. The standard Mallows model belongs to the exponential family and uses two parameters to calculate the probabilities, a center ranking π_0 and a spread parameter θ [23]. The Mallows model is defined by the following formula:

The model assigns the maximum probability to the center ranking π_0 and the larger the Kendall tau distance between a ranking π and the center ranking π_0, the smaller the probability $P(\pi|\theta, \pi_0)$ becomes. The spread parameter θ determines how quickly the probability decreases. For $\theta = 0$, the uniform distribution is obtained, while for $\theta \to \infty$, the distribution converges to the one-point distribution with π_0 having probability 1.

To solve the problem of label ranking, the proposed learner uses the Mallows model in combination with instance-based learning, a popular technique for classification that uses local prediction based on the nearest neighbor estimation principle. Consider a query instance $x \in X$ and let $x_1, \ldots, x_k$ denote the k nearest neighbors of x (according to the Euclidean distance) in the training set, with their corresponding rankings $\pi_1, \ldots, \pi_k \in \Omega$. Assuming that the aforementioned rankings are generated independently of each other by the Mallows model distribution, the learner uses maximum likelihood estimation (MLE) to calculate the parameters (π_0, θ) that maximise the following formula:

Given complete rankings, maximum likelihood estimation is easily computed. In the case of incomplete rankings though, the Mallows model is arguably not ideal, because the probability of a ranking cannot be expressed in closed form. Since we cannot efficiently use maximum likelihood estimation, an approximation alternative is used. More specifically, a variation of the classic EM (expectation-maximization) procedure that replaces the E-step with a maximization step. Starting with an initial π_0 ranking, which is calculated using a method similar to borda-count, for each incomplete ranking, the most probable extension to complete ranking is computed and is used instead. The extended complete rankings are used to calculate a new center ranking π_0 and the whole procedure is iterated until convergance is achieved. The center ranking π_0 is the output of the method.

4.3.2 Placket-Luce model (IB-PL)

Another similar probabilistic approach using the Placket-Luce model was produced the following year by Cheng et al. [4]. Placket-Luce model is a generalization of the well-known Bradley-Terry model. The model assumes a parameterized (conditional) probability distributions on the class of all rankings which is expressed using a parameter vector v. The advantage of this model, comparing to Mallows, is that it is inherently better in handling incomplete rankings.

The Placket-Luce model is a stagewise model, which decomposes the process of generating a ranking of n labels into n sequential stages. At the i-th stage, selecting one from the labels that have not been selected so far, then assigned to the position i according to a probability based on the scores of the unassigned labels. The product of the selection probabilities at all the stages defines the probability of the ranking.

Two methods that depend on the Placket-Luce model have been proposed. The first method uses the idea of instance-based learning. Similar to the aforementioned approach, it fits a locally constant probability model given the k-nearest neighbours observed rankings

The Minorization Maximization algorithm converges to an MLE estimation of the Placket-Luce parameter vector v and the prediction is derived as the ranking with the highest posterior probability

The second approach estimates a global model instead of a local model. The central idea of this method is to define the Placket-Luce parameter vector v as a function of the instance and, consequently, the model parameters can be estimated so as the model fits optimally the training data. The maximization of the log-likelihood can be accomplished by means of gradient-based optimization methods, such as a standard stochastic gradient descent algorithm.

The complexity of these models is polynomial comparing to the Mallows approach which has exponential complexity. Empirical results also suggest that Placket-Luce model performs better, especially for incomplete data.

4.3.3 Benefits and Drawbacks

The probabilistic methods for label ranking allow one to complement predictions by diverse types of statistical information. For example, a benefit of Mallows instance based model is that the spread parameter θ serves as a natural metric of confidence for the model. One drawback of instance-based label ranking approaches with Placket-Luce or Mallows model require store the entire training data in memory. This is a prohibitive factor for source-constrained applications.

4.4 Tree-Based Methods

Tree Based approaches are widely used in machine learning in general because they have high predictive accuracy and they are relatively easy to interpret comparing to the rest state-of-the-art models in machine learning. We can distinguish two popular approaches: the decision tree model, which can be used for predictions and is the main component of tree-based approaches, and the random forest model, which depends on an ensemble of decision trees to make predictions.

4.4.1 Decision Tress (DTR)

Decision tree induction is one of the most extensively studied methods in machine learning for classification and regression and readers can refer to [33] and to [34] for more information.

In his paper [3], Cheng combines Decision Trees with the aforementioned Mallows model to produce a competitive label ranker. To construct a decision tree, the model uses one-dimensional splits for an attribute value and recursively partition the data until a stopping criterion is met. The one-dimensional split is chosen in a manner such that it optimises a criterion of interest. Usually this criterion is an optimisation of a formula that measures the quality of the split and how homogenous the partitions will be after the split. In this label ranker, Cheng et. al only use binary splits and handle numerical attributes. The main modifications of conventional decision tree learning concerns the split criterion at inner nodes, which is described by an optimisation formula based on the Mallows model, and the criterion for stopping the recursive partitioning, which states that the construction of the decision tree stops if the is completely pure or if nodes become too small.

4.4.2 Random Forests (LR-RF)

Random forest is a powerful learning algorithm proposed in Breiman [35] and has been one of the most successful general-purpose algorithms in modern times. It is inspired by the idea of collective intelligence or wisdom of the crowd and combines several randomized decision trees and aggregates their predictions by averaging to achieve improved predictions.

A recent model that uses random forest learning algorithm was proposed by Zhou et al. [5]. The forest consists of a finite number of decision trees, which is a parameter of the algorithm. Each tree is trained independently using a unique data set that is drawn at random, with replacement, from the original data set, and at each node of each tree, a split is performed by maximizing the information

gain over a random subset of the original features. There is a range of criteria to calculate the optimal split point of a node and they highly affect the performance and time complexity of a model. For the proposed algorithm, the top class labels of the rankings associated with each instance serve as supervising information to find the best split and guide the growing of the trees. The process of construction stops until the stopping criterion is met.

After construction, the random forest is used to make predictions given a query instance. At prediction phase, a two step process is implemented. At the first step, we pass the query instance through each tree and reach a leaf node that contains a small group of the training rankings. Using the closest neighbours to the query instance, we aggregate using the generalised Borda count to predict a corresponding rankings. At the second step, we aggregate the predicted rankings of each decision tree to provide a final ranking.

The model produced is efficient in terms of time and space complexity and also achieves high accuracy predictions in terms of Kendall tau coefficient.

4.4.3 Benefits and Drawbacks

A benefit of tree-based approach is that they are easily comprehensible. The decision tree model (LRT) was first to introduce an innovative, tree-based approach bu does not produce competitive results comparing to other state-of-the-art models. The random forest model (LR-RF) is simple yet efficient in terms of time and space complexity and also achieves high accuracy predictions in terms of Kendall tau coefficient. The random forest model (LR-RF) significantly outperforms the decision tree model (LRT) in all cases. This results confirm the conclusion that a ensemble model can produce substantial improvements. LR-RF also demonstrates great performance with partial rankings.

4.5 Reduction Methods

Reduction approaches transform complex prediction problems, such as the label ranking problem, into representations of other already well-researched forms of sub-problems. After using the popular algorithms for solving the sub-problems, they combine their solutions and reconstruct the prediction for the original problem. We are going to describe the constraint classification (CC), log-linear models for label ranking (LL), and ranking by pairwise comparison (RPC).

4.5.1 Constraint Classification (CC) and Log-Linear (LL) models

To describe Constraint Classification and Log-Linear models we first need to define utility functions. The label preferences scenario, a utility function $f_i : X \rightarrow R$ is needed for each of the labels $\lambda_i = 1, \ldots, m$.Here, $f_i(x)$ is the utility assigned to alternative λ_i by instance x. After calculating the utility scores per label, we obtain a ranking for x, by ordering according to these utility scores. For each $\lambda_i \succ_x \lambda_j \Leftrightarrow f_i(x) \geq f_j(x)$. Utility functions provide a simple, natural way to represent preferences by evaluating individual alternatives.

Constraint Classification (CC) is a simple approach that uses utility functions to solve the label ranking problem [6, 11]. In the context of CC, the linear utility functions are define as:

$$f_i(x) = \sum_{k=1}^{n} a_{ik} x_k$$

with a_{ik} being label-specific coefficients and $i = 1, \ldots, m$, where m the number of labels, n the number of instances and l the number of features. A preference relation $\lambda_i \succ_x \lambda_j$ translates into a constraint $f_i(x)$–$f_j(x) > 0 \Leftrightarrow f_j(x)$–$f_i(x) < 0$. Both positive and the negative constraints can be expressed in terms of the sign of an inner product $\langle z, \rangle$, where $= (11...1n, 21...mn)$ is a concatenation of all label-specific coefficients. The vector z is constructed by mapping the original l-dimensional training example into into an $(m \times l)$-dimensional space. After defining the new space,

the corresponding learner tries to find a separating hyperplane in this space, which translates to a suitable vector satisfying all constraints.

The log-linear models also predict rankings using utility functions. They have been proposed in Dekel et al. [7] and utility functions are defined in terms of linear combinations of a set of base ranking functions:

$$f_i(x) = \sum_j a_j h_j(x, \lambda_i)$$

where a base function h_j maps instance/label pairs to real numbers. For the case that base functions are defined as:

$$h_{kj}(x, \lambda) = x_k \; if \; \lambda_i = \lambda_j else 0$$

the approach is essentially equivalent to CC.

4.5.2 Ranking by Pairwise Preferences (RPC)

Reduction approaches decompose the label ranking problem into several simpler binary classification problems, and then the solutions of these classification problems are combined into a predicted ranking. Label ranking by learning pairwise preferences is a state of the art approach that has powerful results despite its simple structure. This scheme will serve as the center of focus for this thesis. Ranking by pairwise comparison (RPC) learns binary models for each pair of labels, and the predictions of these binary models are then aggregated into a ranking. It was proposed by Hullermeier et al. [8] since 2010 and still remains one of the highest quality algorithms on this scientific field. The key idea of this approach is to model the individual preferences directly and the reasoning of this idea comes naturally. The following figure 4.1, shows a visual representation of the two stage process of the RPC approach.

Base Learners At the first stage of the algorithm, for each pair of labels we construct a binary model that is responsible for predicting which label is preferred given an input features vector. We already mentioned that the set of labels is $L = \{\lambda_1, \ldots, \lambda_m\}$. We use M_{ij}, with $1 \leq i < j \leq m$ to denote the base learner which is responsible for learning the preference between the pair (λ_i, λ_j). The total number of models is $\frac{m(m-1)}{2}$. Any popular learning algorithm can be used to model the preferences. Pairwise classification is a popular approach that has been used in areas such as statistics, neural networks and support vectors [36, 37, 38]. The approach is intuitive and usually achieves higher accuracy than one-against-all classification models that split instances of one class as positive and instances of the rest classes as negative. This idea is extended to the label ranking model. Using the transitive property, each ranking is transformed to into $\frac{m(m-1)}{2}$ training examples, each corresponding to a unique base learner. The information of the form $\lambda_a \succ \lambda_b$ is turned into a training example (x, y) for the base learner M_{ij}, where $i = min(a, b), j = max(a, b)$. Moreover, $y = 1$ if $a < b$ and $y = 0$ otherwise.

As a learner, either a classifier or a regressor can be used. In the case of the classifier, the mapping is into the set $0, 1$, where $y = 0$ denotes $\lambda_i \succ \lambda_j$ and $y = 1$ denotes $\lambda_j \succ \lambda_i$. In the case of the regressor, the mapping takes place in the unit interval [0,1]. The number now does not denote strict preference but rather serves as a confidence metric of the preference relation. $y \to 0$ denotes confidence $\lambda_i \succ \lambda_j$ while $y \to 1$ denotes confidence on the preference relation $\lambda_j \succ \lambda_i$. A value of $y = 0.5$ denotes that the preference relations are equally possible.

Aggregation Techniques At classification time, a feature instance is passed to each binary model and the predictions are combined to construct a final full ranking that serves as a prediction. The aggregation technique of the binary preferences could be any algorithm. The original scheme uses a rather simple aggregation approach that is similar to the Borda count method. Each label takes

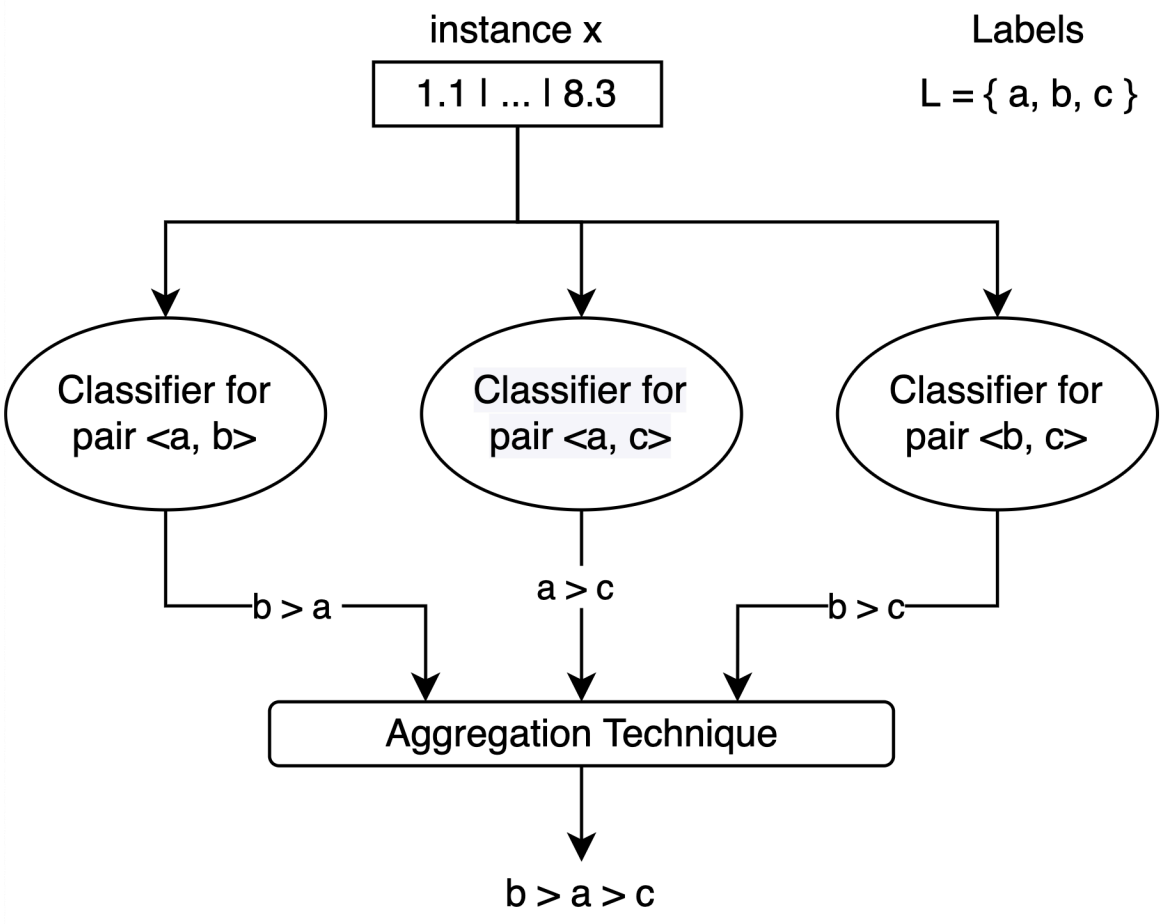


Figure 4.1: RPC approach visualisation

votes equal to the sum of confidences that are in favour of this label. This gives us a quantifiable confidence on each label, more specifically we are 0.8 in favor of label λ_i and 0.2 in favor of label λ_j. Next, we sort the labels in descending based on the number of votes accumulated by each one. Since each pair splits votes of 1 between the two candidates, the total number of votes for all labels is equal to the number of pairs, i.e. $\frac{m(m-1)}{2}$.

4.5.3 Benefits and Drawbacks

In a nutshell, the aforementioned approaches solve label ranking through the use of an ensemble of binary models; the size of this ensemble is linear in the number of labels for LL and CC, but quadratic for RPC. Nevertheless, RPC has experimentally shown better computational efficiency than alternative approaches. Another concern for the ensemble methods is the difficulty of translating theoretical assumptions on the sought "ranking-valued" mapping, which may serve as a proper learning bias, to the corresponding assumptions for classification problems [4]. This problem is apparent in CC approach, where the transformation from rankings to classification strongly exploits the linearity of the underlying utility function without proof that minimizing the classification error is equivalent to maximizing the performance of the label ranking model in terms of the desired loss function on rankings [39].

As for the RPC model, the modularity of the scheme, i.e. that it consist of two independent steps, is an important benefit. For large datasets, where training the models requires a lot of resources and may be difficult to retrain, it is possible to improve the accuracy of the algorithm by experimenting with the aggregation techniques that are used, without the need to retrain the base learners. Another benefit is the abstraction of the method. The overall scheme require neither the use of a

specific algorithm as base learner nor the use of a specific aggregation technique. A significant part of this thesis is dedicated to experimentation with popular learning algorithms, with aggregation alternatives and comparing the results of these models.

One drawback of the complexity of the pairwise comparisons approach is the quadratic number of binary models in relation to number of labels.

Chapter 5

Datasets' Analysis

5.1 Introduction

Label ranking is a learning problem that has numerous practical applications. Due to is generality and significance, there is large interest in the empirical results that an approach like RPC achieves. To conduct the experimental evaluation, we make use of the most popular datasets in recent bibliography.

However, since the label ranking problem has high degree of complexity, further investigation is required, in order to examine the aspects of different learning problems, by studying the charachteristics of the datasets, and provide a reasoning on why the RPC approach achieves these scores.

This chapter is dedicated to the experimental evaluation of the pairwise comparisons approach, as well as other label ranking approaches. In this chapter we make an thorough analysis on the datasets that we used for benchmarking. We focus on the different characteristics of each dataset and comment on the predictive accuracy of the label ranking approaches. Not only will present the performance of the models but, most importantly, we will explain the aspects that make some learning problems more difficult than others and the reasons why some models outperform the rest when particular datasets are encountered. The main objective is to examine the results per dataset, explain why some learning problems (datasets) are more difficult than others and create a system to predict whether a dataset is expected to be difficult.

5.2 Methodology of datasets' analysis

For the experimental evaluations of the datasets, we follow a well structured methodology.

Firstly, we study the procedure of construction of the datasets and we give an overview of the characteristics and simple metrics that will give insights of the differences between the datasets. Our conclusion and explanations will be based upon these metrics, namely the number of instances, labels, features.

We present the performance scores in terms of the kendall tau coefficient for the RPC approach, that uses Random Forest Classifiers (RFC) as the algorithm for the base learners at the first stage and the simple summing of binary predictions (SumBP) as aggregation technique at the second stage. The scores are obtained using five repetitions of ten-fold cross-validation scheme and averaging the scores of each fold.

We also focus on ROC AUC scores metric and unique ranking metric. The ROC AUC scores concern the Random Forest Classifiers (RFC) and are calculated using averaging of the ROC AUC scores of the five repetitions of ten-fold cross-validation scheme. We use diagrams of these metrics to draw conclusions. To make the analysis easier, we split them into three different perfromance groups, according to the performance levels of the RPC approach in terms of kendall tau scores.

Lastly, we use scatter plot diagrams to examine the internal structure of the datasets. We interpret the visualisations of the scatter grids to understand how the datasets of each performance group differ. We also get an intuition of how the internal structure of a dataset affects the performance levels of the label ranking approaches.

5.3 Datasets Overview

The datasets used for evaluation are found in the KEBI Data Repository. These data sets are obtained by transforming multi-class and regression data sets from the UCI repository of machine learning databases and the Statlog collection into label ranking data sets in two different ways:

- (A) For classification data, a naive Bayes classifier is first trained on the complete data set. Afterward, for each instance, all the class labels in the data set are ordered according to their predicted class label probabilities, breaking ties by ranking the class label with lower index first.

- (B) For regression data, some numerical attributes are removed from the set of predictors, and each one is treated as a label. To obtain a ranking, the attributes are standardized and then ordered by size. This type of learning problems will prove more difficult according to the experimental results. Given that the original attributes are correlated, the remaining predictive features will contain information about the ranking thus produced.

To understand better the characteristics of different datasets, we make an extensive analysis.

dataset	instances	features	labels	unique rankings	roc auc	kendall tau
bodyfat (B)	252	7	7	236	0.602±0.1	0.204±0.06
calhousing (B)	20640	4	4	24	0.736±0.051	0.484±0.009
cpu-small (B)	8192	6	5	119	0.73±0.038	0.519±0.014
wisconsin (B)	194	16	16	194	0.767±0.118	0.549±0.034
housing (B)	506	6	6	112	0.909±0.069	0.822±0.026
vehicle (A)	846	18	4	18	0.932±0.051	0.882±0.025
glass (A)	214	9	6	30	0.86±0.164	0.901±0.035
vowel (A)	528	10	11	294	0.945±0.04	0.914±0.016
stock (B)	950	5	5	51	0.96±0.027	0.926±0.014
authorship (A)	841	70	4	17	0.927±0.069	0.935±0.018
wine (A)	178	13	3	5	0.975±0.043	0.941±0.051
iris (A)	150	4	3	5	0.981±0.036	0.965±0.037
pendigits (A)	10992	16	10	2081	0.981±0.011	0.975±0.001
segment (A)	2310	18	7	135	0.985±0.013	0.977±0.005
fried (B)	40768	9	5	120	0.996±0.001	0.991±0.001

Table 5.1: Charachteristics of datasets (ordered by kendall tau score)

In the table 5.1, we can view the number of instances, features, labels. These are characteristics of the datasets that affect models' classification performance. Additionally, we use the number of unique rankings as a simple metric to get a perspective of the diversity of information available for the model to be trained. We also provide ROC AUC metric as a measure of the effectiveness of a random forest classifier in making accurate predictions.

The Area Under the Curve of the Receiver Operating Characteristics Area curve, or simply ROC AUC metric, is popular among the machine learning for quantifying the performance of a classifier. We use a five repetition ten fold cross-validation scheme to train and measure the performance of the random forest clasifiers (RFC). The performance of each classifer is measured using ROC AUC, by averaging we get the ROC AUC scores of each fold and by averaging again we calculate the final ROC AUC score and variance for each dataset. Metrics like classification accuracy lose their meaning on imbalanced datasets. Alternate methods for evaluating predictions are required like ROC AUC metric. It is highly probable that a label a is preferred over the rest of the labels for the majority of the training instances. Practically, if label a precedes label b in the majority of the instances, the class distribution of the binary model responsible for pair of labels $a - b$ would be imbalanced

and thus the ROC AUC metric is preferred over the accuracy score metric or other simple metrics. The ROC AUC metric measures the Area Under the Receiver Operating Characteristic Curve (ROC AUC) from prediction scores. Low ROC AUC scores are equivalent to low-quality binary classifiers. Since the RPC reduction approach is based on the quality of predictions by the classifiers, we expect that models with low ROC AUC scores achieve low-quality ranking predictions.

Lastly, we provide the kendall tau coefficient scores for the predictions of the RPC model that uses random forest for the classifiers of the pairwise comparisons stage and the simple summing operation for the aggregation stage, as proposed in the original work [8]. As we will show in the next chapter, this scheme has the best performance overall. The scores are also obtained using five repetitions of ten fold cross validation scheme and averaging the scores of each fold.

We expect that models with high levels of performance will also have high kendall tau scores, since the final rankings of the RPC models are heavily based on the predictions of the binary classifiers. This expectation is experimentally validated. On the table 5.1, we can view the average ROC AUC scores obtained using the random forest classifier and a five repetition ten-fold cross-validation scheme. The scores of the binary classifiers and the overall predictions have correlation. A slight change in the scores of the pairwise classifiers affects drastically the performance. The ROC AUC score gets values in the range $[0, 1]$ and a classifier that makes random guesses is expected to have a ROC AUC score of 0.5, whereas the Kendall tau coefficient has a value range of $[-1, 1]$ and the value between a random ranking prediction and the true ranking is expected to be 0.

For example, in the bodyfat and calhousing datasets we find that low ROC AUC scores lead to low prediction scores by the RPC scheme. On the contrary, the ROC AUC scores are high in iris and pendigits datasets, thus good predictions are achieved by the classic scheme proposed by Hullermeier.

We should underline that the label ranking problem has high complexity and therefore it is not possible to find a precise relation between the aforementioned metrics and the scores we are expected to obtain, since all the predictions are based on the actual distribution of the training information. For example, calhousing dataset has a similar ROC AUC score with cpu-small dataset but inferior prediction score. Similarly, although the glass dataset has inferior scores to housing dataset, it achieves higher kendall tau quality predictions. The numerical values of the aforementioned metrics serve as an indicator but are not enough to express a strict relation. The performance of the model depends highly on the number of the instances in the features' space and the dimensions of the rankings' space, on how instances are distributed in the features' space and on how the instances are mapped on the rankings' space. The number of features also plays an important role. However, these metrics can give us a good approximation of how well a model is expected to behave. They are measurable aspects of each dataset that can help us arrive to useful conclusion.

5.4 Performance in Bibliography

In chapter 4, we discussed different approaches for solving the label ranking problem. In table 5.2, we provide the type of each dataset in combination with the prediction scores of the methods already described in chapter 3. These results where calculated in [5]. It is important to point out that the RPC scheme in this table uses an SVC classifier for the binary models, which is not the best possible option for the binary model stage, as we will prove in the next chapter. The scores also differ from the ones we showcase in 5.1.

According to the experimental results, we notice that type B datasets appear to be harder than type A datasets. As we described, type B datasets come from regression data. Given that the original attributes are correlated, the remaining predictive features will contain information about the ranking thus produced.

The RPC model, due to its simple yet natrual structure, has really competitive results. Looking at datasets like fried and wisconsin, it has the highest performance. Furthermore, the scores are close to the highest performing model for the rest of datasets as well.

datasets	CC	LL	RPC	IB-M	IB-PL	LRT	LR-RF
authorship (A)	0.920	0.657	0.910	**0.936**	**0.936**	0.882	0.913
bodyfat (B)	0.281	0.266	**0.285**	0.229	0.230	0.117	0.185
calhousing (B)	0.250	0.223	0.243	0.344	0.326	0.324	**0.367**
cpu-small (B)	0.475	0.419	0.450	0.496	0.495	0.447	**0.515**
fried (B)	**0.999**	0.989	**0.999**	0.900	0.894	0.890	0.926
glass (A)	0.846	0.818	0.882	0.842	0.841	0.883	**0.888**
housing (B)	0.660	0.626	0.671	0.736	0.711	**0.797**	0.792
iris (A)	0.836	0.818	0.889	0.925	0.960	0.947	**0.966**
pendigits (A)	0.903	0.814	0.932	**0.941**	0.939	0.935	0.939
segment (A)	0.914	0.810	0.934	0.802	0.950	0.949	**0.961**
stock (B)	0.737	0.696	0.777	**0.925**	0.922	0.895	0.922
vehicle (A)	0.855	0.770	0.854	0.855	0.859	0.827	**0.860**
vowel (A)	0.623	0.601	0.647	**0.882**	0.851	0.794	0.867
wine (A)	0.933	0.942	0.921	0.944	0.947	0.882	**0.953**
wisconsin (B)	0.629	0.542	**0.633**	0.501	0.479	0.343	0.478

Table 5.2: Kendall tau scores for different label ranking models

From the table, the three models that have the best results are RPC, IB-M and LR-RF. It is really interesting that each of the three models belongs to a different category of approach, namely reduction approach, probabilistic approach and tree-based approach accordingly. This suggest that all approaches are equally important and competitive results can be obtained by innovating and creating new approaches.

The RPC approach has the best performance in bodyfat and wisconsin, which are two of the hardest datasets. They are really abstract due to the lack of training instances and the large amount of labels that increases the dimensionality of the problem. This suggests that RPC thrives in lack of data comparing to other models. Furthermore, the scores are close to the highest performing model for the rest of datasets as well.

LL and LRT have the lowest levels of performance among all models. CC has similar but inferior results to RPC. The same happens between the results of IB-PL and IB-M.

Lastly, we should point out that the levels of scores for all approaches are similar per dataset. In other words, it is apparent that some learning problems are more difficult than others. We make interpretations of the aforementioned metrics to explain why it is hard to achieve good scores in specific datasets and to create a system of recognising when a dataset is expected to have low scores given the values of the metrics.

In the following section, we give an in depth analysis of the datasets. We analyse the characteristics of each dataset, the potential of classification and regressor models to achieve high quality classification per dataset and through comparison we make conclusions about them.

5.5 Datasets analysis using diagrams

The number of unique rankings as well as the median and variance of ROC AUC scores may be a bit misleading. They do not grasp the whole information about the dataset. For one, we don't know how close or far apart the unique rankings are and, secondly, we do not know how easily separable the datasets are.

In the following figures (5.1, 5.2, 5.5), for each dataset we provide a visual representation of the distribution of ROC AUC scores for the binary classifiers. This visualisation help us distinguish in which datasets the learning problem is harder and how wide the range of roc auc scores is per dataset. We also visualize the distribution of unique rankings. This is important to know how much variety does a model has in its training data to make good use of it.

The diagrams in the left column are histograms of the ROC AUC scores of the classifiers. We run a five repetitions of a ten-fold cross-validation, each training $\frac{n}{n-1}$ models, thus the number of scores is $25n(n-1)$. This visualization is useful as it provides a sense of the quality of classifiers that is more representative than the median and standard deviation.

The diagrams in the right column show, in sorted order, the number of appearances for each unique ranking in each dataset. The number of appearances of a ranking refers to the number of instances that have the ranking as target. As the curve approaches the form of a horizontal line, i.e. it does not have big fluctuations, the dataset is considered more balanced and homogenous. Respectively, if the curve shows big fluctuations, the dataset is considered more unbalanced, i.e. only a small portion of the rankings space is used as targets. The diagrams present visually how uniform is the distibution in target space Ω. Note that the diagrams placed on the right column show the number appearances including the rankings that have zero appearances (except for the wisconsin dataset).

The datasets are split into three different categories of accuracy in terms of Kendall tau score for the sake of making the analysis easier. The grouping is depicted visually in the table of datasets' characteristics 5.1. Distinguishing these different groups in terms of ROC AUC scores would result to the same split of groups since there is correlation between ROC AUC scores and kendall tau scores.

5.5.1 Low Scores Datasets

In bodyfat, calhousing, cpu-small and wisconsin, our models achieve the poorest performances overall in terms of auc scores of the classifiers and kendall tau distances from the rankings.

Calhousing is an interesting case, where the classifiers are visibly divided into two different score groups. The curve of unique rankings is close to linear shape.

Bodyfat and wisconsin are very similar datasets. The rankings' space is very large comparing to the small number of instances available. Furthermore, the number of apperance of unique rankings is equally distributed, meaning that each ranking appears only a small amount of times and none stands out. This is visible by the right-hand side diagrams. Although this is not always the case, in datasets where the rankings are equally distributed, it is more difficult to make good predictions since there is no small group of rankings that stand out among the rest. In case there was a small group of rankings that appear the majority of the times in the training dataset of a binary classifier / regressor, there is high probability that predicting one ranking out of this group would be correct. Another similarity between the two, is that the range of values of roc auc scores of the models is wide. This happens because of the lack of instances for training. Usually, in machine learning we use large amounts of data in order to capture the underlying distribution (according to the stat). The probability of having a statistical error is big when it uses little number of instances.

As for the cpu-small dataset, the roc auc scores diagram follows a narrow normal distribution around 0.72. Thus the predictions scores are also similarly low. Although the unique rankings curve is exponential-like, the performance of our model is still poor.

Counterintuitevely, cpu-small and calhousing, despite the high number of training instances, have poor performance. This has to do with the inherent difficulty of the training dataset rather than the performance of the RPC approach.

As for the diagram of the unique rankings, we notice that most of the curves have shape similar to the horizontal line. This suggests that the distribution in the rankings' space is uniform. All rankings are equally likely to be predicted, thus the learning problem is more difficult in a sense.

We also observe a direct correspondence between the roc auc score of a model and the kendall scores achieved. Also, in datasets where the roc auc scores are high but the models kendall tau score is not great, we can assume that the problem lies mainly on the aggregation method of the pairwise preferences of our models. We are going to discuss this aspect in depth on the following section. More specifically, in datasets where the range of roc auc scores is wide, we are going to

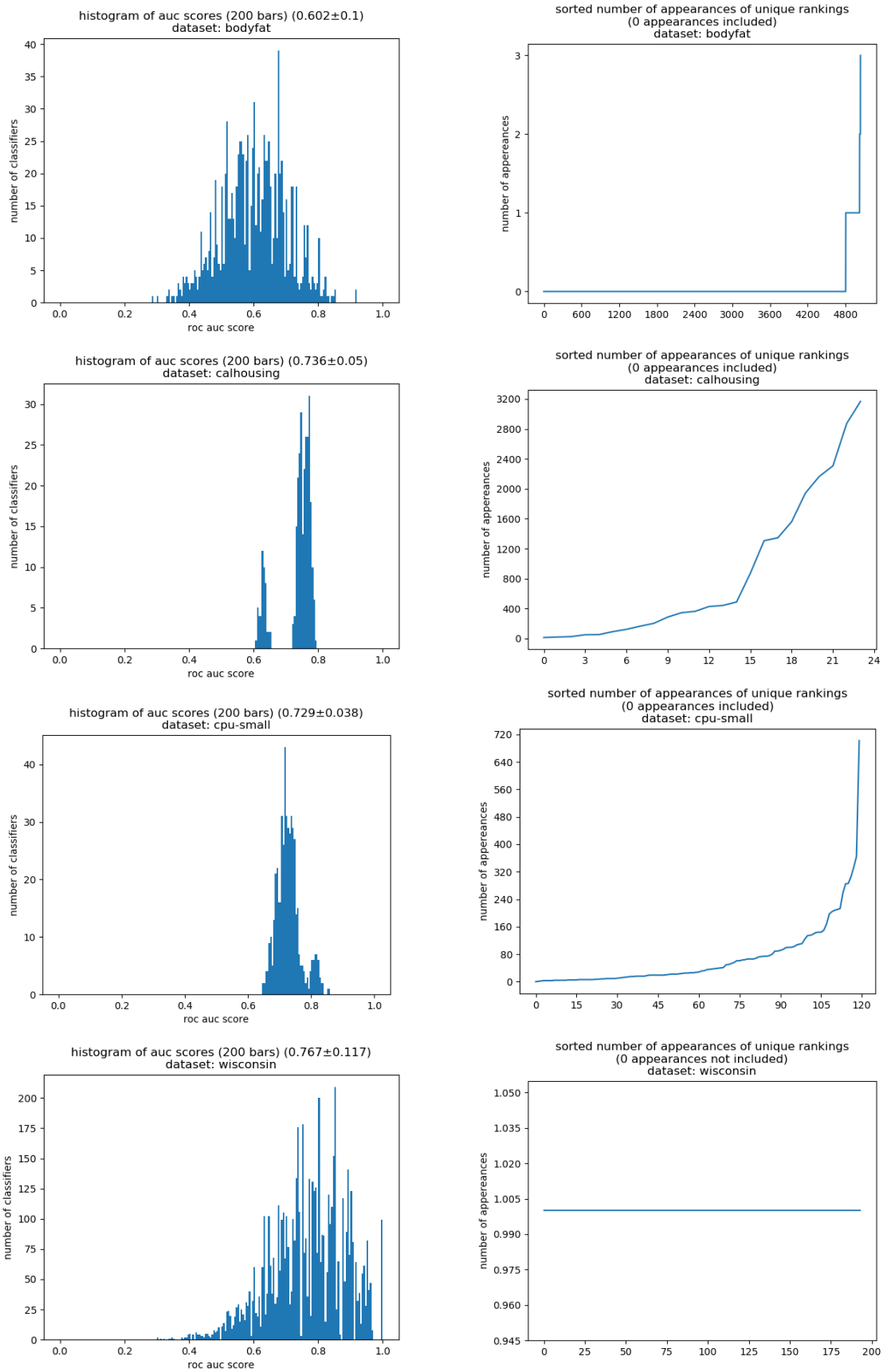


Figure 5.1: Low Scores Datasets

experiment with different aggregation methods that try to take advantage of only the high-quality classifiers/regressors in order to make predictions.

5.5.2 Medium Scores Datasets

In this group of datasets, the classifiers/regressors achieve medium scores in terms of roc auc and the basic pairwise scheme achieves medium kendall tau scores. This group consist of the following datasets: authorship, glass, housing, vehicle, vowel, stock and wine.

We already mentioned that there is important correlation between the two metrics. However, it is interesting to note that the relation is not strict, meaning that higher ROC AUC score does not necessarily mean better performance of the model in general. For example, the glass dataset achieves inferior ROC AUC scores but superior kendall tau scores in comparison to the housing dataset. Looking at the glass diagrams and its standard deviation it is easy to see that the range of scores is wide, hence the good performing regressors can counterbalance the low quality regressors during the aggregation stage. This is not always the case.

As for the appearance of unique rankings diagrams, we notice that most curves have an exponential growth rate and are similar to the exponential curve. This suggests that only a handful of rankings are really prevalent in a dataset, hence practically reducing the size of our search space into a small number of possible target points.

One pattern to be noticed is that low volume datasets (with not many training instances) can be easily tackled in cases where there are few prevalent candidates and the unique rankings diagram is exponential like. In the contrary, comparing with datasets like bodyfat and wisconsin where the distribution is uniform, the prediction scores are inferior. We can also claim that in high volume datasets, the unique rankings diagram is not that important since we have enough information to learn the mapping between the instance / feature space $\mathbb{X}$ and the permutation / ranking space Ω.

The amount of training data compared to the size of the rankings' target space and the inherent difficulty of the mapping function from features' space to the target space are the most important factors that determine the performance of any model against a dataset.

5.5.3 High Scores Datasets

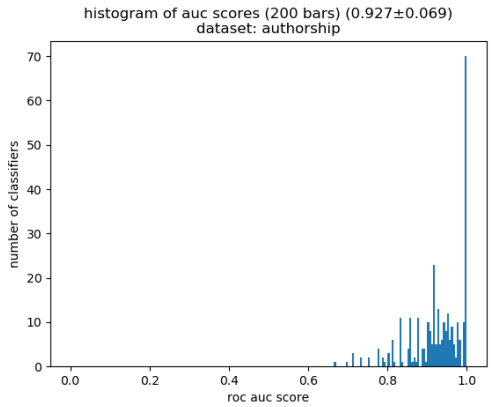
This last group contains the datasets with the best performing base learners and pairwise comparison schemes. Iris, pendigits, segment and fried are in this group. The existing research already performs great in the easy datasets and we expect that experimenting on these datasets will not affect the prediction scores drastically. Since the kendall tau scores for these datasets are so high, they can be considered a bit saturated and we shall focus more on the previous datasets.

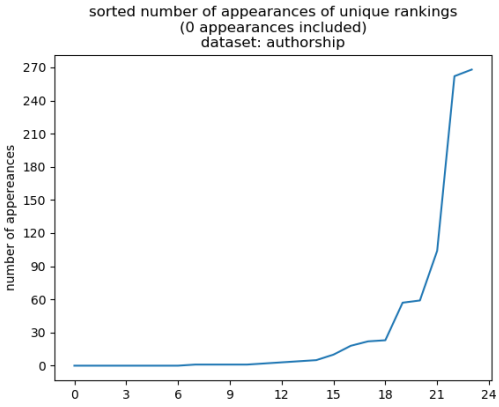
The roc auc scores are close to 1. Also, the standard deviation is really low, thereafter the range of scores for the classifiers / regressors is narrow. Similarly to the previous section, most histograms resemble the exponential curve.

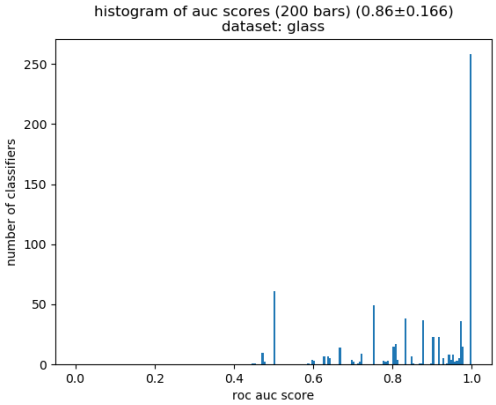
The pendigits and segment datasets have high degree of similarity. The ROC AUC histograms are close to narrow and centered close to one. The unique rankings curve resembles the exponential curve and are really steep. This translates to practically smaller ranking space Ω and thus better results.

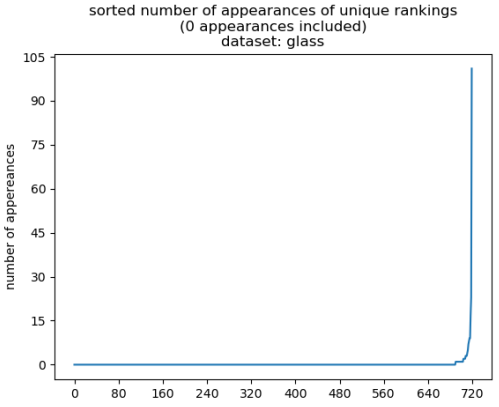
Our models achieve great results also in the fried dataset. These three datasets are actually among the datasets with the highest number of training instances. The scores specifically for the According to 5.2, the reduction approach using RPC scheme has state of the art performance that is superior to other approaches like LR-RF and IB-M to a large extent.

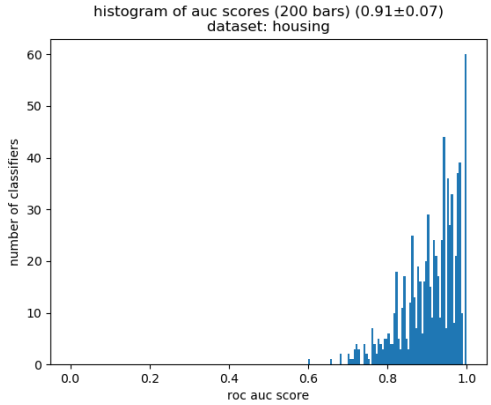
According to the relevant bibliography, similar results are achieved to all research among the aforementioned datasets. The hardness of these datasets is inherent and does not only appear against the pairwise comparisons method that we study in depth here. Knowing the basic characteristics of every dataset can give as useful insight on why a particular approaches can outperform other

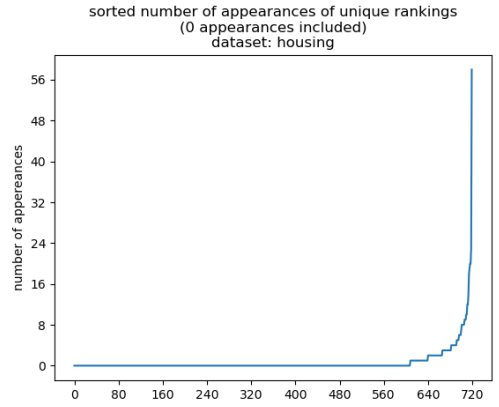

histogram of auc scores (200 bars) (0.927±0.069)
dataset: authorship
number of classifiers
roc auc score

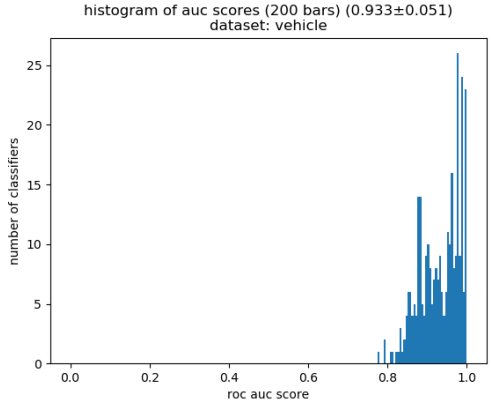

sorted number of appearances of unique rankings
(0 appearances included)
dataset: authorship
number of appereances

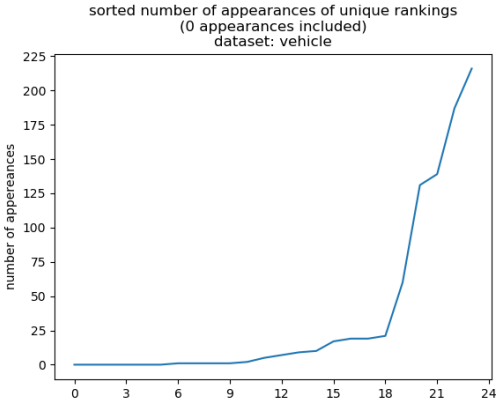

histogram of auc scores (200 bars) (0.86±0.166)
dataset: glass
number of classifiers
roc auc score


sorted number of appearances of unique rankings
(0 appearances included)
dataset: glass
number of appereances


histogram of auc scores (200 bars) (0.91±0.07)
dataset: housing
number of classifiers
roc auc score


sorted number of appearances of unique rankings
(0 appearances included)
dataset: housing
number of appereances


histogram of auc scores (200 bars) (0.933±0.051)
dataset: vehicle
number of classifiers
roc auc score


sorted number of appearances of unique rankings
(0 appearances included)
dataset: vehicle
number of appereances

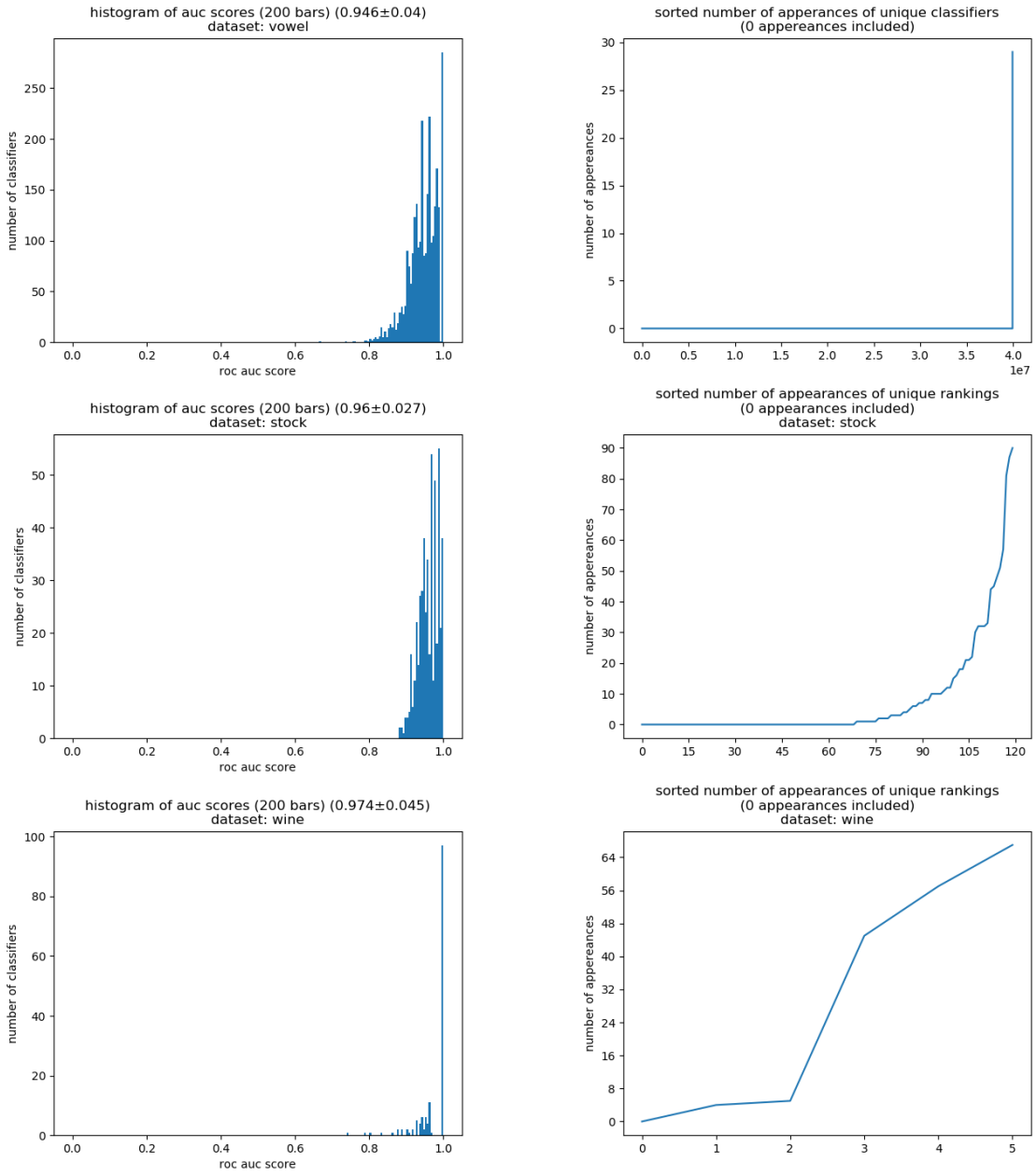


Figure 5.2: Medium Scores Datasets

approaches on specific datasets. In the following chapter, we make use of the dataset analysis to explain why different classifiers or different aggregation methods are better or worse.

5.6 Other Categorisations for Datasets

The categorisation that we offered earlier was based on the kendall tau scores. Based on different parameters though, we find different similarities and potential groupings.

As far as the unique rankings appearances are concerned, we could distinguish three groups:

- Homogenous distribution group: the number of appearances for each ranking is nearly identical. The diagram has similar form to the horizontal line. Bodyfat and wisconsin belong to this group.
- Semi-homogenous distribution group: in this group of datasets the number of appearances of

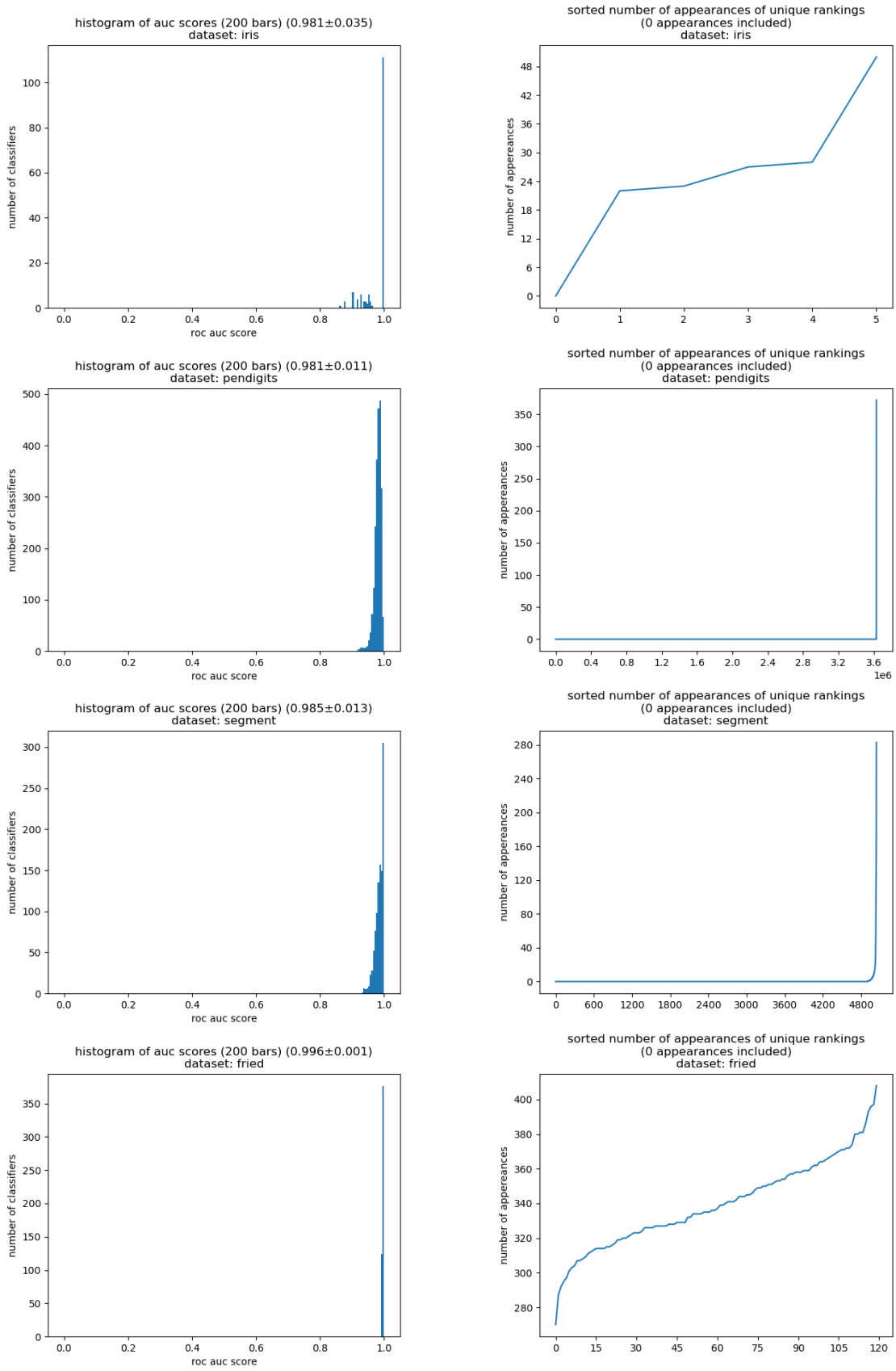
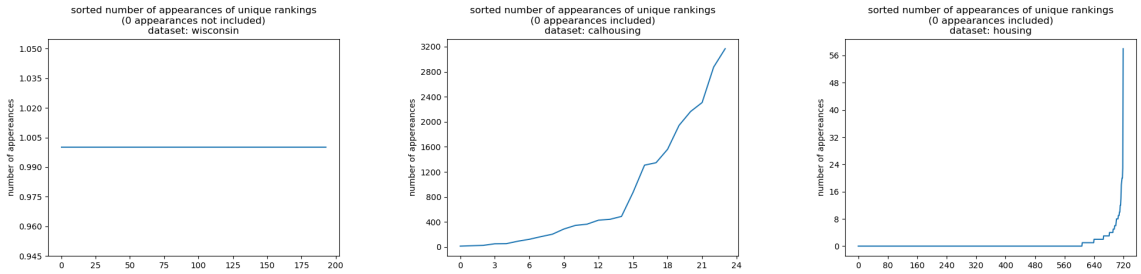
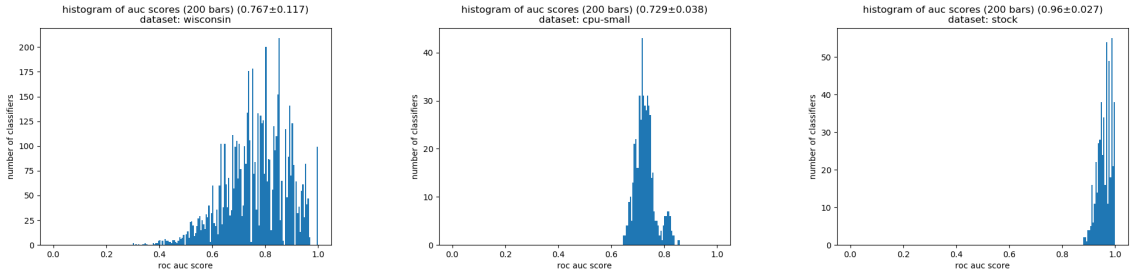


Figure 5.3: High Scores Datasets



(a) Homogenous distribution group (b) Semi-homogenous distribution group (c) Non-homogenous distribution group

Figure 5.4: Split dataset based on shape of unique rankings curve



(a) Uniform-like distribution (b) Normal-like distribution (c) Exponential-like distribution

Figure 5.5: Split dataset based on shape of roc auc scores distribution

unique rankings differ and gradually increments with a steady rate. The diagram has similar form to a linear diagonal line. Calhousing, iris and wine datasets belong to this group.

- Non-homogenous distribution group: this group of datasets is characterised by a majority of rankings that have little to no appearances and a handful of rankings that appear the most. The diagram has an exponential-like form. The rest of the datasets belong to this group.

The unique rankings appearances can be a deceiving metric because it does not contain the whole information. For example, we are not sure whether the unique rankings that show up the most times are close in terms of Kendall tau distance or not. As far as the ROC AUC scores of classifiers are concerned, we could distinguish the following groups in terms of their shapes:

- Uniform-like distribution: the distribution is similar to uniform distribution. This means that classifiers differ a lot and that can have harmful effect on our predictions. Glass and wisconsin belong to this group, due to the fact that their classifiers have varying ROC AUC scores.

- Normal-like distribution: the distribution is similar to normal distribution, with more classifiers achieving a score close to the median. As the achieved ROC AUC scores moves away from this center score, fewer number of classifiers belong to this bucket. Cpu-small and bodyfat datasets belong to this group. Calhousing is a unique case and its' distribution consists of two normal-like distributions.

- Exponential-like distribution (beta-like distribution): Most datasets fall in this category.

There are some dataset where there is huge difference between the unique rankings and the possible rankings based on the labels. These are wisconsin, vowel, pendigits, segment, glass and bodyfat. In contrast, for the datasets of cpu-small, calhousing and fried, the number of unique rankings is almost equal to the possible rankings of appearance.

5.7 Dataset analysis using scatter plots

For further analysis of the datasets we make use of scatter plots. More specifically, we make use of scatter grids. The scatter grid consists of a grid of scatter plots and each scatter plot corresponds to one pair of features. The numeric values of each feature are shown on the y-axes across a single row and on the x-axes across a single column. Each scatter plot contains the geometric representation of instance points on the two dimensional space for the corresponding pair of features that the row and column indicate. The plotted points are colored depending on their corresponding label. We can distinguish two types of scatter grids.

The scatter grids of the first type are binary and only use two colors. Each scatter grid of this type refers to a specific pair $i - j$ of labels and the color of the plotted instance points will be based on the condition of whether label i precedes label j in the corresponding ranking. Therefore, cluster zero contains plotted points where label i precedes label j in the ranking. Respectively, for cluster one, label j precedes label i in the rankings of the plotted points. The diagonal plots of the scatter grids are different. Each diagram of the diagonal represents a univariate distribution plot of the corresponding feature to show the marginal distribution of the data in each column. The smaller the overlap between the appearances of the different groups, the easier it is to distinguish points based on the specific feature of the diagonal. In our training datasets, the features are labeled as $A1, A2, ...$ and the labels have names of the alphabet $a, b, ...$ We may also use number $0, 1, ...$ for labels, to declare the corresponding letter of the alphabet.

For the second type of scatter grid, each cluster numbers of the dataset refers to a unique whole ranking. The number of colors used by the scatter grid in each scatter plot is equal to the amount of unique rankings of the relevant datasets. 5.9 falls into this category. We use the latter diagram in datasets where the number of unique rankings existing is small, and therefore we can easily make conclusions based on the colors of the diagram for the clusters that unique rankings form.

Useful observations can be produced using these visualisations. We maintain the categorisation pattern that we already used in the previous section, which split points according to different range groups of the roc auc metric.

5.7.1 Low scores datasets

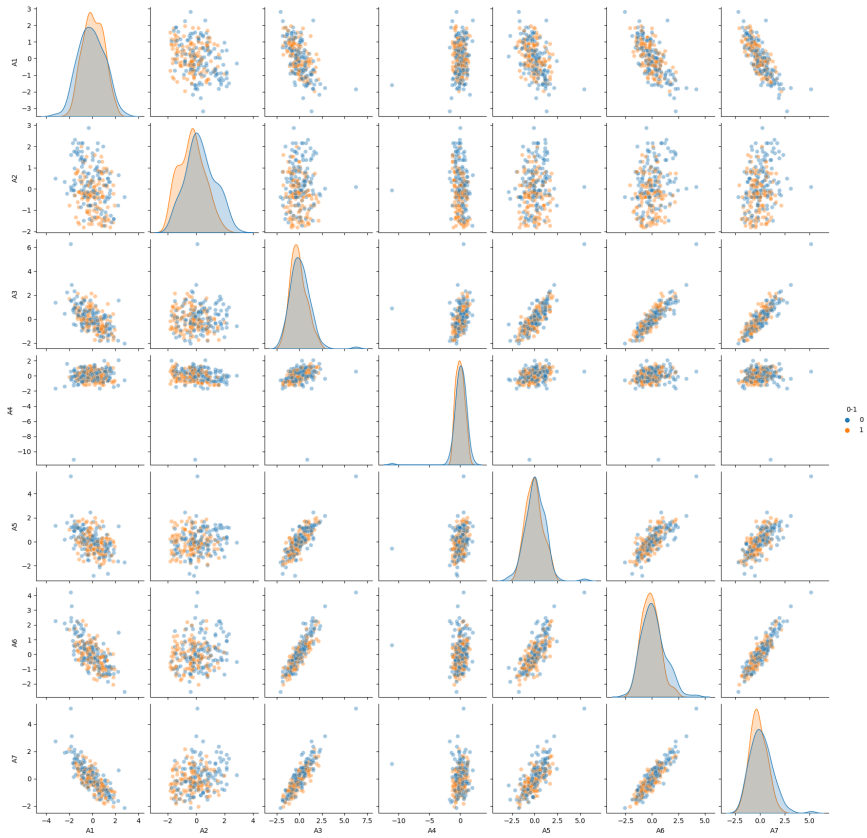
We consider the datasets of this category really important, since there is plenty room for improvement. Using the scatter grids, we make an analysis on the reasons why these datasets have low roc auc scores.

Bodyfat. Bodyfat is a dataset where different labels are hard to seperate. As we see from the scatter grid diagrams 5.6, labels 1 (b) and 2 (c) cannot be divided easily. That is the case for almost all of the pairs for the bodyfat dataset. We can also find some linearly seperable pattern, for example looking at labels 0 and 5, in the scatter plot of A6-A7 features.

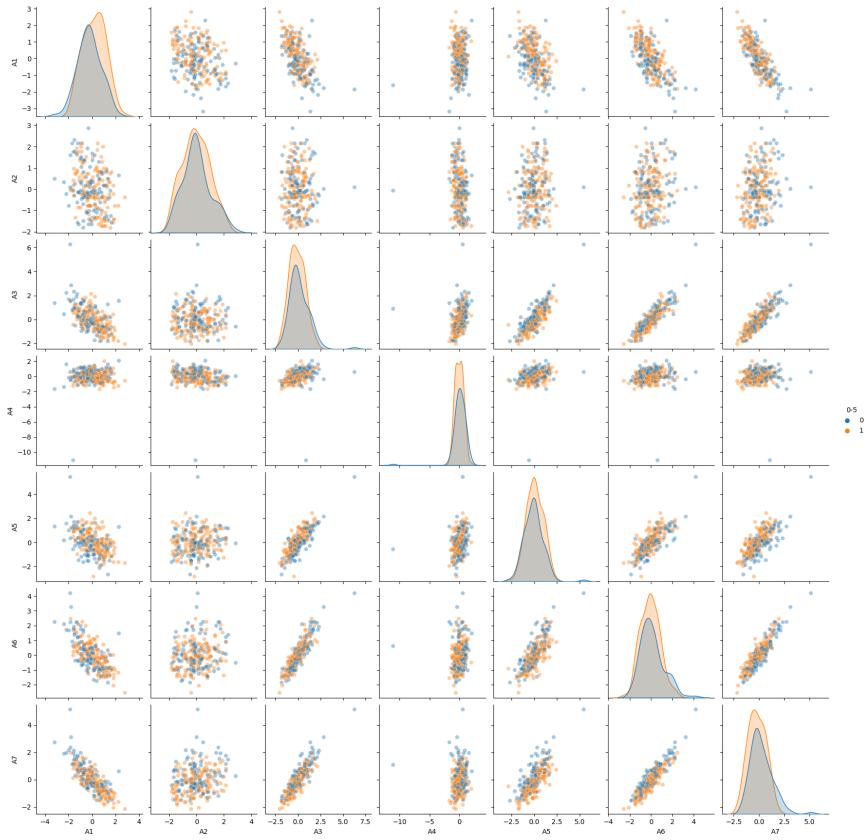
As we see, the datapoints remain the same scatter plots maintain the same shapes but the colors change depending on the features' pair of choice.

By comparing the bodyfat, calhousing and cpu-small datasets, we notice that the original RPC approach achieves poor performance in all of them. Bodyfat actually has the worst performance between the three, while the other two have similar scores. As we see from the scatter plots, the different categories are mixed together, thus making the problem of classification difficult.

Calhousing. About calhousing 5.7, looking at the pair of labels 0 (a) and 1 (b), the two groups are mixed to each other and the clusters are cannot be easily separated. In A2-A3 pair of features we observe mainly blue data points (that represent label 0 preceding label 1) with small high-concentration clusters of orange points. Similarly, for A1-A2 pair of features, we see mainly blue points, with a small orange cluster at the center of the plot.

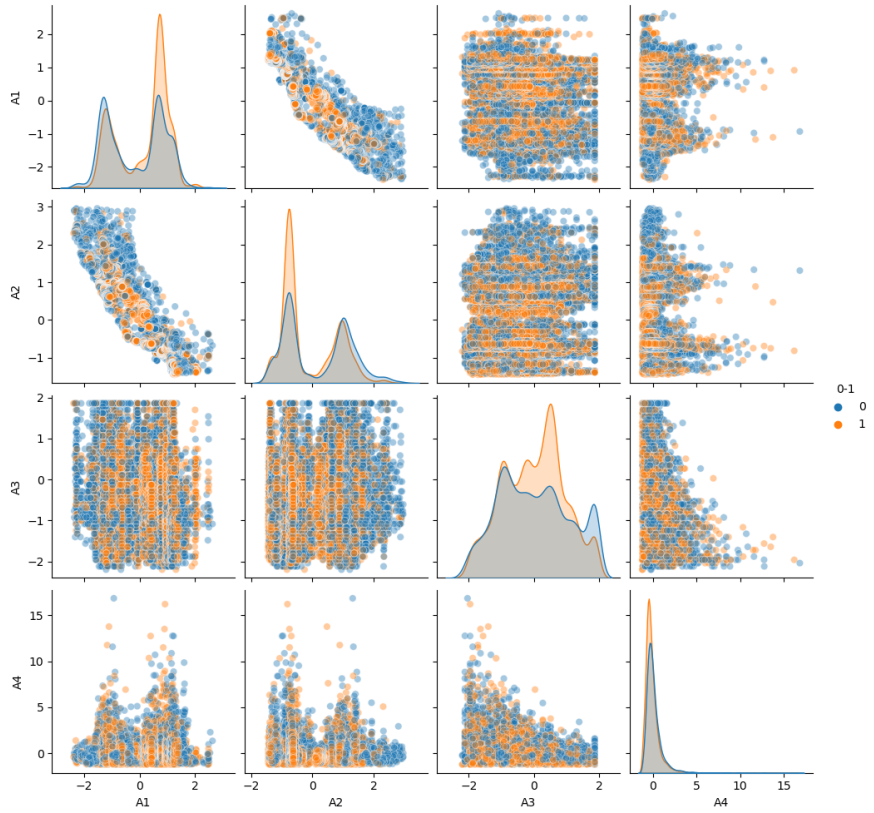


(a) Labels 0 and 1

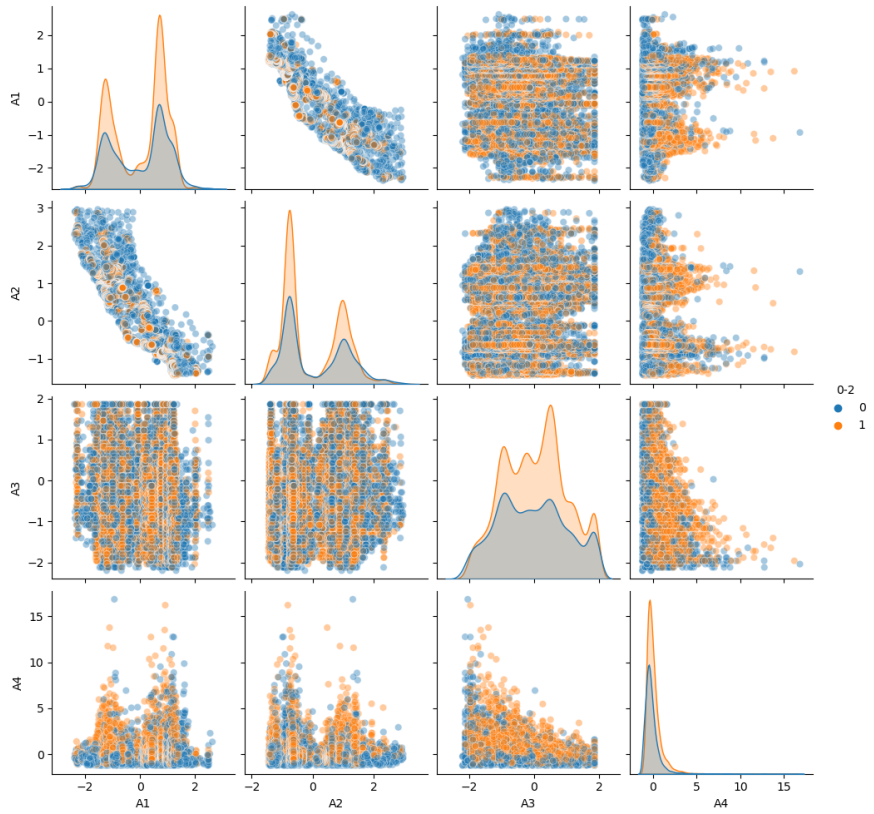


(b) Labels 0 and 5

Figure 5.6: Bodyfat scatter grids for pairs of labels



(a) Labels 0 and 1



(b) Labels 0 and 2

Figure 5.7: Calhousing scatter grids for pairs of labels

In contrast, labels 0 (a) and 2 (c) are more discrete. Focusing on A4 column of the scatter grid, we can clearly see that data points where 2 precedes 0 tend to be at the right side of the diagram. Consequently, we can almost be certain that instances with higher values on A4 feature will have label 2 preceding label 0. Although the dataset is difficult, observations like these shows as that there is a level of order that our system can make good use of in order to facilitate better performance. The rest of the plots follow similar behaviour. We conclude that some features like A4 are especially helpful in solving the problem of classification for the specific dataset. We also expect that the RPC approach will predict the true order between labels 0 and 2 more often than the true order between labels 0 and 1. This helps improve the performance of the model in general.

Cpu-small. About cpu-small dataset, a similar situation is apparent. For pair of labels 0 and 2 (coloring rankings where 0 precedes 2 with orange and blue for the reverse order), we can spot clear trends. For pair of features A3-A6, orange tend to be at the low part of the diagram and blue tend to be at the high part of the big round cluster. For pairs A4-A6 and A5-A6, the cluster is more narrow, following a vertical straight line. The orange instances also tend to be at the bottom part of the diagrams. At the diagonal diagrams, which use univariate distribution plots to show the marginal distribution of the data in each column, we observe big overlap between the orange and the blue distributions. This suggests difficulty in splitting the instances based on that specific feature.

About the bodyfat dataset, looking at all pairs of lables and features, it is difficult to find clear clusters. In most of the diagrams there seems to be one cluster on which two colors emerge. Looking at the A7 column, a sligth trend can be found for splitting the cluster into two different territories. For features A6-A7, a straight line would split the circular shape into un upper orange half and a lower blue half. Nevertheless, for the pairs containing features from A1 to A5, the diagrams contain both orange and blue points without obvious trend for splitting. In contrast to cpu-small and calhousing, there seems to be no trends to make confident splits. This difference in difficulty can also be noticed in the roc auc scores of the models, where cpu-small and calhousing have scores of 0.73 and bodyfat scores of 0.60. The difference is also reflected on the kendall tau scores.

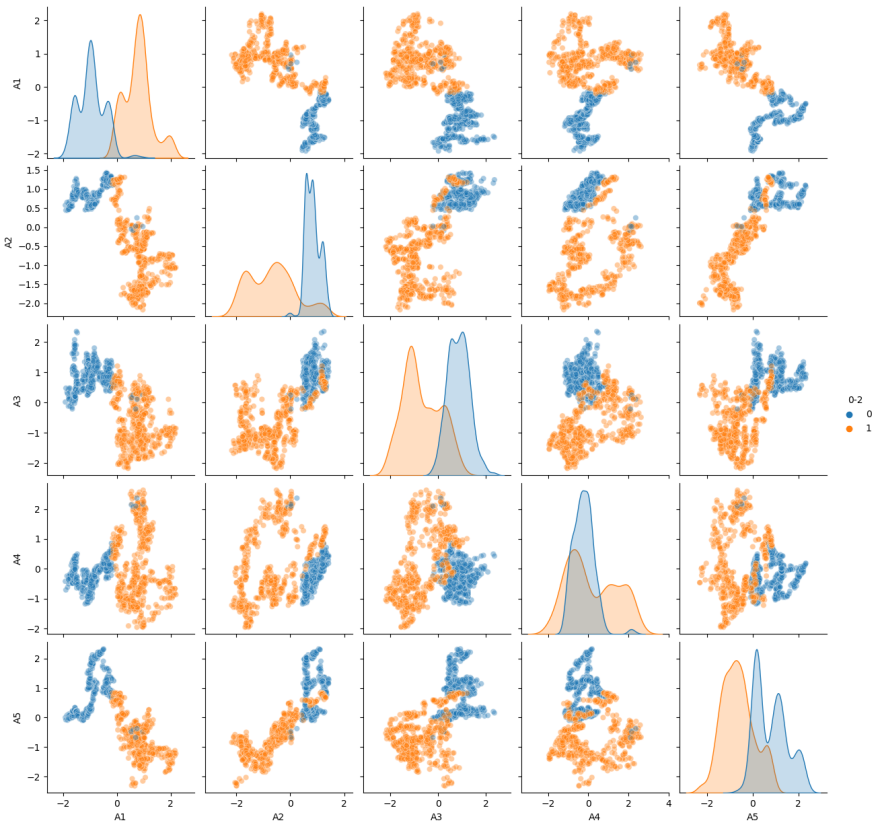
5.7.2 Medium Scores Datasets

In this subsection we are going to focus on the medium scores dataset. The separability between clusters is more clearly depicted comparing to the scatter plots of the previous subsection.

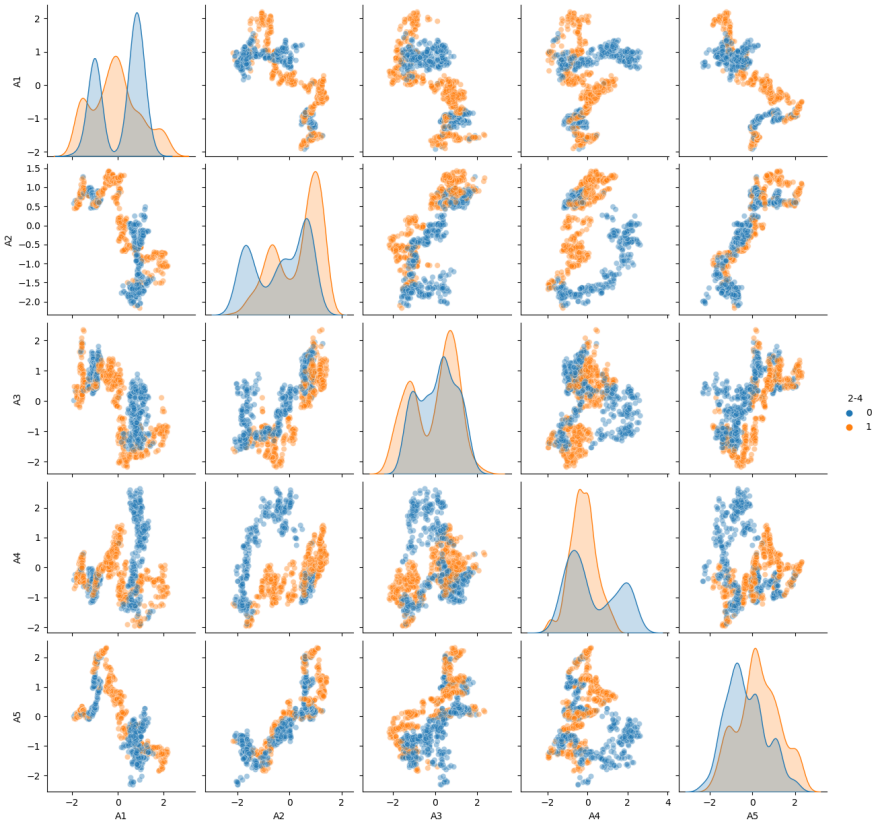
stock About the stock dataset 5.8, the scatter grids for different pairs of labels have the same structure. There are clear clusters to be identified, although there is overlap between the data at different areas. In scatter grid of label 2 and 4, the scatter plot for features A2 and A4 consists of an upper orange cluster and a bottom blue cluster. On column A1 of the scatter plot, we notice that the blue cluster is spread out mainly on the right side of the scatter plots, whereas the orange cluster is at the left. The distributions' plot on the diagonal also represents that. Nevertheless, there is a higher complexity for separability for the clusters of scatter plot A3-A4. We notice a blue cluster in the middle and two smaller orange clusters on either side of the blue cluster. This added complexity makes the problem prevents the RPC model from achieving higher scores. Similarly, looking at the scatter grid of labels 0 and 2, the separability of the clusters is more apparent. The blue cluster in scatter plots for features A1-A3, A1-A4, A2-A3, A2-A4 have circular shapes and are placed close to the edges of the diagrams, while the orange cluster is spread out to the rest of it. We notice that there are marginal points where blue and orange cluster intersect. This noise, produced by the marginal data points, lowers the performance levels of the RPC approach.

5.7.3 High Scores Datasets

In contrast to the bodyfat dataset, while observing the scatter grid for the iris dataset 5.9, it is easy to see why the classifiers/regressors achieve high roc auc scores. The clusters of different rankings



(a) Labels 0 and 2



(b) Labels 2 and 4

Figure 5.8: Stock scatter grids for pairs of labels

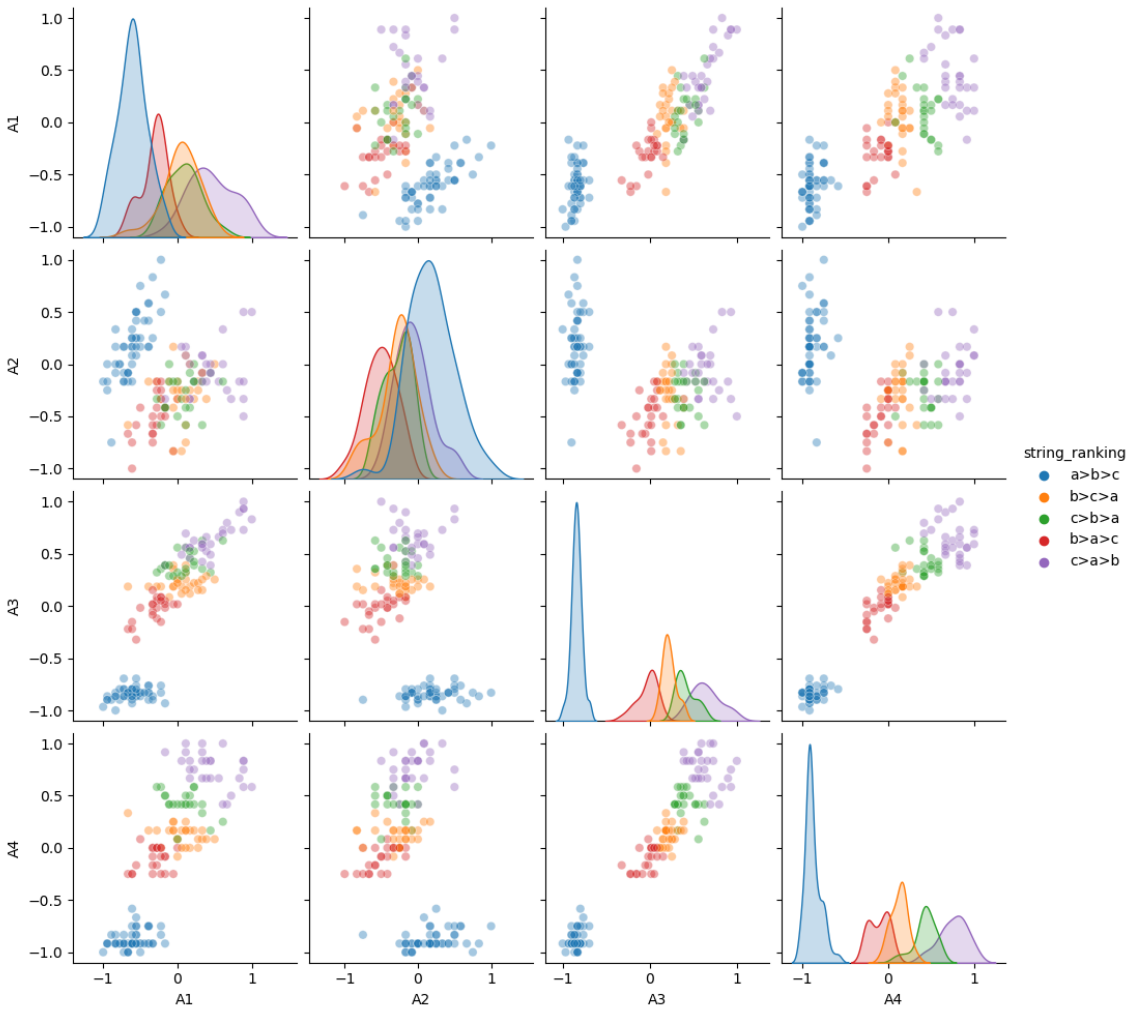


Figure 5.9: Iris scatter plot for discrete rankings

are obvious and the predicted rankings are always close to the true ranking.

The dataset contains five unique rankings, thus five colors are used in each scatter plots. The amount of points always stays she same, since each point corresponds to one training instance. The blue cluster forms a small cluster that is placed far from the other points and can be linearly separated in the 2 dimensional space. Furthermore, looking at the diagonal distribution diagram of feature A3 and A4, it is obvious that the blue cluster has no overlap with the other cluster. Practically, by using a correct threshold value on either feature A3 or feature A4 would serve as a correct rule for classification. Looking at rows A3 and A4 of the scatter grid, we also observe clusters on the top of the diagrams. Each cluster consists of four colors and can be approximately separated with three horizontal lines. The same linearly separable patterns can also be observed on the rows A1 and A2 of the scatter grid.

As wee see, for all three pairs of features, the rankings in all scatter plots are linearly separable already in the two dimensional plane. This suggests that the original three dimension problem is easier to solve. Apart from marginal cases of points that are close to the separation lines, the performance of any popular model like random forests or support vector machines is expected to be top.

Fried. Fried is the dataset where the RPC approach has the best results. Using the scatter grids 5.10, it is to explain this behaviour. Looking at the scatter plots for labels 0 and 1 we observe that apart

from the majority of the scatter plots where the clusters are mixed, there exists one two dimensional plot that is obviously linearly separable. The scatter plot between features A5-A6 contains two perfectly separable clusters that could be split using a diagonal line. This pattern repeats with each pair of labels when using different pairs of features. For example, pairs of labels 2 and 4 are perfectly separated using only features A7 and A9. More specifically, we could match label 0 with feature A5, label 1 with feature A6, ... and label 4 with feature A9. For classification of a pair of labels we shall only focus on the scatter plot between the corresponding features in order to get perfect classification.

The scatter plots for this dataset provides a natural interpretation of the reason why the RPC model can perform great on this dataset and visualises its hidden structure.

Important Distinction. Scatter plots make diagrams of data points using pairs of features and plotting the different categories - labels on the axes of the two features. Since the plots are in 2 dimensions, they give us a general perspective on how hard or easy it is to split different categories of data points. There is an asymmetry worth noting though. If a plot show in 2 dimensions shows distinct, separated clusters, then it is clear evidence for the separability of the data. However, if the groups of data cannot be split in 2 dimensions in an obvious way, it does not necessarily mean that the same holds true for the the original high dimension problem. In other words, an absence of this kind of structure is not evidence for a lack of separability. This may happen because of the choice of angles that are used for visualisation.

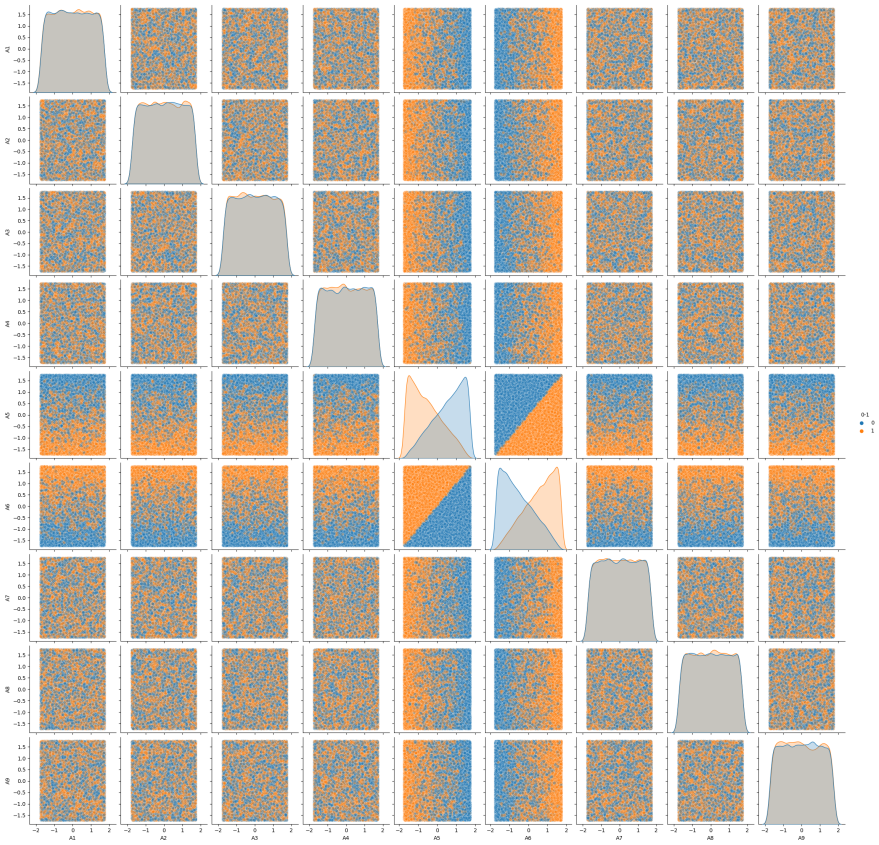
5.8 Conclusions

An observation to be made is that the lack of training instances can lower the performance levels of our models. This is especially true for datasets that have uniform distribution among the apperances of unique rankings in the target space. In datasets like wisconsin or bodyfat, the training instances are few in comparison to the size of the rankings' space, while the unique rankings curve represent a uniform distribution of the training instances in the target space Ω. Therefore, low scores should be expected.

In the contrary, in datasets with high number of training instances and imbalanced distribution of training instances in the target space Ω, the RPC approach performs better. As we already mentioned, unique ranking curves that have exponential growth suggest that the target space Ω has smaller size and thus it is easier for the RPC approach to make correct predictions.

Furthermore, there is strong correlation between the ROC AUC scores and Kendall tau scores. The RPC approach is a two stage process and the first stage aims on training binary models that can solve the binary classification model between two labels. The second stage combines the predictions of the binary models to provide a final complete ranking as a prediction. The functionality of the second stage is based on the assumption that the binary classifiers of the first stage make good predictions. Therefore, the ROC AUC scores, which quantify the performance of the classifiers / regressors of the first stage, are strongly connected with the Kendall tau scores, which quantifies the overall performance of the model by using the final ranking provided by the second stage.

However, the most important aspect for the performance of a model is its inherent difficulty. By inherent difficulty we refer to how the training instances are distributed in the instances' space $\mathbb{X}$ and how the are mapped to the rankings' space Ω. Datasets like calhousing and pendigits are phenomenally similar when comparing the amount of instances or unique rankings, but the kendall tau scores are really different. The scatter grids also support this arguement. The basic assumption on which the learning algorithms are based on is that instances with similar features vectors will be mapped to similar target rankings. This locality assumption though is more straightforward and clear in some datsets than other. For example, the stock and cpu-small dataset are very similar in terms of features and labels and in fact cpu-small has a bigger amount of training instances. However, in practice, the binary classifiers perform much better for the stock dataset.



(a) Labels 0 and 1

(b) Labels 0 and 2

Figure 5.10: Fried scatter grids for pairs of labels

Chapter 6

Implementation - Comparisons - Results

6.1 Programming in Python

Python is a high-level programming language that lets programmers work quickly and integrate systems more effectively. It is a really popular language in the programming world, mainly due to its powerful tools yet simple syntax.

During the work for my thesis, I specifically used python 3.6 to create different programs for analysis, visualisations and train numerous models. Modules in python group related code and by importing modules programmers can expand the available functionalities. The most important modules used during the development were NumPy, Pandas, Matplotlib.

Numpy is the fundamental package for scientific computing in python. It is a Python library that provides a multidimensional array object, various derived objects (such as masked arrays and matrices), and an assortment of routines for fast operations on arrays

Pandas is an open-source, BSD-licensed library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming language.

Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy.

6.2 Methodology - General Setting of Experiment

The purpose of this chapter is to make a comparative evaluation of models that use the RPC approach. We experiment with a variety of classifiers and regressors. The goal is to analyze how different classification algorithms perform and to gain insight on which datasets they perform best.

Our comparative analysis also focuses on aggregation techniques. We study the effects of aggregation techniques by pairwise preferences on the performances of different models.

The evaluation of the models happens in terms of Kendall's tau coefficient and derive from five repetitions of ten-fold cross-validation. Cross-validation is a statistical method used to estimate the skill of machine learning models. We also set fixed values of input seeds to the pseudo-random number generator of Python module "random" for reproducibility of our results.

In the following subsections, we will focus on two different aspects. Firstly, we make extensive analysis on the first stage of the RPC algorithm, that is the binary classifiers. We experiment with different classifiers and regressors and compare their predictions. Secondly, we focus on the second stage of the RPC algorithm, which is the aggregation technique. We experiment with different aggregation methods, not only popular methods but also new and intuitive methods that we propose.

6.3 Stage One: Pairwise Comparisons

The first stage is to train models for each pair of labels that predict which label is preferred most between the two. The key idea is to transform the label ranking problem through reduction to many simpler binary learning problems. The advantage of this approach is that the simple binary classification problem has already been studied very well and many artificial intelligence algorithms can be

used in order to solve it. The question that arises therefore is which model performs experimentally better.

In this section, we present the results of experimenting with different classification algorithms for the base learners and provide conclusions based on their results. For the stage of experimentation we tried both regressors and classifiers.

To simplify the comparison of the models, we use the sum of binary predictions aggregation technique that was proposed in the original paper [8]. This method is a intuitive approach while also simple to implement.

6.3.1 Classifiers

Binary classifiers are models that are trained given a set of inputs with their desired binary output. After training, the classifier is able to predict, given a new input, the most probable output. In our implementation, we have matched all labels with unique identification numbers. For example, the model $M_{i,j}$ where $i < j$, predicts 0 if label i is preferred over label j and 1 if label j is preferred over label i, given an input.

We used the methods of Support Vector Machines, Decision Trees and Random Forests to train classifiers.

- Support vector machines (SVMs) are a set of supervised learning methods used for classification, regression and outliers detection. They are versatile, memory efficient and effective in high dimensional spaces. More information for this algorithm can be found in [40, 41, 42, 43].

- Decision Trees (DTs) are a non-parametric supervised learning method used for classification and regression. They are simple and easily interpretable. More information for this algorithm can be found in [44, 45, 46].

- Random Forests (RFs) fall under the umbrella of ensemble methods. The goal of ensemble methods is to combine the predictions of several base estimators built with a given learning algorithm in order to improve generalizability / robustness over a single estimator. The idea is based on the wisdom of the crowd principle, which states that we should prefer the collective opinion of a group of individuals over that of a single expert. An explanation for this phenomenon is that there is idiosyncratic noise associated with each individual judgment, and taking the average over a large number of responses will go some way toward canceling the effect of this noise. More information for this algorithm can be found in [47, 35].

As it turns out, the Decision Trees algorithm always achieves inferior classification scores to Random Forest. This is to be expected since Random Forests actually consists of many decision trees and ensembles their results to achieve even greater scores. The Random Forest Classifier turned out to be the most consistent in achieving high scores in regards to Kendall tau coefficient among the three classifiers. As for the Support Vector Classifier, it turns out that it achieves pretty good results and in some cases surpassing the results of random forest.

As shown by our experiment, random forests and Support Vector Classifier have comparable results, with random forests achieving a bit more accurate predictions in most datasets. The datasets in which support vector algorithm works better are the ones that are linearly separable. Results for different datasets are given below .

6.3.2 Regressors

Binary regressors differ from binary classifiers in that they don't predict discrete values in the set $\{0, 1\}$, but they predict a number in the range $[0, 1]$. This value of prediction can be interpreted as a measure of confidence. The closest the predicted value of model M_{ij} is to 0, the more confident we are that the corresponding label i precedes label j. Respectively, the closest the predicted value of

dataset	SVC	DTC	RFC100	SVR	DTR	RFR100
authorship	0.943±0.017	0.873±0.022	0.935±0.018	**0.949±0.016**	0.873±0.022	0.924±0.021
bodyfat	0.278±0.059	0.126±0.063	0.204±0.06	**0.287±0.055**	0.127±0.06	0.208±0.061
calhousing	0.298±0.012	0.354±0.011	0.484±0.009	0.299±0.012	0.354±0.011	**0.487±0.01**
cpu-small	0.501±0.014	0.397±0.015	**0.519±0.014**	0.504±0.014	0.398±0.016	0.515±0.013
fried	0.983±0.001	0.987±0.001	0.991±0.001	0.97±0.001	0.987±0.001	**0.993±0.001**
glass	0.862±0.049	0.875±0.04	**0.901±0.035**	0.871±0.046	0.875±0.04	0.894±0.034
housing	0.689±0.027	0.782±0.027	**0.822±0.026**	0.692±0.026	0.782±0.027	0.817±0.023
iris	0.968±0.034	0.949±0.042	0.965±0.037	**0.983±0.025**	0.949±0.042	0.958±0.037
pendigits	0.957±0.002	0.961±0.001	0.975±0.001	0.95±0.002	0.961±0.001	**0.977±0.001**
segment	0.957±0.005	0.969±0.005	**0.977±0.005**	0.951±0.006	0.969±0.005	**0.977±0.004**
stock	0.892±0.013	0.901±0.015	**0.926±0.014**	0.897±0.014	0.901±0.015	0.92±0.013
vehicle	0.866±0.026	0.831±0.032	0.882±0.025	0.861±0.028	0.831±0.032	**0.883±0.027**
vowel	0.897±0.015	0.861±0.015	**0.913±0.015**	0.909±0.014	0.861±0.015	0.902±0.014
wine	0.959±0.043	0.892±0.081	0.941±0.051	**0.965±0.034**	0.892±0.081	0.923±0.061
wisconsin	0.555±0.034	0.512±0.031	0.549±0.034	**0.584±0.032**	0.512±0.032	0.572±0.033

Table 6.1: Table of algorithms for base learners

model M_{ij} is to 1, the more confident we are that the corresponding label j precedes label i. If the value is 0.5, we are equally confident for the two labels.

For the experiments we used the three regressor versions of the algorithms described in the previous subsection.

Similar to the previous results, the decision tree algorithm achieve inferior results comparing to the other two models. Furthermore, random forests and support vector regressors are comparable, with random forests achieving slightly better performance.

6.3.3 Classifiers vs Regressors

We experimented both with the classifier version as well as the regressor version of each algorithmic model. Hence, we made use of six different models in total. In order to compare the algorithms, firstly we use the simple summing method that was proposed in the original paper as the rank aggregation method for the second stage of the RPC. We will refer to our algorithms as follows:

- SVC: classifier that uses support vector machines algorithm
- DTC: classifier that uses decision trees algorithm
- RFC: classifier that uses random forest algorithm
- SVR: regressor that uses support vector machines algorithm
- DTR: regressor that uses decision trees algorithm
- RFR: regressor that uses random forest algorithm

For the random forest classifier and regressor we make use of an ensemble of predictions from 100 decision trees, a number which provides a nice balance between accuracy and speed. For reproducibility of the scores, we set a constant value as seed number for random generators.

The table 6.1 shows the scores per dataset for the different models.

Firstly, the kendall tau scores per dataset have noticeable fluctuations. This suggests that the algorithm of the base learner plays significant role in the performance of the RPC approach in general. Similarly to the previous chapter **??**, the importance of the binary model of the base learners is once more highlighted.

We can observe a similar pattern to the level of Kendall tau scores achieved among all binary models. This is an indicator that the nature of the dataset and the inherent classification difficulty for these datasets affects the scores significantly. This fact was also pointed out in the previous chapter.

Comparing DTC to RFC and DTR to RFR, we notice that the random forest algorithm is superior to the decision tree algorithm in almost all cases. This is to be expected due to the fact that the random forest algorithm uses an ensemble of decision trees that are trained with diversified input data (this is achieved using the bagging technique) and thus has a better performance. We should not forget thought that the random forest algorithm requires more time in order to train the ensemble of decision tree and therefore has increased time complexity.

Counter-intuitively, while comparing RFR to RFC, we notice that the classifier version of random forest algorithm for the base learners seem to have slightly superior scores. We might expect that the regressor would perform better because the predictions of the classifiers are in the range between 0 and 1 and is not forced to be an integer number and thus the predictions are more accurate. However, the scores suggest that the aggregation function which is used in combination with decimal predictions amplifies the classification errors of the base learners and slightly worsens the predictions of the models. Both models though have real good scores that are competitive to the state-of-the-art kendall tau scores.

In contrast, by comparing SVR to SVC model, we observe that the regressor version is slightly superior to the classifier version of the algorithm. One possible explanation for this could by given by considering the prediction score of the regressor as a confidence metric. The support vector machine algorithm uses a separating hyperplane in higher dimensions to classify the data in two different classes. This linearity limits the algorithms capabilities for splitting the data and thus many data points can be marginally wrongly classified. However, due to the linear nature of the algorithm, the regressor confidence prediction has a geometric reasoning can be trusted. Hence, the regressor predictions give a valuable indication of how certain the model is and the aggregation method takes advantage of this confidence to outperform the classifier version of the algorithm.

From the six algorithms we experimenter, the best performing models proved to work using RFC, SVR and RFR. More specifically, RFC has the best overall performance.

From the six algorithms we experimenter, the Random Forest Classifier (RFC) turned out to be the most consistent in achieving high scores in regards to Kendall tau coefficient among the 3 classifiers. The Random Forest Regressor (RFR) is also among the best performing models. As for the Support Vector Regressor, it turns out to be one of the most competitive models, even surpassing in some cases the results of random forest.

The rest of the binary models do not require further investigation. For the next sections, we will make extensive use of RFC, SVR, RFR algorithms for our base learners.

6.4 Stage Two: Aggregation Techniques

At the second stage of the modular RPC approach, the votes of the base learners, have to be combined in order to produce a final complete ranking that will serve as the prediction of the RPC model. The problem of aggregating by the binary predicted preference relations of an instance x has been well studied. In this section, we only consider the complete rankings version of the label ranking problem.

In chapter 3, we focused on explaining different approaches and methods proposed in the bibliography from a theoretical perspective. In this section, we will focus on implementing these aggregation techniques and measuring the performance of each in the given datasets. We compare the differences between datasets and between the base learner algorithm and how this affects the scores of our RPC model.

6.4.1 Summing Binary Predictions

Given the predictions of the base learner M_{ij} for an instance x, each label i is associated with a score that corresponds to the votes that each relevant base learner gives to that label. The mathematical

formula is the following:

$$S(\lambda_i) = \sum_{k=1}^{i-1} M_{ki} + \sum_{k=i+1}^{n} (1 - M_{ik})$$

We should remind that the prediction of a base learner M_{ij} could be considered as measure of confidence on which label precedes the other in the final ranking. When the prediction R_{ij} approaches 0, the base learner is confident that $\lambda_i \succ \lambda_j$ in the final ranking and vice versa.

After that, the labels are sorted in decreasing order in respect to formula $S(\lambda_i)$. The resulting order serves as the predicted ranking of the RPC approach.

We will refer to this aggregation technique as SumBP, which stands for Summing Binary Predictions of base learners. We use SumBP, which was proposed in the original paper, as a baseline for comparisons.

6.4.2 Max Votes of Training Ranking

Firstly, we introduce a new aggregation technique which selects the training instance that maximises the agreement with the base learners predictions. Each corresponding output ranking that we encounter in the training dataset, serves as potential prediction of the aggregation method. Each ranking is associated with a value $S(\tau)$, and similarly to the aforementioned function, gets votes from the base learners which come to an agreement with the pairwise relation that the labels have on ranking τ. Finally, the ranking that maximises formula $S(\tau)$ is used as prediction of the RPC approach.

Although it seems similar to SumBP technique, note that this technique forces the complete ranking prediction to be present of the training datset. We will refer to this approach as MaxVTR, which stands for max votes of of training ranking.

Furthermore, in an effort to improve performance, we introduce a technique to include information about the distribution (frequency of appearance) of each ranking. This way, our aggregation favors rankings that appear in the majority of the time and hopefully are dominant in a dataset.

We consider that each rankings after the predictions has v votes. It also has an f frequency of appearance in the training dataset, which is equal to the number of appearances divided by the number of instances in the training dataset. We introduce a new aggregation methods that selects the ranking with the maximum values for different formulas. We refer to this model as MaxVTR-LF, where LF stands for low frequency and suggests that information about the frequency has impact in low degree on the final score of a ranking. The logic behind this is that we want to increase the probability of frequent rankings in the training dataset being used as predictions. The method select the ranking that maximises the formula $g(f, v) = (1 + f) * v$.

6.4.3 Kwiksort

Another aggregation function that is important to examine is kwiksort. Kwiksort, as described in chapter 2 (3.4.7), is used in tournaments and has similar logic to quicksort. In our problem, we use a variation of the algorithm, that uses the binary predictions of the base learners to arrive to the final complete prediction. It uses a label i as pivot and depending on whether the predictions of the base learners are above 0.5 or below, the rest of the labels are split into two groups, the preceding labels and the succeeding labels. The process then is repeated to the groups until the final ranking is produced. We refer to this first implementation of the kwiksort algorithm as KWIK.

The following table contains the

The following table 6.2 contains the scores for the three different approaches. The RFC and SVR algorithms are used for the base learners since they perform better than the rest according to the results of the previous section. The RFC and RFR approaches use an ensemble of 100 decision trees.

First and foremost, we notice that the scores between the classic approach SumBP and the max votes MaxVT approach with no frequency are competitive. In datasets like cpu-small and housing,

	SumBP		MaxVTI		MaxVTI-LF		KWIK	
dataset	RFC100	SVR	RFC100	SVR	RFC100	SVR	RFC100	SVR
authorship	0.935±0.018	0.949±0.016	0.936±0.018	0.947±0.015	0.929±0.017	0.925±0.016	0.938±0.017	0.947±0.015
bodyfat	0.204±0.06	0.287±0.055	0.193±0.061	0.268±0.059	0.194±0.061	0.265±0.06	0.198±0.067	0.276±0.058
calhousing	0.484±0.009	0.296±0.011	0.488±0.009	0.299±0.012	0.488±0.01	0.294±0.013	0.481±0.009	0.298±0.012
cpu-small	0.519±0.014	0.507±0.014	0.522±0.014	0.503±0.014	0.523±0.015	0.497±0.014	0.519±0.014	0.503±0.013
glass	0.901±0.035	0.871±0.046	0.897±0.035	0.879±0.047	0.84±0.056	0.807±0.062	0.898±0.037	0.875±0.044
housing	0.822±0.026	0.692±0.026	0.823±0.025	0.701±0.029	0.822±0.023	0.695±0.031	0.82±0.026	0.695±0.028
iris	0.965±0.037	0.983±0.025	0.965±0.037	0.961±0.035	0.965±0.037	0.953±0.043	0.965±0.037	0.961±0.035
segment	0.977±0.005	0.951±0.005	0.912±0.016	0.905±0.017	0.97±0.005	0.938±0.006	0.977±0.005	0.95±0.006
stock	0.926±0.014	0.897±0.014	0.926±0.013	0.899±0.013	0.926±0.013	0.893±0.013	0.925±0.014	0.899±0.013
vehicle	0.882±0.025	0.861±0.028	0.883±0.026	0.872±0.028	0.881±0.026	0.853±0.034	0.882±0.026	0.873±0.027
vowel	0.914±0.016	0.895±0.011	0.912±0.016	0.905±0.017	0.91±0.017	0.903±0.017	0.903±0.017	0.898±0.016
wine	0.941±0.051	0.965±0.034	0.941±0.051	0.96±0.036	0.941±0.051	0.955±0.031	0.955±0.036	0.954±0.046
wisconsin	0.549±0.034	0.584±0.032	0.478±0.034	0.496±0.034	0.478±0.034	0.496±0.034	0.528±0.036	0.554±0.031

Table 6.2: Table of scores of different aggregation techniques

max votes method is superior. The two aggregation methods have similar logic, hence we expected that the differences between scores are negligible in most cases (they concern the third decimal place).

However, the simple summing aggregation method performs better on datasets that have low amount of training instances with uniform appearnace of instances, e.g. bodyfat and wisconsin. This is because in the max votes aggregation method, we only use existing rankings as predictions. This fact is limiting and causes increased loss between the predicted and the true ranking, especially in cases where the true ranking is not contained in the training dataset. This aggregation method performs poorly though on datasets that are inherently difficult as we already explained.

We also observe that adding the frequency factor using the MaxVT-LF aggregation process achieves slightly inferior results. One possible explanation is that the frequency information is not combined in the optimal way in order to take advantage of it. The impact of the frequency factor may be too high in some datasets and too low in other datasets. Furthermore, there are datasets where the scores do not change a lot between max votes and max votes with frequency impact, e.g. stock and wine, whereas in other datasets there are noticable drops of performance, e.g. glass. Using the datasets analysis from the previous chapter, we notice that the former group has multiple rankings with many appearances, whereas the later group has small target space with high frequency of appearance. We suspect that when the unique rankings appearances curve approaches a linear shape, max votes aggregation method thrives.

We also notice that the kwiksort aggregation method KWIK achieves similar results with the simple summing method. In datasets like authorship and wine, KWIK achieves slightly superior scores to SumBP performs, but the differences are practically negligible.

In many cases, the differences concern the third decimal digit. The differences between base learners however is considerable. We should underline that the base learner that achieves the best performance is the same among all aggregation methods. This suggests that the effect of the first stage of the RPC is more important comparing to the second stage of aggregating the binary predictions. As we showed earlier, a biased aggregation technique can significantly worsen the performance of our model. However, finding an aggregation technique that can achieve significantly higher results seems inprobable and that is experimentally validated. This is because the aggregation method is mainly used to transform the binary predictions to the right format that the problem requires. Using noisy binary predictions can only lead to noisy ranking, no matter how simple or sophisticated the aggregation techinque. Still, the aggregation method using kwiksort is much more efficient since we use only the $logn * n$ binary prediction instead of $n * (n - 1)$ binary predictions that we use for the original aggregation method.

In conclusion, we notice that the choice of aggregation technique at the second stage does not impact the performance of the RPC model significantly. In contrast to the first stage, where the per-

formance of the model varies depending on the choice of the classifier, at the second stage the results in term of kendall tau score does not seem to vary much. The aggregation technique is just required to have a solid mathematical logic and be simple. Furthermore, using additional information, like the frequency factor, does not necessarily lead to better results. Unless the additional information is integrated in an effective way, it can harm performance, acting as a kind of noise to our model.

6.5 Incomplete Data

So far, we payed attention to the complete rankings version of label ranking problem. However, in many real life scenarios, it is not possible to get complete ranking information about every instance. This may be because of financial constraints of a data collection procedure, the nature of the problem where it is difficult to grasp the whole information for one individual, e.g. it makes sense to find the best 3 and the worst 3 among a set of label choice for a person since the chaotic nature of a problem with many choices can make the experiment intimidating and incorrect to true intentions of a person from a behavioural perspective.

We refer to learning from incomplete preference information label ranking problem when we have access to few of pairs of preferences for an instance.

To produce the incomplete training data, we implement the following procedure. We define a probability p and for each label for each ranking we flip a biased coin. The biased coin deletes the label from the ranking with probability p or keeps it wit probability $1 - p$. After this procedure, the rankings become incomplete with an estimated of p labels missing from the data. We should note that the the probability of a pairwise preference surviving this procedure is $(1 - p)^2$, thus the training data for each base learner decreases quadratically proportional to the number of data missing. For example, a ranking $\lambda_5 \succ \lambda_3 \succ \lambda_2 \succ \lambda_4 \succ \lambda_1$ may be transformed to $\lambda_5 \succ \lambda_2 \succ \lambda_4$. In this case, we originally had $5 * 4/2 = 10$ pairwise preferences, whereas for the incomplete ranking we only have $3 * 2/2 = 3$ pairwise preferences to train the corresponding base learners.

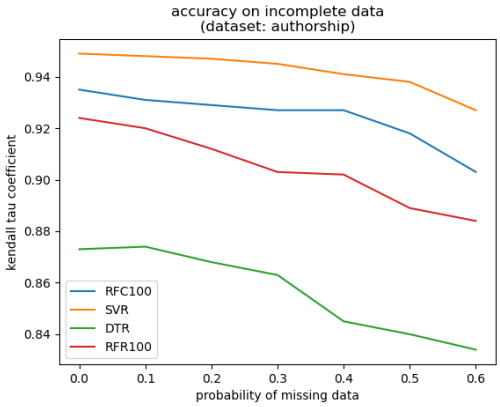
We are going to use RFC, SVR, DTR, RFR as algorithms for the base learners and examine the robustness of each model comparing to the others. The incomplete probability ranges from 0.0 to 0.6 with step of 0.1.

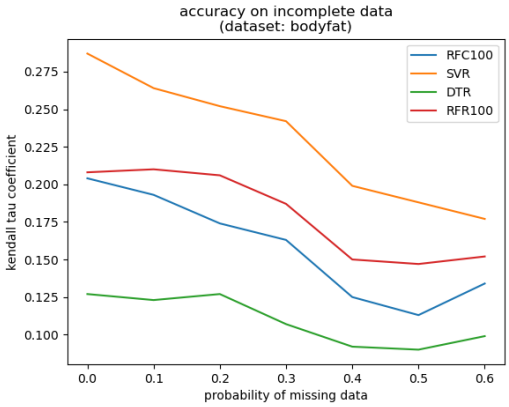
We make the following conclusion:

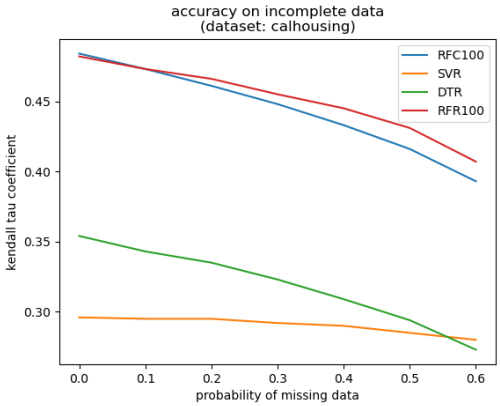
- As we expected, the curves are declining, which means the higher the probability p of incomplete data, the more the the lower the performance of the base learners and in turn the performance of the complete RPC model. More importantly though, the rate of decrement becomes steeper as probability p approaches values close to 1. This is depicted visually by the concave shape of the curves. For example, in the wine and vehicle datasets, the drop of performance between 0.5 and 0.6 is bigger than that between 0.1 and 0.2. The increment of the rate of is to be expected, due to the fact that an increment of the incomplete probability translates to a quadratic fall in the number of training instances for the base learners.

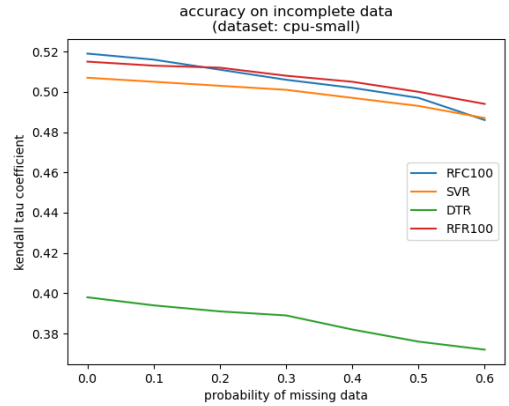
- There is a visual verification for the conclusions made in the previous section about the performance of the algorithms. The RFR and RFC curves achieve similar results since the blue and red curves are almost identical. The DTR model has the worst perfromance, thus the green curve is always below the other three. We also observe that depending on the dataset, either the random forest approach or the support vector approach perfroms better. More specifically, in authorship, bodyfat, wine and wisconsin datasets, SVR has superior performance comparing to the rest of the algorithms.

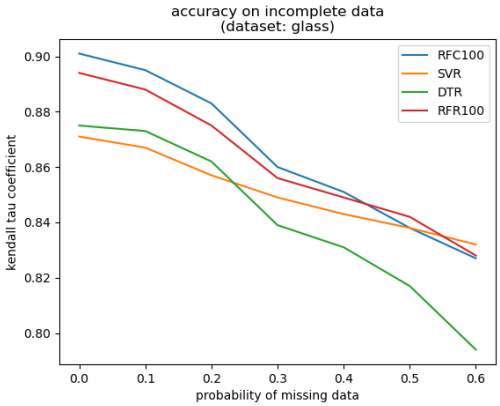
- There is significant variation on the impact of the missing data to the performance of RPC depending on the dataset, e.g. in dataset like authorship and cpu-small there is a drop of 0.02 in kendall tau scores, whereas in the iris and vowel dataset there is an average drop of 0.10 in performance of the RPC models.

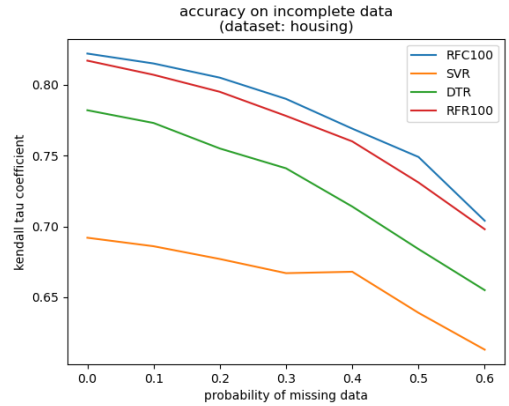

accuracy on incomplete data
(dataset: authorship)
kendall tau coefficient
probability of missing data
RFC100
SVR
DTR
RFR100

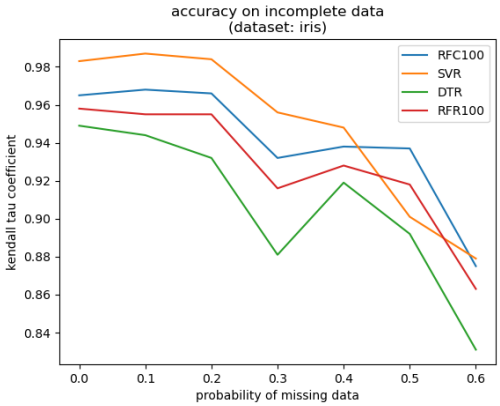

accuracy on incomplete data
(dataset: bodyfat)
kendall tau coefficient
probability of missing data
RFC100
SVR
DTR
RFR100

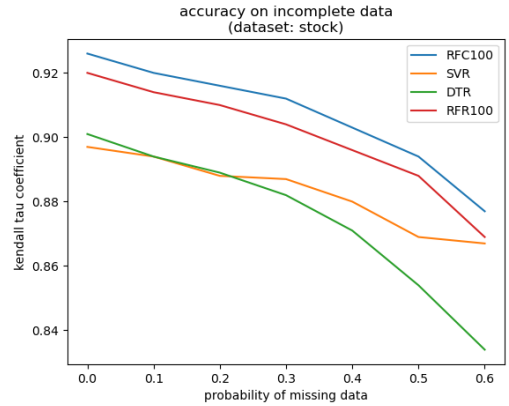

accuracy on incomplete data
(dataset: calhousing)
kendall tau coefficient
probability of missing data
RFC100
SVR
DTR
RFR100


accuracy on incomplete data
(dataset: cpu-small)
kendall tau coefficient
probability of missing data
RFC100
SVR
DTR
RFR100


accuracy on incomplete data
(dataset: glass)
kendall tau coefficient
probability of missing data
RFC100
SVR
DTR
RFR100


accuracy on incomplete data
(dataset: housing)
kendall tau coefficient
probability of missing data
RFC100
SVR
DTR
RFR100


accuracy on incomplete data
(dataset: iris)
kendall tau coefficient
probability of missing data
RFC100
SVR
DTR
RFR100


accuracy on incomplete data
(dataset: stock)
kendall tau coefficient
probability of missing data
RFC100
SVR
DTR
RFR100

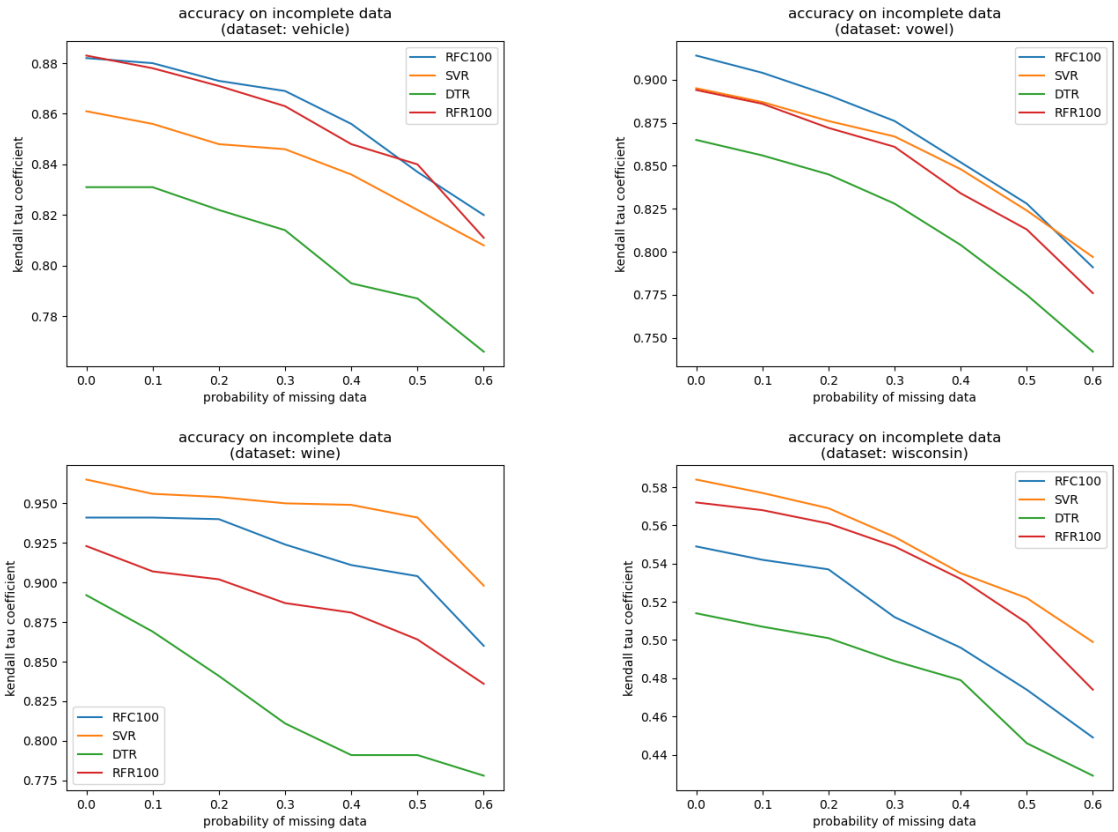


Figure 6.1: Performance scores on incomplete data

- A general rule of thumb is that the support vector approach is more robust comparing to tree-based approaches. The fact is experimentally validated in the diagrams. In datasets like calhousing, glass, stock and vehicle, the effect of the incompleteness of the training data on the performance of SVR is smaller comparing to the other three algorithms.

Chapter 7

Closing remarks

7.1 Conclusions

During this thesis, we introduced the label ranking problem and the importance of it, as well as the practical applications. We discussed the approaches in the recent bibliography and we especially focused on the RPC model, which reduces the problem to learning pairwise comparisons and uses aggregation techniques to combine the binary predictions and produce a ranking that will serve as prediction of the model.

We conducted thorough experimental evaluation on the datasets and the difficulty of the label ranking in general. We noticed that the label ranking problem has high complexity and different aspects, like the number of labels or unique rankings encountered, give us hints on the levels of results we should expect. However, the structure of a learning problem, i.e. how instances are distributed in the features' spaces and mapped in the rankings' space, is the major factor that determines difficulty of the learning problem.

As we underlined in the previous chapter, the RPC approach achieves competitive results comparing to the state-of-the-art approaches, even if the model was originally proposed more than 10 years ago. Since RPC has a modular scheme, we made a comparative evaluation on how different binary classifiers affect performance. The Random Forest Classifier proved to be the most effective algorithm for the base learners. The Support Vector Machine Regressor and Random Forest Regressor also have high quality performance. Lastly, we concluded that the effects of aggregation techniques are negligable comparing to the classifiers. Simple aggregation techniques like summing the binary preferences or predicting the ranking of the training instance with the most votes seemed to perform better than complex techniques. The kwiksort approach also had good performance levels, while being more efficient in terms of algorithmic complexity.

7.2 Future Work

The most important ones were the following:

- Since the algorithms of the base learners had such impact to the performance of the model, an extension of the comparative evaluation that this thesis started by including even more pairwise comparisons algorithms would be interesting from an research perspective.
- There is a need for investigation on possible ways in which the property of transitivity could be used in order to achieve better results. Possible ways would be creating aggregation methods that explicitly force transitivity in cases where the base learners are really confident about their predictions.
- The RPC method reduces the problem to simpler binary learning problems. Taking this logic a step further, a reduction to trinary learning problems could potentially provide even more possibilities for experimentation. More sophisticated aggregation techniques could also be implemented in that case. However, we should note that there is a huge complexity overhead by doing this. More specifically, since we make groups of three, there is a need for n^3 models.

In conclusion, label ranking is a practical problem and has plenty room for exploration. There are numerous applications for it and the scientific interest for it is getting bigger.

RPC has proved to be a very effective method. This thesis made extensive experimetntall evaluation on this model and the label ranking in general and hopefully the conclusion will be considered and studied by other researchers to better evaluate their models and carry further scientific reasearch.

Bibliography / Βιβλιογραφία

[1] Shankar Vembu and Thomas Gärtner. Label ranking algorithms: A survey. In *Preference learning*, pages 45–64. Springer, 2010.

[2] Yangming Zhou, Yangguang Liu, Jiangang Yang, Xiaoqi He, and Liangliang Liu. A taxonomy of label ranking algorithms. *JCP*, 9(3):557–565, 2014.

[3] Weiwei Cheng, Jens Hühn, and Eyke Hüllermeier. Decision tree and instance-based learning for label ranking. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 161–168, 2009.

[4] Weiwei Cheng, Krzysztof Dembczynski, and Eyke Hüllermeier. Label ranking methods based on the plackett-luce model. In *ICML*, 2010.

[5] Yangming Zhou and Guoping Qiu. Random forest for label ranking. *Expert Systems with Applications*, 112:99–109, 2018.

[6] Sariel Har-Peled, Dan Roth, and Dav Zimak. Constraint classification: A new approach to multiclass classification. In *International conference on algorithmic learning theory*, pages 365–379. Springer, 2002.

[7] Ofer Dekel, Yoram Singer, and Christopher D Manning. Log-linear models for label ranking. *Advances in neural information processing systems*, 16:497–504, 2003.

[8] Johannes Fürnkranz and Eyke Hüllermeier. Preference learning and ranking by pairwise comparison. In *Preference learning*, pages 65–82. Springer, 2010.

[9] Guang-Bin Huang, Hongming Zhou, Xiaojian Ding, and Rui Zhang. Extreme learning machine for regression and multiclass classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 42(2):513–529, 2011.

[10] Mohamed Aly. Survey on multiclass classification methods. *Neural Netw*, 19:1–9, 2005.

[11] Sariel Har-Peled, Dan Roth, and Dav Zimak. Constraint classification for multiclass classification and ranking. *Advances in neural information processing systems*, pages 809–816, 2003.

[12] Grigorios Tsoumakas, Ioannis Katakis, and Ioannis Vlahavas. A review of multi-label classification methods. In *Proceedings of the 2nd ADBIS workshop on data mining and knowledge discovery (ADMKD 2006)*, pages 99–109. Citeseer, 2006.

[13] Grigorios Tsoumakas, Ioannis Katakis, and Ioannis Vlahavas. Random k-labelsets for multilabel classification. *IEEE transactions on knowledge and data engineering*, 23(7):1079–1089, 2010.

[14] Grigorios Tsoumakas and Ioannis Katakis. Multi-label classification: An overview. *International Journal of Data Warehousing and Mining (IJDWM)*, 3(3):1–13, 2007.

[15] Johannes Fürnkranz and Eyke Hüllermeier. Pairwise preference learning and ranking. In *European conference on machine learning*, pages 145–156. Springer, 2003.

[16] John I. Marden. *Analyzing and Modeling Rank Data.* 1995.

[17] Miklos Bona. *Combinatorics of Permutations (Discrete Mathematics and Its Applications) (2nd ed.)*. 2012.

[18] Persi Diaconis. Group representations in probability and statistics. *Lecture notes-monograph series*, 11:i–192, 1988.

[19] Maurice George Kendall. Rank correlation methods. 1948.

[20] Charles Spearman. The proof and measurement of association between two things. 1961.

[21] Persi Diaconis and Ronald L Graham. Spearman's footrule as a measure of disarray. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(2):262–268, 1977.

[22] H.J.M. D'Abrera Erich L. Lehmann. *Nonparametrics: Statistical Methods Based on Ranks (1st ed.)*. 1975.

[23] Colin L Mallows. Non-null ranking models. i. *Biometrika*, 44(1/2):114–130, 1957.

[24] Ravi Kumar and Sergei Vassilvitskii. Generalized distances between rankings. In *Proceedings of the 19th international conference on World wide web*, pages 571–580, 2010.

[25] John G Kemeny. Mathematics without numbers. *Daedalus*, 88(4):577–591, 1959.

[26] JL Snell and JG Kemeny. Mathematical models in the social sciences. *Introduction to Higher Mathematics. Ginn, Boston*, 1962.

[27] John Bartholdi, Craig A Tovey, and Michael A Trick. Voting schemes for which it can be difficult to tell who won the election. *Social Choice and welfare*, 6(2):157–165, 1989.

[28] Cynthia Dwork, Ravi Kumar, Moni Naor, and Dandapani Sivakumar. Rank aggregation methods for the web. In *Proceedings of the 10th international conference on World Wide Web*, pages 613–622, 2001.

[29] Kenneth J Arrow, Amartya Sen, and Kotaro Suzumura. *Handbook of social choice and welfare*, volume 2. Elsevier, 2010.

[30] H Peyton Young. An axiomatization of borda's rule. *Journal of economic theory*, 9(1):43–52, 1974.

[31] Tin Kam Ho, Jonathan J. Hull, and Sargur N. Srihari. Decision combination in multiple classifier systems. *IEEE transactions on pattern analysis and machine intelligence*, 16(1):66–75, 1994.

[32] Nir Ailon, Moses Charikar, and Alantha Newman. Aggregating inconsistent information: ranking and clustering. *Journal of the ACM (JACM)*, 55(5):1–27, 2008.

[33] Leo Breiman, Jerome Friedman, Charles J Stone, and Richard A Olshen. *Classification and regression trees*. CRC press, 1984.

[34] J Ross Quinlan. *C4. 5: programs for machine learning*. Elsevier, 2014.

[35] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[36] Sang-Hyeun Park and Johannes Fürnkranz. Efficient pairwise classification. In *European Conference on Machine Learning*, pages 658–665. Springer, 2007.

[37] Trevor Hastie and Robert Tibshirani. Classification by pairwise coupling. *The annals of statistics*, 26(2):451–471, 1998.

[38] Nicolas Usunier, David Buffoni, and Patrick Gallinari. Ranking with ordered weighted pairwise classification. In *Proceedings of the 26th annual international conference on machine learning*, pages 1057–1064, 2009.

[39] Eyke Hüllermeier and Johannes Fürnkranz. On predictive accuracy and risk minimization in pairwise label ranking. *Journal of Computer and System Sciences*, 76(1):49–62, 2010.

[40] 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*, pages 144–152, 1992.

[41] Li Zhang, Weida Zhou, and Licheng Jiao. Wavelet support vector machine. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 34(1):34–39, 2004.

[42] David Meyer, Friedrich Leisch, and Kurt Hornik. The support vector machine under test. *Neurocomputing*, 55(1-2):169–186, 2003.

[43] William S Noble. What is a support vector machine? *Nature biotechnology*, 24(12):1565–1567, 2006.

[44] Sotiris B Kotsiantis. Decision trees: a recent overview. *Artificial Intelligence Review*, 39(4):261–283, 2013.

[45] J. Ross Quinlan. Simplifying decision trees. *International journal of man-machine studies*, 27(3):221–234, 1987.

[46] J. Ross Quinlan. Induction of decision trees. *Machine learning*, 1(1):81–106, 1986.

[47] Thais Mayumi Oshiro, Pedro Santoro Perez, and José Augusto Baranauskas. How many trees in a random forest? In *International workshop on machine learning and data mining in pattern recognition*, pages 154–168. Springer, 2012.

[48] Eyke Hüllermeier, Johannes Fürnkranz, Weiwei Cheng, and Klaus Brinker. Label ranking by learning pairwise preferences. *Artificial Intelligence*, 172(16-17):1897–1916, 2008.