Εθνικο Μετσοβιο Πολυτεχνειο
Σχολη Ηλεκτρολογων Μηχανικων και Μηχανικων Υπολογιστων
Τομεας Τεχνολογιας Πληροφορικης και Υπολογιστων
Εργαστηριο Λογικης και Επιστημης Υπολογιστων

Αποδοτικοί Αλγόριθμοι Εκμάθησης για Γραμμικά Ενθόρυβα Μοντέλα **Label Ranking**

Διπλωματικη Εργασια

ΓΕΩΡΓΙΟΣ ΓΚΡΙΝΙΑΣ

Επιβλέπων: Δημήτριος Φωτάκης
Καθηγητής Ε.Μ.Π.

Αθήνα, Οκτώβριος 2023

Εθνικο Μετσοβιο Πολυτεχνειο
Σχολη Ηλεκτρολογων Μηχανικων και Μηχανικων Υπολογιστων
Τομεας Τεχνολογιας Πληροφορικης και Υπολογιστων
Εργαστηριο Λογικης και Επιστημης Υπολογιστων

Linear Label Ranking with Noisy Samples

Διπλωματικη Εργασια

ΓΕΩΡΓΙΟΣ ΓΚΡΙΝΙΑΣ

Επιβλέπων: Δημήτριος Φωτάκης
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 24η Οκτωβρίου 2023.

Δημήτριος Φωτάκης Καθηγητής Ε.Μ.Π.	Αριστείδης Παγουρτζής Καθηγητής Ε.Μ.Π.	Χρήστος Τζάμος Αν. Καθηγητής Ε.Κ.Π.Α.

Αθήνα, Οκτώβριος 2023

.....

Γεώργιος Γκρίνιας

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

Περίληψη

Στην παρούσα διπλωματική εργασία, μελετάμε το πρόβλημα Label Ranking (LR), ήτοι το επιβλεπόμενο πρόβλημα εκμάθησης μιας συνάρτησης που αντιστοιχίζει παραδείγματα σε κατατάξεις ενός πεπερασμένου συνόλου προκαθορισμένων ετικετών. Το LR κατέχει καίρια θέση στην περιοχή της εκμάθησης προτιμήσεων και συνιστά αντικείμενο αυξανόμενου ενδιαφέροντος λόγω της εμπλοκής του σε μια πληθώρα τομέων, όπως η στοχευμένη διαφήμιση, η βιοπληροφορική και η μετα-μάθηση. Η συντριπτική πλειοψηφία των εργασιών γύρω από το LR, υιοθετεί μια πρακτική προσέγγιση για το εν λόγω πρόβλημα, προτείνοντας αλγορίθμους στη βάση πειραματικής αξιολόγησης και όχι θεωρητικών αποτελεσμάτων. Ως εκ τούτου, μια από τις κύριες προκλήσεις στην περιοχή του LR αφορά στην κατασκευή αλγορίθμων, οι οποίοι να υποστηρίζονται τόσο από στατιστικές εγγυήσεις, διασφαλίζοντας ικανότητα γενίκευσης σε νέα δεδομένα, όσο και από εχέγγυα αποδοτικότητας, εξασφαλίζοντας υπολογιστική διαχειρισιμότητα. Ο σκοπός της παρούσας εργασίας έχει δύο πτυχές. Ο πρώτος μας στόχος σχετίζεται με τη θεωρητική μελέτη του LR και την επέκταση ορισμένων υπαρχόντων βιβλιογραφικών αποτελεσμάτων. Συγκεκριμένα, επικεντρωνόμαστε στη θεμελιώδη κλάση των Γραμμικών Ταξινομητικών Συναρτήσεων (Linear Sorting Functions ή LSFs), η οποία αντιστοιχεί στη γραμμική εκδοχή του LR, και, βασιζόμενοι στη δουλεία των Fotakis et al. [2022b], παρέχουμε έναν αποδοτικό, κανονικό (proper) αλγόριθμο εκμάθησης LSFs κατά το μοντέλο PAC. Ο αλγόριθμός μας συνοδεύεται από εχέγγυα υπό το καθεστώς των ισοτροπικών λογαριθμικώς κοίλων κατανομών πιθανότητας, ως προς την απόσταση Tau του Kendall και ως προς μοντέλα θορύβου που επεκτείνουν τα μοντέλα θορύβου δυαδικής ταξινόμησης Massart και Tsybakov στην περίπτωση του LR. Ο δεύτερος στόχος μας είναι να διερευνήσουμε πειραματικώς την επίδοση LR αλγορίθμων βασισμένων σε γραμμικές συναρτήσεις σε σύγκριση με αλγορίθμους, οι οποίοι βασίζονται σε δέντρα απόφασης και τυχαία δάση, δεδομένου, ότι οι τελευταίες μέθοδοι θεωρούνται υπερσύγχρονες στην περιοχή του LR. Η πειραματική μας αξιολόγηση λαμβάνει χώρα τόσο σε θορυβώδη σύνολα εκπαίδευσης προερχόμενα από LSFs, όσο και σε τυπικά σύνολα αναφοράς για το LR.

Λέξεις Κλειδιά

Κατάταξη Ετικετών, Μηχανική Μάθηση, Στατιστική Μάθηση, Θεωρία Μάθησης, PAC Εκμάθηση, Μάθηση από Δεδομένα με Θόρυβο, Ημιδιάστημα, Γραμμική Ταξινομητική Συνάρτηση

Abstract

In this thesis, we study the problem of Label Ranking (LR), that is, the supervised task of learning a hypothesis that maps instances to rankings over a finite set of predefined labels. LR holds a dominant position in the field of preference learning and constitutes a topic of increasing attention due to its involvement in a large number of areas, such as targeted advertising, bioinformatics and meta-learning. The vast majority of works on LR adopt a practical approach to this problem, proposing algorithms on the basis of experimental evaluation rather than theoretical results. Hence, one of the main challenges in LR concerns the development of algorithms that are supported by both statistical guarantees, ensuring generalization capability over new data, and efficiency assurances, guaranteeing computational tractability. The purpose of this thesis is twofold. Our first goal is to address the theoretical aspects of LR and extend some of the current literature results. In particular, we focus on the fundamental concept class of Linear Sorting Functions (LSFs), which corresponds to the linear variant of LR, and, building upon the work of Fotakis et al. [2022b], we provide an efficient algorithm that learns LSFs properly in the distribution-dependent PAC model. Our algorithm is accompanied by guarantees under the regime of isotropic logarithmically concave probability distributions, with respect to the Kendall's Tau distance and with respect to noise models that extend the Massart and Tsybakov binary classification noise models to the LR setting. Our second goal is to experimentally investigate the performance of LR algorithms based on linear predictors against LR algorithms based on decision trees and random forests, given that the latter constitute state-of-the-art techniques for LR. The evaluation we conduct is both on noisy data sets originating from LSFs and on standard LR benchmarks.

Keywords

Label Ranking, Machine Learning, Statistical Learning, Learning Theory, PAC Learning, Learning from Noisy Data, Halfspace, Linear Sorting Function

Ευχαριστίες

Κατ' αρχάς, θα ήθελα να ευχαριστήσω τον κύριο Φωτάκη για την ευκαιρία που μου έδωσε να ασχοληθώ με ένα τόσο ενδιαφέρον αντικείμενο και για την εμπιστοσύνη, την ενθάρρυνση και την υποστήριξή του, τόσο ως προς τη διεκπεραίωση της παρούσας εργασίας, όσο και σε προσωπικό επίπεδο. Επίσης, θα ήθελα να ευχαριστήσω θερμά την Ελένη Ψαρουδάκη, τον Άλκη Καλαβάση και τον Βασίλη Κοντονή για την πολύτιμη βοήθεια, τις συμβουλές και την καθοδήγηση που μου παρείχαν κατά τη διάρκεια εκπόνησης της εργασίας.

Γιώργος Γκρίνιας,

Αθήνα, 24η Οκτωβρίου 2023

Contents

Εκτεταμένη Ελληνική Περίληψη **17**
0.1 Εισαγωγή . . . 17
0.1.1 Προγενέστερες Δουλειές πάνω στο Label Ranking . . . 17
0.1.2 Η Συμβολή Μας . . . 19
0.2 Θεωρία Μάθησης . . . 19
0.2.1 Το μοντέλο εκμάθησης PAC . . . 20
0.2.2 Μοντέλα Θορύβου για Δυαδική Ταξινόμηση . . . 21
0.3 Μάθηση Γραμμικών Συναρτήσεων Κατωφλίου . . . 22
0.3.1 Μάθηση Γραμμικών Συναρτήσεων Κατωφλίου χωρίς Θόρυβο . . . 22
0.3.2 Μάθηση Γραμμικών Συναρτήσεων Κατωφλίου με Θόρυβο Massart . . . 22
0.4 Label ranking . . . 24
0.4.1 Μοντέλα Θορύβου για Label Ranking . . . 24
0.4.2 Τεχνικές για Label Ranking . . . 25
0.5 Μάθηση Γραμμικών Ταξινομητικών Συναρτήσεων . . . 27
0.5.1 Μάθηση Γραμμικών Ταξινομητικών Συναρτήσεων χωρίς Θόρυβο . . . 28
0.5.2 Μάθηση Ομογενών Γραμμικών Ταξινομητικών Συναρτήσεων με Θόρυβο . . . 28
0.6 Πειραματικά Αποτελέσματα . . . 29

1 Introduction **31**
1.1 Prior Work . . . 31
1.2 Our Contributions . . . 33
1.3 Organization . . . 33
1.4 Notation . . . 34

2 Learning Theory **35**
2.1 The Statistical Learning Framework . . . 35
2.2 Empirical Risk Minimization . . . 37
2.3 The PAC Learning Model . . . 37
2.3.1 Realizable PAC Learning . . . 37
2.3.2 Agnostic PAC Learning . . . 38
2.3.3 Extensions of the PAC Model . . . 39
2.4 The No-Free-Lunch Theorem . . . 40
2.5 VC Dimension . . . 40
2.6 The Fundamental Theorem of Statistical Learning . . . 41
2.7 Noise Models in Binary Classification . . . 42
2.7.1 The Random Classification Noise Model . . . 42
2.7.2 The Massart Noise Model . . . 43
2.7.3 The Tsybakov Noise Model . . . 44

3 Learning Halfspaces **45**
3.1 Learning Halfspaces in the noiseless setting . . . 46
3.1.1 Learning Halfspaces using Linear Programming . . . 46
3.1.2 Learning Halfspaces using the Perceptron Algorithm . . . 47
3.2 Learning Halfspaces in the noisy setting . . . 48
3.2.1 Prior Work on Halfspace Learning with Noise . . . 48
3.2.2 Learning Halfspaces with Massart Noise . . . 48

4 Label Ranking **53**
4.1 About Rankings . . . 53
4.2 The Label Ranking Problem . . . 54
4.3 Noise Models in Label Ranking . . . 56
4.3.1 Ranking Probability Distributions . . . 56
4.3.2 Label Ranking Probability Distributions . . . 57
4.4 Label Ranking Techniques . . . 58
4.4.1 Label Ranking by Pairwise Decomposition . . . 59
4.4.2 Label Ranking by Labelwise Decomposition . . . 63

5 Learning Linear Sorting Functions **65**
5.1 Learning Linear Sorting Functions in the Noiseless Setting . . . 66
5.2 Learning Homogeneous Linear Sorting Functions in the Noisy Setting . . . 67

6 Experimental Results **75**
6.1 Label Ranking Algorithms under Comparison . . . 75
6.2 Results on Synthetic Data Sets . . . 76
6.3 Results on Semi-synthetic and Real-world Data Sets . . . 81

A Logarithmically Concave Probability Distributions **89**

B The Ellipsoid Method **93**

C Omitted Proofs **95**
C.1 The Proof of Lemma 2.7.1 . . . 95
C.2 The Proof of Lemma 2.7.2 . . . 95
C.3 The Proof of Lemma 5.2.1 . . . 96

List of Figures

1 Αξιολόγηση ως προς τον μέσο συντελεστή συσχέτισης KT σε δεδομένα με θόρυβο Mallows 30

3.1 Visualization of a homogeneous halfspace in $\mathbb{R}^2$ 45
3.2 The step function and the logistic function 49

5.1 Visualization of a homogeneous linear sorting function in $\mathbb{R}^2$ with 3 labels 65

6.1 Evaluation in terms of mean KT coefficient on SFN datasets with Mallows noise 77
6.2 Evaluation in terms of mean KT coefficient on LFN datasets with Mallows noise 77
6.3 Evaluation in terms of mean KT coefficient on SFN datasets with Gaussian additive noise 78
6.4 Evaluation in terms of mean KT coefficient on LFN datasets with Gaussian additive noise 78
6.5 Evaluation of LWLR in terms of mean KT coefficient 79
6.6 Evaluation of LWDT in terms of mean KT coefficient 79
6.7 Evaluation of LWRF in terms of mean KT coefficient 79
6.8 Evaluation of PWHH in terms of mean KT coefficient 80
6.9 Evaluation of PWDT in terms of mean KT coefficient 80
6.10 Evaluation of PWRF in terms of mean KT coefficient 80

List of Tables

1 Αξιολόγηση ως προς τον μέσο συντελεστή συσχέτισης KT σε ημισυνθετικά δεδομένα . . . 30

6.1 Semi-synthetic datasets . . . 81
6.2 Real-world datasets . . . 81
6.3 Evaluation in terms of mean KT coefficient on semi-synthetic datasets . . . 82
6.4 Evaluation in terms of mean KT coefficient on real-world datasets . . . 82

Εκτεταμένη Ελληνική Περίληψη

Το βασικό σκέλος της αυτής της διπλωματικής εργασίας έχει αποδοθεί στην αγγλική γλώσσα. Στο παρόν κεφάλαιο, οι ενότητες του οποίου είναι σε πλήρη αντιστοιχία με τα υπόλοιπα κεφάλαια, παρουσιάζουμε συνοπτικώς το περιεχόμενό της εργασίας, παραλείποντας αποδείξεις και τεχνικές λεπτομέρειες.

0.1 Εισαγωγή

Το πρόβλημα Label Ranking (LR) (κατάταξη ετικετών) αποτελεί ένα αυξανόμενα δημοφιλές αντικείμενο στον τομέα της μηχανικής μάθησης με κεντρικό ρόλο στην περιοχή της μάθησης προτιμήσεων (preference learning) (Fürnkranz and Hüllermeier [2010]). Στόχος του είναι η εύρεση μιας συνάρτησης, η οποία να αντιστοιχίζει χαρακτηριστικά σε κατατάξεις ενός πεπερασμένου συνόλου ετικετών. Το LR έχει λάβει ιδιαίτερη προσοχή τα τελευταία χρόνια, καθώς ανακύπτει σε μια πληθώρα πρακτικών εφαρμογών. Μια από τις πιο χαρακτηριστικές εφαρμογές του αφορούν στην στον τομέα της στοχευμένης διαφήμισης (targeted advertising) (Djuric et al. [2014]), όπου ζητούμενο είναι η εύρεση μιας κατάταξης ενός συνόλου διαφημίσεων για κάθε χρήστη, ώστε να του προβληθούν οι πιο σχετικές βάσει των ενδιαφερόντων του.

Τα τελευταία χρόνια, έχουν υπάρξει σημαντικές δουλείες γύρω από το LR που προτείνουν καινοτόμους αλγορίθμους. Η συντριπτική πλειοψηφία αυτών των αλγορίθμων συνοδεύεται από πειραματική τους αξιολόγηση που αναδεικνύει την επίδοση τους σε πρακτικό επίπεδο, αλλά από λιγοστές θεωρητικές εγγυήσεις. Ως εκ τούτου, μία από τις μεγαλύτερες προκλήσεις σχετικά με το LR αφορά στην υποστήριξη των προαναφερθέντων αποτελεσμάτων από στατιστικά και υπολογιστικά εχέγγυα. Μια άλλη σημαντική πρόκληση αφορά στην κατασκευή αλγορίθμων για το LR, οι οποίοι να μπορούν να διαχειριστούν περιπτώσεις, όπως παραδείγματα εκπαίδευσης με κατατάξεις, στις οποίες δεν είναι όλες οι ετικέτες παρούσες, κατατάξεις με ισοπαλίες μεταξύ των στοιχείων τους ή κατατάξεις που έχουν αλλοιωθεί, ήτοι έχουν τροποποιηθεί, λόγω θορύβου. Ο βασικός στόχος αυτής της διπλωματικής είναι να επεκτείνουμε κάποια από τα υπάρχοντα θεωρητικά αποτελέσματα στην θεμελιώδη περίπτωση του γραμμικού LR και να διερευνήσουμε πειραματικώς την επίδοση αλγορίθμων προσαρμοσμένων στην περίπτωση του γραμμικού LR συγκριτικώς με κάποιους από τους πιο σύγχρονους αλγορίθμους για το γενικό LR που έχουν προταθεί στη βιβλιογραφία.

0.1.1 Προγενέστερες Δουλειές πάνω στο Label Ranking

Κατά την πάροδο των χρόνων, έχουν υπάρξει ποικίλες προσεγγίσεις για το πρόβλημα Label Ranking, οι περισσότερες εκ των οποίων παρουσιάζονται συλλογικώς στις δουλειές των Fürnkranz and Hüllermeier [2010], Vembu and Gärtner [2011], Zhou et al. [2014a]. Οι προσεγγίσεις αυτές μπορούν να ομαδοποιηθούν στις ακόλουθες κατηγορίες.

Μέθοδοι αποσύνθεσης Μια από τις πρώτες LR τεχνικές που υπάγεται στις μεθόδους αποσύνθεσης είναι *ταξινόμηση μέσω περιορισμών* (constraint classification) (Har-Peled et al. [2002]). Ο στόχος της είναι να βρει μια γραμμική συνάρτηση, η οποία να αντιστοιχίζει διανύσματα χαρακτηριστικών σε κατατάξεις, το οποίο επιτυγχάνεται μέσω μετασχηματισμού του αρχικού προβλήματος σε ένα πρόβλημα εύρεσης ομογενών ημιδιαστημάτων (homogeneous halfspaces) εντός ενός χώρου μεγαλύτερης διαστατικότητας.

Μια άλλη τεχνική αποσύνθεσης είναι αυτή των *λογαριθμικώς γραμμικών μοντέλων* (log-linear models) (Dekel et al. [2003]), η οποία επεκτείνει την προηγούμενη μέθοδο, υπό την έννοια, ότι επιχειρεί να μάθει συναρτήσεις που αποτελούν γραμμικό συνδυασμό ενός συνόλου συναρτήσεων βάσης.

Μια γενικότερη τεχνική για το LR είναι η αυτή της *αποσύνθεσης κατά ζεύγη* (pairwise decomposition) ή, αλλιώς, της *κατάταξης μέσω ανά ζεύγη προτιμήσεων* (ranking by pairwise preferences ή RPC) (Fürnkranz and Hüllermeier [2003], Hüllermeier et al. [2008]). Η βασική ιδέα είναι να χωρίσουμε το πρόβλημα LR σε πολλαπλά προβλήματα δυαδικής ταξινόμησης, ένα για κάθε ζεύγος ετικετών, στο οποίο στόχος είναι η

πρόβλεψη της σειράς προτίμησης για το συγκεκριμένο ζεύγος. Τα αποτελέσματα των επιμέρους προβλέψεων μπορούν να συγχωνευτούν σε μια τελική κατάταξη μέσω διαφόρων τρόπων που θα αναλυθούν στη συνέχεια.

Δύο αξιοσημείωτες δουλείες πάνω σε αυτή τη μέθοδο είναι αυτές των Vogel and Clémençon [2020] και Fotakis et al. [2022a]. Η πρώτη παρέχει στατιστικές εγγυήσεις, στην περίπτωση που ο αλγόριθμος έχει πρόσβαση μόνο στην πρώτη ετικέτα κάθε κατάταξης, ενώ η δεύτερη παρέχει παρόμοιες εγγυήσεις για μια γενικότερη περίπτωση ελλιπών διατάξεων υπό ορισμένες παραδοχές. Επιπλέον, αμφότερα τα αποτελέσματα ισχύουν υπό την παρουσία θορύβου. Μια άλλη σημαντική δουλεία σε αυτή την κατεύθυνση είναι αυτή των Fotakis et al. [2022b] για την περίπτωση του γραμμικού LR. Συγκεκριμένα, οι Fotakis et al. [2022b] έδειξαν, ότι η κλάση των γραμμικών ταξινομητικών συναρτήσεων (linear sorting functions) είναι αποδοτικώς PAC εκμαθήσιμη κατά κανονικό (proper) τρόπο στην περίπτωση της πολυδιάστατης τυπικής κανονικής κατανομής, ως προς δύο δημοφιλείς για κατατάξεις συναρτήσεις και υπό την παρουσία θορύβου. Τελικώς, επισημαίνουμε, ότι έχουν προταθεί και πιο περίπλοκες μέθοδοι αποσύνθεσης ανά ζεύγη (Gurrieri et al. [2014]) που λαμβάνουν, επίσης, υπόψιν τη συσχέτιση μεταξύ ετικετών.

Ως εναλλακτική στη μέθοδο αποσύνθεσης κατά ζεύγη, οι Cheng et al. [2013], Cheng and Hüllermeier [2013] πρότειναν την μέθοδο της *αποσύνθεσης κατά ετικέτες* (labelwise decomposition), η οποία χωρίζει το πρόβλημα LR σε πολλαπλά υποπροβλήματα, αυτή τη φορά, ένα για κάθε ετικέτα, στο οποίο στόχος είναι η πρόβλεψης της θέσης της στην τελική κατάταξη. Μια σημαντική σχετική δουλεία είναι αυτή των Fotakis et al. [2022a], η οποία παρείχε τους πρώτους LR αλγορίθμους με εγγυήσεις αποδοτικότητας στο PAC μοντέλο για τη χρήση δέντρων απόφασης.

Μέθοδοι βασισμένες σε στιγμιότυπα Μια σημαντική τεχνική που συχνά χρησιμοποιείται ως τμήμα Label Ranking αλγορίθμων (Cheng and Hüllermeier [2008], Cheng et al. [2009, 2010], Cheng and Hüllermeier [2013]) είναι η τεχνική *μάθησης που βασίζεται σε στιγμιότυπα* (instance-based learning) *instance-based learning* (Brinker and Hüllermeier [2006]). Η βασική της ιδέα είναι, ότι η πρόβλεψη της κλάσης ενός δοθέντος στιγμιοτύπου στηρίζεται σε τοπική πληροφορία, ήτοι στις κλάσεις γειτονικών στιγμιοτύπων. Ο απλούστερος τρόπος για να πραγματοποιηθεί αυτό είναι μέσω του γνωστού αλγορίθμου k-κοντινότερων γειτόνων (k-NN).

Πιθανοτικές μέθοδοι Ένας άλλος δημοφιλής τρόπος αντιμετώπισης του Label Ranking προβλήματος είναι να αναπτύξουμε μεθόδους πρόβλεψης βασισμένες σε στατιστικά μοντέλα πάνω σε κατατάξεις, όπως το μοντέλο Mallows (Mallows [1957]) και το μοντέλο Plackett-Luce (Plackett [1975]), ή άλλα μοντέλα όπως μοντέλα μίξης Γκαουσιανών (Gaussian Mixture Models). Έχουν υπάρξει αρκετές δουλείες πάνω σε αυτήν την κατεύθυνση (Cheng and Hüllermeier [2008], Cheng et al. [2009, 2010, 2012], Cheng and Hüllermeier [2012], Grbovic et al. [2012], Zhou et al. [2014b]), οι περισσότερες εκ των οποίων περιλαμβάνουν ένα instance-based στάδιο και υιοθετούν μεθόδους όπως εκτίμηση μέγιστης πιθανοφάνειας (maximum likelihood estimation) ή μεγιστοποίηση αναμενόμενης τιμής (expectation-maximization) για προσδιορισμό παραμέτρων.

Μέθοδοι βασισμένες σε δέντρα απόφασης Η χρήση Label Ranking αλγορίθμων που είναι βασισμένοι σε δέντρα απόφασης (decision trees) αποτελεί ακόμα μία καινοτόμο τεχνική, η οποία, όπως καταδεικνύεται από πειραματικά αποτελέσματα, είναι υψηλά ανταγωνιστική σε σχέση με τις προαναφερθείσες μεθόδους. Μερικές από τις πιο αξιοσημείωτες δουλείες σε αυτήν την κατεύθυνση περιλαμβάνουν προσαρμογή δέντρων απόφασης (Cheng et al. [2009]), τυχαία δάση (random forests) (Zhou and Qiu [2016]) και σύνολα δέντρων απόφασης (decision tree ensembles) (de Sá et al. [2015], de Sá et al. [2017], Aledo et al. [2017]). Επιπροσθέτως, όπως προαναφέραμε, η δουλεία των Fotakis et al. [2022a], που βασίζεται στην τεχνική αποσύνθεσης βάσει ετικετών, ήταν η πρώτη που υποστήριξε τη χρήση δέντρων απόφασης στο LR με θεωρητικές εγγυήσεις πέραν πειραματικής αξιολόγησης.

Άλλες μέθοδοι Μία καινοτόμος δουλεία που δεν υπάγεται στις ανωτέρω κατηγορίες είναι αυτή των Korba et al. [2018], η οποία ακολουθεί μια δομημένη προσέγγιση πρόβλεψης που αποτελείται από δύο βήματα. Το πρώτο βήμα είναι ένα βήμα παλινδρόμησης σε έναν χώρο Hilbert, όπου οι κατατάξεις αναπαρίστανται από διανύσματα μέσω κατάλληλων συναρτήσεων (embeddings). Το δεύτερο βήμα είναι ένα βήμα αποκωδικοποίησης, το οποίο βοηθά στην ανάκτηση μιας κατάταξης από κάθε πρόβλεψη που κείται σε έναν χώρο Hilbert. Η εν λόγω δουλεία υποστηρίζεται και από θεωρητικές εγγυήσεις για ορισμένες επιλογές από embeddings.

Εν κατακλείδι, υπάρχουν επιπλέον προσεγγίσεις για το LR που βασίζονται σε *μέτρα ομοιότητας* (similarity measures) (Aiguzhinov et al. [2010], de Sá et al. [2011], Ribeiro et al. [2012]), *μεθόδους βασισμένες σε κανόνες* (rule-based methods) (Gurrieri et al. [2012]) ή *επιβλεπόμενη συσταδοποίηση* (supervised clustering) (Grbovic et al. [2013]).

0.1.2 Η Συμβολή Μας

Οι βασικές συνεισφορές αυτής της διπλωματικής έγκεινται στην αποκόμιση ενός νέου θεωρητικού αποτελέσματος στην περιοχή του γραμμικού LR και στη διεξαγωγή μιας πειραματικής αξιολόγησης LR αλγορίθμων. Σχετικά με το θεωρητικό μέρος, στηριζόμαστε στη δουλεία των Fotakis et al. [2022b] και επεκτείνουμε ένα από τα αποτελέσματά τους για μια ευρύτερη οικογένεια κατανομών πιθανότητας. Συγκεκριμένα, δείχνουμε, ότι η κλάση των γραμμικών ταξινομητικών συναρτήσεων (LSFs) είναι αποδοτικά και κανονικά (properly) εκμαθήσιμη κατά το μοντέλο PAC ως προς την απόσταση τ του Kendall υπό την παρουσία ορισμένων μοντέλων θορύβου και υπό την παραδοχή ισοτροπικών λογαριθμικώς κοίλων (isotropic log-concave) κατανομών πιθανότητας.

Όσον αφορά το πειραματικό μέρος, συγκρίνουμε έξι LR αλγορίθμους, συμπεριλαμβανομένων αλγορίθμων που προτάθηκαν από τους Fotakis et al. [2022b], Fotakis et al. [2022a] ως προς την ικανότητα γενίκευσής τους και ως προς την ανθεκτικότητά τους σε θορυβώδη δεδομένα. Στόχος μας είναι να αποκομίσουμε μια εικόνα για την επίδοση LR αλγορίθμων βασισμένων σε γραμμικές συναρτήσεις σε σχέση με ορισμένους state-of-the-art LR αλγορίθμους γενικού σκοπού που βασίζονται σε δέντρα απόφασης και τυχαία δάση.

0.2 Θεωρία Μάθησης

Το γενικό πλαίσιο ενός προβλήματος μηχανικής μάθησης περιλαμβάνει τα εξής.

- Έναν *χώρο παραδειγμάτων* (instance space) $\mathcal{X}$, δηλαδή ένα σύνολο αντικειμένων στα οποία θέλουμε να αποδώσουμε μια ετικέτα. Για παράδειγμα, σε ένα πρόβλημα ταξινόμησης σκύλων βάσει της ράτσας τους, το σύνολο $\mathcal{X}$ θα ταυτιζόταν με το σύνολο όλων των σκύλων.
- Έναν *χώρο ετικετών* (label space) $\mathcal{Y}$, δηλαδή ένα σύνολο που αντιστοιχεί στις *ετικέτες* (labels) που μπορούν να αντιστοιχηθούν σε κάθε αντικείμενο του χώρου παραδειγμάτων. Για παράδειγμα, στο προαναφερθέν πρόβλημα ταξινόμησης σκύλων, το σύνολο $\mathcal{Y}$ θα ταυτιζόταν με το σύνολο όλων των ρατσών. Η περίπτωση που το $\mathcal{Y}$ περιλαμβάνει μόνο δύο ετικέτες, όπου συνήθως επιλέγεται $\mathcal{Y} = \{0, 1\}$ ή $\mathcal{Y} = \{-1, 1\}$, αντιστοιχεί στη θεμελιώδη κατηγορία των προβλημάτων δυαδικής ταξινόμησης (binary classification).
- Συναρτήσεις $h \colon \mathcal{X} \to \mathcal{Y}$ που καλούνται *υποθέσεις* (hypotheses) και οι οποίες αντιστοιχίζουν τα στοιχεία του χώρου παραδειγμάτων σε ετικέτες του χώρου ετικετών. Ένα σύνολο υποθέσεων $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ καλείται *κλάση υποθέσεων* (hypothesis class).
- *Αλγορίθμους εκμάθησης* (learning algorithms ή learners), οι οποίοι λαμβάνουν ως είσοδο ένα πεπερασμένο *σύνολο εκπαίδευσης* με στοιχεία $(x, y) \in \mathcal{X} \times \mathcal{Y}$ και επιστρέφουν μια υπόθεση εντός του $\mathcal{Y}^{\mathcal{X}}$. Συμβολίζουμε με $\mathcal{A}_{\mathcal{X},\mathcal{Y}}$ το σύνολο αυτών των αλγορίθμων.

Όσον αφορά τα δεδομένα εκπαίδευσης, θεωρούμε, ότι υπάρχει κάποια άγνωστη κατανομή πιθανότητας $\mathcal{D}$, από την οποία τα παράγονται τα δεδομένα. Επιπλέον, θεωρούμε, ότι για κάθε σύνολο εκπαίδευσης S τα στοιχεία του αποτελούν ανεξάρτητα και ταυτόσημα κατανεμημένες (independently and identically distributed ή i.i.d.) τυχαίες μεταβλητές που ακολουθούν την $\mathcal{D}$, το οποίο συμβολίζεται ως $S \sim \mathcal{D}^m$, όπου m ο πληθάριθμος του S.

Ανεπίσημα, ο στόχος ενός αλγορίθμου εκμάθησης είναι να επιστρέψει μια υπόθεση, της οποίας οι προβλέψεις τείνουν να είναι ορθές. Για να επισημοποιήσουμε την έννοια της επιτυχίας ενός αλγορίθμου εκμάθησης, χρειάζεται να ορίσουμε πρώτα μια συνάρτηση σφάλματος (loss function) $\ell \colon \mathcal{Y}^2 \to \mathbb{R}_{\geq 0}$, η οποία θα ποσοτικοποιεί πόσο απέχει μια πρόβλεψη από μια τιμή αναφοράς. Για προβλήματα ταξινόμησης (classification), η πιο συνήθης και απλή επιλογή είναι η $0-1$ απώλεια, που ορίζεται ως $\ell_{0-1}(\hat{y}, y) = \mathbb{1}\{\hat{y} \neq y\}$. Για προβλήματα παλινδρόμησης (regression), μια καταλληλότερη επιλογή θα ήταν το σφάλμα $\ell_1(\hat{y}, y) = |\hat{y} - y|$ ή το $\ell_2(\hat{y}, y) = (\hat{y} - y)^2$, εφόσον τα τελευταία αποδίδουν καλύτερα την απόσταση μιας πρόβλεψης από κάποια τιμή αναφοράς.

Έχοντας καθορίσει μια κατάλληλη συνάρτηση απώλειας ℓ για το πρόβλημά μας, ορίζουμε το *σφάλμα* (error) μιας υπόθεσης h ως προς την κατανομή $\mathcal{D}$ ως $L_{\mathcal{D},\ell}(h) \triangleq \mathbf{E}_{(x,y)\sim\mathcal{D}}[\ell(h(x), y)]$, ήτοι, ως την αναμενόμενη απώλεια ως προς την κατανομή των δεδομένων. Επιπλέον, για τις περιπτώσεις, όπου υπάρχει μια συνάρτηση στόχος f, την οποία θέλουμε να προσεγγίσουμε, ορίζουμε το σφάλμα της h ως προς $\mathcal{D}$ και f ως $L_{\mathcal{D}_x,f,\ell}(h) \triangleq \mathbf{E}_{x\sim\mathcal{D}_x}[\ell(h(x), f(x))]$, όπου $\mathcal{D}_x$ η περιθώρια κατανομή της $\mathcal{D}$ στον $\mathcal{X}$. Τότε, πιο επίσημα, ο στόχος ενός αλγορίθμου εκμάθησης είναι να επιστρέψει μια υπόθεση που ελαχιστοποιεί το $L_{\mathcal{D},\ell}$ ή $L_{\mathcal{D}_x,f,\ell}$ (αναλόγως του εκάστοτε προβλήματος).

Τέλος, ένα ακόμη χρήσιμο μέτρο σφάλματος είναι η εμπειρικό σφάλμα που ορίζεται ως $\hat{L}_{S,\ell}(h) \triangleq m^{-1}\sum_{t=1}^{m}\ell\left(h\left(x^{(t)}\right), y^{(t)}\right)$, όπου $S = \left(\left(x^{(1)}, y^{(1)}\right), \ldots, \left(x^{(m)}, y^{(m)}\right)\right) \in (\mathcal{X}\times\mathcal{Y})^m$ ένα σύνολο εκπαίδευσης.

0.2.1 Το μοντέλο εκμάθησης PAC

Ακολούθως, παρουσιάζουμε το μοντέλο της *πιθανώς προσεγγιστικώς ορθής εκμάθησης* (Probably Approximately Correct ή PAC learning model) που εισήχθη από τον Valiant [1984] και που αποτελεί έναν τρόπο επισημοποίησης της έννοιας της εκμαθησιμότητας (learnability). Για τη συνέχεια της ανάλυσης, θεωρούμε έναν χώρο στιγμιοτύπων $\mathcal{X}$, έναν χώρο ετικετών $\mathcal{Y}$, μια κλάση υποθέσεων $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$, μια συνάρτηση απωλειών $\ell \colon \mathcal{Y}^2 \to \mathbb{R}_{\geq 0}$ και έναν αλγόριθμο εκμάθησης $A \in \mathcal{A}_{\mathcal{X},\mathcal{Y}}$.

Στην παρούσα ενότητα, περιοριζόμαστε στην απλούστερη *πραγματοποιήσιμη* (realizable) έκδοση του μοντέλου εκμάθησης PAC, όπου θεωρείται πως υπάρχει μια συνάρτηση στόχος που επιφέρει μηδενικό σφάλμα ως προς την $\mathcal{D}$ στην κλάση υποθέσεων $\mathcal{H}$ υπό μελέτη. Ο στόχος μας είναι να βρούμε μια υπόθεση, η οποία, με μεγάλη πιθανότητα, να επιτυγχάνει σφάλμα κοντά στο μηδέν.

Ορισμός 0.2.1 (Πραγματοποιησιμότητα)**.** *Μια κατανομή πιθανότητας $\mathcal{D}$ στον $\mathcal{X} \times \mathcal{Y}$ καλείται πραγαματοποιήσιμη (realizable) από την $\mathcal{H}$, αν υπάρχει κάποια υπόθεση $h \in \mathcal{H}$, τέτοια, ώστε $L_{\mathcal{D},\ell}(h) = 0$.*

Ορισμός 0.2.2 (Πραγματοποιήσιμη Δειγματική Πολυπλοκότητα ενός Αλγορίθμου Εκμάθησης)**.** *Η πραγματοποιήσιμη δειγματική πολυπλοκότητα (realizable sample complexity) του A ως προς $\mathcal{H}$ και ℓ είναι η συνάρτηση $m^r_{A,\mathcal{H},\ell} \colon (0,\infty)^2 \to \mathbb{N}$ που ορίζεται ως εξής: Για κάθε $\epsilon, \delta > 0$, το $m^r_{A,\mathcal{H},\ell}(\epsilon,\delta)$ ταυτίζεται με τον ελάχιστο ακέραιο, για τον οποίο, για κάθε ακέραιο $m \geq m^r_{A,\mathcal{H},\ell}(\epsilon,\delta)$ και κάθε κατανομή πιθανότητας $\mathcal{D}$ στον $\mathcal{X} \times \mathcal{Y}$ που είναι πραγματοποιήσιμη από την $\mathcal{H}$, ισχύει:*

$$\mathop{\mathbf{Pr}}_{S \sim \mathcal{D}^m} \left[L_{\mathcal{D},\ell}\left(A(S)\right) > \epsilon \right] \leq \delta$$

Για κάθε $\epsilon, \delta > 0$, τέτοια, ώστε η ανωτέρω ανισότητα να μην ικανοποιείται από κάποιον ακέραιο, ορίζουμε $m^r_{A,\mathcal{H},\ell}(\epsilon,\delta) = \infty$.

Ορισμός 0.2.3 (Πραγματοποιήσιμη PAC Εκμαθησιμότητα)**.** *Η $\mathcal{H}$ λέγεται PAC εκμαθήσιμη κατά πραγματοποιήσιμο τρόπο με τον A ως προς ℓ, αν $m^r_{A,\mathcal{H},\ell}(\epsilon,\delta) < \infty$ για κάθε $\epsilon, \delta > 0$.*

Η παράμετρος ϵ (παράμετρος ορθότητας) στους ανωτέρω ορισμούς καθορίζει το πόσο μεγάλο μπορεί να γίνει το σφάλμα μιας υπόθεσης που επιστρέφεται από τον αλγόριθμο A (και αντιστοιχεί στο "approximately" κομμάτι του PAC). Η παράμετρος δ (παράμετρος εμπιστοσύνης) σχετίζεται με την πιθανότητα να ισχύει το παραπάνω, δηλαδή το σφάλμα να φράσσεται άνω από ϵ, (και αντιστοιχεί στο "probably" κομμάτι του PAC). Συγκεκριμένα, η παράμετρος δ αποδίδει την εξάρτηση της διαδικασίας της εκπαίδευσης στο συγκεκριμένο σύνολο εκπαίδευσης S που χρησιμοποιήθηκε. Εφόσον το S είναι πεπερασμένο, υπάρχει πάντοτε μια πιθανότητα το S να μην είναι αντιπροσωπευτικό της $\mathcal{D}$, κατά τρόπο που να μην μπορεί να διασφαλιστεί, ότι το σφάλμα είναι μικρότερο από ϵ.

Κανονική και Μη Κανονική PAC Εκμάθηση Στον ανωτέρω ορισμό της PAC εκμαθησιμότητας μιας κλάσης $\mathcal{H}$, οποιοσδήποτε αλγόριθμος ζητείται να επιστρέψει μια υπόθεση h εντός του $\mathcal{Y}^{\mathcal{X}}$, αλλά όχι απαραιτήτως εντός του $\mathcal{H}$. Αυτός ο τύπος μάθησης καλείται *μη κανονικός* (improper). Ωστόσο, κάποιες φορές είναι προτιμότερο η υπόθεση που επιστρέφεται να ανήκει στην κλάση $\mathcal{H}$ (π.χ. για υπολογιστικούς λόγους). Αν επιβάλουμε ένας αλγόριθμος να επιστρέφει υποθέσεις εντός της $\mathcal{H}$, τότε, ο αντίστοιχος τύπος μάθησης καλείται *κανονικός* (proper).

Αποδοτική PAC Εκμάθηση Ο προαναφερθείς ορισμός της PAC εκμαθησιμότητας έχει αμιγώς στατιστικό χαρακτήρα και δεν συμπεριλαμβάνει τις υπολογιστικές πτυχές της εκμαθησιμότητας. Συγκεκριμένα, κατά την κατασκευή αλγορίθμων εκμάθησης, πρέπει επίσης να φροντίσουμε τόσο η αποκόμιση μίας υπόθεσης, όσο και η χρήση της για την πρόβλεψη της ετικέτας ενός νέου παραδείγματος να γίνονται με αποδοτικό τρόπο. Για παράδειγμα, στην ειδική περίπτωση των προβλημάτων δυαδικής ταξινόμησης, όπου $\mathcal{X} = \mathbb{R}^d$, συνήθως, απαιτούμε η δειγματική πολυπλοκότητα, ο χρόνος εκτέλεσης ενός αλγορίθμου και ο χρόνος πρόβλεψης μιας ετικέτας για οποιοδήποτε νέο στιγμιότυπο από την υπόθεση που επιστρέφει ο αλγόριθμος να φράσσονται από ένα πολυώνυμο στη διάσταση d των στιγμιοτύπων, στο μέγεθος αναπαράστασης των στιγμιοτύπων και στις PAC παραμέτρους $1/\epsilon$ και $1/\delta$. Στην περίπτωση που το πρόβλημα μας χαρακτηρίζεται και από επιπρόσθετες παραμέτρους, επιθυμούμε τα ανωτέρω μεγέθη να φράσσονται πολυωνυμικώς και από αυτές.

Εξαρτώμενη από την Κατανομή PAC Εκμάθηση Το τυπικό μοντέλο της πραγματοποιήσιμης PAC εκμάθησης αναφέρεται συνήθως ως *ανεξάρτητο της κατανομής* (distribution-independent), διότι δεν υπάρχει κάποιος περιορισμός για την περιθώρια κατανομή $\mathcal{D}_x$ της $\mathcal{D}$ στον $\mathcal{X}$. Στην πράξη, αν δεν επιβάλουμε

περιορισμούς στη $\mathcal{D}_x$, τότε μπορεί να καταστεί δύσκολο να αποδείξουμε την PAC εκμαθησιμότητα των κλάσεων, ειδικά υπό την παρουσία θορύβου. Για αυτό, σε αρκετές περιπτώσεις, στρεφόμαστε στο λεγόμενο μοντέλο PAC εκμάθησης *με εξάρτηση από την κατανομή* (distribution-dependent), το οποίο επιβάλλει η $\mathcal{D}_x$ να ανήκει σε μια συγκεκριμένη οικογένεια $\mathcal{F}$ κατανομών πιθανότητας στον $\mathcal{X}$.

0.2.2 Μοντέλα Θορύβου για Δυαδική Ταξινόμηση

Το πρώτο και απλούστερο εκ των μοντέλων θορύβου που θα παρουσιάσουμε, είναι το μοντέλο *τυχαίου θορύβου ταξινόμησης* (random classification noise ή RCN) (Angluin and Laird [1988]).

Ορισμός 0.2.4 (Τυχαίος Θόρυβος Ταξινόμησης)**.** *Μια κατανομή πιθανότητας $\mathcal{D}$ στον $\mathcal{X} \times \{\pm 1\}$ ικανοποιεί τη συνθήκη τυχαίου θορύβου ταξινόμησης, αν υπάρχει κάποια (μοναδική) συνάρτηση $f \colon \mathcal{X} \to \{\pm 1\}$ και κάποιο $\eta \in [0, 1/2)$, έτσι, ώστε για κάθε $x \in \mathcal{X}$, να ισχύει $\mathbf{Pr}_{y \sim \mathcal{D}_{y|x}}[y \neq f(x) \mid x] = \eta$. Τότε, η $\mathcal{D}$ καλείται, επίσης, (η, f)-RCN κατανομή.*

Το μοντέλο RCN μπορεί να ερμηνευθεί διαισθητικώς ως εξής: Διαθέτουμε ένα σύνολο εκπαίδευσης δίχως θόρυβο, το οποίο, δηλαδή, περιέχει παραδείγματα των οποίων οι ετικέτες καθορίζονται ντετερμινιστικά από μια συνάρτηση f, και κάποιος αντίπαλος (adversary) αλλάζει την ετικέτα κάθε παραδείγματος ανεξάρτητα με πιθανότητα $\eta \in [0, 1/2)$.

Ένα μειονέκτημα του μοντέλου RCN είναι, ότι υιοθετεί τη μη ρεαλιστική υπόθεση, ότι η πιθανότητα $\mathbf{Pr}_{y \sim \mathcal{D}_{y|x}}[y \neq f(x) \mid x]$ ο αντίπαλος να αλλάξει την ετικέτα κάποιου παραδείγματος είναι η ίδια για όλα τα παραδείγματα. Σε ένα πραγματικό σενάριο, ο αντίπαλος θα μπορούσε να αλλάζει με διαφορετική πιθανότητα την ετικέτα κάθε παραδείγματος. Αυτό αποτελεί κίνητρο για την μελέτη του γενικότερου μοντέλου θορύβου Massart (Massart and Nédélec [2006]).

Ορισμός 0.2.5 (Θόρυβος Massart)**.** *Μια κατανομή πιθανότητας $\mathcal{D}$ στον $\mathcal{X} \times \{\pm 1\}$ ικανοποιεί τη συνθήκη θορύβου Massart, αν υπάρχει κάποια (μοναδική) συνάρτηση $f \colon \mathcal{X} \to \{\pm 1\}$ και κάποιο $\eta \in [0, 1/2)$, έτσι, ώστε για κάθε $x \in \mathcal{X}$, να ισχύει $\mathbf{Pr}_{y \sim \mathcal{D}_{y|x}}[y \neq f(x) \mid x] \leq \eta$. Τότε, η $\mathcal{D}$ καλείται, επίσης, (η, f)-Massart κατανομή.*

Το μοντέλο Massart μπορεί να ερμηνευθεί διαισθητικώς ως εξής: Διαθέτουμε ένα σύνολο εκπαίδευσης δίχως θόρυβο, το οποίο, δηλαδή, περιέχει παραδείγματα των οποίων οι ετικέτες καθορίζονται ντετερμινιστικά από μια συνάρτηση f, και κάποιος αντίπαλος (adversary) αλλάζει την ετικέτα κάθε παραδείγματος ανεξάρτητα με πιθανότητα το πολύ $\eta \in [0, 1/2)$. Συνεπώς, στο τρέχον μοντέλο επιβάλουμε μόνο ένα άνω φράγμα στην πιθανότητα αλλαγής της ετικέτας του κάθε δείγματος από τον αντίπαλο και δεν προσδιορίζουμε επακριβώς την τιμή της, όπως στο μοντέλο RCN.

Παρόλο που το μοντέλο Massart αποτελεί σημαντική γενίκευση του μοντέλου RCN, το γεγονός, ότι θέτει ένα άνω φράγμα στην πιθανότητα αλλαγής της ετικέτας του κάθε δείγματος από τον αντίπαλο εξακολουθεί να είναι περιοριστικό. Για παράδειγμα, το μοντέλο Massart αποτυγχάνει να αποδώσει το ρεαλιστικό σενάριο, στο οποίο η ανωτέρω πιθανότητα μπορεί να φτάσει αυθαιρέτως κοντά στο $1/2$ για κάποια στοιχεία του χώρου παραδειγμάτων. Ως εκ τούτου, καθίσταται αναγκαία η μελέτη και ενός ακόμη γενικότερου μοντέλου, του μοντέλου Tsybakov (Mammen and Tsybakov [1999], Tsybakov [2004]).

Ορισμός 0.2.6. *Μια κατανομή πιθανότητας $\mathcal{D}$ στον $\mathcal{X} \times \{\pm 1\}$ ικανοποιεί τη συνθήκη θορύβου Tsybakov, αν υπάρχει κάποια συνάρτηση $f \colon \mathcal{X} \to \{\pm 1\}$ και κάποια $\alpha \in [0, 1)$ και $B \geq 1$, έτσι, ώστε για κάθε $t \geq 0$, να ισχύει $\mathbf{Pr}_{x \sim \mathcal{D}_x}\left[\mathbf{Pr}_{y \sim \mathcal{D}_{y|x}}[y \neq f(x) \mid x] \geq 1/2 - t/2\right] \leq B t^{\frac{\alpha}{1-\alpha}}$. Τότε, η $\mathcal{D}$ καλείται, επίσης, (α, B, f)-Tsybakov κατανομή.*

Στο μοντέλο Tsybakov, η ποσότητα $\mathbf{Pr}_{y \sim \mathcal{D}_{y|x}}[y \neq f(x) \mid x]$ μπορεί να λάβει τιμές αυθαίρετα κοντά στο $1/2$, αλλά αυτό συμβαίνει με πιθανότητα που τείνει στο μηδέν, όσο περισσότερο πλησιάζουμε στο $1/2$.

Ακολούθως προσαρμόζουμε το PAC μοντέλο για την περίπτωση του θορύβου Massart. Κατά εντελώς ανάλογο τρόπο αυτό μπορεί να γίνει και για τα μοντέλα RCN και Tsybakov (βλ. Κεφάλαιο 2). Έστω $\mathcal{X}$ ένας χώρος παραδειγμάτων, $\mathcal{H} \subseteq \{\pm 1\}^{\mathcal{X}}$, $\mathcal{F}$ μια οικογένεια κατανομών στον $\mathcal{X}$ και $A \in \mathcal{A}_{\mathcal{X}, \{\pm 1\}}$.

Ορισμός 0.2.7 (Δειγματική Πολυπλοκότητα Massart)**.** *Η δειγματική πολυπλοκότητα Massart του A ως προς $\mathcal{H}$ και $\mathcal{F}$ είναι η συνάρτηση $m^{\text{Massart}}_{A,\mathcal{H},\mathcal{F}} \colon (0, \infty) \times (0, \infty) \times [0, 1/2) \to \mathbb{N}$ που ορίζεται ως εξής: Για κάθε $\epsilon, \delta > 0$ και $\eta \in [0, 1/2)$, το $m^{\text{Massart}}_{A,\mathcal{H},\mathcal{F}}(\epsilon, \delta, \eta)$ ταυτίζεται με τον ελάχιστο ακέραιο, για τον οποίο, για κάθε ακέραιο $m \geq m^{\text{Massart}}_{A,\mathcal{H},\mathcal{F}}(\epsilon, \delta, \eta)$, $f \in \mathcal{H}$ και (η, f)-Massart κατανομή, της οποίας η περιθώρια κατανομή $\mathcal{D}_x$ στον $\mathcal{X}$ ανήκει στην $\mathcal{F}$, ισχύει*[1]*:*

$$\underset{S_x \sim \mathcal{D}_x^m}{\mathbf{Pr}}\left[L_{\mathcal{D}_x, f, \ell_{0-1}}\left(A\left(S_x, h|_{S_x}\right)\right) > \epsilon\right] \leq \delta\,.$$

[1] Συμβολίζουμε $A\left(S_x, h|_{S_x}\right) = A(S)$, όπου $S = ((x_1, h(x_1)), \ldots, (x_m, h(x_m)))$ και $S_x = (x_1, \ldots, x_m)$.

Για κάθε $\epsilon, \delta > 0$ και $\eta \in [0, 1/2)$, για τα οποία η ανωτέρω ανισότητα δεν ικανοποιείται από κάποιον ακέραιο, ορίζουμε $m^{\text{Massart}}_{A,\mathcal{H},\mathcal{F}}(\epsilon, \delta, \eta) = \infty$.

Ορισμός 0.2.8 (PAC Εκμαθησιμότητα με Massart Θόρυβο)**.** *Η $\mathcal{H}$ λέγεται PAC εκμαθήσιμη με τον A ως προς $\mathcal{F}$ υπό την παρουσία Massart θορύβου, αν $m^{\text{Massart}}_{A,\mathcal{H},\mathcal{F}}(\epsilon, \delta, \eta) < \infty$ για κάθε $\epsilon, \delta > 0$ και $\eta \in [0, 1/2)$.*

0.3 Μάθηση Γραμμικών Συναρτήσεων Κατωφλίου

Ορισμός 0.3.1 (Γραμμική Συνάρτηση Κατωφλίου)**.** *Ως ημιδιάστημα (halfspace) ή γραμμική συνάρτηση κατωφλίου (linear threshold function) στον χώρο $\mathbb{R}^d$ ορίζεται οποιαδήποτε συνάρτηση $h_{\boldsymbol{w},b} \colon \mathbb{R}^d \to \{\pm 1\}$ της μορφής $h_{\boldsymbol{w},b}(\boldsymbol{x}) = \text{sign}(\langle \boldsymbol{w}, \boldsymbol{x} \rangle + b)$, όπου $\boldsymbol{w} \in \mathbb{R}^d$ και $b \in \mathbb{R}$.*

Ένα ημιδιάστημα $h_{\boldsymbol{w},b}$ λέγεται ομογενές, αν $b = 0$, δηλαδή αν το υπερεπίπεδο που ορίζει περιέχει την αρχή των αξόνων, όπου χρησιμοποιούμε τον συμβολισμό $h_{\boldsymbol{w}} = h_{\boldsymbol{w},0}$. Συμβολίζουμε με $\mathcal{H}^d_{\text{LTF}}$ (αντιστοίχως $\mathcal{H}^d_{\text{HLTF}}$) την κλάση των ημιδιαστημάτων (αντιστοίχως ομογενών ημιδιαστημάτων) στον $\mathbb{R}^d$.

0.3.1 Μάθηση Γραμμικών Συναρτήσεων Κατωφλίου χωρίς Θόρυβο

Αποδεικνύεται, ότι η κλάση των γραμμικών συναρτήσεων κατωφλίου στον $\mathbb{R}^d$ είναι αποδοτικώς PAC εκμαθήσιμη κατά πραγματοποιήσιμο τρόπο (δηλαδή στην απουσία θορύβου) ως προς ℓ_{0-1}. Έστω ένα σύνολο εκπαίδευσης $S = \left(\left(\boldsymbol{x}^{(1)}, y^{(1)} \right), \ldots, \left(\boldsymbol{x}^{(m)}, y^{(m)} \right) \right)$ με δείγματα στον $\mathbb{R}^d \times \{\pm 1\}$. Θεωρούμε το ακόλουθο γραμμικό πρόγραμμα, το οποίο συμβολίζουμε με LP1:

$$\text{Find} \qquad \boldsymbol{w} \in \mathbb{R}^d,\ b \in \mathbb{R}$$

$$\text{subject to} \quad y^{(t)} \left(\left\langle \boldsymbol{w}, \boldsymbol{x}^{(t)} \right\rangle + b \right) \geq 1 \quad \forall t \in [m]$$

Μπορούμε εύκολα να διαπιστώσουμε, ότι κάτω από την υπόθεση της πραγματοποιησιμότητας, το παραπάνω γραμμικό πρόγραμμα είναι σχεδόν βεβαίως εφικτό και οποιαδήποτε λύση του εξασφαλίζει μηδενικό σφάλμα στο σύνολο εκπαίδευσης S. Παράλληλα, μπορούμε να δείξουμε (Θεώρημα 0.3.1), ότι χρησιμοποιώντας έναν επαρκή αριθμό δειγμάτων εκπαίδευσης, επιτυγχάνεται η ζητούμενη (ϵ, δ)-PAC εγγύηση.

Αλγόριθμος 1 Μάθηση Γραμμικών Συναρτήσεων Κατωφλίου με Γραμμικό Προγραμματισμό

Είσοδος: Σύνολο εκπαίδευσης $S = \left(\left(\boldsymbol{x}^{(1)}, y^{(1)} \right), \ldots, \left(\boldsymbol{x}^{(m)}, y^{(m)} \right) \right)$
Έξοδος: LTF $h_{\boldsymbol{w},b} \colon \mathbb{R}^d \to \{\pm 1\}$

1: Κατασκεύασε το LP1 από το S
2: $(\boldsymbol{w}, b) \leftarrow$ Ellipsoid(LP1) ▷ Βλ. Appendix B
3: **return** $h_{\boldsymbol{w},b}$

Θεώρημα 0.3.1. *Η κλάση $\mathcal{H}^d_{\text{LTF}}$ είναι PAC εκμαθήσιμη κατά κανονικό και πραγματοποιήσιμο τρόπο με τον Αλγόριθμο 1 (ως προς ℓ_{0-1}) με δειγματική πολυπλοκότητα $O\left(\left(d \log(1/\epsilon) + \log(1/\delta) \right) / \epsilon \right)$ και πολυωνυμικό χρόνο εκτέλεσης ως προς τη διάσταση d, τον αριθμό των δειγμάτων και το μέγεθος αναπαράστασης των πραγματικών αριθμών.*

0.3.2 Μάθηση Γραμμικών Συναρτήσεων Κατωφλίου με Θόρυβο Massart

Το πρώτο μοντέλο θορύβου μεταξύ των RCN, Massart και Tsybakov, για το οποίο βρέθηκαν θεωρητικές εγγυήσεις σχετικά με τη μάθηση ημιδιαστημάτων ήταν το RCN. Συγκεκριμένα, έχει αποδειχθεί, ότι η κλάση των ημιδιαστημάτων είναι αποδοτικώς και κανονικώς εκμαθήσιμη κατά το PAC μοντέλο με RCN και ανεξαρτήτως κατανομής. Αμέσως μετά ως προς τη δυσκολία απόκτησης θεωρητικών εγγυήσεων, έρχεται το μοντέλο Massart, για το οποίο όλες οι σχετικές στη βιβλιογραφία δουλείες συνοδεύονται από παραδοχές, όπως περιορισμό σε ομογενή ημιδιαστήματα ή σε συγκεκριμένες οικογένειες κατανομών.

Στην παρούσα εργασία, εμβαθύνουμε στη δουλειά των Diakonikolas et al. [2020a], οι οποίοι παρείχαν έναν αποδοτικό αλγόριθμο για την κανονική εκμάθηση ομογενών ημιδιαστημάτων με Massart θόρυβο κατά το PAC μοντέλο, κάτω από μια ευρεία οικογένεια κατανομών πιθανότητας που περιέχει και τις ισοτροπικές λογαριθμικώς κοίλες κατανομές. Χάριν απλότητας, το αντίστοιχο θεώρημα (Θεώρημα 0.3.2) θα παρατεθεί για την περίπτωση της κλάσης των ισοτροπικών λογαριθμικώς κοίλων κατανομών[2].

[2] Με $\mathcal{F}^d_{\text{LC}}$ συμβολίζουμε την κλάση των ισοτροπικών λογαριθμικώς κοίλων κατανομών πιθανότητας στον $\mathbb{R}^d$.

Η προσέγγιση των Diakonikolas et al. [2020a] βασίζεται στην ακόλουθη ιδέα. Για να βρούμε κάποιο $\boldsymbol{w}$ που ελαχιστοποιεί το $L_{\mathcal{D}_{\boldsymbol{x}},f,\ell_{0-1}}(h_{\boldsymbol{w}})$, μια φυσιολογική προσέγγιση είναι να επιχειρήσουμε να ελαχιστοποιήσουμε το $L_{\mathcal{D},\ell_{0-1}}(h_{\boldsymbol{w}})$, χρησιμοποιώντας δείγματα από την ψορυβώδη κατανομή $\mathcal{D}$. Για να το πετύχουμε αυτό, αρκεί να ελαχιστοποιήσουμε τη συνάρτηση $\mathbf{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}\left[\mathbb{1}\{-y\langle \boldsymbol{w},\boldsymbol{x}\rangle/\|\boldsymbol{w}\|_2\}\geq 0\right]$. Ωστόσο, είναι δύσκολο να ελαχιστοποιήσουμε αποδοτικά μια τέτοια μη κυρτή συνάρτηση. Αντί αυτού, οι Diakonikolas et al. [2020a] εντοπίζουν ένα μη κυρτό (nonconvex), αλλά ομαλό (smooth) υποκατάστατο της ανωτέρω συνάρτησης με την ιδιότητα, ότι κάθε προσεγγιστικώς στάσιμο σημείο αυτού του υποκαταστάτου αντιστοιχεί σε κάποιο ημιδιάστημα που είναι κοντά στο ημιδιάστημα στόχο. Συγκεκριμένα, παρακινούμενοι από το γεγονός, ότι η συνάρτηση $S_\sigma(t)\triangleq\frac{1}{1+e^{-t/\sigma}}$, όπου $\sigma>0$, αποτελεί μια καλή προσέγγιση της $\mathbb{1}\{t\geq 0\}$ όταν $\sigma\to 0$, χρησιμοποιούν ως υποκατάστατο τη συνάρτηση

$$\mathcal{L}_\sigma(\boldsymbol{w})=\mathbf{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}\left[g_\sigma\left((\boldsymbol{x},y),\boldsymbol{w}\right)\right],$$

όπου $g_\sigma\left((\boldsymbol{x},y),\boldsymbol{w}\right)=S_\sigma\left(-y\langle\boldsymbol{w},\boldsymbol{x}\rangle/\|\boldsymbol{w}\|_2\right)$.

Αλγόριθμος 2 Προβαλλόμενη Στοχαστική Κατάβαση Δυναμικού για την $\mathbf{E}_{\boldsymbol{z}\sim\mathcal{D}}[g(\boldsymbol{z},\boldsymbol{w})]$

Είσοδος: Συνάρτηση $g(\boldsymbol{z},\boldsymbol{w})$, σύνολο εκπαίδευσης $S=\left(\boldsymbol{z}^{(1)},\ldots,\boldsymbol{z}^{(T)}\right)$ και βήμα β
Έξοδος: Πλειάδα από T διανύσματα στον $\mathbb{S}^{d-1}$

procedure PSGD(g,S,β)
  $\boldsymbol{w}^{(0)}\leftarrow(1,0,\ldots,0)$
  for $t=1,\ldots,T$ **do**
    $\boldsymbol{v}^{(t)}\leftarrow\boldsymbol{w}^{(t-1)}-\beta\nabla_{\boldsymbol{w}}g\left(\boldsymbol{z}^{(t)},\boldsymbol{w}^{(t-1)}\right)$
    $\boldsymbol{w}^{(t)}\leftarrow\boldsymbol{v}^{(t)}/\left\|\boldsymbol{v}^{(t)}\right\|_2$
  end for
  return $\left(\boldsymbol{w}^{(1)},\ldots,\boldsymbol{w}^{(T)}\right)$
end procedure

Έπειτα, συνεχίζουμε εκτελώντας Προβαλλόμενη Στοχαστική Κατάβαση Δυναμικού (Projected Stochastic Gradient Descent ή PSGD) πάνω στη συνάρτηση $\mathcal{L}_\sigma$, με προβολή στην μοναδιαία σφαίρα S^{d-1}, με στόχο να βρούμε ένα προσεγγιστικώς στάσιμο σημείο. Η PSGD επιστρέφει μια συλλογή διανυσμάτων βαρών, που εγγυημένα περιέχουν κάποιο προσεγγιστικώς στάσιμο σημείο (δηλαδή ένα διάνυσμα $\boldsymbol{w}$, τέτοιο, ώστε το $\|\nabla_{\boldsymbol{w}}\mathcal{L}_\sigma(\boldsymbol{w})\|_2$ να έχει μικρή τιμή). Για να βρούμε το καλύτερο διάνυσμα στην ανωτέρω συλλογή, αρκεί να αξιολογήσουμε κάθε υποψήφιο διάνυσμα σε έναν μικρό αριθμό ανεξάρτητων δειγμάτων της $\mathcal{D}$ χρησιμοποιώντας την εμπειρική εκδοχή του $L_{\mathcal{D},\ell_{0-1}}$. Αποδεικνύεται, ότι αυτό επαρκεί για να αποκομίσουμε το επιθυμητό PAC αποτέλεσμα (βλ. Θεώρημα 0.3.2). Τα ανωτέρω βήματα περιγράφονται λεπτομερώς στον Αλγόριθμο 3.

Αλγόριθμος 3 Κανονική Μάθηση Ομογενών Γραμμικών Συναρτήσεων Κατωφλίου με Θόρυβο Massart

Input: Σύνολο εκπαίδευσης $S=\left(\left(\boldsymbol{x}^{(1)},y^{(1)}\right),\ldots,\left(\boldsymbol{x}^{(T)},y^{(T)}\right)\right)$, $\sigma>0$, $T'\in[T]$ και βήμα $\beta>0$
Output: Ομογενής LTF $h_{\hat{\boldsymbol{w}}}:\mathbb{R}^d\to\{\pm1\}$

$\left(\boldsymbol{w}^{(1)},\ldots,\boldsymbol{w}^{(T)}\right)\leftarrow$ PSGD(g_σ,S_1,β) ▷ Βλ. Προβαλλόμενη Στοχαστική Κατάβαση Δυναμικού
$L\leftarrow\left(\boldsymbol{w}^{(1)},\ldots,\boldsymbol{w}^{(T)},-\boldsymbol{w}^{(1)},\ldots,-\boldsymbol{w}^{(T)}\right)$ ▷ Σύνολο υποψήφιων διανυσμάτων
$S'\leftarrow\left(\left(\boldsymbol{x}^{(1)},y^{(1)}\right),\ldots,\left(\boldsymbol{x}^{(T')},y^{(T')}\right)\right)$
$\hat{\boldsymbol{w}}\leftarrow\operatorname{argmin}_{\boldsymbol{w}\in L}\hat{L}_{S',\ell_{0-1}}(h_{\boldsymbol{w}})$
return $h_{\hat{\boldsymbol{w}}}$

Θεώρημα 0.3.2 (Diakonikolas et al. [2020a]). *Έστω $\eta\in[0,1/2)$, $\epsilon,\delta\in(0,1)$ και $\boldsymbol{w}^*\in\mathbb{R}^d\setminus\{\mathbf{0}\}$ ένα διάνυσμα στόχος. Έστω, επίσης, $\mathcal{D}$ μια $(\eta,h_{\boldsymbol{w}^*})$-Massart κατανομή, τέτοια, ώστε $\mathcal{D}_{\boldsymbol{x}}\in\mathcal{F}^d_{\mathrm{LC}}$. Ο Αλγόριθμος 3 έχει την ακόλουθη εγγύηση επίδοσης: Με είσοδο $T\in\Theta\left(\frac{\log^8(\epsilon(1-2\eta))}{\epsilon^4(1-2\eta)^{10}}\left(d+\log(1/\delta)\right)\right)$ ανεξάρτητα δείγματα από την $\mathcal{D}$, $\sigma\in\Theta\left(\frac{\epsilon(1-2\eta)^{3/2}}{\log^2(\epsilon(1-2\eta))}\right)$, $T'\in\Theta\left(\frac{\log(T/\delta)}{\epsilon^2(1-2\eta)^2}\right)$ και βήμα $\beta\in\Theta\left(\frac{\sigma^2}{\sqrt{dT}}\right)$, τερματίζει σε χρόνο $\Theta(TG+dTT')$, όπου G ένα άνω φράγμα σε κάθε υπολογισμό του gradient, και επιστρέφει ένα διάνυσμα $\hat{\boldsymbol{w}}\in S^{d-1}$, τέτοιο, ώστε, με πιθανότητα τουλάχιστον $1-\delta$, $L_{\mathcal{D}_{\boldsymbol{x}},h_{\boldsymbol{w}^*},\ell_{0-1}}(h_{\hat{\boldsymbol{w}}})\leq\epsilon$.*

0.4 Label ranking

Γενικώς, μια *κατάταξη* (ranking) πάνω σε ένα πεπερασμένο σύνολο S αναφέρεται σε μια αυστηρή μερική διάταξη $\succ$ πάνω στο S. Αν μια κατάταξη $\succ$ είναι αυστηρή ολική διάταξη, τότε θα λέγεται *πλήρης* (complete), ειδάλλως *ελλιπής* (incomplete). Κάθε πλήρης κατάταξη πάνω στο $[k]$, όπου $k \geq 1$, μπορεί να μοντελοποιηθεί σαν μία μετάθεση $\pi \in \mathbb{S}_k$[3], τέτοια, ώστε το $\pi(i)$ να δίδει τη θέση της ετικέτας i στην κατάταξη για κάθε $i \in [k]$ και, κατά συνέπεια, το $\pi^{-1}(i)$ να δίδει το στοιχείο του $[k]$ που βρίσκεται στην i-οστή θέση της κατάταξης για κάθε $i \in [k]$.

Ορισμός προβλήματος Έστω S ένα πεπερασμένο σύνολο $k \geq 2$ ετικετών, $\mathcal{X}$ ένας χώρος παραδειγμάτων (συνήθως $\mathcal{X} \subseteq \mathbb{R}^d$, όπου $d \geq 1$), $\mathcal{Y}$ το σύνολο των αυστηρών μερικών διατάξεων πάνω στο S και $\mathcal{Y}'$ το σύνολο των αυστηρών ολικών διατάξεων πάνω στο S. Το πρόβλημα Label Ranking (κατάταξη ετικετών) είναι ένα επιβλεπόμενο πρόβλημα πρόβλεψης που αφορά τη χρήση ενός συνόλου εκπαίδευσης με στοιχεία εντός του $\mathcal{X} \times \mathcal{Y}$ προς εύρεση μιας υπόθεσης από το $\mathcal{X}$ στο $\mathcal{Y}'$. Ο στόχος ενός LR αλγορίθμου είναι να εξασφαλίσει, ότι η υπόθεση που επιστρέφει ελαχιστοποιεί κάποια έννοια σφάλματος, σχετιζόμενη με κάποια συνάρτηση απώλειας $\ell \colon \mathcal{Y}^2 \to \mathbb{R}_{\geq 0}$.

Εφεξής, θεωρούμε ότι το σύνολο των ετικετών ταυτίζεται με $S = [k]$ χωρίς βλάβη της γενικότητας. Στην περιπτώσεις που μεταχειριζόμαστε *πλήρεις* κατατάξεις, τότε αυτές θα μοντελοποιούνται ως στοιχεία του $\mathbb{S}_k$.

Αναπαράσταση προτιμήσεων μέσω συναρτήσεων χρησιμότητας Όπως έχει αναφερθεί σε μια πληθώρα δουλειών (Har-Peled et al. [2002], Dekel et al. [2003], Hüllermeier et al. [2008], Fotakis et al. [2022a,b]), ένας φυσικός τρόπος αναπαράστασης προτιμήσεων μεταξύ k ετικετών είναι μέσω μιας *συνάρτησης χρησιμότητας* (utility ή score function) $\boldsymbol{m} \colon \mathcal{X} \to \mathbb{R}^k$, η οποία αξιολογεί κάθε ετικέτα αναθέτοντάς της μια τιμή. Όσο μεγαλύτερη είναι αυτή η τιμή, τόσο μεγαλύτερη είναι και η προτίμηση για τη συγκεκριμένη ετικέτα και τόσο υψηλότερη είναι η θέση της ετικέτας στην υποκείμενη κατάταξη. Συγκεκριμένα, κάθε διάνυσμα χρησιμοτήτων αντιστοιχίζεται σε μια κατάταξη μέσω της συνάρτησης $\mathfrak{S} \colon \mathbb{R}^k \to \mathbb{S}_k$, η οποία δρα ως εξής. Λαμβάνει ως είσοδο ένα διάνυσμα $\boldsymbol{v} = (v_1, \ldots, v_k) \in \mathbb{R}^k$, του οποίου το i-οστό στοιχείο αποτελεί την τιμή χρησιμότητας της i-οστής ετικέτας, και επιστρέφει την (μοναδική) μετάθεση $\pi = \mathfrak{S}(\boldsymbol{v}) \in \mathbb{S}_k$, για την οποία για κάθε $1 \leq i < j \leq k$ ισχύει $\pi(i) < \pi(j)$ αν και μόνο αν $v_i \geq v_j$.

Συναρτήσεις απώλειας για το Label Ranking Ακολούθως, παραθέτουμε ορισμένες από τις πιο ευρέως χρησιμοποιούμενες συναρτήσεις απώλειας στο πρόβλημα Label Ranking για την περίπτωση των πλήρων κατατάξεων. Μια από αυτές είναι η απόσταση τ του Kendall (KT distance), η οποία ορίζεται για οποιαδήποτε $\pi, \sigma \in \mathbb{S}_k$ ως $d_\tau(\pi, \sigma) \triangleq \sum_{1 \leq i < j \leq k} \mathbb{1}\left\{(\pi(i) - \pi(j))\,(\sigma(i) - \sigma(j)) < 0\right\}$. Δηλαδή, η απόσταση KT μετρά το πλήθος των ζευγών ετικετών που η μεταξύ τους διάταξη είναι διαφορετική στις δύο κατατάξεις. Η απόσταση KT σχετίζεται άμεσα με τον συντελεστή συσχέτισης κατατάξεων του Kendall (KT correlation coefficient) που ορίζεται ως $\tau(\pi, \sigma) \triangleq 1 - \frac{4 d_\tau(\pi, \sigma)}{k(k-1)}$ και η οποία ποσοτικοποιεί την ομοιότητα δύο κατατάξεων μέσω του αριθμού των σύμφωνων και ασύμφωνων ζευγών ετικετών. Δύο άλλες εξίσου συχνά χρησιμοποιούμενα μέτρα απόστασης για κατατάξεις είναι τα $d_1(\pi, \sigma) \triangleq \sum_{i=1}^k |\pi(i) - \sigma(i)|$ (Spearman's footrule) και $d_2(\pi, \sigma) \triangleq \sum_{i=1}^k (\pi(i) - \sigma(i))^2$ (Spearman's distance). Στο Κεφάλαιο 4, παρατίθενται μερικές ακόμη δημοφιλείς συναρτήσεις απώλειας για το LR.

Συμβολισμοί Έστω $(i, j) \in [k]^2$ με $i \neq j$ και έστω $\mathcal{D}$ μια κατανομή στον $\mathcal{X} \times \mathbb{S}_k$. Για κάθε $\pi \in S_k$, ορίζουμε $\pi_{ij} = \operatorname{sign}(\pi(j) - \pi(i))$. Επιπλέον, για κάθε $\sigma : \mathcal{X} \to \mathbb{S}_k$, ορίζουμε τη συνάρτηση $\sigma_{ij} \colon \mathcal{X} \to \{\pm 1\}$, για την οποία $\sigma_{ij}(x) = \operatorname{sign}(\sigma(x)(j) - \sigma(x)(i))$ για κάθε $x \in \mathcal{X}$. Τέλος, συμβολίζουμε με $\mathcal{D}_{ij}$ την από κοινού κατανομή του ζεύγους (x, π_{ij}), όπου $(x, \pi) \sim \mathcal{D}$.

0.4.1 Μοντέλα Θορύβου για Label Ranking

Εν συνεχεία ορίζουμε κάποιες Label Ranking κατανομές, δηλαδή κατανομές πιθανότητας στον $\mathcal{X} \times \mathbb{S}_k$, οι οποίες ποσοτικοποιούν την ύπαρξη θορύβου.

Ορισμός 0.4.1 (Κατανομή Mallows (Mallows [1957])). *Έστω $\phi \in (0, 1]$, $\pi_0 \in \mathbb{S}_k$ και $d \colon \mathbb{S}_k^2 \to \mathbb{R}_{\geq 0}$. Η κατανομή Mallows $\mathcal{M}_{\mathrm{Mal}}(d, \phi, \pi_0)$ με κεντρική κατάταξη π_0 και παράμετρο εξάπλωσης ϕ είναι ένα μέτρο*

[3] Με $\mathbb{S}_k$ συμβολίζουμε το σύνολο όλων των μεταθέσεων του $[k]$.

πιθανότητας στον $\mathbb{S}_k$ με συνάρτηση μάζας πιθανότητας:

$$\Pr_{\pi\sim\mathcal{M}_{\text{Mal}}(d,\phi,\pi_0)}[\pi] = \frac{\phi^{d(\pi,\pi_0)}}{\sum_{\sigma\in\mathbb{S}_k}\phi^{d(\sigma,\pi_0)}}$$

Η πρώτη κατανομή LR που θα ορίσουμε, βασίζεται στο μοντέλο Mallows, το οποίο αποτελεί ένα από τα δημοφιλέστερα μοντέλα σε προβλήματα με κατατάξεις. Συγκεκριμένα, μια φυσική προσέγγιση θα ήταν να μοντελοποιήσουμε την υπό συνθήκη κατανομή των κατατάξεων δεδομένου ενός στιγμιοτύπου $x \in \mathcal{X}$ ως μια κατανομή Mallows με κεντρική κατάταξη $\sigma^*(x)$, όπου $\sigma^*\colon \mathcal{X} \to \mathbb{S}_k$ μια συνάρτηση στόχος. Αυτό, διαισθητικά, εξασφαλίζει, ότι, για κάθε $x \in \mathcal{X}$, η συχνότερα παρατηρούμενη κατάταξη θα είναι η "ορθή" κατάταξη $\sigma^*(x)$.

Ορισμός 0.4.2 (Κατανομή Label Ranking με Θόρυβο Mallows)**.** *Έστω $\phi \in (0,1]$, $d\colon \mathbb{S}_k^2 \to \mathbb{R}_{\geq 0}$ και $\sigma\colon \mathcal{X} \to \mathbb{S}_k$. Μια κατανομή πιθανότητας $\mathcal{D}$ στον $\mathcal{X}\times\mathbb{S}_k$ λέγεται (ϕ, d, σ)-LR κατανομή με θόρυβο Mallows, αν $\mathcal{D}_{\pi|x} = \mathcal{M}_{\text{Mal}}(d, \phi, \sigma(x))$ για κάθε $x \in \mathcal{X}$.*

Ακολουθεί μια άλλη οικογένεια κατανομών, η οποία εισήχθη από τους Fotakis et al. [2022b] και η οποία επεκτείνει το μοντέλο Massart στην περίπτωση του LR. Αποδεικνύεται δε, ότι κάτω από πολύ ήπιες παραδοχές, συμπεριλαμβάνει και την ανωτέρω οικογένεια των LR κατανομών με θόρυβο Mallows, το οποίο νοηματοδοτεί τη χρήση της στην περίπτωση του LR.

Ορισμός 0.4.3 (Label Ranking Κατανομή με Massart Θόρυβο)**.** *Μια κατανομή πιθανότητας $\mathcal{D}$ στον $\mathcal{X} \times \mathbb{S}_k$ λέγεται LR κατανομή με Massart θόρυβο, αν υπάρχει κάποια (μοναδική) συνάρτηση $\sigma\colon \mathcal{X} \to \mathbb{S}_k$ και κάποιο $\eta \in [0, 1/2)$, έτσι ώστε η κατανομή $\mathcal{D}_{ij}$ να αποτελεί (η, σ_{ij})-Massart κατανομή για κάθε $(i,j) \in [k]^2$ με $i \neq j$. Τότε, η $\mathcal{D}$ καλείται, επίσης, (η, σ)-LR κατανομή με Massart θόρυβο.*

Επισημαίνουμε, ότι και το μοντέλο Tsybakov μπορεί να γενικευτεί στην περίπτωση του LR με εντελώς ανάλογο τρόπο (βλ. Κεφάλαιο 4).

Όπως προαναφέραμε, κάθε συνάρτηση LR (από το $\mathcal{X}$ στο $\mathbb{S}_k$) μπορεί να συσχετιστεί με μια (άπειρες για την ακρίβεια) συνάρτηση χρησιμοτήτων. Αυτό αποτελεί κίνητρο για να ορίσουμε ένα ακόμη είδος κατανομών LR, στο οποίο ο θόρυβος προστίθεται απευθείας σε κάθε διάνυσμα χρησιμοτήτων πριν τη δημιουργία μιας κατάταξης.

Ορισμός 0.4.4 (Label Ranking Κατανομή με Προσθετικό Θόρυβο)**.** *Έστω $\mathcal{E}$ μια κατανομή πιθανότητας στον $\mathbb{R}^k$ και $\boldsymbol{m}\colon \mathcal{X} \to \mathbb{R}^k$ μια συνάρτηση χρησιμότητας. Μια κατανομή πιθανότητας $\mathcal{D}$ στον $\mathcal{X} \times \mathbb{S}_k$ λέγεται $(\boldsymbol{m}, \mathcal{E})$-LR κατανομή με προσθετικό θόρυβο, αν για κάθε $(x, \pi) \sim \mathcal{D}$, ισχύει $\pi = \mathfrak{S}(\boldsymbol{m}(x) + \boldsymbol{\xi})$, όπου τα x και $\boldsymbol{\xi}$ είναι ανεξάρτητα και $\boldsymbol{\xi} \sim \mathcal{E}$.*

0.4.2 Τεχνικές για Label Ranking

Αποσύνθεση Κατά Ζεύγη

Μια από τις δημοφιλέστερες τεχνικές στο LR είναι η τεχνική της *αποσύνθεσης κατά ζεύγη* (pairwise decomposition) ή, αλλιώς, της *κατάταξης μέσω ανά ζεύγη προτιμήσεων* (ranking by pairwise preferences ή RPC). Η κεντρική ιδέα έγκειται στην αποσύνθεση του προβλήματος σε $\binom{k}{2}$ υποπροβλήματα δυαδικής ταξινόμησης, ένα για κάθε (μη διατεταγμένο) ζεύγος (i,j) ετικετών, όπου στόχος είναι η πρόβλεψη της μεταξύ τους σειράς εντός της κατάταξης. Αυτό επιτυγχάνεται, χρησιμοποιώντας έναν αλγόριθμο δυαδικής ταξινόμησης για κάθε υποπρόβλημα, τον οποίο τρέχουμε πάνω σε μια προσαρμοσμένη εκδοχή των δεδομένων εκπαίδευσης, όπως φαίνεται ακολούθως (Αλγόριθμος 4).

Ένα βασικό πλεονέκτημα της μεθόδου αποσύνθεσης κατά ζεύγη είναι, ότι μπορεί να διαχειριστεί περιπτώσεις ελλιπών κατατάξεων, εφόσον κάθε υποπρόβλημα αφορά ένα συγκεκριμένο ζεύγος ετικετών και έχει μηδενική εξάρτηση από τις υπόλοιπες. Ωστόσο, η ανεξάρτητη μεταχείριση των υποπροβλημάτων έχει το μειονέκτημα, ότι οι ανά ζεύγη προβλέψεις μπορεί να είναι μεταξύ τους αντιφατικές (δηλαδή, οι ανά ζεύγη προτιμήσεις να μην ικανοποιούν τη μεταβατική ιδιότητα), κατά τρόπο που να μην είναι εφικτή η άμεση αποκόμιση μιας κατάταξης. Ως εκ τούτου, είναι αναγκαία η εύρεση μεθόδων συνάθροισης (aggregation), οι οποίες να αξιοποιούν την πληροφορία των ανά ζεύγη προβλέψεων, κατά τρόπο που να μπορεί να καταστεί εφικτή η εξαγωγή μιας κατάταξης.

Ένας από τους απλούστερους τρόπους για να γίνει αυτό, είναι να χρησιμοποιήσουμε ένα σχήμα ψηφοφορίας, το οποίο ρυθμίζει τη θέση μιας ετικέτας στην τελική κατάταξη ανάλογα με το πλήθος των ανά ζεύγη "νικών" (Αλγόριθμος 5). Για κάθε ετικέτα, όσο μεγαλύτερο είναι το πλήθος των ζευγών στα οποία αυτή

Αλγόριθμος 4 Label Ranking μέσω Αποσύνθεσης κατά Ζεύγη

Είσοδος: Σύνολο εκπαίδευσης $T \subseteq \mathcal{X} \times \mathcal{Y}$, αλγόριθμος δυαδικής ταξινόμησης $A \in \mathcal{A}_{\mathcal{X},\{\pm 1\}}$
Έξοδος: Υπόθεση $h \colon \mathcal{X} \to \mathbb{S}_k$

procedure PairwiseDecomposition(T, A)
  for $1 \leq i < j \leq k$ **do**
    $T_{ij} \leftarrow \varnothing$
    for $(x, \succ) \in T$ **do**
      if $i \succ j$ **then**
        $T_{ij} \leftarrow T_{ij} \cup (x, 1)$
      end if
      if $j \succ i$ **then**
        $T_{ij} \leftarrow T_{ij} \cup (x, -1)$
      end if
    end for
    $g_{ij} \leftarrow A(T_{ij})$
  end for
  return $(g_{ij})_{1 \leq i < j \leq k}$
end procedure
$C \leftarrow$ PairwiseDecomposition(T, A)
return EstimateAggregation(C) ▷ Βλ. συνάθροιση μέσω ψηφοφορίας και μέσω γραφήματος

Αλγόριθμος 5 Συνάθροιση μέσω Ψηφοφορίας στην Αποσύνθεση κατά Ζεύγη

Είσοδος: Συλλογή από δυαδικούς ταξινομητές $(g_{ij})_{1 \leq i < j \leq k}$
Έξοδος: Υπόθεση $h \colon \mathcal{X} \to \mathbb{S}_k$

procedure VotingAggregation($(g_{ij})_{1 \leq i < j \leq k}$)
  for $1 \leq i \leq k$ **do**
    $s_i(x) \leftarrow 1 + \sum_{j \in [k] \setminus \{i\}} \mathbb{1}\{g_{ij}(x) = -1\}$
  end for
  $\boldsymbol{s} \leftarrow (s_1, \ldots, s_k)$
  return argsort $\circ$ argsort $\circ\, \boldsymbol{s}$
end procedure

συμμετέχει και προηγείται της άλλης ετικέτας του ζεύγους, τόσο υψηλότερη θα είναι και η θέση της στην τελική κατάταξη.

Αλγόριθμος 6 Συνάθροιση μέσω Γραφήματος στην Αποσύνθεση κατά Ζεύγη

Είσοδος: Συλλογή από δυαδικούς ταξινομητές $(g_{ij})_{1 \leq i < j \leq k}$, αλγόριθμος A για μετατροπή πλήρους κατευθυνόμενου γραφήματος σε ακυκλικό
Έξοδος: Υπόθεση $h \colon \mathcal{X} \to \mathbb{S}_k$

procedure TournamentAggregation($(g_{ij})_{1 \leq i < j \leq k}, A$)
  $V \leftarrow [k]$
  $E_x \leftarrow \{(i, j) \in [k]^2 : i \neq j \ \wedge \ g_{ij}(x) = 1\}$
  $G_x \leftarrow (V, E_x)$
  $G'_x \leftarrow A(G_x)$
  Έστω $\hat{\sigma}(x)$ η κατάταξη που επάγεται από το G'_x
  return $\hat{\sigma}(\cdot)$
end procedure

Ένας άλλος τρόπος συνάθροισης των επιμέρους προβλέψεων, είναι να κατασκευάσουμε ένα πλήρες κατευθυνόμενο γράφημα, η κατεύθυνση των ακμών του οποίου θα είναι βάσει των ανά ζεύγη προβλέψεων για κάθε νέο στιγμιότυπο (Αλγόριθμος 6). Μια φυσική προσέγγιση θα ήταν να αφαιρέσουμε όσο το δυνατόν λιγότερες ακμές, ώστε να καταστήσουμε το γράφημα ακυκλικό, και, εν συνεχεία, να επιστρέψουμε την κατάταξη που αντιστοιχεί στην τοπολογική διάταξη των κορυφών του. Αυτό μας παραπέμπει στο πρόβλημα εύρεσης ενός *ελάχιστου συνόλου ανάδρασης τόξων* (minimum feedback arc set). Το τελευταίο πρόβλημα είναι NP-hard, αλλά υπάρχουν αποδοτικοί προσεγγιστικοί αλγόριθμοι στους οποίους μπορούμε να στραφούμε.

Αποσύνθεση Κατά Ετικέτες

Μια άλλη εξίσου δημοφιλής τεχνική στο LR είναι η τεχνική της *αποσύνθεσης κατά ετικέτες* (labelwise decomposition). Η κεντρική ιδέα έγκειται στην αποσύνθεση του προβλήματος σε k υποπροβλήματα, ένα για κάθε ετικέτα, όπου στόχος είναι η πρόβλεψη της θέσης της εντός της κατάταξης. Στην παρούσα διπλωματική, υιοθετούμε την προσέγγιση των Fotakis et al. [2022a], όπου κάθε υποπρόβλημα αποτελεί ένα πρόβλημα παλινδρόμησης, στο οποίο, δηλαδή, οι προβλέψεις μπορούν να λάβουν πραγματικές τιμές. Κατά αντιστοιχία με την περίπτωση της ανά ζεύγη ταξινόμησης, για κάθε υποπρόβλημα χρησιμοποιούμε έναν αλγόριθμο παλινδρόμησης, τον οποίο τρέχουμε πάνω σε μια προσαρμοσμένη εκδοχή των δεδομένων εκπαίδευσης. Εν προκειμένω, ο πιο προφανής τρόπος συνάθροισης των επιμέρους προβλέψεων είναι να κατατάξουμε τις ετικέτες κατά αύξουσα σειρά προβλεπόμενης θέσης. Η ανωτέρω διαδικασία περιγράφεται επακριβώς παρακάτω (Αλγόριθμος 7).

Αλγόριθμος 7 Label Ranking μέσω Αποσύνθεσης κατά Ετικέτες

Είσοδος: Σύνολο εκπαίδευσης $T \subset \mathcal{X} \times \mathbb{S}_k$, αλγόριθμος $A \in \mathcal{A}_{\mathcal{X},\mathbb{R}}$
Έξοδος: Υπόθεση $h \colon \mathcal{X} \to \mathbb{S}_k$

procedure LabelwiseDecomposition(T, A)
  for $1 \leq i \leq k$ **do**
    $T_i \leftarrow \varnothing$
    for $(x, \pi) \in T$ **do**
      $T_i \leftarrow T_i \cup (x, \pi(i))$
    end for
    $g_i \leftarrow A(T_i)$
  end for
  return $(g_1, \ldots, g_k)$
end procedure
$\boldsymbol{g} \leftarrow$ LabelwiseDecomposition(T, A)
return argsort $\circ$ argsort $\circ\, \boldsymbol{g}$

Επισημαίνουμε, ότι αμφότερες οι τεχνικές της κατά ζεύγη και της κατά ετικέτες αποσύνθεσης συνοδεύονται από στατιστικές εγγυήσεις υπό ορισμένες παραδοχές, οι οποίες αναλύονται στο Κεφάλαιο 4.

0.5 Μάθηση Γραμμικών Ταξινομητικών Συναρτήσεων

Στην παρούσα ενότητα, υποθέτουμε, ότι οι κατατάξεις παράγονται από κάποια γραμμική συνάρτηση χρησιμότητας, την οποία επιθυμούμε να μάθουμε. Σε αυτήν την περίπτωση, η υποκείμενη κλάση υποθέσεων ταυτίζεται με την κλάση των γραμμικών ταξινομητικών συναρτήσεων (LSFs), η οποία εισήχθη από τους Har-Peled et al. [2002]. Αυτή η ειδική περίπτωση του LR αναφέρεται και ως γραμμικό LR (Fotakis et al. [2022b]). Μια σημαντική ιδιότητα των LSFs είναι, ότι συνδέονται στενά με την κλάση των ημιδιαστημάτων, γεγονός το οποίο μας επιτρέπει να αποκομίσουμε θεωρητικά αποτελέσματα για το γραμμικό LR τόσο στην πραγματοποιήσιμη περίπτωση (δίχως θόρυβο), όσο και υπό την παρουσία θορύβου.

Ορισμός 0.5.1 (Γραμμική Ταξινομητική Συνάρτηση)**.** *Ως γραμμική ταξινομητική συνάρτηση (linear sorting function ή LSF) στον $\mathbb{R}^d$ με $k \geq 2$ ετικέτες ορίζεται οποιαδήποτε συνάρτηση $\sigma_{\boldsymbol{W},\boldsymbol{b}} \colon \mathbb{R}^d \to \mathbb{S}_k$ με τύπο $\sigma_{\boldsymbol{W},\boldsymbol{b}}(\boldsymbol{x}) = \mathfrak{S}(\boldsymbol{W}\boldsymbol{x} + \boldsymbol{b})$, όπου $\boldsymbol{W} \in \mathbb{R}^{k \times d}$ και $\boldsymbol{b} \in \mathbb{R}^k$.*

Μια LSF $\sigma_{\boldsymbol{W},\boldsymbol{b}}$ λέγεται ομογενής, αν $\boldsymbol{b} = \boldsymbol{0}$, στην οποία περίπτωση συμβολίζεται ως $\sigma_{\boldsymbol{W}}$. Συμβολίζουμε με $\mathcal{H}^{d,k}_{\mathrm{LSF}}$ (αντιστοίχως $\mathcal{H}^{d,k}_{\mathrm{HLSF}}$) την κλάση των LSFs (αντιστοίχως ομογενών LSFs) στον $\mathbb{R}^d$ με k ετικέτες.

Παρατηρούμε, ότι για $k = 2$, υπάρχει ισοδυναμία μεταξύ της κλάσης των LSFs και της κλάσης των ημιδιαστημάτων στον $\mathbb{R}^d$. Επιπροσθέτως, για κάθε $\sigma_{\boldsymbol{W},\boldsymbol{b}} \in \mathcal{H}^{d,k}_{\mathrm{LSF}}$, η συνάρτηση $(\sigma_{\boldsymbol{W},\boldsymbol{b}})_{ij}$ που εμπλέκεται στο υποπρόβλημα που αφορά το ζεύγος ετικετών (i, j) για κάθε $1 \leq i < j \leq k$, μπορεί να γραφεί ως

$$(\sigma_{\boldsymbol{W},\boldsymbol{b}})_{ij}(\boldsymbol{x}) = \operatorname{sign}\left(\sigma_{\boldsymbol{W},\boldsymbol{b}}(j)(\boldsymbol{x}) - \sigma_{\boldsymbol{W},\boldsymbol{b}}(i)(\boldsymbol{x})\right) = \operatorname{sign}\left(\langle \boldsymbol{w}_i - \boldsymbol{w}_j, \boldsymbol{x}\rangle + b_i - b_j\right) = h_{\boldsymbol{w}_i - \boldsymbol{w}_j, b_i - b_j}(\boldsymbol{x}),$$

για κάθε $\boldsymbol{x} \in \mathbb{R}^d$. Δηλαδή, αποτελεί ένα ημιδιάστημα που παραμετροποιείται από το διάνυσμα βαρών $\boldsymbol{w}_i - \boldsymbol{w}_j$ και τον όρο $b_i - b_j$. Αυτή η ιδιότητα μας καθοδηγεί προς τη μέθοδο αποσύνθεσης κατά ζεύγη για την επίλυση του γραμμικού LR, όπου κάθε επιμέρους πρόβλημα θα αφορά τη μάθηση ενός ημιδιαστήματος.

0.5.1 Μάθηση Γραμμικών Ταξινομητικών Συναρτήσεων χωρίς Θόρυβο

Αποδεικνύεται, ότι η κλάση των γραμμικών ταξινομητικών συναρτήσεων στον $\mathbb{R}^d$ με k ετικέτες είναι αποδοτικώς PAC εκμαθήσιμη κατά πραγματοποιήσιμο τρόπο ως προς d_τ. Έστω ένα σύνολο εκπαίδευσης $S = \left(\left(\boldsymbol{x}^{(1)}, \pi^{(1)}\right), \ldots, \left(\boldsymbol{x}^{(m)}, \pi^{(m)}\right)\right)$ με δείγματα στον $\mathbb{R}^d \times \mathbb{S}_k$. Θεωρούμε το ακόλουθο γραμμικό πρόγραμμα, το οποίο συμβολίζουμε με LP2:

$$\text{Βρες τα} \qquad \boldsymbol{W} \in \mathbb{R}^{k \times d},\ \boldsymbol{b} \in \mathbb{R}^k$$

$$\text{έτσι ώστε} \quad \pi_{ij}^{(t)} \left(\left\langle \boldsymbol{w}_i - \boldsymbol{w}_j, \boldsymbol{x}^{(t)} \right\rangle + b_i - b_j \right) \geq 1 \quad \forall\ 1 \leq i < j \leq k,\ t \in [m]$$

Ακολουθώντας μια παρόμοια διαδικασία με αυτή στην περίπτωση της μάθησης ημιδιαστημάτων μέσω γραμμικού προγραμματισμού, μπορούμε να δείξουμε (Θεώρημα 0.5.1), ότι χρησιμοποιώντας έναν επαρκή αριθμό δειγμάτων εκπαίδευσης, επιτυγχάνεται η ζητούμενη (ϵ, δ)-PAC εγγύηση.

Αλγόριθμος 8 Κανονική Εκμάθηση LSFs μέσω Γραμμικού Προγραμματισμού

Είσοδος: Σύνολο εκπαίδευσης $S = \left(\left(\boldsymbol{x}^{(1)}, \pi^{(1)}\right), \ldots, \left(\boldsymbol{x}^{(m)}, \pi^{(m)}\right)\right)$
Έξοδος: LSF $\sigma_{\boldsymbol{W},\boldsymbol{b}} \colon \mathbb{R}^d \to \mathbb{S}_k$
1: Κατασκεύασε το LP2 από το S
2: $(\boldsymbol{W}, \boldsymbol{b}) \leftarrow$ ELLIPSOID(LP2) ▷ Βλ. Appendix B
3: **return** $\sigma_{\boldsymbol{W},\boldsymbol{b}}$

Θεώρημα 0.5.1. *Η κλάση $\mathcal{H}_{\text{LSF}}^{d,k}$ είναι PAC εκμαθήσιμη κατά κανονικό και πραγματοποιήσιμο τρόπο με τον Αλγόριθμο 8 ως προς την απόσταση KT με δειγματική πολυπλοκότητα $O\left((d\log(k/\epsilon) + \log(k/\delta))k^2/\epsilon\right)$ και πολυωνυμικό χρόνο εκτέλεσης ως προς τη διάσταση d, το πλήθος των ετικετών k, τον αριθμό των δειγμάτων και το μέγεθος αναπαράστασης των πραγματικών αριθμών.*

0.5.2 Μάθηση Ομογενών Γραμμικών Ταξινομητικών Συναρτήσεων με Θόρυβο

Ακολούθως, δείχνουμε ότι η κλάση των ομογενών LSFs στον $\mathbb{R}^d$ είναι αποδοτικώς και κανονικά PAC εκμαθήσιμη ως προς d_τ και $\mathcal{F}_{\text{LC}}^d$, υπό την παρουσία LR-Massart θορύβου. Ο αλγόριθμος μας ακολουθεί την μέθοδο της αποσύνθεσης ανά ζεύγη ετικετών, αλλά χρησιμοποιεί μια εναλλακτική μέθοδο συνάθροισης των επιμέρους ταξινομητών, βασιζόμενη στη δουλεία των Fotakis et al. [2022b], ώστε η προκύπτουσα υπόθεση να είναι κανονική.

Συγκεκριμένα, έστω $\boldsymbol{W}^* \in \mathbb{R}^{k \times d}$, όπου $\boldsymbol{w}_i^* \neq \boldsymbol{w}_j^*$ για κάθε $1 \leq i < j \leq k$, ο πίνακας που αντιστοιχεί στη συνάρτηση στόχο $\sigma_{\boldsymbol{W}^*}$, και $\mathcal{D}_{\boldsymbol{x}} \in \mathcal{F}_{\text{LC}}^d$. Ο αλγόριθμός μας ξεκινά εφαρμόζοντας της διαδικασία αποσύνθεσης κατά ζεύγη χρησιμοποιώντας για κάθε υποπρόβλημα έναν proper αλγόριθμο A εκμάθησης ομογενών ημιδιαστημάτων, ο οποίος ικανοποιεί τη συνθήκη PAC εκμαθησιμότητας με Massart θόρυβο ως προς $\mathcal{F}_{\text{LC}}^d$. Κατά συνέπεια, για κάθε $\epsilon, \delta > 0$, μπορούμε να αποκομίσουμε $\binom{k}{2}$ διανύσματα $\boldsymbol{v}_{ij}$ (τα οποία θεωρούμε μοναδιαία χωρίς βλάβη της γενικότητας), τέτοια, ώστε για $1 \leq i < j \leq k$, να ισχύει

$$\Pr_{\boldsymbol{x} \sim \mathcal{D}_{\boldsymbol{x}}} \left[\text{sign}\left(\left\langle \boldsymbol{w}_i^* - \boldsymbol{w}_j^*, \boldsymbol{x} \right\rangle \right) \neq \text{sign}\left(\left\langle \boldsymbol{v}_{ij}, \boldsymbol{x} \right\rangle \right) \right] \leq \epsilon$$

με πιθανότητα τουλάχιστον $1 - \delta$.

Ο στόχος μας είναι να εκμεταλλευτούμε την πληροφορία που παρέχεται από τη συλλογή των διανυσμάτων $\boldsymbol{v}_{ij}$, ώστε να αποκομίσουμε έναν πίνακα $\boldsymbol{W} \in \mathbb{R}^{k \times d}$, τέτοιο, ώστε το σφάλμα $L_{\mathcal{D}_{\boldsymbol{x}}, \sigma_{\boldsymbol{W}^*}, d_\tau}(\sigma_{\boldsymbol{W}})$ να είναι μικρό. Διαισθητικά, αυτό επιτυγχάνεται, αν $\theta\left(\boldsymbol{w}_i - \boldsymbol{w}_j, \boldsymbol{v}_{ij}\right) \approx 0$ και $\boldsymbol{w}_i \neq \boldsymbol{w}_j$ για κάθε $1 \leq i < j \leq k$. Αυτές οι συνθήκες μπορούν να διατυπωθούν πιο επίσημα μέσω του ακόλουθου κυρτού προγράμματος, το οποίο συμβολίζουμε με CP1:

$$\text{Βρες έναν} \qquad \boldsymbol{W} \in \mathbb{R}^{k \times d}$$

$$\text{έτσι ώστε} \quad \left\langle \boldsymbol{w}_i - \boldsymbol{w}_j, \boldsymbol{v}_{ij} \right\rangle \geq \max\left\{ \xi, (1 - \phi) \left\| \boldsymbol{w}_i - \boldsymbol{w}_j \right\|_2 \right\} \quad \forall\ 1 \leq i < j \leq k$$

$$\|\boldsymbol{W}\|_{\text{F}} \leq 1$$

Οι σταθερές $\phi \in [0, 1)$ και $\xi > 0$ στο ανωτέρω κυρτό πρόγραμμα χρήζουν καθορισμού. Αποδεικνύεται (βλ. Θεώρημα 0.5.2), ότι η κλάση των ομογενών γραμμικών ταξινομητικών συναρτήσεων στον $\mathbb{R}^d$ είναι

Αλγόριθμος 9 Κανονική Εκμάθηση Ομογενών Γραμμικών Ταξινομητικών Συναρτήσεων με Θόρυβο

Είσοδος: Σύνολο εκπαίδευσης $S \subset \mathbb{R}^d \times \mathbb{S}_k$, αλγόριθμος κανονικής εκμάθησης ομογενών ημιδιαστημάτων A, $\phi \in (0,1)$ και $\xi > 0$
Έξοδος: Ομογενής LSF $\sigma_{\boldsymbol{W}} : \mathbb{R}^d \to \mathbb{S}_k$
1: $\left(h_{\boldsymbol{v}_{ij}}\right)_{1 \le i < j \le k} \leftarrow$ PairwiseDecomposition(S, A)
2: Κατασκεύασε το CP1 μέσω των ϕ, ξ και $(\boldsymbol{v}_{ij})_{1 \le i < j \le k}$
3: $\boldsymbol{W} \leftarrow$ Ellipsoid(CP1) ▷ Βλ. Appendix B
4: **return** $\sigma_{\boldsymbol{W}}$

αποδοτικώς PAC εκμαθήσιμη κατά κανονικό τρόπο ως προς d_τ και $\mathcal{F}^d_{\mathrm{LC}}$, υπό την παρουσία θορύβου LR-Massart.

Θεώρημα 0.5.2. *Έστω $\eta \in [0, 1/2)$, $\epsilon, \delta > 0$ και $\boldsymbol{W}^* \in \mathbb{R}^{k \times d}$, όπου $\boldsymbol{w}_i^* \neq \boldsymbol{w}_j^*$ για κάθε $1 \le i < j \le k$. Έστω $\mathcal{D}$ μια $(\eta, \sigma_{\boldsymbol{W}^*})$-LR κατανομή με Massart θόρυβο, τέτοια, ώστε $\mathcal{D}_{\boldsymbol{x}} \in \mathcal{F}^d_{\mathrm{LC}}$. Έστω A οποιοσδήποτε αλγόριθμος που μαθαίνει, κατά το PAC μοντέλο, την κλάση $\mathcal{H}^d_{\mathrm{HLTF}}$ κατά κανονικό τρόπο ως προς την κλάση $\mathcal{F}^d_{\mathrm{LC}}$ υπό την παρουσία θορύβου Massart και έχει πολυωνυμικό ως προς d και ως προς το πλήθος των δειγμάτων εκπαίδευσης χρόνο εκτέλεσης. Ο Αλγόριθμος 9 έχει την ακόλουθη εγγύηση επίδοσης: Με είσοδο A, $\phi \in \Theta(\epsilon^2)$, $\xi = 2^{-\mathrm{poly}(d,k,1/\epsilon)}$ και $m \ge m^{\mathrm{Massart}}_{A, \mathcal{H}^d_{\mathrm{LTF}}, \mathcal{F}^d_{\mathrm{LC}}}\left(\epsilon/O(k^2), \delta/\binom{k}{2}, \eta\right)$ ανεξάρτητα δείγματα από την $\mathcal{D}$, τρέχει σε χρόνο $\mathrm{poly}(m)$ και επιστρέφει έναν πίνακα $\boldsymbol{W} \in \mathbb{R}^{k \times d}$, τέτοιο, ώστε, με πιθανότητα τουλάχιστον $1 - \delta$, να ισχύει $L_{\mathcal{D}_{\boldsymbol{x}}, \sigma_{\boldsymbol{W}^*}, d_\tau}(\sigma_{\boldsymbol{W}}) \le \epsilon$.*

0.6 Πειραματικά Αποτελέσματα

Στο τελευταίο μέρος αυτής της εργασίας διεξάγουμε μια πειραματική αξιολόγηση LR αλγορίθμων. Στόχος μας είναι να διερευνήσουμε την επίδοση αλγορίθμων προσαρμοσμένων στην περίπτωση του γραμμικού LR σε σχέση με state-of-the-art αλγορίθμους, οι οποίοι έχουν προταθεί για το γενικό LR, και, συγκεκριμένα, αλγορίθμους που βασίζονται σε δέντρα απόφασης (decision trees) και τυχαία δάση (random forests). Η αξιολόγησή μας λαμβάνει χώρα τόσο πάνω σε συνθετικά δεδομένα, τα οποία έχουν παραχθεί γραμμικώς, ήτοι μέσω μιας γραμμικής ταξινομητικής συνάρτησης, και στα οποία έχει προστεθεί θόρυβος, όσο και σε ημισυνθετικά και πραγματικά δεδομένα αναφοράς, τα οποία έχουν χρησιμοποιούνται συχνά στη βιβλιογραφία του LR. Οι αλγόριθμοι που συγκρίνουμε είναι οι εξής:

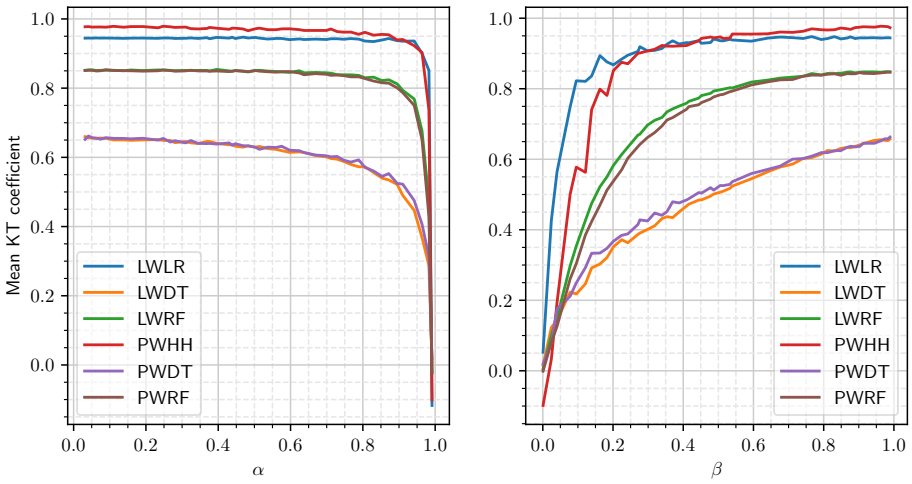
- Η μέθοδος αποσύνθεσης κατά ζεύγη σε συνδυασμό με τον αλγόριθμο μάθησης ομογενών ημιδιαστημάτων των Diakonikolas et al. [2020a] και την μέθοδο συνάθροισης προβλέψεων μέσω γραφήματος. Συμβολίζουμε αυτόν τον αλγόριθμο με PWHH.
- Η μέθοδος αποσύνθεσης κατά ζεύγη σε συνδυασμό με ταξινόμηση μέσω δέντρων απόφασης και την μέθοδο συνάθροισης προβλέψεων μέσω γραφήματος. Συμβολίζουμε αυτόν τον αλγόριθμο με PWDT.
- Η μέθοδος αποσύνθεσης κατά ζεύγη σε συνδυασμό με ταξινόμηση μέσω τυχαίων δασών και την μέθοδο συνάθροισης προβλέψεων μέσω γραφήματος. Συμβολίζουμε αυτόν τον αλγόριθμο με PWRF.
- Η μέθοδος αποσύνθεσης κατά ετικέτες σε συνδυασμό με γραμμική παλινδρόμηση. Συμβολίζουμε αυτόν τον αλγόριθμο με LWLR.
- Η μέθοδος αποσύνθεσης κατά ετικέτες σε συνδυασμό με παλινδρόμηση μέσω δέντρων απόφασης. Συμβολίζουμε αυτόν τον αλγόριθμο με LWDT.
- Η μέθοδος αποσύνθεσης κατά ετικέτες σε συνδυασμό με παλινδρόμηση μέσω τυχαίων δασών. Συμβολίζουμε αυτόν τον αλγόριθμο με LWRF.

Όσον αφορά τη δημιουργία των συνθετικών δεδομένων εκπαίδευσης, αυτά κατασκευάστηκαν μέσω μιας γραμμικής ταξινομητικής συνάρτησης και αλλοιώθηκαν βάσει των προαναφερθέντων LR μοντέλων θορύβου Mallows (Ορισμός 0.4.2) και προσθετικού θορύβου (Ορισμός 0.4.4). Για την ποσοτικοποίηση του θορύβου στα συνθετικά δεδομένα εκπαίδευσης χρησιμοποιούμε τις ακόλουθες έννοιες διαστρέβλωσης, οι οποίες χρησιμοποιήθηκαν και στην πειραματική αξιολόγηση των Fotakis et al. [2022a].

Ορισμός 0.6.1. *Ένα σύνολο εκπαίδευσης $S \subset \mathbb{R}^d \times \mathbb{S}_k$ ικανοποιεί:*

- *την ιδιότητα α-ασυνέπειας, αν* $\frac{1}{|S|}\sum_{(\boldsymbol{x},\pi)\in S}\mathbb{1}\left\{\sigma^*(\boldsymbol{x})\neq\pi\right\}=\alpha\in[0,1]$ *και*
- *την ιδιότητα β-KT διακένου, αν* $\frac{1}{|S|}\sum_{(\boldsymbol{x},\pi)\in S}\tau\left(\sigma^*(\boldsymbol{x}),\pi\right)=\beta\in[-1,1]$.

Η επίδοση κάθε αλγορίθμου αποτιμάται βάσει του μέσου KT συντελεστή συσχέτισης πάνω σε ένα αθόρυβο σύνολο αξιολόγησης, το οποίο έχει κατασκευαστεί από την ίδια γραμμική ταξινομητική συνάρτηση με αυτή των δεδομένων εκπαίδευσης. Ενδεικτικώς, παραθέτουμε τα προκύπτοντα αποτελέσματα στην περίπτωση του θορύβου Mallows για την περίπτωση διάστασης $d = 10$ και πλήθους ετικετών $k = 5$.



Σχήμα 1: Αξιολόγηση ως προς τον μέσο συντελεστή συσχέτισης KT σε δεδομένα με θόρυβο Mallows

Επιπλέον, παραθέτουμε ενδεικτικά αποτελέσματα για την αξιολόγηση των ανωτέρω αλγορίθμων σε ημισυνθετικά δεδομένα LR.

Dataset	LWRF	PWRF	LWDT	PWDT	LWLR	PWHH
authorship	0.926 ± 0.018	0.936 ± 0.014	0.871 ± 0.026	0.870 ± 0.022	$\mathbf{0.940 \pm 0.012}$	0.472 ± 0.060
bodyfat	0.191 ± 0.068	0.188 ± 0.053	0.108 ± 0.072	0.105 ± 0.066	$\mathbf{0.283 \pm 0.054}$	0.136 ± 0.073
calhousing	$\mathbf{0.491 \pm 0.010}$	0.486 ± 0.010	0.351 ± 0.011	0.356 ± 0.010	0.235 ± 0.009	0.169 ± 0.011
cpu-small	0.513 ± 0.011	$\mathbf{0.518 \pm 0.011}$	0.372 ± 0.014	0.369 ± 0.015	0.424 ± 0.012	0.402 ± 0.010
elevators	0.779 ± 0.005	$\mathbf{0.803 \pm 0.006}$	0.675 ± 0.007	0.694 ± 0.007	0.702 ± 0.005	0.576 ± 0.011
fried	0.985 ± 0.001	0.991 ± 0.001	0.964 ± 0.001	0.987 ± 0.001	$\mathbf{0.995 \pm 0.001}$	0.975 ± 0.001
glass	$\mathbf{0.906 \pm 0.032}$	0.905 ± 0.039	0.872 ± 0.043	0.865 ± 0.040	0.815 ± 0.052	0.759 ± 0.072
housing	$\mathbf{0.833 \pm 0.025}$	0.825 ± 0.028	0.783 ± 0.028	0.769 ± 0.030	0.583 ± 0.036	0.581 ± 0.035
iris	0.959 ± 0.043	$\mathbf{0.968 \pm 0.041}$	0.962 ± 0.047	0.946 ± 0.051	0.799 ± 0.085	0.476 ± 0.149
pendigits	$\mathbf{0.976 \pm 0.001}$	0.975 ± 0.001	0.957 ± 0.001	0.959 ± 0.002	0.855 ± 0.002	0.664 ± 0.007
segment	0.976 ± 0.004	$\mathbf{0.977 \pm 0.004}$	0.964 ± 0.005	0.968 ± 0.005	0.877 ± 0.008	0.847 ± 0.010
stock	0.922 ± 0.011	$\mathbf{0.924 \pm 0.011}$	0.902 ± 0.014	0.898 ± 0.015	0.685 ± 0.021	0.495 ± 0.027
vehicle	$\mathbf{0.886 \pm 0.017}$	0.884 ± 0.022	0.837 ± 0.030	0.824 ± 0.026	0.804 ± 0.031	0.745 ± 0.034
vowel	0.892 ± 0.014	$\mathbf{0.908 \pm 0.014}$	0.834 ± 0.020	0.831 ± 0.019	0.596 ± 0.026	0.484 ± 0.034
wine	0.925 ± 0.065	$\mathbf{0.956 \pm 0.042}$	0.888 ± 0.069	0.898 ± 0.065	0.950 ± 0.046	0.213 ± 0.133
wisconsin	0.552 ± 0.034	0.524 ± 0.036	0.415 ± 0.039	0.411 ± 0.045	$\mathbf{0.619 \pm 0.029}$	0.287 ± 0.061

Πίνακας 1: Αξιολόγηση ως προς τον μέσο συντελεστή συσχέτισης KT σε ημισυνθετικά δεδομένα

Τα πλήρη αποτελέσματα και ο σχολιασμός αυτών παρατίθενται στο Κεφάλαιο 6.

Chapter 1

Introduction

Label Ranking (LR) is an increasingly popular topic in the machine learning literature, with an undoubtedly principal role in the area of preference learning (Fürnkranz and Hüllermeier [2010]). Its goal is to find a mapping from an instance space to rankings over a finite set of predefined labels. The Label Ranking problem has received a lot of attention in recent years, since it arises in a plethora of real-world tasks. One of its most characteristic applications is in the area of targeted advertising (Djuric et al. [2014]), where we want to identify a ranking over ads for each individual user and present them with the most relevant based on their interests, with the aim of maximizing the advertisers' revenue. Other applications, where Label Ranking emerges, include: in bioinformatics (Balasubramaniyan et al. [2005], Hestilow et al. [2009]), ranking a set of genes according to their expression level based on the features of each phylogenetic profile; in meta-learning (Aiguzhinov et al. [2010], Brazdil et al. [2003]), ranking a set of available algorithms according to their appropriateness based on the characteristics of each data set; in sentiment analysis (Wang et al. [2011]), ranking a set of social emotions manifested by individuals when exposed to news articles; in document categorization (Jindal and Shweta [2015]), ranking a set of class labels for each particular text document.

Over the years, there have been several seminal works, which are discussed in the following section, proposing state-of-the-art algorithms for the Label Ranking problem. The overriding majority of these algorithms is supported by experimental evaluation indicating their practical performance, but comes with few to none theoretical assurances. Thus, one of the biggest challenges in Label Ranking concerns supporting these results on the basis of statistical and computational guarantees. Another major challenge is whether a Label Ranking algorithm can handle the existence of rankings with missing labels, rankings with ties among their elements or rankings that have been corrupted, i.e. altered, by noise. The main goal of this thesis is to extend some of the existing theoretical results in the noisy linear Label Ranking setting and to experimentally investigate the performance of algorithms customized to the linear Label Ranking setting, against some of the state-of-the-art general Label Ranking algorithms that have been proposed in literature.

1.1 Prior Work

There are multiple approaches to the Label Ranking problem, most of which are collectively presented in the works of Fürnkranz and Hüllermeier [2010], Vembu and Gärtner [2011], Zhou et al. [2014a]. These works can be roughly grouped in the following categories.

Decomposition methods One of the first Label Ranking techniques to be proposed that can fall under the umbrella of decomposition methods is the *constraint classification* technique (Har-Peled et al. [2002]). Its goal is to find a linear utility function for each label that maps feature vectors to score values. Given the score value of each label for some specific feature vector, the construction of a ranking comes naturally by sorting the labels by decreasing score value, so that labels with higher score are ranked higher and vice versa. As for how to obtain the linear utility functions, the constraint classification technique transforms the original Label Ranking problem with d-dimensional instances into a single homogeneous halfspace learning problem in an expanded kd-dimensional space, where k is the number of labels.

Another decomposition technique is that of *log-linear models*, proposed in the work of Dekel et al. [2003]. This method extends the constraint classification technique, in the sense that it attempts to learn

utility functions, which are expressed as a linear combination of a set of general base functions. Algorithmically, though, the estimation of the model's parameters is accomplished by means of a boosting-based algorithm, which seeks to minimize a generalized ranking error iteratively.

A more general technique that models preferences directly instead of attempting to construct utility functions is the *pairwise decomposition* technique, more commonly known as *ranking by pairwise comparison* (RPC) (Fürnkranz and Hüllermeier [2003], Hüllermeier et al. [2008]). The main idea is to split the Label Ranking problem into multiple binary classification subproblems, one for each pair of labels, which concerns finding a model that predicts the preference order for that specific pair of labels given a new instance. The results of the individual pairwise models can be aggregated into a single ranking in multiple ways, which will be analyzed in the following chapters. Two remarkable works that are based on this pairwise approach are those of Vogel and Clémençon [2020] and Fotakis et al. [2022a]. The first one provides statistical guarantees, when the learner observes only the top label of each ranking, while the second provides similar guarantees for the more general case of incomplete rankings under specific assumptions. Moreover, both results hold in the presence of noise. Another noteworthy result based on the pairwise approach was given by Fotakis et al. [2022b] for the linear Label Ranking setting and, specifically, for the concept class of Linear Sorting Functions (Har-Peled et al. [2002]) (which essentially comprises linear utility functions as the ones mentioned in the constraint classification technique). In particular, Fotakis et al. [2022b] showed that the concept class of Linear Sorting Functions is efficiently and properly learnable in the distribution-dependent PAC model (specifically, under the standard multivariate normal distribution), with respect to the well-known Kendall's Tau and top-r ranking distances under a customized for rankings noise model that extends the Massart model (Massart and Nédélec [2006]). Lastly, we remark that more complex pairwise decomposition techniques have been proposed by Gurrieri et al. [2014], which take into account label correlation as well.

As an alternative to pairwise decomposition, Cheng et al. [2013], Cheng and Hüllermeier [2013] proposed a *labelwise decomposition* technique, that splits the Label Ranking problem into multiple subproblems in a different manner. Each subproblem concerns a specific label and we seek to find a model that predicts its position in the final ranking. Like in the pairwise decomposition method, one has to aggregate the labelwise estimates in an appropriate way to get a final ranking, which will be analyzed later in this thesis. An important work in this direction is that of Fotakis et al. [2022a], which provided the first LR algorithm using decision trees in a black box manner with efficiency guarantees in the PAC model.

Instance-based methods A crucial technique that is often used as part of Label Ranking algorithms (Cheng and Hüllermeier [2008], Cheng et al. [2009, 2010], Cheng and Hüllermeier [2013]) is *instance-based learning* (Brinker and Hüllermeier [2006]). Its main idea is to predict the class for a given instance based on local information, that is, the classes of neighboring rankings. The arguably simplest way to do that is using the well-known k-nearest neighbor algorithm (k-NN), assuming that the instance space is endowed with a distance metric. While in the standard classification setting, k-NN sets each new instance's class to be the most frequent class among its k nearest neighbors, in the context of Label Ranking, it is preferable that the structured nature of rankings be incorporated into the prediction process. Namely, we have to devise an appropriate method of aggregating the rankings corresponding to the k nearest neighbors into a single ranking, which is closely related to the *ranking aggregation* problem (Korba et al. [2017], Clémençon et al. [2018]).

Probabilistic methods A highly popular way to tackle the Label Ranking problem is to develop predictive methods on the basis of statistical models on rankings such as the Mallows model (Mallows [1957]) and the Plackett-Luce model (Plackett [1975]) or other models such as Gaussian Mixture Models (GMMs). There have been several works in this direction (Cheng and Hüllermeier [2008], Cheng et al. [2009, 2010, 2012], Cheng and Hüllermeier [2012], Grbovic et al. [2012], Zhou et al. [2014b]), most of which embody an instance-based approach and adopt methods such as maximum likelihood estimation (MLE), expectation-maximization (EM) or majorization-minimization (MM) for estimating the distribution parameters.

Decision tree methods The use of decision tree based Label Ranking algorithms constitutes another novel Label Ranking technique, which turns out to be highly competitive to the aforementioned methods, as experimental evaluation has indicated. Some of the most notable works in this area include adaptation of decision trees (Cheng et al. [2009]), random forests (Zhou and Qiu [2016]), ensembles of decision trees (de Sá et al. [2015], de Sá et al. [2017]) and bagging weak tree-based learners (Aledo et al. [2017]).

Moreover, as mentioned above, the work of Fotakis et al. [2022a], which is based on the labelwise decomposition method, was the first to support the use of decision trees in Label Ranking with theoretical guarantees rather than solely with an experimental evaluation.

Other methods A novel work that does not pertain to the aforementioned categories is that of Korba et al. [2018], which follows a structured prediction approach constituting of two steps. The first step is a regression step in a Hilbert space, where rankings are represented by vectors through an appropriate embedding. The second step is a decoding step helping to retrieve a ranking from each prediction that lies in the Hilbert space. This work is also supported by theoretical guarantees for several embedding choices.

Finally, there are additional works that adapt existing machine learning methods based on similarity measures (Aiguzhinov et al. [2010], de Sá et al. [2011], Ribeiro et al. [2012]) and works that focus on rule-based methods (Gurrieri et al. [2012]) or supervised clustering (Grbovic et al. [2013]).

1.2 Our Contributions

The main contributions of this thesis lie in the acquirement of a new theoretical result concerning the linear Label Ranking setting and in an experimental evaluation of Label Ranking algorithms. Regarding the theoretical part, we build upon the work of Fotakis et al. [2022b] and extend one of their results, which applies to the case of the multivariate standard normal distribution, to the broader family of isotropic logarithmically concave probability distributions. In particular, we show that the concept class of Linear Sorting Functions is efficiently and properly learnable in the PAC model under isotropic log-concave marginals, with respect to the Kendall's Tau distance and with respect to two noise models that constitute extensions of the fundamental Massart and Tsybakov binary classification noise models respectively to the Label Ranking setting.

As for the experimental part, we compare six LR algorithms, including those proposed in Fotakis et al. [2022b], Fotakis et al. [2022a] in terms of their generalization capability and their robustness in the presence of noise. Our goal is to get an understanding of how LR algorithms based on linear predictors perform against some of the state-of-the-art general-purpose LR algorithms based on decision trees and random forests. The comparison takes place on synthetic data sets and on semi-synthetic and real data sets that constitute standard LR benchmarks.

1.3 Organization

In Chapter 2, we provide some of the theoretical foundations of learning theory, centering on the fundamental category of prediction problems. Specifically, we focus on the popular PAC learning framework, which will be extensively used in the next chapters to quantify the notion of learnability and to obtain theoretical guarantees. Moreover, we define some basic noise models that will be used, when we consider the learnability of classes in noisy settings.

In Chapter 3, we focus on the fundamental concept class of halfspaces. We begin by providing some well-known algorithms concerning the learnability of halfspaces in the noiseless setting and proceed by studying their learnability in the more challenging noisy setting and, specifically, in the presence of Massart noise. In particular, we present the work of Diakonikolas et al. [2020a] that provides an efficient halfspace learning algorithm, tolerant in the existence of Massart noise.

In Chapter 4, we address the main topic of this thesis, that is, the Label Ranking problem. Initially, we provide a formal definition of LR, discuss its association with other learning settings such as binary, multiclass and multilabel classification and present some of the most popular loss functions used in LR. Afterwards, we present several noise models for LR, some of which stem from well-known probability distributions on rankings. Finally, we expand on some of the most popular label ranking techniques proposed in literature, while discussing the theoretical guarantees they are associated with.

In Chapter 5, we concentrate on the concept class of Linear Sorting Functions, a specialization of the Label Ranking problem under a linear setting, whose learnability is studied both in the noiseless and the noisy setting. In the noisy setting, we present the first main contribution of this thesis, that is, the aforementioned theoretical result that extends the work of Fotakis et al. [2022b].

In Chapter 6, we present the second main contribution of this thesis, that is, the experimental evaluation that was discussed before.

1.4 Notation

We will use $\mathbf{Pr}[\mathcal{E}]$ to denote the probability of an event $\mathcal{E}$ and $\mathbf{E}[X]$ to denote the expected value of a random variable X. For any event $\mathcal{E}$, we let $\mathbb{1}\{\mathcal{E}\} = 1$ if $\mathcal{E}$ occurs, otherwise 0. Moreover, we define $\operatorname{sign}(z) \triangleq \mathbb{1}\{z > 0\} - \mathbb{1}\{z < 0\}$.

For any $n \in \mathbb{Z}_{>0}$, let $[n] = \{1, \dots, n\}$. We will use small boldface italic characters for vectors and capital boldface italic characters for matrices. Fix any $n, m \in \mathbb{Z}_{>0}$. For any $\boldsymbol{x} \in \mathbb{R}^n$, we denote by x_i the i-th element of $\boldsymbol{x}$, where $i \in [n]$. For any $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$, we denote by $\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \sum_{i=1}^n x_i y_i$ the Euclidean inner product between $\boldsymbol{x}$ and $\boldsymbol{y}$. For any $\boldsymbol{x} \in \mathbb{R}^n$, we denote by $\|\boldsymbol{x}\|_2 = \sqrt{\langle \boldsymbol{x}, \boldsymbol{x} \rangle}$ the Euclidean norm of $\boldsymbol{x}$. For any nonzero $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$, let $\theta(\boldsymbol{x}, \boldsymbol{y}) = \arccos \frac{\langle \boldsymbol{x}, \boldsymbol{y} \rangle}{\|\boldsymbol{x}\|_2 \|\boldsymbol{y}\|_2} \in [0, \pi]$, that is the angle between $\boldsymbol{x}$ and $\boldsymbol{y}$. For any $r \in \mathbb{R}_{\geq 0}$, let $B^n(r) = \{\boldsymbol{x} \in \mathbb{R}^n : \|\boldsymbol{x}\|_2 \leq r\}$, that is, the d-dimensional unit ball centered at the origin. Moreover, let $S^{n-1}(r) = \{\boldsymbol{x} \in \mathbb{R}^n : \|\boldsymbol{x}\|_2 = r\}$, that is, the boundary of $B^n(r)$. We also define $B^n = B^n(1)$ and $S^{n-1} = S^{n-1}(1)$. For any $\boldsymbol{A} \in \mathbb{R}^{m \times n}$, we denote by a_{ij} the element in the i-th row and j-th column of $\boldsymbol{A}$ and we denote by $\boldsymbol{a}_i$ the vector corresponding to the i-the row of $\boldsymbol{A}$, where $i \in [m]$ and $j \in [n]$. The Frobenius norm of $\boldsymbol{A}$ is defined as $\|\boldsymbol{A}\|_{\mathrm{F}} = \sqrt{\sum_{i=1}^n \sum_{j=1}^m |a_{ij}|^2}$.

Chapter 2

Learning Theory

The term *machine learning* (or simply *learning*) refers to the automated detection of meaningful patterns in data (Shalev-Shwartz and Ben-David [2022]). The area of learning has received significant attention over the last decades, due to its substantial contribution in tasks associated with the extraction of information from large data sets and the consequent enhancement of numerous scientific and financial applications. The importance of learning stems from the need to develop adaptive mechanisms with the capability of constant improvement and transformation of their experience into expertise, as opposed to the rigidity of standard methods.

In this chapter, we focus on the fundamental category of *prediction problems*, where our goal is to learn how to make correct predictions on the basis of some predefined context. We begin by defining the basic entities and notions related to a prediction problem. Afterwards, we present the PAC learning model, a well-known framework that formalizes the notion of learning, and cover some of the theoretical fundaments associated with it. Lastly, we discuss the learnability in the presence of noise, which is related to the modern challenge that concerns the development of robust learning algorithms that can handle the existence of corrupted data.

2.1 The Statistical Learning Framework

Domain set A *domain set* (also called *instance space* or *input space*), usually denoted as $\mathcal{X}$, is a set, which represents the set of objects, which we want to label. It is often the case that these objects are represented through vectors of *features* related to them. This is why we also refer to $\mathcal{X}$ as *feature space*. For example, suppose that we are studying the problem of classifying dogs according to their breed. In such a scenario, $\mathcal{X}$ would correspond to the set of all dogs and could possibly contain vectors with features such as the height, the weight and the skin color of a dog.

Label set A *label set* (also called *label space* or *output space*), usually denoted as $\mathcal{Y}$, is a set, which represents the *labels* each element of the domain set can be assigned to. In our dogs example, $\mathcal{Y}$ would be the set of all dog breeds. The case when there exist only two labels, corresponds to the fundamental setting of *binary classification*. In that case, we usually choose $\mathcal{Y} = \{0, 1\}$ or $\mathcal{Y} = \{-1, 1\}$. A more general case that extends binary classification is that of *multiclass classification*, where $\mathcal{Y}$ is a finite set with $|\mathcal{Y}| \geq 2$. Another setting is that of *regression* problems, where the output can take any real value within a specific range.

Hypothesis A *hypothesis* (also called *classifier*, *predictor* or *prediction rule*) is a function $h \colon \mathcal{X} \to \mathcal{Y}$ that maps elements of the instance space to the label space, namely predicts the label of the element it is given as input. Moreover, a set $\mathcal{H}$ of hypotheses in $\mathcal{Y}^{\mathcal{X}}$ is referred to as a *hypothesis class* or *concept class*.

Learning algorithm A *learning algorithm* or *learner* is an algorithm that takes a tuple of labeled examples $S \in \bigcup_{n=0}^{\infty}(\mathcal{X} \times \mathcal{Y})^n$, called a *training set*, as input and returns a hypothesis from $\mathcal{X}$ to $\mathcal{Y}$, that should be capable of predicting the label of any new instance in $\mathcal{X}$. In that sense, a learner can be thought of as some function $A \colon \bigcup_{n=0}^{\infty}(\mathcal{X} \times \mathcal{Y})^n \to \mathcal{Y}^{\mathcal{X}}$. We denote by $\mathcal{A}_{\mathcal{X},\mathcal{Y}}$ the set of all learning algorithms of

the aforementioned form. We also denote $A(S_x, h|_{S_x}) = A(S)$, where $S = ((x_1, h(x_1)), \ldots, (x_m, h(x_m)))$ and $S_x = (x_1, \ldots, x_m)$.

Data generation As far as the training data are concerned, we assume that there is some underlying probability distribution $\mathcal{D}$ on $\mathcal{X} \times \mathcal{Y}$, which they are generated from. It is often useful to decompose $\mathcal{D}$ into two parts: the distribution $\mathcal{D}_x$, which denotes the marginal of $\mathcal{D}$ on $\mathcal{X}$, and $\mathcal{D}_{y|x}$, which denotes the conditional distribution of $\mathcal{D}$ on $\mathcal{Y}$ given $x \in \mathcal{X}$. Observe, that each $x \in \mathcal{X}$ is not deterministically labeled, but is assigned each label with some probability dictated by $\mathcal{D}_{y|x}$. Namely, the labels are not required to be fully determined by the features describing the elements of $\mathcal{X}$. In our dogs example, this can be interpreted as the features chosen to describe a dog, being inadequate to uniquely determine its breed.

Nevertheless, as we will later see, it is sometimes assumed that $\mathcal{D}_{y|x}$ is degenerate for all $x \in \mathcal{X}$, that is, there exists a target labeling function $f\colon \mathcal{X} \to \mathcal{Y}$ such that $y = f(x)$ almost surely, where $(x, y) \sim \mathcal{D}$. Moreover, we will be assuming that the elements of any training set S given as input to a learner are independently and identically distributed (i.i.d.), which assumption will be denoted as $S \sim \mathcal{D}^m$[1], where m is the size of S.

Success criteria Informally, the goal of a learner is to return a hypothesis, whose predictions tend to be correct. For instance, in our dogs example, a learner is successful, if it is able to classify an unlabeled dog example to the breed category it actually belongs. To formalize the notion of success, we first need to define a *loss function* $\ell\colon \mathcal{Y}^2 \to \mathbb{R}_{\geq 0}$ that quantifies how successful is a prediction with respect to some ground truth value. For binary and multiclass classification problems, the most natural choice is the $0-1$ loss, defined as $\ell_{0-1}(\hat{y}, y) \triangleq \mathbb{1}\{\hat{y} \neq y\}$, which simply says that we have zero loss, if the predicted label is the correct one, otherwise one. For regression problems, where $\mathcal{Y} \subseteq \mathbb{R}$, more appropriate choices would be the absolute loss $\ell_1(\hat{y}, y) \triangleq |\hat{y} - y|$ or the squared loss $\ell_2(\hat{y}, y) \triangleq (\hat{y} - y)^2$, which are able to reflect the dependence of the loss on some sort of distance between the prediction and the ground truth value.

Having determined some appropriate loss ℓ for our learning problem, we define the error of a hypothesis with respect to $\mathcal{D}$ as

$$L_{\mathcal{D},\ell}(h) \triangleq \mathop{\mathbf{E}}_{(x,y)\sim\mathcal{D}} [\ell(h(x), y)] \,,$$

that is, the expected loss over the data distribution[2]. Additionally, for cases, where there exists some target function f we want to approximate, we define the error of a hypothesis with respect to $\mathcal{D}_x$ and f as

$$L_{\mathcal{D}_x,f,\ell}(h) \triangleq \mathop{\mathbf{E}}_{x\sim\mathcal{D}_x} [\ell(h(x), f(x))] \,.$$

Then, in a more formal manner, the goal of a learning algorithm is to output a hypothesis that minimizes $L_{\mathcal{D},\ell}$ or $L_{\mathcal{D}_x,f,\ell}$ (depending on the context of the learning problem). A useful property that relates the aforementioned errors is as follows.

Proposition 2.1.1. *If ℓ satisfies the triangle inequality, then*

$$|L_{\mathcal{D},\ell}(h) - L_{\mathcal{D}_x,f,\ell}(h)| \leq L_{\mathcal{D},\ell}(f)$$

for any $h, f \in \mathcal{Y}^{\mathcal{X}}$.

Bayes predictor It can be shown that any hypothesis $h^* \in \mathcal{Y}^{\mathcal{X}}$, where

$$h^*(x) \in \mathop{\operatorname{argmin}}_{\hat{y}\in\mathcal{Y}} \mathop{\mathbf{E}}_{y\sim\mathcal{D}_{y|x}} [\ell(\hat{y}, y) \mid x] \,,$$

minimizes $L_{\mathcal{D},\ell}$ among all hypotheses in $\mathcal{Y}^{\mathcal{X}}$. Such a hypothesis is said to be a *Bayes predictor*. For instance, if $\mathcal{Y}$ is discrete and the loss is the $0-1$ loss, we can show that any hypothesis

$$h^*(x) \in \mathop{\operatorname{argmax}}_{\hat{y}\in\mathcal{Y}} \mathop{\mathbf{Pr}}_{y\sim\mathcal{D}_{y|x}} [y = \hat{y} \mid x]$$

is a Bayes predictor.

[1] $\mathcal{D}^m$ denotes the probability distribution over m-tuples on $(\mathcal{X} \times \mathcal{Y})^m$ induced by drawing each element of the tuple from $\mathcal{D}$, independently of the other members of the tuple.

[2] Since the loss ℓ is treated as a random variable, we require that the function $g_h\colon \mathcal{X} \times \mathcal{Y} \to \mathbb{R}_{\geq 0}$, where $g_h(x, y) = \ell(h(x), y)$, is measurable for all $h \in \mathcal{H}$.

Remark. *We stress that any learner is assumed to have no knowledge of the data distribution $\mathcal{D}$, but only of the specific training set that was given as input to it. Had $\mathcal{D}$ been known, we would be able to calculate a Bayes predictor and minimize $L_{\mathcal{D},\ell}$ with any learning procedure being unnecessary.*

2.2 Empirical Risk Minimization

As mentioned before, a learning algorithm receives a training set sampled according to $\mathcal{D}$ and outputs a hypothesis from $\mathcal{X}$ to $\mathcal{Y}$ that should minimize the error with respect to $\mathcal{D}$. To this end, it is vital that the learning algorithm has the ability to calculate some measure of error, through which it will be able to distinguish good from bad hypotheses during the learning process. Since the learner has no knowledge of $\mathcal{D}$, it is incapable of using $L_{\mathcal{D},\ell}$ to evaluate hypotheses. This leads us to introducing the *training error* (also called *empirical error* or *empirical risk*) that is defined as

$$\hat{L}_{S,\ell}(h) \triangleq \frac{1}{m}\sum_{t=1}^{m} \ell\left(h\left(x^{(t)}\right), y^{(t)}\right),$$

where $S = \left(\left(x^{(1)}, y^{(1)}\right), \ldots, \left(x^{(m)}, y^{(m)}\right)\right) \in (\mathcal{X} \times \mathcal{Y})^m$, with $m \geq 1$, is the training set. The process of finding a hypothesis that minimizes the training error corresponds to the learning paradigm that is referred to as *Empirical Risk Minimization* (ERM). The intuition behind the ERM paradigm is that if some hypothesis h achieves a small training error on a training set S, which is a proxy of the unknown data distribution $\mathcal{D}$, then we can hope that h will also achieve a small error with respect to $\mathcal{D}$, which is our actual goal.

However natural this approach might seem, we can show that it may perform poorly, if applied without any further restrictions. In particular, one can construct hypotheses that ensure zero training error on S, but fail to minimize the error with respect $\mathcal{D}$, namely fail to *generalize* over new data, which is known as the *overfitting* phenomenon. To solve this problem, we restrict the search space of hypotheses to a specific hypothesis class $\mathcal{H}$, whose choice should be due to some prior knowledge about the learning problem in consideration. This procedure is called *inductive bias*.

Definition 2.2.1. *A learning algorithm $A \in \mathcal{A}_{\mathcal{X},\mathcal{Y}}$ is an ERM learner for a hypothesis class $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ with respect to a loss ℓ, if $A(S) \in \operatorname{argmin}_{h\in\mathcal{H}} \hat{L}_{S,\ell}(h)$ for any training set $S \in \bigcup_{n=1}^{\infty}(\mathcal{X} \times \mathcal{Y})^n$.*

We denote by $\mathrm{ERM}_{\mathcal{H},\ell}$ the set of all ERM learners for $\mathcal{H}$ with respect to ℓ. In a few sections, we will formulate some conditions a hypothesis class should satisfy so that an ERM learner is guaranteed not to overfit in the binary classification setting.

2.3 The PAC Learning Model

In the previous section, we described the context of a learning problem, but did not give a precise definition of learnability. We now present the Probably Approximately Correct (PAC) learning model, first introduced by Valiant [1984], which formalizes the notion of learnability. Specifically, we use its adjusted form to prediction problems and general loss functions, following the notation of Daniely et al. [2014], Hopkins et al. [2023] and Shalev-Shwartz and Ben-David [2022]. In what follows, we let $\mathcal{X}$ be a domain space, $\mathcal{Y}$ be a label space, $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ be a hypothesis class, $\ell\colon \mathcal{Y}^2 \to \mathbb{R}_{\geq 0}$ be a loss function and $A \in \mathcal{A}_{\mathcal{X},\mathcal{Y}}$ be a learning algorithm.

2.3.1 Realizable PAC Learning

We first consider the simplest *realizable* version of PAC learning, where a target function that incurs zero expected loss on $\mathcal{D}$ is assumed to exist in the hypothesis class $\mathcal{H}$ into consideration. Our goal is to find a hypothesis that, with high probability, achieves error close to zero.

Definition 2.3.1 (Realizability Assumption)**.** *A probability distribution $\mathcal{D}$ on $\mathcal{X} \times \mathcal{Y}$ is said to be realizable by $\mathcal{H}$, if there exists some $h \in \mathcal{H}$ such that $L_{\mathcal{D},\ell}(h) = 0$.*

Definition 2.3.2 (Realizable Sample Complexity of a Learning Algorithm)**.** *The realizable sample complexity of A with respect to $\mathcal{H}$ and ℓ is the function $m^r_{A,\mathcal{H},\ell}\colon (0,\infty)^2 \to \mathbb{N}$ defined as follows: For every $\epsilon, \delta > 0$, $m^r_{A,\mathcal{H},\ell}(\epsilon,\delta)$ is the minimal integer such that for every*

- *integer* $m \geq m^r_{A,\mathcal{H},\ell}(\epsilon, \delta)$ *and*
- *probability distribution* $\mathcal{D}$ *on* $\mathcal{X} \times \mathcal{Y}$ *that is realizable by* $\mathcal{H}$*, it holds:*

$$\underset{S \sim \mathcal{D}^m}{\mathbf{Pr}} \left[L_{\mathcal{D},\ell} \left(A(S) \right) > \epsilon \right] \leq \delta$$

For every $\epsilon, \delta > 0$ *such that no integer satisfies the above inequality, we define* $m^r_{A,\mathcal{H},\ell}(\epsilon, \delta) = \infty$*.*

Remark. *If* ℓ *satisfies the (reasonable for a loss function) property* $\ell(x, y) = 0 \iff \ell(z, x) = \ell(z, y)$ *(which will always be the case for the loss functions considered in this thesis), then the second sentence in the above definition can be restated as follows: For every* $\epsilon, \delta > 0$*,* $m^r_{A,\mathcal{H},\ell}(\epsilon, \delta)$ *is the minimal integer such that for every integer* $m \geq m^r_{A,\mathcal{H},\ell}(\epsilon, \delta)$*, every probability distribution* $\mathcal{D}_x$ *on* $\mathcal{X}$ *and every* $f \in \{\pm 1\}^{\mathcal{X}}$*, it holds*

$$\underset{S_x \sim \mathcal{D}_x^m}{\mathbf{Pr}} \left[L_{\mathcal{D}_x, f, \ell} \left(A \left(S_x, h|_{S_x} \right) \right) > \epsilon \right] \leq \delta \, .$$

This equivalent statement is often preferred in literature when dealing with binary classification problems, as it makes explicit the existence of a target function.

Definition 2.3.3 (Realizable Sample Complexity of a Hypothesis Class)**.** *The realizable sample complexity of* $\mathcal{H}$ *with respect to* ℓ *is the function* $m^r_{\mathrm{PAC},\mathcal{H},\ell} \colon (0, \infty)^2 \to \mathbb{N}$ *defined as*

$$m^r_{\mathrm{PAC},\mathcal{H},\ell}(\epsilon, \delta) = \inf_{A \in \mathcal{A}_{\mathcal{X},\mathcal{Y}}} m^r_{A,\mathcal{H},\ell}(\epsilon, \delta)$$

for all $\epsilon, \delta > 0$*.*

Namely, the realizable sample complexity of $\mathcal{H}$ is the best (lowest) realizable sample complexity that an algorithm can achieve.

Definition 2.3.4 (Realizable PAC learnability)**.** $\mathcal{H}$ *is said to be realizably PAC learnable with respect to* ℓ*, if it holds that* $m^r_{\mathrm{PAC},\mathcal{H},\ell}(\epsilon, \delta) < \infty$ *for all* $\epsilon, \delta > 0$*. Moreover,* $\mathcal{H}$ *is said to be realizably PAC learnable with* A *with respect to* ℓ*, if it holds that* $m^r_{A,\mathcal{H},\ell}(\epsilon, \delta) < \infty$ *for all* $\epsilon, \delta > 0$*.*

Some comments are in order. The parameter ϵ (accuracy parameter) in the above definitions determines how much the error of the hypothesis returned by A can exceed its optimal value (which is zero, due to the realizability assumption) and corresponds to the "approximately" part of PAC. The parameter δ (confidence parameter) is related to the probability that the aforementioned approximation condition is satisfied and corresponds to the "probably" part of PAC. In particular, δ captures the dependence of the training procedure on the specific training set S that was used. Since S is finite, it could be the case that S is nonrepresentative of $\mathcal{D}$, in a way that the we would be unable to guarantee the approximation condition for the error.

Definition 2.3.5 (Realizable ERM sample complexity)**.** *The realizable ERM sample complexity of* $\mathcal{H}$ *with respect to* ℓ *is the function* $m^r_{\mathrm{ERM},\mathcal{H},\ell} \colon (0, \infty)^2 \to \mathbb{N}$ *defined as*

$$m^r_{\mathrm{ERM},\mathcal{H},\ell}(\epsilon, \delta) = \sup_{A \in \mathrm{ERM}_{\mathcal{H},\ell}} m^r_{A,\mathcal{H},\ell}(\epsilon, \delta)$$

for all $\epsilon, \delta > 0$*.*

Namely, the realizable ERM sample complexity of $\mathcal{H}$ is the realizable sample complexity that can be guaranteed for any ERM learner. Obviously, we have that $m^r_{\mathrm{PAC},\mathcal{H},\ell}(\epsilon, \delta) \leq m^r_{\mathrm{ERM},\mathcal{H},\ell}(\epsilon, \delta)$ for any $\epsilon, \delta > 0$.

2.3.2 Agnostic PAC Learning

We now consider the more general *agnostic* version of PAC Learning (Haussler [1992], Kearns et al. [1992]), in which the realizablity assumption is waived. Here, the goal is to find a hypothesis that, with arbitrarily high probability, achieves error arbitrarily close to the minimum one achieved by any hypothesis in $\mathcal{H}$.

Definition 2.3.6 (Agnostic Sample Complexity of a Learning Algorithm)**.** *The agnostic sample complexity of a* A *with respect to* $\mathcal{H}$ *and* ℓ *is the function* $m^a_{A,\mathcal{H},\ell} \colon (0, \infty)^2 \to \mathbb{N}$ *defined as follows: For every* $\epsilon, \delta > 0$*,* $m^a_{A,\mathcal{H},\ell}(\epsilon, \delta)$ *is the minimal integer such that for every*

- *integer* $m \geq m^r_{A,\mathcal{H},\ell}(\epsilon, \delta)$ *and*
- *probability distribution* $\mathcal{D}$ *on* $\mathcal{X} \times \mathcal{Y}$*, it holds:*

$$\underset{S\sim\mathcal{D}^m}{\mathbf{Pr}}\left[L_{\mathcal{D},\ell}\left(A(S)\right) > \inf_{h\in\mathcal{H}} L_{\mathcal{D},\ell}(h) + \epsilon\right] \leq \delta$$

For every $\epsilon, \delta > 0$ *such that no integer satisfies the above inequality, we define* $m^a_{A,\mathcal{H},\ell}(\epsilon, \delta) = \infty$*.*

Definition 2.3.7 (Agnostic Sample Complexity of a Hypothesis Class)**.** *The agnostic sample complexity of* $\mathcal{H}$ *with respect to* ℓ *is the function* $m^a_{\mathrm{PAC},\mathcal{H},\ell}\colon (0,\infty)^2 \to \mathbb{N}$ *defined as*

$$m^a_{\mathrm{PAC},\mathcal{H},\ell}(\epsilon, \delta) = \inf_{A\in\mathcal{A}_{\mathcal{X},\mathcal{Y}}} m^a_{A,\mathcal{H},\ell}(\epsilon, \delta)$$

for all $\epsilon, \delta > 0$*.*

Definition 2.3.8 (Agnostic PAC learnability)**.** $\mathcal{H}$ *is said to be agnostically PAC learnable with respect to* ℓ*, if it holds that* $m^a_{\mathrm{PAC},\mathcal{H},\ell}(\epsilon, \delta) < \infty$ *for all* $\epsilon, \delta > 0$*. Moreover,* $\mathcal{H}$ *is said to be agnostically PAC learnable with* A *with respect to* ℓ*, if it holds that* $m^a_{A,\mathcal{H},\ell}(\epsilon, \delta) < \infty$ *for all* $\epsilon, \delta > 0$*.*

Definition 2.3.9 (Agnostic ERM sample complexity)**.** *The agnostic ERM sample complexity of* $\mathcal{H}$ *with respect to* ℓ *is the function* $m^a_{\mathrm{ERM},\mathcal{H},\ell}\colon (0,\infty)^2 \to \mathbb{N}$ *defined as*

$$m^a_{\mathrm{ERM},\mathcal{H},\ell}(\epsilon, \delta) = \sup_{A\in\mathrm{ERM}_{\mathcal{H},\ell}} m^a_{A,\mathcal{H},\ell}(\epsilon, \delta)$$

for all $\epsilon, \delta > 0$*.*

2.3.3 Extensions of the PAC Model

We now discuss some variations of the standard realizable and agnostic PAC models, which are more often than not considered with an eye to making the acquirement of theoretical guarantees easier or ensuring computational efficiency.

Proper versus Improper PAC Learning Notice that in the preceding definitions of PAC learnability of a class $\mathcal{H}$, any learner is required to output a hypothesis $h \in \mathcal{Y}^{\mathcal{X}}$, but not necessarily within $\mathcal{H}$. Such a choice is justified, if our mere goal is to minimize the expected loss irrespective of the representation of the output hypothesis. This type of learning is referred to as *improper* or *representation-independent*. Nevertheless, as it will become more clear afterwards, it is sometimes preferable that $h \in \mathcal{H}$ for computational reasons, i.e. in order to ensure smaller representation size of h and smaller time to compute the output of h. If we require that $h \in \mathcal{H}$, then the corresponding learning type is referred to as *proper* or *representation-dependent*.

Efficient PAC Learning The aforementioned definitions of PAC learnability focus on the sample complexity to carry out a learning task, which covers the statistical aspects of learning. When it comes to creating learning algorithms, one has to take the computational aspects of learning into consideration as well. Namely, it is crucial that both obtaining a hypothesis from a training set and predicting labels with that hypothesis can be done in an efficient manner. To this end, it is critical that we define some efficiency criterion, which should naturally be related to the parameters of the learning task, and extend the standard definitions of PAC learnability accordingly. For instance, in the special case of binary classification problems with $\mathcal{X} = \mathbb{R}^d$, we normally require that the sample and computational complexity of learning algorithms be bounded by a polynomial in the dimension d of the instances, in the bit complexity of the examples and in $1/\epsilon$ and $1/\delta$, where $\epsilon, \delta > 0$ are the accuracy and confidence PAC parameters respectively. If our learning problem happens to be associated with more parameters (as in the presence of noise that will be studied later), it is desirable that the sample and computational complexity has a polynomial dependence on them as well. Additionally, it is critical that the time a learner's output hypothesis takes to label a new instance is also polynomially bounded by the aforementioned parameters (which constitutes an indication of the importance of proper learning). For a more general and rigorous quantification of the notion of efficiency in the PAC learning model we refer to Kearns and Vazirani [1994] and Shalev-Shwartz and Ben-David [2022].

Distribution-dependent PAC Learning The standard realizable and agnostic PAC learning models are often referred to as *distribution-independent* or *distribution-free* since no assumption about the marginal distribution $\mathcal{D}_x$ of $\mathcal{D}$ on $\mathcal{X}$ is made. In practice, though, the distributions on instances tend to manifest "nice" concentration properties that make algorithms outperform the worst-case generalization bounds provided by the standard PAC learning models. Furthermore, imposing no restrictions on $\mathcal{D}_x$ might make it difficult to obtain efficiency guarantees, especially in the presence of noise, as we will see later. This is why we often consider a variant of the standard PAC model, the *distribution-dependent* or *distribution-specific* model, which requires $\mathcal{D}_x$ to belong to a specific family $\mathcal{F}$ of probability distributions on $\mathcal{X}$.

Relaxation of the standard PAC guarantees Finally, another way, besides the distribution-dependent PAC model, to overcome arising difficulties in the acquirement of generalization bounds, is to compromise on weaker guarantees than the ones required by the standard realizable and agnostic PAC models. Such an example is the c-agnostic PAC model, where we require that for any $\epsilon, \delta > 0$ the learner outputs a hypothesis $g \in \mathcal{Y}^{\mathcal{X}}$ such that, with probability at least $1 - \delta$, it holds $L_{\mathcal{D},\ell}(g) \leq c \inf_{h \in \mathcal{H}} L_{\mathcal{D},\ell}(h) + \epsilon$, where $c \geq 1$ is a constant.

2.4 The No-Free-Lunch Theorem

Notice that PAC learnability has been defined with respect to some hypothesis class. Moreover, we have seen that the ERM rule might fail, if not applied within some specific hypothesis class (ideally one we believe is appropriate for our learning problem). A natural arising question is whether such a restriction is necessary. In other words, does there exist some universal learner that can perform well in all learning tasks, without restriction within any hypothesis class, or rather ensure arbitrarily small error with arbitrarily high probability, with respect to any distribution, using finite training samples? The answer to this question is negative, as the following theorem suggests.

Theorem 2.4.1 (No-Free-Lunch (Shalev-Shwartz and Ben-David [2022]))**.** *For any learning algorithm $A \in \mathcal{A}_{\mathcal{X},\{\pm 1\}}$ and $m < |\mathcal{X}|/2$, there exist a distribution $\mathcal{D}$ on $\mathcal{X} \times \{\pm 1\}$ and a hypothesis $h \in \{\pm 1\}^{\mathcal{X}}$ such that $L_{\mathcal{D},\ell_{0-1}}(h) = 0$ and*

$$\Pr_{S \sim \mathcal{D}^m} \left[L_{\mathcal{D},\ell_{0-1}} \left(A(S) \right) \geq 1/8 \right] \geq 1/7 \,.$$

Intuitively, the No-Free-Lunch theorem states that for each learner there exists some learning task (distribution) in which it fails, whereas there exists some other learner that succeeds in the same task. A direct corollary is that if $\mathcal{X}$ is an infinite domain, then $\{\pm 1\}^{\mathcal{X}}$ is not PAC learnable with respect to ℓ_{0-1}, which justifies in a more formal manner the necessity of restricting our attention to specific hypothesis classes, if we hope of obtaining any PAC learnability guarantees.

2.5 VC Dimension

We now focus our attention on binary classification problems with the loss function being the $0-1$ loss. We study the conditions under which classes are PAC learnable and try to quantify how easy it is to learn a hypothesis class in terms of the required sample complexity. To this end, we define the Vapnik-Chervonenkis dimension (VC dimension), a combinatorial notion that characterizes the learnability of classes. Then, we state the fundamental theorem of statistical learning theory, which relates the VC dimension of a class with the sample complexity of learning it.

We assume, without loss of generality, that our binary label space is the set $\{\pm 1\}$ and proceed by defining some prerequisite notions, before providing the definition of the VC dimension.

Definition 2.5.1 (Restriction of $\mathcal{H}$ to C)**.** *Let $\mathcal{H} \subseteq \{\pm 1\}^{\mathcal{X}}$ be a hypothesis class and let $C \subseteq \mathcal{X}$ be a finite set. The restriction of $\mathcal{H}$ to C is the set $\mathcal{H}|_C = \{h|_C : h \in \mathcal{H}\}$, that is, the set of functions from C to $\{\pm 1\}$ that can be derived from $\mathcal{H}$.*

Definition 2.5.2 (Shattering)**.** *Let $\mathcal{H} \subseteq \{\pm 1\}^{\mathcal{X}}$ be a hypothesis class and let $C \subseteq \mathcal{X}$ be a finite set. We say that $\mathcal{H}$ shatters C, if $\mathcal{H}|_C = \{\pm 1\}^C$, that is, the restriction of $\mathcal{H}$ to C is the set of all functions from C to $\{\pm 1\}$.*

To understand the usefulness of the above definitions, we need to get some deeper insight into the No-Free-Lunch theorem. A more detailed statement of it (according to the proof of Shalev-Shwartz and Ben-David [2022]) would be that for any learning algorithm $A \in \mathcal{A}_{\mathcal{X},\{\pm 1\}}$ and any subset $C = \{x_1, \ldots, x_{2m}\}$ of $\mathcal{X}$ of size $2m$, there exist a hypothesis $h \in \{0,1\}^C$ and a uniform distribution $\mathcal{D}$ on $\{(x_1, h(x_1)), \ldots, (x_{2m}, h(x_{2m}))\}$ such that $\mathbf{Pr}_{S\sim\mathcal{D}^m}\left[L_{\mathcal{D},\ell_{0-1}}(A(S)) \geq 1/8\right] \geq 1/7$.

Corollary 2.5.1. *Let $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$. If there exists a set $C \subseteq \mathcal{X}$ of size $2m$ that is shattered by $\mathcal{H}$, then for any learning algorithm $A \in \mathcal{A}_{\mathcal{X},\{\pm 1\}}$, there exist a distribution $\mathcal{D}$ on $\mathcal{X} \times \{\pm 1\}$ and a hypothesis $h \in \mathcal{H}$ such that $L_{\mathcal{D},\ell_{0-1}}(h) = 0$ and $\mathbf{Pr}_{S\sim\mathcal{D}^m}\left[L_{\mathcal{D},\ell_{0-1}}(A(S)) \geq 1/8\right] \geq 1/7$.*

Intuitively, this means that the larger the maximal size of a set that $\mathcal{H}$ can shatter becomes, the larger becomes the lower bound on samples required to learn some hypothesis that can achieve the PAC learning guarantee. This fact naturally leads to the following definition.

Definition 2.5.3 (VC Dimension)**.** *The VC dimension of a hypothesis class $\mathcal{H} \subseteq \{\pm 1\}^{\mathcal{X}}$, denoted $\mathrm{VC}(\mathcal{H})$, is the maximal size of a finite set $C \subseteq \mathcal{X}$ that can be shattered by $\mathcal{H}$. If $\mathcal{H}$ can shatter sets of arbitrarily large size, we define $\mathrm{VC}(\mathcal{H}) = \infty$.*

A direct corollary of the definition of shattering is that if $\mathcal{H}$ shatters a finite set C, then it also shatters all subsets of C. Contrapositively, if every set of size $d \geq 1$ is not shattered by $\mathcal{H}$, then every set of size greater than d is also not shattered by $\mathcal{H}$. Therefore, to show that $\mathrm{VC}(\mathcal{H}) = d$, we need to show that

1. There exists a set $C \subseteq \mathcal{X}$ with $|C| = d$ that is shattered by $\mathcal{H}$.
2. Every set $C \subseteq \mathcal{X}$ with $|C| = d + 1$ is not shattered by $\mathcal{H}$.

We remark that any nonempty class $\mathcal{H}$ trivially shatters a set of size zero, so $\mathrm{VC}(\mathcal{H}) \geq 0$. Moreover, the condition $\mathrm{VC}(\mathcal{H}) = 0$ holds, if and only if $\mathcal{H}$ contains a single hypothesis, a constant function.

Lemma 2.5.1. *If $\mathrm{VC}(\mathcal{H}) = \infty$, then $\mathcal{H}$ is not PAC learnable with respect to ℓ_{0-1}.*

Proof. Since $\mathrm{VC}(\mathcal{H}) = \infty$, there exists a set of arbitrarily large size shattered by $\mathcal{H}$. Hence, from Corollary 1, we get that for any $0 < \epsilon < 1/8$ and $0 < \delta < 1/7$, it holds that $m^r_{A,\mathcal{H},\ell}(\epsilon, \delta) = \infty$, which means that $\mathcal{H}$ is not realizably PAC learnable (and therefore neither agnostically PAC learnable) with respect to ℓ_{0-1}. □

2.6 The Fundamental Theorem of Statistical Learning

The following theorems relate the VC dimension of classes with their learnability in the PAC learning model.

Theorem 2.6.1 (The Fundamental Theorem of Statistical Learning)**.** *Let $\mathcal{H}$ be a hypothesis class of functions from a domain set $\mathcal{X}$ to $\{\pm 1\}$. Then, the following are equivalent.*

1. *$\mathcal{H}$ has finite VC dimension.*
2. *$\mathcal{H}$ is realizably PAC learnable with respect to ℓ_{0-1}.*
3. *$\mathcal{H}$ is agnostically PAC learnable with respect to ℓ_{0-1}.*
4. *$\mathcal{H}$ is realizably PAC learnable with any $A \in \mathrm{ERM}_{\mathcal{H},\ell_{0-1}}$ with respect to ℓ_{0-1}.*
5. *$\mathcal{H}$ is agnostically PAC learnable with any $A \in \mathrm{ERM}_{\mathcal{H},\ell_{0-1}}$ with respect to ℓ_{0-1}.*

Theorem 2.6.2 (The Fundamental Theorem of Statistical Learning - Quantitative Version)**.** *Let $\mathcal{H}$ be a hypothesis class of functions from a domain set $\mathcal{X}$ to $\{\pm 1\}$. There exist universal constants $C_1, C_2 > 0$ such that*

$$C_1 \frac{\mathrm{VC}(\mathcal{H}) + \ln(1/\delta)}{\epsilon} \leq m^r_{\mathrm{PAC},\mathcal{H},\ell_{0-1}}(\epsilon, \delta) \leq m^r_{\mathrm{ERM},\mathcal{H},\ell_{0-1}}(\epsilon, \delta) \leq C_2 \frac{\mathrm{VC}(\mathcal{H}) \ln(1/\epsilon) + \ln(1/\delta)}{\epsilon}$$

and

$$C_1 \frac{\mathrm{VC}(\mathcal{H}) + \ln(1/\delta)}{\epsilon^2} \leq m^a_{\mathrm{PAC},\mathcal{H},\ell_{0-1}}(\epsilon, \delta) \leq m^a_{\mathrm{ERM},\mathcal{H},\ell_{0-1}}(\epsilon, \delta) \leq C_2 \frac{\mathrm{VC}(\mathcal{H}) + \ln(1/\delta)}{\epsilon^2}$$

for all $\epsilon, \delta \in (0,1)$.

2.7 Noise Models in Binary Classification

The arguably simplest starting point when studying the learnability of a hypothesis class in the binary classification setting, is to consider the realizable case, where all instances are assumed to be labeled by some target function within the class. This setting is also referred to as noiseless, in the sense that there is no corrupting factor that could modify the ground truth labels (the ones dictated by the target function). However, in real world applications, it is often the case that the learner has access to examples, which might contain noise, namely, there might be some portion of the examples consisting of instances accompanied by a wrong label. This is the motivation for formalizing the presence of noise in the data, namely making distributional assumptions that extend the realizable case so as to capture more general scenarios. In this section, we present three noise models in ascending order of generality that will be used throughout the thesis.

2.7.1 The Random Classification Noise Model

The first and simplest among the noise models we will present, is the random classification noise (RCN) model, which was introduced by Angluin and Laird [1988].

Definition 2.7.1 (Random Classification Noise)**.** *A probability distribution $\mathcal{D}$ on $\mathcal{X} \times \{\pm 1\}$ is said to satisfy the random classification noise (RCN) condition, if there exists some $\eta \in [0, 1/2)$ such that $\min\left\{\mathbf{Pr}_{y\sim\mathcal{D}_{y|x}}[y = 1 \mid x], \mathbf{Pr}_{y\sim\mathcal{D}_{y|x}}[y \neq 1 \mid x]\right\} = \eta$ for all $x \in \mathcal{X}$.*

Proposition 2.7.1. *A probability distribution $\mathcal{D}$ on $\mathcal{X} \times \{\pm 1\}$ satisfies the random classification noise condition, if and only if there exists some (unique) function $f\colon \mathcal{X} \to \{\pm 1\}$ and some $\eta \in [0, 1/2)$ such that for all $x \in \mathcal{X}$, it holds that $\mathbf{Pr}_{y\sim\mathcal{D}_{y|x}}[y \neq f(x) \mid x] = \eta$. Then, we also say that $\mathcal{D}$ is an (η, f)-RCN distribution.*

This alternative definition, which is in fact the original given by Angluin and Laird [1988], leads to interpreting the RCN model as having some noiseless dataset, where all samples are deterministically labeled by f, and an adversary flips each label independently with some constant probability $\eta \in [0, 1/2)$. The constraint $\eta \in [0, 1/2)$ is due to the fact that if $\eta = 1/2$, then we have zero information about which class each point belongs to, so every learning procedure should be expected to fail. Furthermore, if $\eta > 1/2$, then we can simply flip every label and reduce the problem to the case with flipping probability $1 - \eta < 1/2$.

The following lemma relates $L_{\mathcal{D}_x,f,\ell_{0-1}}$ (error with respect to the target function) with $L_{\mathcal{D},\ell_{0-1}}$ (misclassification error). Its proof can be found in Appendix C.

Lemma 2.7.1. *Let $\mathcal{D}$ be an (η, f)-RCN distribution, where $\eta \in [0, 1/2)$ and $f \in \{\pm 1\}^{\mathcal{X}}$. For any $h \in \{\pm 1\}^{\mathcal{X}}$, it holds that*

$$L_{\mathcal{D}_x,f,\ell_{0-1}}(h) = \frac{L_{\mathcal{D},\ell_{0-1}}(h) - \eta}{1 - 2\eta}$$

and f is a minimizer of $L_{\mathcal{D},\ell_{0-1}}$.

We now adjust the PAC model to the distribution dependent setting in the presence of random classification noise. Our objective is to minimize the expected $0-1$ loss with respect to $\mathcal{D}_x$ and the function, with respect to which an RCN distribution is defined. In what follows, we let $\mathcal{X}$ be a domain space, $\mathcal{H} \subseteq \{\pm 1\}^{\mathcal{X}}$ be a hypothesis class, $A \in \mathcal{A}_{\mathcal{X},\{\pm 1\}}$ be a learning algorithm and $\mathcal{F}$ be a family of probability distributions on $\mathcal{X}$.

Definition 2.7.2 (Distribution-dependent RCN Sample Complexity)**.** *The distribution-dependent RCN sample complexity of A with respect to $\mathcal{H}$ and $\mathcal{F}$ is the function $m^{\mathrm{RCN}}_{A,\mathcal{H},\mathcal{F}}\colon (0,\infty) \times (0,\infty) \times [0, 1/2) \to \mathbb{N}$ defined as follows: For every $\epsilon, \delta > 0$ and $\eta \in [0, 1/2)$, $m^{\mathrm{RCN}}_{A,\mathcal{H},\mathcal{F}}(\epsilon, \delta, \eta)$ is the minimal integer such that for every*

- *integer $m \geq m^{\mathrm{RCN}}_{A,\mathcal{H},\mathcal{F}}(\epsilon, \delta, \eta)$,*
- *target function $f \in \mathcal{H}$ and*
- *(η, f)-RCN probability distribution with $\mathcal{D}_x \in \mathcal{F}$, it holds:*

$$\underset{S_x \sim \mathcal{D}_x^m}{\mathbf{Pr}} \left[L_{\mathcal{D}_x, f, \ell_{0-1}} \left(A \left(S_x, h|_{S_x} \right) \right) > \epsilon \right] \leq \delta$$

For every $\epsilon, \delta > 0$ and $\eta \in [0, 1/2)$ such that no integer satisfies the above inequality, we define $m^{\mathrm{RCN}}_{A,\mathcal{H},\mathcal{F}}(\epsilon, \delta, \eta) = \infty$.

Definition 2.7.3 (Distribution-dependent PAC Learnability with RCN)**.** *$\mathcal{H}$ is said to be PAC learnable with A with respect to $\mathcal{F}$ in the presence of RCN, if it holds that $m^{\mathrm{RCN}}_{A,\mathcal{H},\mathcal{F}}(\epsilon, \delta, \eta) < \infty$ for all $\epsilon, \delta > 0$ and $\eta \in [0, 1/2)$.*

2.7.2 The Massart Noise Model

A drawback of the RCN model is that its assumption is too strong to be realistic, in the sense that in a real case scenario, $\mathbf{Pr}_{y \sim \mathcal{D}_{y|x}}[y \neq f(x) \mid x]$ would probably not have the same value for all points x of the feature space. This is the motivation for studying a much more general model, the Massart noise model, introduced by Massart and Nédélec [2006], which is defined below.

Definition 2.7.4 (Massart Noise)**.** *A probability distribution $\mathcal{D}$ on $\mathcal{X} \times \{\pm 1\}$ is said to satisfy the Massart (or bounded) noise condition, if there exists some $\beta > 0$ such that $\left| \mathbf{Pr}_{y \sim \mathcal{D}_{y|x}}[y = 1 \mid x] - 1/2 \right| \geq \beta$ for all $x \in \mathcal{X}$.*

Proposition 2.7.2. *A probability distribution $\mathcal{D}$ on $\mathcal{X} \times \{\pm 1\}$ satisfies the Massart noise condition, if and only if there exists some (unique) function $f \colon \mathcal{X} \to \{\pm 1\}$ and some $\eta \in [0, 1/2)$ such that for all $x \in \mathcal{X}$, it holds that $\mathbf{Pr}_{y \sim \mathcal{D}_{y|x}}[y \neq f(x) \mid x] \leq \eta$. Then, we also say that $\mathcal{D}$ is an (η, f)-Massart distribution.*

This alternative definition leads to interpreting the Massart noise model as having some noiseless dataset, where all samples are deterministically labeled by f, and an adversary flips each label independently with probability at most $\eta \in [0, 1/2)$. Obviously, since the Massart noise model imposes only an upper bound on the flipping probability rather than a constant value, it constitutes a significant generalization of the RCN model.

The following lemma relates $L_{\mathcal{D}_x, f, \ell_{0-1}}$ (error with respect to the target function) with $L_{\mathcal{D}, \ell_{0-1}}$ (misclassification error). Its proof can be found in Appendix C.

Lemma 2.7.2. *Let $\mathcal{D}$ be an (η, f)-Massart distribution, where $\eta \in [0, 1/2)$ and $f \in \{0, 1\}^{\mathcal{X}}$. For any $h \in \{0, 1\}^{\mathcal{X}}$, it holds that*

$$L_{\mathcal{D}_x, f, \ell_{0-1}}(h) \leq \frac{L_{\mathcal{D}, \ell_{0-1}}(h) - L_{\mathcal{D}, \ell_{0-1}}(f)}{1 - 2\eta}$$

and f is a minimizer of $L_{\mathcal{D}, \ell_{0-1}}$.

Definition 2.7.5 (Distribution-dependent Massart Sample Complexity)**.** *The distribution-dependent Massart sample complexity of A with respect to $\mathcal{H}$ and $\mathcal{F}$ is the function $m^{\mathrm{Massart}}_{A,\mathcal{H},\mathcal{F}} \colon (0, \infty) \times (0, \infty) \times [0, 1/2) \to \mathbb{N}$ defined as follows: For every $\epsilon, \delta > 0$ and $\eta \in [0, 1/2)$, $m^{\mathrm{Massart}}_{A,\mathcal{H},\mathcal{F}}(\epsilon, \delta, \eta)$ is the minimal integer such that for every*

- *integer $m \geq m^{\mathrm{Massart}}_{A,\mathcal{H},\mathcal{F}}(\epsilon, \delta, \eta)$,*
- *target function $f \in \mathcal{H}$ and*
- *(η, f)-Massart probability distribution with $\mathcal{D}_x \in \mathcal{F}$, it holds:*

$$\underset{S_x \sim \mathcal{D}_x^m}{\mathbf{Pr}} \left[L_{\mathcal{D}_x, f, \ell_{0-1}} \left(A \left(S_x, h|_{S_x} \right) \right) > \epsilon \right] \leq \delta$$

For every $\epsilon, \delta > 0$ and $\eta \in [0, 1/2)$ such that no integer satisfies the above inequality, we define $m^{\mathrm{Massart}}_{A,\mathcal{H},\mathcal{F}}(\epsilon, \delta, \eta) = \infty$.

Definition 2.7.6 (Distribution-dependent PAC Learnability with Massart Noise)**.** *$\mathcal{H}$ is said to be PAC learnable with A with respect to $\mathcal{F}$ in the presence of Massart noise, if it holds that $m^{\mathrm{Massart}}_{A,\mathcal{H},\mathcal{F}}(\epsilon, \delta, \eta) < \infty$ for all $\epsilon, \delta > 0$ and $\eta \in [0, 1/2)$.*

2.7.3 The Tsybakov Noise Model

The Massart noise model generalizes the RCN model to a large extent, but still fails to capture scenarios, where the flipping probability can be arbitrarily close to $1/2$ for some points of the instance space. This is the motivation for defining the even more general Tsybakov noise model, which was originally proposed by Mammen and Tsybakov [1999] and later refined by Tsybakov [2004].

Definition 2.7.7 (Tsybakov Noise)**.** *A probability distribution $\mathcal{D}$ on $\mathcal{X} \times \{\pm 1\}$ is said to satisfy the Tsybakov noise condition, if there exist some $\alpha \in [0, 1)$ and $B \geq 1$ such that*

$$\mathop{\mathbf{Pr}}_{x \sim \mathcal{D}_x} \left[\left| \mathop{\mathbf{Pr}}_{y \sim \mathcal{D}_{y|x}} [y = 1 \mid x] - 1/2 \right| \leq t/2 \right] \leq B t^{\frac{\alpha}{1-\alpha}}$$

for all $t \geq 0$.

Proposition 2.7.3. *A probability distribution $\mathcal{D}$ on $\mathcal{X} \times \{\pm 1\}$ satisfies the Tsybakov noise condition, if and only if there exists some function $f \colon \mathcal{X} \to \{\pm 1\}$ and some $\alpha \in [0, 1)$ and $B \geq 1$ such that for all $t \geq 0$, it holds that $\mathbf{Pr}_{x \sim \mathcal{D}_x} \left[\mathbf{Pr}_{y \sim \mathcal{D}_{y|x}} [y \neq f(x) \mid x] \geq 1/2 - t/2 \right] \leq B t^{\frac{\alpha}{1-\alpha}}$. Then, we also say that $\mathcal{D}$ is an (α, B, f)-Tsybakov distribution.*

The intuition behind the Tsybakov noise model is as follows. While in Massart case there exists some universal $\eta \in [0, 1/2)$ such that the event $\mathbf{Pr}_{y \sim \mathcal{D}_{y|x}} [y \neq f(x) \mid x] > \eta$ can never occur, here $\mathbf{Pr}_{y \sim \mathcal{D}_{y|x}} [y \neq f(x) \mid x]$ can get arbitrarily close to $1/2$ for some instances, but the probability of observing the corresponding regions decays to zero, as we get closer to $1/2$. The aforementioned decay is controlled by the parameter α. In particular, when α tends to 1 we observe a more intense decay, while values of α close to zero, correspond to a slower decay, yielding a noisier distribution overall.

Definition 2.7.8 (Distribution-dependent Tsybakov Sample Complexity of a Learning Algorithm)**.** *The distribution-dependent Tsybakov sample complexity of a learning algorithm $A \in \mathcal{A}_{\mathcal{X},\mathcal{Y}}$ with respect to $\mathcal{H}$ and $\mathcal{F}$ is the function $m^{\text{Tsybakov}}_{A,\mathcal{H},\mathcal{F}} \colon (0, \infty) \times (0, \infty) \times [0, 1) \times [1, \infty) \to \mathbb{N}$ defined as follows: For every $\epsilon, \delta > 0$, $\alpha \in [0, 1)$ and $B \geq 1$, $m^{\text{Tsybakov}}_{A,\mathcal{H},\mathcal{F}}(\epsilon, \delta, \alpha, B)$ is the minimal integer such that for every*

- *integer $m \geq m^{\text{Tsybakov}}_{A,\mathcal{H},\mathcal{F}}(\epsilon, \delta, \alpha, B)$,*
- *target function $f \in \mathcal{H}$ and*
- *(α, B, f)-Tsybakov probability distribution with $\mathcal{D}_x \in \mathcal{F}$, it holds:*

$$\mathop{\mathbf{Pr}}_{S_x \sim \mathcal{D}_x^m} \left[L_{\mathcal{D}_x, f, \ell_{0-1}} \left(A \left(S_x, h|_{S_x} \right) \right) > \epsilon \right] \leq \delta$$

For every $\epsilon, \delta > 0$, $\alpha \in [0, 1)$ and $B \geq 1$ such that no integer satisfies the above inequality, we define $m^{\text{Tsybakov}}_{A,\mathcal{H},\mathcal{F}}(\epsilon, \delta, \alpha, B) = \infty$.

Definition 2.7.9 (Distribution-dependent PAC Learnability with Tsybakov Noise)**.** *$\mathcal{H}$ is said to be PAC learnable with A with respect to $\mathcal{F}$ in the presence of Tsybakov noise, if it holds that $m^{\text{Tsybakov}}_{A,\mathcal{H},\mathcal{F}}(\epsilon, \delta, \alpha, B) < \infty$ for all $\epsilon, \delta > 0$, $\alpha \in [0, 1)$ and $B \geq 1$.*

Chapter 3

Learning Halfspaces

In this chapter, we study the hypothesis class of *halfspaces* or *linear threshold functions*. This is one of the most fundamental concept classes within the binary classification setting and has been vastly used over the years either standalone, or as part of more sophisticated structures such as Support Vector Machines (SVMs) or Neural Networks. The extensive use of halfspaces in learning tasks is due to the capability to learn them efficiently, whilst maintaining simplicity and intuition. We begin by providing some basic definitions and finding the VC dimension of the class of halfspaces. Then, we study algorithms for learning halfspaces both in the noiseless and the noisy PAC setting.

Definition 3.0.1 (Halfspace)**.** *A halfspace or linear threshold function (LTF) in $\mathbb{R}^d$ is any function $h_{\boldsymbol{w},b}\colon \mathbb{R}^d \to \{\pm 1\}$ of the form*

$$h_{\boldsymbol{w},b}(\boldsymbol{x}) = \operatorname{sign}(\langle \boldsymbol{w}, \boldsymbol{x} \rangle + b)\,,$$

where $\boldsymbol{w} \in \mathbb{R}^d$ is a weight vector, $b \in \mathbb{R}$ is a bias term.

A halfspace $h_{\boldsymbol{w},b}$ is said to be homogeneous if $b = 0$, namely if the hyperplane that defines it contains the origin and we denote it as $h_{\boldsymbol{w}} = h_{\boldsymbol{w},0}$. We denote by $\mathcal{H}^d_{\mathrm{LTF}}$ (resp. $\mathcal{H}^d_{\mathrm{HLTF}}$) the class of halfspaces (resp. homogeneous halfspaces) in $\mathbb{R}^d$.

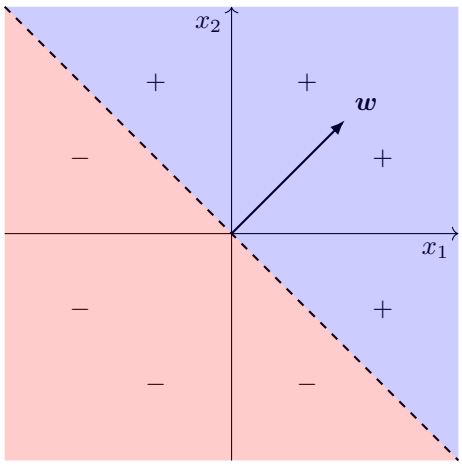


Figure 3.1: Visualization of a homogeneous halfspace in $\mathbb{R}^2$

Definition 3.0.2 (Linear separability)**.** *A set $S \subset \mathbb{R}^d \times \{-1, 1\}$ is said to be linearly separable, if there exist some $\boldsymbol{w} \in \mathbb{R}^d$ and $b \in \mathbb{R}$ such that $y(\langle \boldsymbol{w}, \boldsymbol{x} \rangle + b) > 0$ for all $(\boldsymbol{x}, y) \in S$.*

Remark. *Notice that each halfspace $h_{\boldsymbol{w},b}(\boldsymbol{x}) = \operatorname{sign}(\langle \boldsymbol{w}, \boldsymbol{x} \rangle + b)$ in $\mathbb{R}^d$ can be rewritten as $h_{\boldsymbol{w}'}(\boldsymbol{x}') = \operatorname{sign}(\langle \boldsymbol{w}', \boldsymbol{x}' \rangle)$, where $\boldsymbol{w}' = (w_1, \ldots, w_d, b)$ and $\boldsymbol{x}' = (x_1, \ldots, x_d, 1)$. Namely, it can be expressed as a homogeneous halfspace in $\mathbb{R}^{d+1}$ applied over the transformation that appends the constant 1 to each input vector. This reduction is very useful and can sometimes (when the aforementioned transformation complies with distributional or other assumptions made about the instances) be applied to maintain simplicity.*

Theorem 3.0.1. *The VC dimension of the class of homogeneous halfspaces in $\mathbb{R}^d$ is d.*

Proof. First, we will show that $\mathrm{VC}\left(\mathcal{H}^d_{\mathrm{HLTF}}\right) \geq d$. Consider the set $C = \{\boldsymbol{e}_1, \ldots, \boldsymbol{e}_d\}$, where $\boldsymbol{e}_i \in \mathbb{R}^d$ is the vector, whose i-th element is 1 and the rest 0, for $i \in [d]$. For any labeling $y_1, \ldots, y_d$ of the elements of C, if we set $\boldsymbol{w} = (y_1, \ldots, y_d)$, then $\mathrm{sign}\left(\langle \boldsymbol{w}, \boldsymbol{e}_i \rangle\right) = y_i$ for all $i \in [d]$, namely the labeling can be derived from $\mathcal{H}^d_{\mathrm{HLTF}}$.

Next, we will show that $\mathrm{VC}\left(\mathcal{H}^d_{\mathrm{HLTF}}\right) < d+1$. Let $C' = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{d+1}\}$ be any set of $d+1$ vectors in $\mathbb{R}^d$. Then, there exist real numbers $\lambda_1, \ldots, \lambda_{d+1}$, not all of them zero, such that $\sum_{i=1}^{d+1} \lambda_i \boldsymbol{x}_i = \boldsymbol{0}$. Let $P = \{i \in [d] : \lambda_i > 0\}$ and $N = \{i \in [d] : \lambda_i < 0\}$. At least one of P and N is nonempty and it holds that $\sum_{i \in P} \lambda_i \boldsymbol{x}_i = \sum_{i \in N} |\lambda_i| \boldsymbol{x}_i$. Suppose that $\mathcal{H}^d_{\mathrm{HLTF}}$ shatters C'. Then, there must exist some $\boldsymbol{w}_1 \in \mathbb{R}^d$ such that $\langle \boldsymbol{w}_1, \boldsymbol{x}_i \rangle \geq 0$, if and only if $i \in P$, and some $\boldsymbol{w}_2 \in \mathbb{R}^d$ such that $\langle \boldsymbol{w}_2, \boldsymbol{x}_i \rangle \geq 0$, if and only if $i \in N$. If N is nonempty, we get that $0 \leq \sum_{i \in P} \lambda_i \langle \boldsymbol{w}, \boldsymbol{x}_i \rangle = \sum_{i \in N} |\lambda_i| \langle \boldsymbol{w}, \boldsymbol{x}_i \rangle < 0$, which leads to a contradiction. Similarly, if P is nonempty, we get that $0 \leq \sum_{i \in N} |\lambda_i| \langle \boldsymbol{w}, \boldsymbol{x}_i \rangle = \sum_{i \in P} \lambda_i \langle \boldsymbol{w}, \boldsymbol{x}_i \rangle < 0$, which leads again to a contradiction. Hence, $\mathcal{H}^d_{\mathrm{HLTF}}$ cannot shatter C', which concludes the proof. □

Theorem 3.0.2. *The VC dimension of the class of (nonhomogeneous) halfspaces in $\mathbb{R}^d$ is $d+1$.*

Proof. First, we will show that $\mathrm{VC}\left(\mathcal{H}^d_{\mathrm{LTF}}\right) \geq d+1$. Consider the set $C = \{\boldsymbol{e}_1, \ldots, \boldsymbol{e}_d, \boldsymbol{0}\}$. For any labeling $y_1, \ldots, y_{d+1}$ of the elements of C, if we set $\boldsymbol{w} = (y_1, \ldots, y_d)$ and $b = y_{d+1}/2$, then $\mathrm{sign}\left(\langle \boldsymbol{w}, \boldsymbol{e}_i \rangle + b\right) = y_i$ for all $i \in [d+1]$, namely the labeling can be derived from $\mathcal{H}^d_{\mathrm{LTF}}$.

Next, we will show that $\mathrm{VC}\left(\mathcal{H}^d_{\mathrm{LTF}}\right) < d+2$. Let $C' = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{d+2}\}$ be any set of $d+2$ vectors in $\mathbb{R}^d$. Suppose that $\mathcal{H}^d_{\mathrm{LTF}}$ shatters C'. Then, applying the aforementioned reduction from nonhomogeneous to homogeneous halfspaces, it follows that the set $C'' = \{(x_1, \ldots, x_d, 1) : (x_1, \ldots, x_d) \in C\}$ is shattered by $\mathcal{H}^{d+1}_{\mathrm{HLTF}}$, which leads to contradiction since $\mathrm{VC}\left(\mathcal{H}^{d+1}_{\mathrm{HLTF}}\right) = d+1$ and $|C''| = d+2$. Hence, $\mathcal{H}^d_{\mathrm{LTF}}$ cannot shatter C', which concludes the proof. □

3.1 Learning Halfspaces in the noiseless setting

In this section, we consider the learnability of halfspaces in the noiseless setting. This the realizable case, where the learner is given access to instances (almost surely) labeled by some target halfspace. We provide two well-known algorithms for learning halfspaces in the realizable case, showing that the hypothesis class of halfspaces is properly and efficiently realizably PAC learnable (with respect to ℓ_{0-1}).

3.1.1 Learning Halfspaces using Linear Programming

In Chapter 2, we have seen that any hypothesis class of finite VC dimension is realizably PAC learnable with any ERM learner with respect to the $0-1$ loss. We will show that an efficient ERM learner for $\mathcal{H}^d_{\mathrm{LTF}}$ can be implemented through linear programming. In particular, let $S = \left(\left(\boldsymbol{x}^{(1)}, y^{(1)}\right), \ldots, \left(\boldsymbol{x}^{(m)}, y^{(m)}\right)\right)$ be any training set of samples from $\mathbb{R}^d \times \{\pm 1\}$. Consider the following linear program, which we denote by $\mathsf{LP1}$[1]:

$$\text{Find} \qquad \boldsymbol{w} \in \mathbb{R}^d,\ b \in \mathbb{R}$$

$$\text{subject to} \quad y^{(t)}\left(\left\langle \boldsymbol{w}, \boldsymbol{x}^{(t)} \right\rangle + b\right) \geq 1 \quad \forall t \in [m]$$

In Theorem 3.1.1, we show that $\mathsf{LP1}$ combined with the Ellipsoid method (Algorithm 1) yields a proper and efficient realizable PAC learner for the class of LTFs.

Algorithm 1 Properly Learning Halfspaces with Linear Programming

Input: Training set $S = \left(\left(\boldsymbol{x}^{(1)}, y^{(1)}\right), \ldots, \left(\boldsymbol{x}^{(m)}, y^{(m)}\right)\right)$
Output: LTF $h_{\boldsymbol{w},b} \colon \mathbb{R}^d \to \{\pm 1\}$
1: Construct $\mathsf{LP1}$ from S
2: $(\boldsymbol{w}, b) \leftarrow \textsc{Ellipsoid}(\mathsf{LP1})$ ▷ See Appendix B
3: **return** $h_{\boldsymbol{w},b}$

Theorem 3.1.1. *$\mathcal{H}^d_{\mathrm{LTF}}$ is properly realizably PAC learnable with Algorithm 1 (with respect to the 0-1 loss) with sample complexity $O\left(\left(d\log(1/\epsilon) + \log(1/\delta)\right)/\epsilon\right)$ and polynomial runtime in d, in the number of samples and in the representation size of real numbers.*

[1] $\mathsf{LP1}$ can be typically formulated by arranging the unknown variables in a $(d+1)$-dimensional vector.

Proof. Fix any $\epsilon, \delta \in (0,1)$ and let $\mathcal{D}$ be any probability distribution on $\mathbb{R}^d \times \{\pm 1\}$ that is realizable by $\mathcal{H}^d_{\text{LTF}}$. Let $S = \left(\left(\boldsymbol{x}^{(1)}, y^{(1)}\right), \ldots, \left(\boldsymbol{x}^{(m)}, y^{(m)}\right)\right)$ be any training set of i.i.d. samples from $\mathcal{D}$. Since we are in the realizable case, the linear program LP1 constructed from S as above, is almost surely feasible (due to linear separability) and any solution of it, obviously, corresponds to an ERM learner, since it ensures zero empirical error on S.

Since $\text{VC}\left(\mathcal{H}^d_{\text{LTF}}\right) = d+1$, we have that $m \in O\left(\left(d\log(1/\epsilon) + \log(1/\delta)\right)/\epsilon\right)$ samples suffice (due to the realizable ERM sample complexity of $\mathcal{H}^d_{\text{LTF}}$ (see Theorem 2.6.2)) to get the (ϵ, δ)-realizable PAC learning guarantee with respect to $\mathcal{D}$. Moreover, LP1 can be efficiently solved, namely in time polynomial in m, d and in the representation size of real numbers using the Ellipsoid method. Finally, the properness comes trivially from the fact that LP1 returns a weight vector and a bias term, which define an LTF. These facts conclude the proof.

□

3.1.2 Learning Halfspaces using the Perceptron Algorithm

Another way to implement the ERM rule is the well-known Perceptron algorithm (Rosenblatt [1957, 1958]). The Perceptron is an iterative algorithm that starts with $\boldsymbol{w} = \boldsymbol{0}$, updates $\boldsymbol{w}$ in each iteration using an update rule based on the misclassified examples and terminates when all training examples are correctly classified.

Algorithm 2 Perceptron

Input: Training set $S = \left(\left(\boldsymbol{x}^{(1)}, y^{(1)}\right), \ldots, \left(\boldsymbol{x}^{(m)}, y^{(m)}\right)\right)$
Output: Homogeneous LTF $h_{\boldsymbol{w}} \colon \mathbb{R}^d \to \{\pm 1\}$

$t \leftarrow 0$
$\boldsymbol{w}^{(0)} \leftarrow \boldsymbol{0}$
while $\exists i \in [m] : y^{(i)} \left\langle \boldsymbol{w}^{(t)}, \boldsymbol{x}^{(i)} \right\rangle \leq 0$ **do**
    $t \leftarrow t + 1$
    $\boldsymbol{w}^{(t)} \leftarrow \boldsymbol{w}^{(t-1)} + y^{(i)} \boldsymbol{x}^{(i)}$
end while
return $h_{\boldsymbol{w}^{(t)}}$

We remark that the Perceptron algorithm in the formulation above is adjusted for homogeneous halfspaces, but it can perfectly be used to learn general halfspaces, if we use the aforementioned reduction from general to homogeneous halfspaces (by transformation of the instances). As for the intuition behind the Perceptron's update rule, notice that it can be derived from the update rule of subgradient descent on the loss $\ell(\boldsymbol{w}) = \sum_{i=1}^m \max\left\{0, -y^{(i)} \left\langle \boldsymbol{w}, \boldsymbol{x}^{(i)} \right\rangle\right\}$ (using only one example for the update). Moreover, if $\boldsymbol{x}^{(i)}$ is the instance used for the update at the t-th iteration, we have

$$y^{(i)} \left\langle \boldsymbol{w}^{(t)}, \boldsymbol{x}^{(i)} \right\rangle - y^{(i)} \left\langle \boldsymbol{w}^{(t-1)}, \boldsymbol{x}^{(i)} \right\rangle = \left\| \boldsymbol{x}^{(i)} \right\|_2^2 \geq 0\,,$$

which is an indication of Perceptron tending to adjust the current halfspace so that $\boldsymbol{x}_i$ is no longer misclassified. The following theorem guarantees that, under the realizable setting, the Perceptron algorithm is able to find a vector corresponding to a halfspace that classifies all samples correctly, being, therefore, a successful ERM learner.

Theorem 3.1.2 (Shalev-Shwartz and Ben-David [2022])**.** *Let $S = \left(\left(\boldsymbol{x}^{(1)}, y^{(1)}\right), \ldots, \left(\boldsymbol{x}^{(m)}, y^{(m)}\right)\right)$ be a linearly separable training set, let $R = \max_{i \in [m]} \left\| \boldsymbol{x}^{(i)} \right\|_2$ and let $B = \min\{\|\boldsymbol{w}\|_2 : \boldsymbol{w} \in \mathbb{R}^d \ \wedge \ \forall i \in [m],\ y^{(i)} \left\langle \boldsymbol{w}, \boldsymbol{x}^{(i)} \right\rangle \geq 1\}$. Then, the Perceptron algorithm, after at most $\left\lfloor (RB)^2 \right\rfloor$ iterations, returns a vector $\boldsymbol{w}$ such that $y^{(i)} \langle \boldsymbol{w}, \boldsymbol{x}^{(i)} \rangle > 0$ for all $i \in [m]$.*

The arguments concerning the sample complexity required needed to get the (ϵ, δ) realizable PAC learning guarantee are of course the same as in the linear programming case. However, Theorem 3.1.2 provides no efficiency guarantee in terms of runtime since $\left\lfloor (RB)^2 \right\rfloor$ might be exponentially large in d. This means that however practical and simple in implementation the Perceptron might be, one has to resort to the linear programming implementation of the ERM rule (Algorithm 1), if efficiency must be ensured.

3.2 Learning Halfspaces in the noisy setting

While learning halfspaces in the noiseless setting can be achieved using the aforementioned methods (Linear Programming and Perceptron), as mentioned in the Chapter 2, the zero noise assumption could be considered unrealistic. This motivates us to study the learnability of halfspaces in the presence of noise. In that case, the training set becomes not linearly separable, which makes the previous methods to fail, since they rely on the linear separability assumption. Hence, it becomes challenging to study the learnability of halfspaces in noisy settings both statistically and computationally.

In this section, we begin by making a brief reference on literature results concerning the learnability of halfspaces in the PAC model with RCN, Massart and Tsybakov noise, as defined in Chapter 2, and proceed by elaborating on the Massart case.

3.2.1 Prior Work on Halfspace Learning with Noise

The first noise model among the aforementioned (RCN, Massart and Tsybakov), which efficiency guarantees were obtained for, was the RCN model. In particular, it has been shown (Cohen [1997], Vempala et al. [1996]) that the class of halfspaces is properly learnable in the distribution-independent PAC model with RCN (Definition 2.7.3), with sample complexity $O\left(\text{poly}\left(d, 1/\epsilon, \log(1/\delta), 1/(1-2\eta), b\right)\right)$, where $\epsilon, \delta > 0$ are the PAC accuracy and confidence parameters respectively, $\eta \in [0, 1/2)$ is the noise rate of the RCN model and b is the bit complexity of the examples, and runtime polynomial in d and in the number of training samples.

The next noise model for which the first efficiency guarantees were obtained, was the Massart model. Recall that in the RCN model, we have some constant flipping probability η for each instance's label, whereas in the Massart noise model, the parameter η transforms into an upper bound on the flipping probability, which implies a less noisy distribution overall. In that sense, one would justifiably expect that acquiring guarantees for the learnability of halfspaces under Massart noise should be easier than the RCN case. Unfortunately, this is not case. In fact, the ignorance of the exact flipping probability for each instance's label in the Massart model raises technical difficulties, that make it harder to design efficient algorithms, without making compromises such restriction to homogeneous halfspaces, restriction to the distribution-dependent setting, acquirement of weaker PAC guarantees or improper hypotheses. There are many relevant recent works (Awasthi et al. [2015, 2016], Yan and Zhang [2017], Zhang et al. [2017, 2020], Zhang and Li [2021], Diakonikolas et al. [2019, 2020a, 2021a, 2022, 2023]) that provide halfspace learning algorithms with Massart noise, each involving a trade-off between the aforementioned criteria. Indicatively, the works of Zhang et al. [2020], Diakonikolas et al. [2020a] provide efficient homogeneous halfspace learners in the distribution dependent PAC model (under isotropic log-concave marginals) with Massart noise (Definition 2.7.6), with $O\left(\text{poly}\left(d, 1/\epsilon, \log(1/\delta), 1/(1-2\eta)\right)\right)$ sample complexity and runtime polynomial in d and in the number of training samples.

As for the Tsybakov model, the fact that it generalizes the Massart noise model makes it even harder to obtain efficiency guarantees, which is indicated by the smaller number and the recency of relative works (Diakonikolas et al. [2020b, 2021b], Zhang and Li [2021]). Indicatively, Diakonikolas et al. [2021b] showed that the class of homogeneous halfspaces is properly learnable in the distribution-dependent PAC model (specifically, assuming isotropic log-concave marginals) with Tsybakov noise (Definition 2.7.9), with sample complexity $O\left(\text{poly}(d)\left(\frac{B}{\epsilon}\right)^{O(1/\alpha^2)}\log(1/\delta)\right)$, where $\alpha \in [0, 1)$ and $B \geq 1$ are the Tsybakov noise parameters, and runtime polynomial in d and in the number of training samples.

3.2.2 Learning Halfspaces with Massart Noise

We now focus on the problem of learning halfspaces with Massart noise and, specifically, on the algorithm proposed in the relative work of Diakonikolas et al. [2020a]. In this work, an extremely simple optimization approach of some appropriate function related to the actual objective, namely the expected $0-1$ loss, is adopted. Since part of the experimental section in Chapter 6 is based on the algorithm of Diakonikolas et al. [2020a], we proceed by explaining the main aspects of the aforementioned work, including a few technical details.

In the work of Diakonikolas et al. [2020a], it was shown that the class of homogeneous halfspaces is properly and efficiently learnable in the distribution-dependent PAC model with Massart noise (see Definition 2.7.6). Specifically, their result holds for the family of bounded probability distributions, which subsumes the general family of isotropic log-concave probability distributions.

Fix any $\eta \in [0, 1/2)$, $\epsilon, \delta \in (0, 1)$, $U, R > 0$, $t \colon \mathbb{R}_{>0} \to \mathbb{R}_{\geq 0}$ and nonzero target vector $\boldsymbol{w}^* \in \mathbb{R}^d$. Let $\mathcal{D}$ be any $(\eta, h_{\boldsymbol{w}^*})$-Massart distribution such that $\mathcal{D}_{\boldsymbol{x}}$ is (U, R, t)-bounded. The approach of Diakonikolas et al. [2020a] is based on the following idea. In order to find some $\boldsymbol{w}$ that minimizes our actual objective $L_{\mathcal{D}_{\boldsymbol{x}}, f, \ell_{0-1}}(h_{\boldsymbol{w}})$, a natural approach (since f is unknown) would be attempting to minimize $L_{\mathcal{D}, \ell_{0-1}}(h_{\boldsymbol{w}})$, using samples from the noisy distribution $\mathcal{D}$. To achieve that, it is sufficient to minimize the function $\mathbf{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}\left[\mathbb{1}\{-y\langle \boldsymbol{w}, \boldsymbol{x}\rangle / \|\boldsymbol{w}\|_2 \geq 0\}\right]$. However, it is unclear how to efficiently optimize such a nonconvex objective. This obstacle is overcome in the aforementioned work by finding a nonconvex, but smooth surrogate with the property that any approximate stationary of that surrogate, corresponds to a halfspace close to the target one. Specifically, motivated by the fact that the logistic function $S_\sigma(t) \triangleq \frac{1}{1+e^{-t/\sigma}}$,

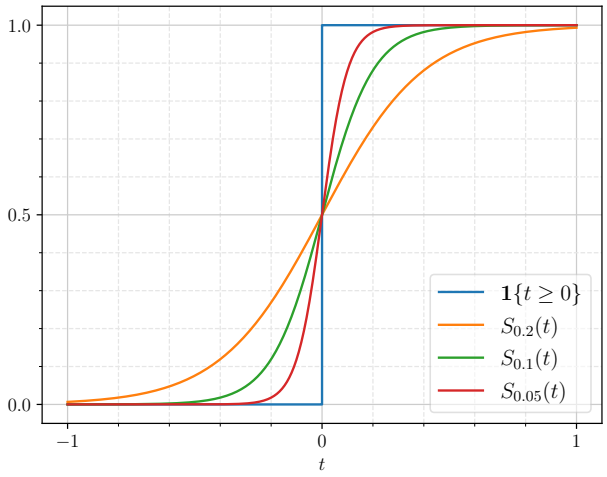


Figure 3.2: The step function and the logistic function

where $\sigma > 0$, seems to be a good approximation of the step function $\mathbb{1}\{t \geq 0\}$ when $\sigma \to 0$ (see Figure 3.2), they introduce the surrogate loss

$$\mathcal{L}_\sigma(\boldsymbol{w}) = \mathbf{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}\left[S_\sigma\left(-\frac{y\langle \boldsymbol{w}, \boldsymbol{x}\rangle}{\|\boldsymbol{w}\|_2}\right)\right].$$

For simplification of the notation, we denote $g_\sigma\left((\boldsymbol{x}, y), \boldsymbol{w}\right) = S_\sigma\left(-y\langle \boldsymbol{w}, \boldsymbol{x}\rangle / \|\boldsymbol{w}\|_2\right)$. It can be shown that $\nabla_{\boldsymbol{w}} \mathcal{L}_\sigma(\boldsymbol{w}) = \mathbf{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}\left[\nabla_{\boldsymbol{w}} g_\sigma\left((\boldsymbol{x}, y), \boldsymbol{w}\right)\right]$, where

$$\nabla_{\boldsymbol{w}} g_\sigma\left((\boldsymbol{x}, y), \boldsymbol{w}\right) = \frac{1}{\sigma} g_\sigma\left((\boldsymbol{x}, y), \boldsymbol{w}\right)\left(1 - g_\sigma\left((\boldsymbol{x}, y), \boldsymbol{w}\right)\right)\left(\frac{y\langle \boldsymbol{w}, \boldsymbol{x}\rangle}{\|\boldsymbol{w}\|_2^3}\boldsymbol{w} - \frac{y}{\|\boldsymbol{w}\|_2}\boldsymbol{x}\right).$$

A crucial property of this surrogate loss is that for any $\epsilon > 0$ and for any unit vector $\boldsymbol{w}$, whose angle with $\boldsymbol{w}^*$ and $-\boldsymbol{w}^*$ is greater than ϵ, there exists a sufficiently small choice of σ (dependent on ϵ), such that the norm of the gradient of $\mathcal{L}$ at $\boldsymbol{w}$ is sufficiently large. Contrapositively, for any $\epsilon > 0$, setting σ sufficiently small (dependent on ϵ), we can ensure that all points with sufficiently small gradient norm have angle at most ϵ with $\boldsymbol{w}^*$ or $-\boldsymbol{w}^*$. In other words, as long as this surrogate loss approximates the step function to a sufficient extent, the norm of its gradient can indicate whether a point is close in terms of angle to $\boldsymbol{w}^*$ or not. The following lemma formalizes the above claim.

Lemma 3.2.1 (Diakonikolas et al. [2020a]). *For any $\theta \in (0, \pi/2)$ and $\boldsymbol{w} \in S^{d-1}$ such that $\theta(\boldsymbol{w}, \boldsymbol{w}^*) \in (\theta, \pi - \theta)$, if $\sigma \leq \frac{R}{8U}\sqrt{1-2\eta}\sin(\theta)$, we have that $\|\nabla_{\boldsymbol{w}} \mathcal{L}_\sigma(\boldsymbol{w})\|_2 \geq \frac{R^2}{32U}(1-2\eta)$.*

Then, we can proceed by Projected Stochastic Gradient Descent (PSGD) for $\mathcal{L}_\sigma$, with projection on the unit sphere S^{d-1}, with a view to finding an approximate stationary point, namely a point $\boldsymbol{w}$ such that $\|\nabla_{\boldsymbol{w}} \mathcal{L}_\sigma(\boldsymbol{w})\|_2$ is sufficiently small. To guarantee that, we will use the following lemma, which concerns the convergence of PSGD for $\mathcal{L}_\sigma$.

Lemma 3.2.2 (Diakonikolas et al. [2020a]). *For any $\epsilon, \delta > 0$, running PSGD for the function $\mathcal{L}_\sigma$ for at least $T \in \Theta\left(\frac{d+\log(1/\delta)}{(\epsilon\sigma)^4}\right)$ iterations, using step size $\beta \in \Theta\left(\frac{\sigma^2}{\sqrt{dT}}\right)$, yields an output $\left(\boldsymbol{w}^{(1)}, \ldots, \boldsymbol{w}^{(T)}\right)$ of unit vectors such that, with probability at least $1 - \delta$, it holds $\min_{t\in[T]} \|\nabla_{\boldsymbol{w}} \mathcal{L}_\sigma\left(\boldsymbol{w}^{(t)}\right)\|_2 \leq \epsilon$.*

Algorithm 3 Projected Stochastic Gradient Descent for $\mathbf{E}_{\boldsymbol{z}\sim\mathcal{D}}[g(\boldsymbol{z},\boldsymbol{w})]$

Input: Function $g(\boldsymbol{z},\boldsymbol{w})$, training set $S=\left(\boldsymbol{z}^{(1)},\dots,\boldsymbol{z}^{(T)}\right)$ and step size $\beta>0$
Output: Tuple of T vectors in $\mathbb{S}^{d-1}$

procedure PSGD(g,S,β)
  $\boldsymbol{w}^{(0)}\leftarrow(1,0,\dots,0)$
  for $t=1,\dots,T$ **do**
    $\boldsymbol{v}^{(t)}\leftarrow\boldsymbol{w}^{(t-1)}-\beta\nabla_{\boldsymbol{w}}g\left(\boldsymbol{z}^{(t)},\boldsymbol{w}^{(t-1)}\right)$
    $\boldsymbol{w}^{(t)}\leftarrow\boldsymbol{v}^{(t)}/\left\|\boldsymbol{v}^{(t)}\right\|_2$
  end for
  return $\left(\boldsymbol{w}^{(1)},\dots,\boldsymbol{w}^{(T)}\right)$
end procedure

The PSGD algorithm returns a collection of weight vectors, which is guaranteed (assuming a sufficient number of steps and an appropriate step size) to contain a vector that is an approximate stationary point. To get the best vector $\boldsymbol{w}^{(t^*)}$ (or rather find some $\boldsymbol{w}^{(t^*)}$ such that the value of $\|\nabla_{\boldsymbol{w}}\mathcal{L}_\sigma\left(\boldsymbol{w}^{(t^*)}\right)\|_2$ is close to $\min_{t\in[T]}\|\nabla_{\boldsymbol{w}}\mathcal{L}_\sigma\left(\boldsymbol{w}^{(t)}\right)\|_2$ with high probability) within the aforementioned collection, we can evaluate all candidate hypotheses on a small number of samples drawn from $\mathcal{D}$ using the empirical version of $L_{\mathcal{D},\ell_{0-1}}$. It turns out that this is adequate to get the desired PAC result (see Theorem 3.2.1). The above steps are encapsulated in Algorithm 4.

Algorithm 4 Properly Learning Homogeneous Halfspaces with Massart Noise

Input: Training set $S=\left(\left(\boldsymbol{x}^{(1)},y^{(1)}\right),\dots,\left(\boldsymbol{x}^{(T)},y^{(T)}\right)\right)$, $\sigma>0$, $T'\in[T]$ and step size $\beta>0$
Output: Homogeneous LTF $h_{\hat{\boldsymbol{w}}}\colon\mathbb{R}^d\to\{\pm1\}$

$\left(\boldsymbol{w}^{(1)},\dots,\boldsymbol{w}^{(T)}\right)\leftarrow\mathrm{PSGD}(g_\sigma,S,\beta)$ ▷ See Projected Stochastic Gradient Descent
$L\leftarrow\left(\boldsymbol{w}^{(1)},\dots,\boldsymbol{w}^{(T)},-\boldsymbol{w}^{(1)},\dots,-\boldsymbol{w}^{(T)}\right)$ ▷ Set of candidate vectors
$S'\leftarrow\left(\left(\boldsymbol{x}^{(1)},y^{(1)}\right),\dots,\left(\boldsymbol{x}^{(T')},y^{(T')}\right)\right)$
$\hat{\boldsymbol{w}}\leftarrow\mathrm{argmin}_{\boldsymbol{w}\in L}\hat{L}_{S',\ell_{0-1}}(h_{\boldsymbol{w}})$
return $h_{\hat{\boldsymbol{w}}}$

Theorem 3.2.1 (Diakonikolas et al. [2020a])**.** *Fix any $\eta\in[0,1/2)$, $\epsilon,\delta\in(0,1)$, $U,R>0$, $t\colon\mathbb{R}_{>0}\to\mathbb{R}_{\geq0}$ and nonzero target vector $\boldsymbol{w}^*\in\mathbb{R}^d$. Let $\mathcal{D}$ be any $(\eta,h_{\boldsymbol{w}^*})$-Massart distribution such that $\mathcal{D}_{\boldsymbol{x}}$ is (U,R,t)-bounded. Algorithm 4 has the following performance guarantee: If given as input $\sigma\in\Theta\left(\frac{R\epsilon(1-2\eta)^{3/2}}{U^2t(\epsilon(1-2\eta)/4)^2}\right)$, $T\in\Theta\left(\frac{U^{12}t(\epsilon(1-2\eta)/4)^8}{R^{12}\epsilon^4(1-2\eta)^{10}}\left(d+\log(1/\delta)\right)\right)$ i.i.d. samples drawn from $\mathcal{D}$, $T'\in\Theta\left(\frac{\log(T/\delta)}{\epsilon^2(1-2\eta)^2}\right)$ and step size $\beta\in\Theta\left(\frac{\sigma^2}{\sqrt{dT}}\right)$, it runs in $\Theta(TG+dTT')$ time, where G is an upper bound on the time of each gradient evaluation, and outputs a vector $\hat{\boldsymbol{w}}\in S^{d-1}$ such that, with probability at least $1-\delta$, it holds $L_{\mathcal{D}_{\boldsymbol{x}},h_{\boldsymbol{w}^*},\ell_{0-1}}(h_{\hat{\boldsymbol{w}}})\leq\epsilon$.*

Proof. From Lemma 3.2.3, we have that with $T\in\Theta\left(\left(d+\log(1/\delta)\right)U^4/\left(R^8(1-2\eta)^4\sigma^4\right)\right)$ iterations and step size $\beta\in\Theta\left(\frac{\sigma^2}{\sqrt{dT}}\right)$, PSGD yields an output $\left(\boldsymbol{w}^{(1)},\dots,\boldsymbol{w}^{(T)}\right)$ of unit vectors such that, with probability at least $1-\delta$, it holds $\min_{t\in[T]}\left\|\nabla_{\boldsymbol{w}}\mathcal{L}_\sigma\left(\boldsymbol{w}^{(t)}\right)\right\|_2<\frac{1}{32U}R^2(1-2\eta)$ for any $\delta\in(0,1)$. Assuming that the last holds, the contrapositive of Lemma 3.2.2 implies that, for any (sufficiently small) angle θ, if $\sigma\in\Theta(\theta RU^{-1}\sqrt{1-2\eta})$, then we have $\min_{\boldsymbol{w}\in L}\theta(\boldsymbol{w},\boldsymbol{w}^*)\leq\theta$, where $L=\left(\boldsymbol{w}^{(1)},\dots,\boldsymbol{w}^{(T)},-\boldsymbol{w}^{(1)},\dots,-\boldsymbol{w}^{(T)}\right)$. Hence, from Lemma A.0.5, we get that $\min_{\boldsymbol{w}\in L}L_{\mathcal{D}_{\boldsymbol{x}},f,\ell_{0-1}}(h_{\boldsymbol{w}})\leq Ut(r)^2\theta+r$ for any $r>0$.

If S' is a tuple of T' independent samples drawn from $\mathcal{D}$, then, from Hoeffding's inequality, we have

$$\mathop{\mathbf{Pr}}_{S'\sim\mathcal{D}^{T'}}\left[\left|\hat{L}_{S',\ell_{0-1}}(h_{\boldsymbol{w}})-L_{\mathcal{D},\ell_{0-1}}(h_{\boldsymbol{w}})\right|\geq t\right]\leq2e^{-2T't^2}$$

for all $\boldsymbol{w}\in L$ and $t>0$. By application of the union bound we get that, for all $t>0$, with probability at least $1-2|L|e^{-2T't^2}$, it holds $L_{\mathcal{D},\ell_{0-1}}(h_{\hat{\boldsymbol{w}}})\leq2t+\min_{\boldsymbol{w}\in L}L_{\mathcal{D},\ell_{0-1}}(h_{\boldsymbol{w}})$, where $\hat{\boldsymbol{w}}\in\mathrm{argmin}_{\boldsymbol{w}\in L}\hat{L}_{S',\ell_{0-1}}(h_{\boldsymbol{w}})$. Therefore, from Lemma 2.7.2 and Proposition 2.1.1, we have that

$$L_{\mathcal{D}_{\boldsymbol{x}},f,\ell_{0-1}}(h_{\hat{\boldsymbol{w}}})\leq\frac{2t+\min_{\boldsymbol{w}\in L}L_{\mathcal{D}_{\boldsymbol{x}},f,\ell_{0-1}}(h_{\boldsymbol{w}})}{1-2\eta}\leq\frac{Ut(r)^2\theta+r+2t}{1-2\eta}\,.$$

Consequently, for any (sufficiently small) $\epsilon > 0$ and $\delta \in (0,1)$, if we set $r = t = \epsilon(1-2\eta)$, $\theta = \epsilon(1-2\eta)/\left(Ut(\epsilon(1-2\eta))^2\right)$ and $T' = \left\lceil \ln\left(4T/\delta\right)/\left(2\epsilon^2(1-2\eta)^2\right)\right\rceil$, we have that, with probability at least $1-2\delta$, it holds $L_{\mathcal{D}_{\boldsymbol{x}},h_{\boldsymbol{w}^*},\ell_{0-1}}(h_{\hat{\boldsymbol{w}}}) \leq 4\epsilon$. The above choices, impose that $\sigma \in \Theta\left(\frac{R\epsilon(1-2\eta)^{3/2}}{U^2 t(\epsilon(1-2\eta))^2}\right)$ and $T \in \Theta\left(\frac{U^{12}t(\epsilon(1-2\eta))^8}{R^{12}\epsilon^4(1-2\eta)^{10}}\left(d+\log(1/\delta)\right)\right)$. Finally, the runtime of the algorithm is due to the T PSGD iterations and the $\Theta(dTT')$ time needed to find the best among the candidate vector set L. □

Chapter 4

Label Ranking

In this chapter, we address the main topic of this thesis, namely the Label Ranking problem. We begin by providing a formal definition for Label Ranking. Afterwards, we present several noise models for LR, some of them stemming from well-known probability distributions on rankings and others constituting extensions of the binary classification Massart and Tsybakov noise models. Finally, we expand on the two basic pairwise and labelwise decomposition techniques for Label Ranking, which were briefly discussed in Chapter 1, and elaborate on some of the learnability aspects associated with them, both in the noiseless and the noisy PAC model.

4.1 About Rankings

Before formally defining the Label Ranking problem, we provide some important definitions and notation about rankings.

Definition 4.1.1 (Binary relation)**.** *A binary relation R from a set X to a set Y is a subset of $X \times Y$.*

For any $(x, y) \in X \times Y$, the statement $(x, y) \in R$ is denoted by xRy. The relation $R^\intercal$ from Y to X defined as $R^\intercal = \{(y, x) \in Y \times X : (x, y) \in R\}$ is the *converse* or *transpose* relation of R. Moreover, if $X = Y$, the binary relation R is said to be a *homogeneous* relation on X.

Definition 4.1.2 (Strict partial order)**.** *A strict partial order on a set S is a homogeneous relation $<$ on S such that the following are satisfied for all $a, b, c \in S$:*

1. *Not $a < a$ (irreflexivity).*
2. *If $a < b$, then not $b < a$ (asymmetry).*
3. *If $a < b$ and $b < c$, then $a < c$ (transitivity).*

Definition 4.1.3 (Strict total order)**.** *A strict total order on a set S is a strict partial order on S such that for all $a, b \in S$, if $a \neq b$, then $a < b$ or $b < a$.*

Definition 4.1.4 (Permutation)**.** *A permutation of a set S is a bijection from S to itself.*

In general, a *ranking* over a finite set S refers to a strict partial order $\succ$ on S[1]. If a ranking $\succ$ is a strict total order, then it will be referred to as *complete*, otherwise *incomplete*. For any $a, b \in S$, the interpretation of the condition $a \succ b$ (resp. $a \prec b$) is that a is ranked higher (resp. lower) than b in the ranking, with $\prec$ denoting the converse relation of $\succ$.

There exists a one-to-one correspondence between the set of complete rankings over S and the set of permutations of S, which motivates us to model complete rankings through permutations. In particular, any complete ranking over $[k]$, where $k \geq 1$, can be conveniently modeled as a permutation $\pi \in \mathbb{S}_k$[2], such that $\pi(i)$ is the position of element i in the ranking for all $i \in [k]$ and, consequently, $\pi^{-1}(i)$ is the element of $[k]$ in the i-th position of the ranking for all $i \in [k]$. For example, if $k = 4$, the complete ranking $4 \succ 1 \succ 3 \succ 2$ corresponds to the permutation π, with $\pi(1) = 2$, $\pi(2) = 4$, $\pi(3) = 3$ and $\pi(4) = 1$, which is also denoted by

$$\pi = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 4 & 3 & 1 \end{pmatrix}.$$

[1] Another case of interest is that of *partial rankings*, where ties among their elements are allowed, but will not be studied in this thesis.

[2] $\mathbb{S}_k$ denotes the set of permutations of $[k]$.

4.2 The Label Ranking Problem

Problem statement Let S be a finite set of $k \geq 2$ labels[3], let $\mathcal{X}$ be an instance space (usually $\mathcal{X} \subseteq \mathbb{R}^d$, where $d \geq 1$), let $\mathcal{Y}$ be the set of strict partial orders on S and let $\mathcal{Y}'$ be the set of strict total orders on S. The Label Ranking problem is a supervised prediction problem that concerns using a training set of examples in $\mathcal{X} \times \mathcal{Y}$ to find a hypothesis from $\mathcal{X}$ to $\mathcal{Y}'$. The goal of a label ranking algorithm is to ensure that its output hypothesis minimizes some notion of error, related to some loss function $\ell \colon \mathcal{Y}^2 \to \mathbb{R}_{\geq 0}$.

Remark. *We emphasize that, unlike the classic setting of prediction problems, a hypothesis acquired by a label ranking algorithm is required to output predictions that belong to a proper subset $\mathcal{Y}'$ (strict total orders) of the original output space $\mathcal{Y}$ (strict partial orders).*

For the rest of this thesis, we assume that the set of labels is $S = [k]$ without loss of generality. Moreover, in most parts of this chapter, we will focus on the case, where a label ranking algorithm is given as input only examples that are promised to contain only complete rankings (in which case, as mentioned before, rankings may be modeled as elements of $\mathbb{S}_k$).

The more general case, where the learner's input might also consist of incomplete rankings, the underlying examples' distribution can be generally modeled through a randomized mechanism that acts on examples drawn from a distribution over $\mathcal{X} \times \mathbb{S}_k$ (with complete rankings) and transforms them to examples consisting of possibly incomplete rankings (see Fotakis et al. [2022a], Vogel and Clémençon [2020]).

Notation Let $(i, j) \in [k]^2$ with $i \neq j$ and let any probability distribution $\mathcal{D}$ over $\mathcal{X} \times \mathbb{S}_k$. We denote by $\mathcal{D}_x$ the marginal of $\mathcal{D}$ on $\mathcal{X}$ and by $\mathcal{D}_{\pi|x}$ the conditional distribution of $\mathcal{D}$ on $\mathbb{S}_k$ given $x \in \mathcal{X}$. For every $\pi \in S_k$, we define $\pi_{ij} = \operatorname{sign}(\pi(j) - \pi(i))$. Moreover, for every label ranking function $\sigma \colon \mathcal{X} \to \mathbb{S}_k$, we define the function $\sigma_{ij} \colon \mathcal{X} \to \{\pm 1\}$, which satisfies that $\sigma_{ij}(x) = \operatorname{sign}(\sigma(x)(j) - \sigma(x)(i))$ for all $x \in \mathcal{X}$. We let $\mathcal{D}_{ij}$ denote the joint distribution on (x, π_{ij}) and $\mathcal{D}_i$ denote the joint distribution on $(x, \pi(i))$, where $(x, \pi) \sim \mathcal{D}$. For all $\mathcal{H} \subseteq \mathbb{S}_k^{\mathcal{X}}$, we define $\mathcal{H}_{ij} = \{\sigma_{ij} : \sigma \in \mathcal{H}\}$.

Relation to multiclass and multilabel classification It is interesting to examine the association of Label Ranking with some other fundamental supervised learning problems, such as *multiclass classification* and *multilabel classification*.

Firstly, one can easily see that when dealing only with complete rankings, Label Ranking can be thought of as a standard multiclass classification problem with $k! = |\mathbb{S}_k|$ classes. Conversely, as observed by Hüllermeier et al. [2008], a multiclass classification problem may by formulated as a Label Ranking problem as well. In particular, consider a multiclass classification problem with input and output spaces $\mathcal{X}$ and $\mathcal{Y}$ respectively. Any training example $(x, i) \in \mathcal{X} \times \mathcal{Y}$ of this multiclass classification implicitly defines the label ranking example $(x, \succ)$, where $\succ = \{(i, j) : j \in \mathcal{Y} \setminus \{i\}\}$. Using the aforementioned reduction to solve a multiclass classification problem by means of a label ranking algorithm would require that each output ranking be projected to a class of $\mathcal{Y}$. A natural and reasonable way to do that is to output the top element of the ranking.

Similarly, consider a multilabel classification problem with input space $\mathcal{X}$ and set of labels $\mathcal{L}$. Here, the output space is the powerset of $\mathcal{L}$, that is, each instance is mapped to any subset of labels in $\mathcal{L}$ that are considered relevant. Any training example $(x, L) \in \mathcal{X} \times 2^{\mathcal{L}}$ of this multilabel classification problem implicitly defines the label ranking example $(x, \succ)$, where $\succ = \{(i, j) : i \in L \wedge j \in \mathcal{L} \setminus \{L\}\}$. Now, though, it is not so obvious how to accomplish the projection from a ranking to a subset of $\mathcal{L}$ or rather how many of a ranking's top elements to keep. A solution to this was proposed by Fürnkranz et al. [2008], where a slight modification of the aforementioned reduction was considered.

Representing preferences through utility functions As mentioned in several works (Har-Peled et al. [2002], Dekel et al. [2003], Hüllermeier et al. [2008], Fotakis et al. [2022a,b]), a natural way to represent preferences among k labels is to use a *utility* or *score function* $\boldsymbol{m} \colon \mathcal{X} \to \mathbb{R}^k$ that evaluates alternatives by assigning score values to them. The greater the score value for some alternative is, the higher is the preference for that alternative and so does its rank in the underlying ranking. Specifically, score vectors are mapped to rankings through the function $\mathfrak{S} \colon \mathbb{R}^k \to \mathbb{S}_k$ that works as follows. It takes as input a score vector $\boldsymbol{v} = (v_1, \ldots, v_k) \in \mathbb{R}^k$, whose i-th element contains a score value for the i-th label,

[3]The term *label* here is different from what was defined as a label (or class) in the general setting of prediction problems and should not be confused.

and outputs the unique permutation $\pi = \mathfrak{S}(\boldsymbol{v}) \in \mathbb{S}_k$ with the property that, for all $1 \leq i < j \leq k$, it holds $\pi(i) < \pi(j)$ if and only if $v_i \geq v_j$. Namely, $\mathfrak{S}$ sorts the elements of $\boldsymbol{v}$ in decreasing score order to obtain an ordering $(i_1, \ldots, i_k)$ of the alternatives such that $v_{i_1} \geq \cdots \geq v_{i_k}$, and outputs the permutation associated with the ranking $i_1 \succ \cdots \succ i_k$.

It is straightforward to see that for every ranking function $\sigma \colon \mathcal{X} \to \mathbb{S}_k$, there exists some score function (infinitely many actually) $\boldsymbol{m} \colon \mathcal{X} \to \mathbb{R}^k$ such that $\sigma = \mathfrak{S} \circ \boldsymbol{m}$. We remark that if the training data offer the score values directly, then the Label Ranking problem is reduced to a standard regression problem. However, this is a too strong condition that can rarely be assumed. One of the simplest types of utility functions for $\mathcal{X} \subseteq \mathbb{R}^d$, is a linear utility function, namely one assigning a score to each label, which is a linear combinations of the features. This is the linear Label Ranking setting that will be studied in Chapter 5.

Loss functions in Label Ranking We now refer to some of the most commonly used loss functions for Label Ranking, that are defined for complete rankings. One of the most post popular loss functions for rankings is the Kendall's τ (KT) distance, which is defined for any $\pi, \sigma \in \mathbb{S}_k$ as

$$d_\tau(\pi, \sigma) \triangleq \sum_{1 \leq i < j \leq k} \mathbb{1}\left\{(\pi(i) - \pi(j))\,(\sigma(i) - \sigma(j)) < 0\right\} .$$

Namely, the KT distance measures the number of discordant pairs of labels in the rankings π and σ, i.e. the number of pairs of labels, whose (pairwise) ordering is not the same in the two rankings. The KT distance is directly related to the Kendall's rank correlation coefficient or Kendall's τ (KT) coefficient defined as

$$\tau(\pi, \sigma) \triangleq 1 - \frac{4 d_\tau(\pi, \sigma)}{k(k-1)} ,$$

which quantifies the similarity between two rankings through the number of concordant and discordant pairs. It is straightforward to see that $d_\tau(\pi, \sigma) = 0$ (minimum d_τ) and $\tau(\pi, \sigma) = 1$ (maximum τ), if and only if $\pi = \sigma$. Moreover, $d_\tau(\pi, \sigma) = \binom{k}{2}$ (maximum d_τ) and $\tau(\pi, \sigma) = -1$ (minimum τ), if and only if σ is the reverse ranking of π, that is, $\sigma(i) + \pi(i) = k + 1$ for all $i \in [k]$. Two other equally used measures are the Spearman's footrule

$$d_1(\pi, \sigma) \triangleq \sum_{i=1}^{k} |\pi(i) - \sigma(i)|$$

and the Spearman's distance

$$d_2(\pi, \sigma) \triangleq \sum_{i=1}^{k} (\pi(i) - \sigma(i))^2$$

that are reminiscent of the standard l_1 and l_2 distances respectively, thinking of π and σ as vectors. The Spearman's distance is associated with another important similarity measure between rankings, the Spearman's rank correlation coefficient, defined as

$$\rho(\pi, \sigma) \triangleq 1 - \frac{6 d_2(\pi, \sigma)}{k(k^2 - 1)} .$$

Like in the case of the KT distance, we have that $d_2(\pi, \sigma) = 0$ (minimum d_2) and $\rho(\pi, \sigma) = 1$ (maximum ρ), if and only if $\pi = \sigma$. Moreover, $d_2(\pi, \sigma) = \frac{k(k^2-1)}{3}$ (maximum d_2) and $\rho(\pi, \sigma) = -1$ (minimum ρ), if and only if σ is the reverse ranking of π. Another useful distance measure is the Hamming distance

$$d_{\mathrm{H}}(\pi, \sigma) \triangleq \sum_{i=1}^{k} \mathbb{1}\left\{\pi(i) \neq \sigma(i)\right\}$$

that measures the number of elements in which π and σ disagree. Finally, another important measure is the top-r disagreement, where $r \in [k]$, defined as

$$d_{\mathrm{top}-r}(\pi, \sigma) \triangleq 1 - \prod_{i=1}^{r} \mathbb{1}\left\{\pi^{-1}(i) = \sigma^{-1}(i)\right\} .$$

Namely, the top-r disagreement is zero if and only if the elements of the two rankings are identical (the same elements and in the same order) in the top r positions. Notice that in the extreme case where $r = k$, we get the top-k disagreement that is equivalent to the $0 - 1$ loss.

The aforementioned loss functions are prevalent in the evaluation of label ranking algorithms due to them being simple and intuitively interpretable. Nonetheless, as observed by Zhou et al. [2014a], when it comes to real world applications, one might need to resort to more sophisticated and ad hoc loss functions. For instance, there might be cases where emphasis should be laid on predicting the position of high-importance labels. Similarly, there might be positions (e.g. the top part of ranking) of higher importance, where errors should be costlier. For such generalizations of the standard ranking distance measures, we refer to Kumar and Vassilvitskii [2010].

4.3 Noise Models in Label Ranking

4.3.1 Ranking Probability Distributions

We begin by defining two popular probability models on rankings, the Mallows model and the Bradley-Terry-Mallows model, introduced by Mallows [1957].

Definition 4.3.1 (Mallows distribution (Mallows [1957]))**.** *Fix some $\phi \in (0,1]$, $\pi_0 \in \mathbb{S}_k$ and a ranking distance $d\colon \mathbb{S}_k^2 \to \mathbb{R}_{\geq 0}$. The Mallows distribution $\mathcal{M}_{\text{Mal}}(d,\phi,\pi_0)$ with central ranking π_0 and spread parameter ϕ is a probability measure over $\mathbb{S}_k$ with probability mass function:*

$$\underset{\pi\sim\mathcal{M}_{\text{Mal}}(d,\phi,\pi_0)}{\mathbf{Pr}}[\pi] = \frac{\phi^{d(\pi,\pi_0)}}{\sum_{\sigma\in\mathbb{S}_k}\phi^{d(\sigma,\pi_0)}}$$

If the ranking distance is the KT distance, which is the only case that will be considered in this thesis, it can be shown that

$$\underset{\pi\sim\mathcal{M}_{\text{Mal}}(d_\tau,\phi,\pi_0)}{\mathbf{Pr}}[\pi] = \frac{\phi^{d_\tau(\pi,\pi_0)}}{\prod_{i=1}^{k-1}\sum_{j=0}^{i}\phi^j},$$

namely the partition function is independent of the central ranking, and that π_0 is the mode of the distribution. Moreover, as $\phi \to 1$ the Mallows distribution tends to become uniform, while as $\phi \to 0$ the mass of the distribution tends to concentrate around π_0.

Definition 4.3.2 (Bradley-Terry-Mallows distribution (Mallows [1957]))**.** *Fix some $\boldsymbol{w} \in \mathbb{R}_{>0}^k$ and a ranking distance $d\colon \mathbb{S}_k^2 \to \mathbb{R}_{\geq 0}$. The Bradley-Terry-Mallows distribution $\mathcal{M}_{\text{BTM}}(\boldsymbol{w})$ is a probability measure over $\mathbb{S}_k$ with probability mass function:*

$$\underset{\pi\sim\mathcal{M}_{\text{BTM}}(\boldsymbol{w})}{\mathbf{Pr}}[\pi] = \frac{\prod_{\pi(i)<\pi(j)}\frac{w_i}{w_i+w_j}}{\sum_{\sigma\in\mathbb{S}_k}\prod_{\sigma(i)<\sigma(j)}\frac{w_i}{w_i+w_j}}$$

Intuitively, for any $i \neq j$, the larger their score difference is in aid of i, the higher becomes the probability of observing rankings where i is ranked higher than j. Assuming that $\boldsymbol{w}$ has k distinct values, it can be shown that the mode of the distribution is $\mathfrak{S}(\boldsymbol{w})$ (we remind that this is the ranking induced by sorting the elements of $\boldsymbol{w}$ in decreasing order).

We finally present another probability model on rankings, proposed by Fotakis et al. [2022b], which subsumes the aforementioned models under some very mild assumptions (see Proposition 2 and Proposition 3).

Definition 4.3.3 (Noisy Ranking Distribution (Fotakis et al. [2022b]))**.** *Let $\mathcal{M}$ be a probability measure over $\mathbb{S}_k$ with the following property: There exists some (unique) $\pi^* \in \mathbb{S}_k$ and $\eta \in [0,1/2)$ such that for any $(i,j) \in [k]^2$ with $\pi^*(i) < \pi^*(j)$, it holds that $\mathbf{Pr}_{\pi\sim\mathcal{M}}[\pi(i) > \pi(j)] \leq \eta$. Then, $\mathcal{M}$ is said to be an η-noisy ranking distribution with ground truth ranking π^*.*

Proposition 4.3.1 (Fotakis et al. [2022b])**.** *For all $\phi \in (0,1)$ and $\pi_0 \in \mathbb{S}_k$, $\mathcal{M}_{\text{Mal}}(d_\tau,\phi,\pi_0)$ is an $\frac{1+\phi}{4}$-noisy ranking distribution with ground truth ranking π_0.*

Proposition 4.3.2 (Fotakis et al. [2022b])**.** *For all $\boldsymbol{w} \in \mathbb{R}_{>0}^k$ with k distinct values, there exists some $\eta \in [0,1/2)$ such that $\mathcal{M}_{\text{BTM}}(\boldsymbol{w})$ is an η-noisy ranking distribution with ground truth ranking $\mathfrak{S}(\boldsymbol{w})$.*

4.3.2 Label Ranking Probability Distributions

We now define some Label Ranking distributions, namely distributions over $\mathcal{X} \times \mathbb{S}_k$, that quantify the presence of noise. A natural approach, also adopted by Fotakis et al. [2022a,b], would be to model the conditional distribution of rankings given an instance $x \in \mathcal{X}$ as one of the aforementioned ranking distributions, choosing the central ranking to be the ground truth ranking for x (intuitively, we set the most often appearing ranking to be the ground truth one). According to that principle, we define the following Label Ranking distributions.

Definition 4.3.4 (Label Ranking Distribution with Mallows Noise)**.** *Let $\phi \in (0,1]$, let $d\colon \mathbb{S}_k^2 \to \mathbb{R}_{\geq 0}$ and let $\sigma\colon \mathcal{X} \to \mathbb{S}_k$ be a target label ranking function. A probability distribution $\mathcal{D}$ on $\mathcal{X} \times \mathbb{S}_k$ is said to be a (ϕ, d, σ)-LR distribution with Mallows noise, if $\mathcal{D}_{\pi|x} = \mathcal{M}_{\mathrm{Mal}}(d, \phi, \sigma(x))$ for all $x \in \mathcal{X}$.*

Definition 4.3.5 (Label Ranking Distribution with Bradley-Terry-Mallows Noise)**.** *Let $\boldsymbol{m}\colon \mathcal{X} \to \mathbb{R}_{>0}^k$ be a target score function. A probability distribution $\mathcal{D}$ on $\mathcal{X} \times \mathbb{S}_k$ is said to be an $\boldsymbol{m}$-LR distribution with Bradley-Terry-Mallows noise, if $\mathcal{D}_{\pi|x} = \mathcal{M}_{\mathrm{BTM}}(\boldsymbol{m}(x))$ for all $x \in \mathcal{X}$.*

Another family of label ranking distributions, introduced by Fotakis et al. [2022b], which extends the Massart condition to the LR setting, is as follows.

Definition 4.3.6 (Label Ranking Distribution with Massart Noise)**.** *A probability distribution $\mathcal{D}$ on $\mathcal{X} \times \mathbb{S}_k$ is said to be a label ranking distribution with Massart noise, if there exists some (unique) function $\sigma\colon \mathcal{X} \to \mathbb{S}_k$ and some $\eta \in [0, 1/2)$ such that $\mathcal{D}_{ij}$ is an (η, σ_{ij})-Massart distribution for all $(i,j) \in [k]^2$ with $i \neq j$. Then, we also say that $\mathcal{D}$ is an (η, σ)-LR distribution with Massart noise.*

Lemma 4.3.1. *$\mathcal{D}$ is an (η, σ)-LR distribution with Massart noise, if and only if $\mathcal{D}_{\pi|x}$ is an η-noisy ranking distribution with ground truth ranking $\sigma(x)$ for all $x \in \mathcal{X}$.*

Proof. If $\mathcal{D}$ is an η-noisy ranking distribution with ground truth ranking $\sigma(x)$ for all $x \in \mathcal{X}$, then for all $(i,j) \in [k]^2$ with $i \neq j$ and for all $x \in \mathcal{X}$ it holds,

$$\begin{aligned}
\Pr_{\pi \sim \mathcal{D}_{\pi|x}} \left[\pi(i) > \pi(j) \wedge \sigma(x)(i) < \sigma(x)(j) \vee \pi(i) < \pi(j) \wedge \sigma(x)(i) > \sigma(x)(j) \;\middle|\; x\right] \leq \eta \iff \\
\Pr_{\pi \sim \mathcal{D}_{\pi|x}} \left[\operatorname{sign}(\pi(j) - \pi(i)) \neq \operatorname{sign}\left(\sigma(x)(j) - \sigma(x)(i)\right) \mid x\right] \leq \eta \iff \\
\Pr_{y \sim (\mathcal{D}_{ij})_{y|x}} \left[y \neq \sigma_{ij}(x) \mid x\right] \leq \eta\,,
\end{aligned}$$

namely $\mathcal{D}_{ij}$ is an (η, σ_{ij})-Massart distribution. □

Lemma 4.3.2. *Let $\phi \in (0,1)$ and let $\sigma\colon \mathcal{X} \to \mathbb{S}_k$ be a target label ranking function. Any (ϕ, d_τ, σ)-LR distribution with Mallows noise is a $\left(\frac{1+\phi}{4}, \sigma\right)$-LR distribution with Massart noise.*

Proof. It is a direct implication of Lemma 4.3.1 and Proposition 4.3.1. □

The above lemma indicates that the family of LR distributions with Massart noise captures any LR distribution $\mathcal{D}$ with Mallows noise under the extremely mild assumption that $\mathcal{D}_{\pi|x}$ is not uniform for any $x \in \mathcal{X}$. One can show as well that a large subset of the family of LR distributions with Bradley-Terry-Mallows noise falls under the umbrella of LR distributions with Massart noise. In particular, some assumptions concerning the underlying target score function (see Definition 4.3.5 and Proposition 4.3.2) need to be made.

The motivation for defining the family of LR distributions with Massart noise is as follows. As observed by Fotakis et al. [2022b], if we assume that the data distribution is an LR distribution with Massart noise, then, using the pairwise decomposition method for LR (which we will be defined afterwards formally), the individual subproblems are binary classification problems in the presence of Massart noise. As we will shortly see, one can leverage this property and obtain theoretical guarantees for LR in the presence of noise. For this reason and to facilitate the forthcoming analysis, we adjust the PAC model to the aforementioned family of distributions.

In what follows, we let $\mathcal{X}$ be a domain set, $\mathcal{H} \subseteq \mathbb{S}_k^{\mathcal{X}}$ be a hypothesis class, $\mathcal{F}$ be a family of probability distributions on $\mathcal{X}$, $\ell\colon \mathbb{S}_k^2 \to \mathbb{R}_{\geq 0}$ be a loss function and $A \in \mathcal{A}_{\mathcal{X}, \mathbb{S}_k}$ be a learning algorithm.

Definition 4.3.7 (Distribution-dependent LR-Massart Sample Complexity of a Learning Algorithm)**.** *The distribution-dependent LR-Massart sample complexity of A with respect to $\mathcal{H}$, $\mathcal{F}$ and ℓ is the function $m_{A,\mathcal{H},\mathcal{F},\ell}^{\mathrm{LR-Massart}}\colon (0,\infty) \times (0,\infty) \times [0,1/2) \to \mathbb{N}$ defined as follows: For every $\epsilon, \delta > 0$ and $\eta \in [0, 1/2)$, $m_{A,\mathcal{H},\mathcal{F},\ell}^{\mathrm{LR-Massart}}(\epsilon, \delta, \eta)$ is the minimal integer such that for every*

- *integer* $m \geq m_{A,\mathcal{H},\mathcal{F},\ell}^{\text{LR}-\text{Massart}}(\epsilon, \delta, \eta)$,
- *target function* $\sigma \in \mathcal{H}$ *and*
- (η, σ)*-LR distribution with Massart noise* $\mathcal{D}$ *with* $\mathcal{D}_x \in \mathcal{F}$*, it holds:*

$$\underset{S_x \sim \mathcal{D}_x^m}{\mathbf{Pr}} \left[L_{\mathcal{D}_x, \sigma, \ell} \left(A \left(S_x, \sigma|_{S_x} \right) \right) > \epsilon \right] \leq \delta$$

For every $\epsilon, \delta > 0$ *and* $\eta \in [0, 1/2)$ *such that no integer satisfies the above inequality, we define* $m_{A,\mathcal{H},\mathcal{F},\ell}^{\text{LR}-\text{Massart}}(\epsilon, \delta, \eta) = \infty$.

Definition 4.3.8 (Distribution-dependent PAC Learnability with LR-Massart Noise)**.** *$\mathcal{H}$ is said to be PAC learnable with A with respect to $\mathcal{F}$ and ℓ in the presence of LR-Massart noise, if it holds that* $m_{A,\mathcal{H},\mathcal{F},\ell}^{\text{LR}-\text{Massart}}(\epsilon, \delta, \eta) < \infty$ *for all* $\epsilon, \delta > 0$ *and* $\eta \in [0, 1/2)$.

In the same vein, one can consider the even more general family of LR distributions, based on the Tsybakov noise condition, which is defined below.

Definition 4.3.9 (Label Ranking Distribution with Tsybakov Noise)**.** *A probability distribution $\mathcal{D}$ on $\mathcal{X} \times \mathbb{S}_k$ is said to be a label ranking distribution with Tsybakov Noise, if there exists some function* $\sigma \colon \mathcal{X} \to \mathbb{S}_k$ *and some* $\alpha \in [0, 1)$ *and* $B \geq 1$ *such that* $\mathcal{D}_{ij}$ *is an* (α, B, σ_{ij})*-Tsybakov distribution for all* $(i, j) \in [k]^2$ *with* $i \neq j$*. Then, we also say that $\mathcal{D}$ is an (α, B, σ)-LR distribution with Tsybakov noise.*

Definition 4.3.10 (Distribution-dependent LR-Tsybakov Sample Complexity of a Learning Algorithm)**.** *The distribution-dependent LR-Tsybakov sample complexity of A with respect to $\mathcal{H}$, $\mathcal{F}$ and ℓ is the function* $m_{A,\mathcal{H},\mathcal{F},\ell}^{\text{LR}-\text{Tsybakov}} \colon (0, \infty) \times (0, \infty) \times [0, 1) \times [1, \infty) \to \mathbb{N}$ *defined as follows: For every* $\epsilon, \delta > 0$, $\alpha \in [0, 1)$ *and* $B \geq 1$, $m_{A,\mathcal{H},\mathcal{F},\ell}^{\text{LR}-\text{Tsybakov}}(\epsilon, \delta, \alpha, B)$ *is the minimal integer such that for every*

- *integer* $m \geq m_{A,\mathcal{H},\mathcal{F},\ell}^{\text{LR}-\text{Tsybakov}}(\epsilon, \delta, \alpha, B)$,
- *target function* $\sigma \in \mathcal{H}$ *and*
- (α, B, σ)*-LR distribution with Tsybakov noise* $\mathcal{D}$ *with* $\mathcal{D}_x \in \mathcal{F}$*, it holds:*

$$\underset{S_x \sim \mathcal{D}_x^m}{\mathbf{Pr}} \left[L_{\mathcal{D}_x, \sigma, \ell} \left(A \left(S_x, \sigma|_{S_x} \right) \right) > \epsilon \right] \leq \delta$$

For every $\epsilon, \delta > 0$, $\alpha \in [0, 1)$ *and* $B \geq 1$ *such that no integer satisfies the above inequality, we define* $m_{A,\mathcal{H},\mathcal{F},\ell}^{\text{LR}-\text{Tsybakov}}(\epsilon, \delta, \alpha, B) = \infty$.

Definition 4.3.11 (Distribution-dependent PAC Learnability with LR-Tsybakov Noise)**.** *$\mathcal{H}$ is said to be PAC learnable with A with respect to $\mathcal{F}$ and ℓ in the presence of LR-Tsybakov noise, if it holds that* $m_{A,\mathcal{H},\mathcal{F},\ell}^{\text{LR}-\text{Tsybakov}}(\epsilon, \delta, \alpha, B) < \infty$ *for all* $\epsilon, \delta > 0$, $\alpha \in [0, 1)$ *and* $B \geq 1$.

Finally, recall that for every target ranking function $\sigma \colon \mathcal{X} \to \mathbb{S}_k$, there exists some score function $\boldsymbol{m} \colon \mathcal{X} \to \mathbb{R}^k$ such that $\sigma(x) = \mathfrak{S}(\boldsymbol{m}(x))$ for all $x \in \mathcal{X}$. This is the motivation for proposing an additional way of modeling the presence of noise. We assume that the noise affects the score values, which the final ranking is induced from, in an additive way. This approach, which was also considered by Fotakis et al. [2022a], is described formally below.

Definition 4.3.12 (Label Ranking Distribution with Additive Noise)**.** *Let $\mathcal{E}$ be a probability distribution on $\mathbb{R}^k$ and let $\boldsymbol{m} \colon \mathcal{X} \to \mathbb{R}^k$ be a target utility function. A probability distribution $\mathcal{D}$ on $\mathcal{X} \times \mathbb{S}_k$ is said to be an $(\boldsymbol{m}, \mathcal{E})$-label Ranking distribution with additive noise, if for any $(x, \pi) \sim \mathcal{D}$, it holds that $\pi = \mathfrak{S}(\boldsymbol{m}(x) + \boldsymbol{\xi})$, where x and $\boldsymbol{\xi}$ are independent and $\boldsymbol{\xi} \sim \mathcal{E}$.*

The noise model above will be used along with the Mallows model for LR (Definition 4.3.4) in the experimental section of Chapter 6.

4.4 Label Ranking Techniques

According to what was discussed before, a straightforward idea in the case of complete rankings would be to address the LR problem as a standard multiclass classification problem with $k!$ classes using the $0 - 1$ loss. However, as pointed out in several works, this approach might rise computational issues due to the massive number of classes. Furthermore, such an approach fails to leverage the structure of $\mathbb{S}_k$, since the inherent relation between classes (permutations) is untapped. In this section, we expand on the general *pairwise decomposition* and *labelwise decomposition* techniques and discuss some of the theoretical guarantees associated with them.

4.4.1 Label Ranking by Pairwise Decomposition

The pairwise decomposition technique (Fürnkranz and Hüllermeier [2003], Hüllermeier et al. [2008]) is a Label Ranking technique that extends the pairwise classification technique (Hastie and Tibshirani [1997], Allwein et al. [2000], Fürnkranz [2002]). Essentially, it constitutes a One-Versus-One (OVO) approach that reduces the LR problem to multiple and presumably simpler binary classification subproblems. In particular, we decompose the LR problem into $\binom{k}{2}$ subproblems, one for each (unordered) pair of labels (i, j), where we aim to learn the order of the labels in that specific pair, namely which label among i and j is preferred for each instance in $\mathcal{X}$. This is achieved by using a binary classification algorithm for each subproblem and running it on some adjusted version of the training data that is constructed as follows.

Algorithm 5 Label Ranking by Pairwise Decomposition

Input: Training set $T \subseteq \mathcal{X} \times \mathcal{Y}$, binary classification algorithm $A \in \mathcal{A}_{\mathcal{X},\{\pm 1\}}$
Output: Hypothesis $h\colon \mathcal{X} \to \mathbb{S}_k$

procedure PairwiseDecomposition(T, A)
  for $1 \leq i < j \leq k$ **do**
    $T_{ij} \leftarrow \varnothing$
    for $(x, \succ) \in T$ **do**
      if $i \succ j$ **then**
        $T_{ij} \leftarrow T_{ij} \cup (x, 1)$
      end if
      if $j \succ i$ **then**
        $T_{ij} \leftarrow T_{ij} \cup (x, -1)$
      end if
    end for
    $g_{ij} \leftarrow A(T_{ij})$
  end for
  return $(g_{ij})_{1 \leq i < j \leq k}$
end procedure
$C \leftarrow$ PairwiseDecomposition(T, A)
return EstimateAggregation(C) ▷ See voting and tournament aggregation

Pairwise decomposition stage Let T be a training set with elements in $\mathcal{X} \times \mathcal{Y}$. For all $1 \leq i < j \leq k$, we assign the training set

$$T_{ij} = \{(x, 1) : (x, \succ) \in T \wedge i \succ j\} \cup \{(x, -1) : (x, \succ) \in T \wedge j \succ i\}$$

to the subproblem concerning the label pair (i, j) and run an algorithm $A \in \mathcal{A}_{\mathcal{X},\{\pm 1\}}$[4] on T_{ij} to obtain a binary classifier $g_{ij} : \mathcal{X} \to \{-1, 1\}$ that is hopefully able to predict correctly the pairwise order between i and j[5]. In the case of having a training set T in $\mathcal{X} \times \mathbb{S}_k$ (namely, assuming examples with complete rankings), each subproblem's training set can be alternatively expressed as $T_{ij} = \{(x, \pi_{ij}) : (x, \pi) \in T\}$.

As the above suggests, a major advantage of the pairwise decomposition technique is that it can handle the case of observing incomplete rankings, since each subproblem focuses on some specific pair of labels and has zero dependence on the rest of them. Unfortunately, though, treating each subproblem independently comes at the cost of the pairwise predictions being potentially conflicting. Namely, there might be cases where the pairwise orders induce preferential cycles, rendering a direct induction of a ranking impossible, since the transitivity property is violated. To overcome this obstacle, we have to use some aggregation method that leverages the information provided by the pairwise predictions in a way that a valid ranking can occur.

Voting aggregation One of the simplest and most commonly used aggregation techniques is a voting scheme, which we refer to as voting aggregation, that works as follows. Given an instance $x \in \mathcal{X}$, for each pairwise duel, we cast a vote for the label ranked lower. Obviously, higher number of votes for a

[4] Here, we assume for the sake of simplicity that the same binary classification algorithm is used for each subproblem, but one could use a different algorithm for each subproblem.

[5] For convenience, we also define $g_{ji} = -g_{ij}$ for all $1 \leq i < j \leq k$.

label implies that the label should be ranked lower in the final ranking. Therefore, for each label $i \in [k]$, its position in the ranking is estimated as

$$s_i(x) = 1 + \sum_{j \in [k] \setminus \{i\}} \mathbb{1}\{g_{ij}(x) = -1\}\,.$$

The final ranking $\hat{\sigma}(x)$ results from sorting the estimations for every label, breaking ties arbitrarily (e.g. in alphabetical order), namely $\hat{\sigma}(x)$ should satisfy that $s_i(x) < s_j(x) \implies \hat{\sigma}(x)(i) < \hat{\sigma}(x)(j)$ for all $i \neq j$. The latter can be expressed through the operation $\hat{\sigma} = \text{argsort} \circ \text{argsort} \circ \boldsymbol{s}$[6], where $\boldsymbol{s} = (s_1, \ldots, s_k)$.

Algorithm 6 Voting Aggregation for Pairwise Decomposition

Input: Collection of binary classifiers $(g_{ij})_{1 \leq i < j \leq k}$
Output: Hypothesis $h \colon \mathcal{X} \to \mathbb{S}_k$

```
procedure VotingAggregation((g_ij)_{1≤i<j≤k})
    for 1 ≤ i ≤ k do
        s_i(x) ← 1 + Σ_{j∈[k]\{i}} 𝟙{g_ij(x) = −1}
    end for
    s ← (s_1, ..., s_k)
    return argsort ∘ argsort ∘ s
end procedure
```

We now make a brief reference to wherein the pairwise decomposition method combined with the voting aggregation technique can provide statistical guarantees.

Definition 4.4.1 (Stochastic Transitivity)**.** *Let $\mathcal{D}$ be any probability distribution on $\mathcal{X} \times \mathbb{S}_k$ and let $p_{ij}(x) = \mathbf{Pr}_{\pi \sim \mathcal{D}_{\pi|x}}[\pi(i) < \pi(j) \mid x]$ for all $x \in \mathcal{X}$ and $(i,j) \in [k]^2$. $\mathcal{D}_{\pi|x}$ is said to be stochastically transitive, if for any $x \in \mathcal{X}$ and $(i,j,l) \in [k]^3$, it holds that $p_{ij}(x) \geq 1/2 \wedge p_{jl}(x) \geq 1/2 \implies p_{il}(x) \geq 1/2$.*

Definition 4.4.2 (Strict Stochastic Transitivity)**.** *Let $\mathcal{D}$ be any probability distribution on $\mathcal{X} \times \mathbb{S}_k$ and let $p_{ij}(x) = \mathbf{Pr}_{\pi \sim \mathcal{D}_{\pi|x}}[\pi(i) < \pi(j) \mid x]$ for all $x \in \mathcal{X}$ and $(i,j) \in [k]^2$. $\mathcal{D}_{\pi|x}$ is said to be strictly stochastically transitive, if it is stochastically transitive and for any $x \in \mathcal{X}$ and $(i,j) \in [k]^2$, it holds that $p_{ij}(x) \neq 1/2$.*

Lemma 4.4.1 (Korba et al. [2017])**.** *Let $\mathcal{D}$ be any probability distribution on $\mathcal{X} \times \mathbb{S}_k$ and let $p_{ij}(x) = \mathbf{Pr}_{\pi \sim \mathcal{D}_{\pi|x}}[\pi(i) < \pi(j) \mid x]$ for all $x \in \mathcal{X}$ and $(i,j) \in [k]^2$. If $\mathcal{D}_{\pi|x}$ is strictly stochastically transitive, then the minimizer of $L_{\mathcal{D},d_\tau}$ is almost surely unique and given by $\sigma^* : \mathcal{X} \to \mathbb{S}_k$, where*

$$\sigma^*(x)(i) = 1 + \sum_{j \in [k] \setminus \{i\}} \mathbb{1}\{p_{ij}(x) < 1/2\}$$

for all $i \in [k]$.

Lemma 4.4.2. *Let $\mathcal{D}$ be any probability distribution on $\mathcal{X} \times \mathbb{S}_k$. Moreover, for all $(i,j) \in [k]^2$ with $i \neq j$, let $g^*_{ij} \colon \mathcal{X} \to \{\pm 1\}$ be any Bayes optimal classifier for the binary classification task of predicting the pairwise order for the label pair (i,j). Let $(g_{ij})_{1 \leq i < j \leq k}$ be any collection of classifiers from $\mathcal{X}$ to $\{-1,1\}$. Assume that $\mathcal{D}_{\pi|x}$ is strictly stochastically transitive and let σ^* be a minimizer of $L_{\mathcal{D},d_\tau}$. Running the voting aggregation algorithm with $(g_{ij})_{1 \leq i < j \leq k}$ as input, yields a hypothesis $\hat{\sigma} : \mathcal{X} \to \mathbb{S}_k$ such that*

$$\mathop{\mathbf{Pr}}_{x \sim \mathcal{D}_x}\left[\hat{\sigma}(x) \neq \sigma^*(x)\right] \leq \sum_{1 \leq i < j \leq k} \mathop{\mathbf{Pr}}_{x \sim \mathcal{D}_x}\left[g_{ij}(x) \neq g^*_{ij}(x)\right]\,.$$

Proof. Let $\hat{\sigma}^* \colon \mathcal{X} \to \mathbb{S}_k$ be the hypothesis generated by aggregating the classifiers $\{g^*_{ij}\}_{(i,j) \in [k]^2}$ using the voting aggregation algorithm. It holds that

$$\bigcap_{1 \leq i < j \leq k} \left\{x \in \mathcal{X} : g_{ij}(x) = g^*_{ij}(x)\right\} \subset \left\{x \in \mathcal{X} : \hat{\sigma}(x) = \hat{\sigma}^*(x)\right\}\,,$$

or, equivalently,

$$\left\{x \in \mathcal{X} : \hat{\sigma}(x) \neq \hat{\sigma}^*(x)\right\} \subset \bigcup_{1 \leq i < j \leq k} \left\{x \in \mathcal{X} : g_{ij}(x) \neq g^*_{ij}(x)\right\}\,,$$

[6] The function argsort : $\mathbb{R}^k \to \mathbb{S}_k$ takes as input a vector $\boldsymbol{v} = (v_1, \ldots, v_k) \in \mathbb{R}^k$ and outputs the unique permutation $\sigma^{-1} \in \mathbb{S}_k$ with the property that, for all $1 \leq i < j \leq k$, it holds $i \succ_\sigma j$ if and only if $v_i \leq v_j$.

which implies, by an application of the union bound, that

$$\mathop{\mathbf{Pr}}_{x\sim\mathcal{D}_x}\left[\hat{\sigma}(x)\neq\hat{\sigma}^*(x)\right]\leq\sum_{1\leq i<j\leq k}\mathop{\mathbf{Pr}}_{x\sim\mathcal{D}_x}\left[g_{ij}(x)\neq g_{ij}^*(x)\right].$$

Since the strict stochastic transitivity assumption holds, Lemma 4.4.2 implies that

$$\sigma^*(x)(i)=1+\sum_{j\in[k]\setminus\{i\}}\mathbb{1}\{g_{ij}^*(x)=-1\}=\hat{\sigma}^*(x)(i)\,.$$

for all $x\in\mathcal{X}$ and $i\in[k]$ almost surely, which concludes the proof. □

The aforementioned lemma indicates that to bound the error probability against the median hypothesis $\sigma^*\in\operatorname{argmin}_{\sigma\in\mathbb{S}_k^{\mathcal{X}}}L_{\mathcal{D},d_\tau}(\sigma)$, it is sufficient to focus our attention on each binary subproblem, where one can use standard tools to obtain generalization bounds (see Bousquet et al. [2004], Boucheron et al. [2005]). Such an approach has been adopted by Vogel and Clémençon [2020] and Fotakis et al. [2022a] to provide (under some additional distributional assumptions) statistical guarantees for the LR problem given incomplete rankings and using empirical risk minimization as the binary classification algorithm for each subproblem.

Tournament aggregation Another way of aggregating the pairwise predictions into a ranking is by creating a tournament (complete directed graph) that indicates the pairwise preferences (Algorithm 7). In particular, for each instance $x\in\mathcal{X}$, we construct a tournament $G_x=(V,E_x)$ with vertices $V=[k]$ and edges $E_x=\{(i,j)\in[k]^2:i\neq j\ \wedge\ g_{ij}(x)=1\}$. That is, for each pair of labels (i,j), the direction of the corresponding edge in G is from i to j, if and only if for the instance x, label i is preferred over label j, according to the binary classifier g_{ij}[7].

Algorithm 7 Tournament Aggregation for Pairwise Decomposition

Input: Collection of binary classifiers $(g_{ij})_{1\leq i<j\leq k}$, algorithm A that converts a tournament to a DAG
Output: Hypothesis $h\colon\mathcal{X}\to\mathbb{S}_k$

procedure TournamentAggregation($(g_{ij})_{1\leq i<j\leq k}, A$)
  $V\leftarrow[k]$
  $E_x\leftarrow\{(i,j)\in[k]^2:i\neq j\ \wedge\ g_{ij}(x)=1\}$
  $G_x\leftarrow(V,E_x)$
  $G'_x\leftarrow A(G_x)$
  Let $\hat{\sigma}(x)$ be the ranking induced by G'_x
  return $\hat{\sigma}(\cdot)$
end procedure

If G_x is acyclic, then we can obtain a ranking by its topological order in a straightforward way, since the transitivity property holds. If G_x contains cycles, a natural approach would be to flip (or remove) as few edges as possible to render G_x acyclic and then proceed as before, which is equivalent to finding a minimum feedback arc set (MFAS) in G_x. The latter problem is NP-hard, but there exist efficient approximation algorithms which we can turn to. Such an algorithm, is the KwikSort algorithm (Ailon et al. [2008]), which is a randomized MFAS algorithm for tournaments with expected approximation ratio 3 and $O(n^3)$ runtime, where n is the number of vertices. Moreover, as shown by van Zuylen et al. [2007], there exists a deterministic algorithm, based on KwikSort, with the same approximation ratio and runtime, which we denote by MFAS3.

The following lemma, which constitutes a slight generalization of a result shown by Fotakis et al. [2022b], opens the way for acquiring a series of PAC guarantees for the pairwise decomposition method combined with the tournament aggregation technique.

Lemma 4.4.3. *Let $(g_{ij})_{1\leq i<j\leq k}$ be any collection of classifiers from $\mathcal{X}$ to $\{-1,1\}$. Running the tournament aggregation algorithm with $(g_{ij})_{1\leq i<j\leq k}$ and* MFAS3 *as input, yields a hypothesis $\hat{\sigma}\colon\mathcal{X}\to\mathbb{S}_k$ such that*

$$L_{\mathcal{D},d_\tau}(\hat{\sigma})\leq 4\sum_{1\leq i<j\leq k}L_{\mathcal{D}_{ij},\ell_{0-1}}(g_{ij})\,.$$

[7] Notice that the estimation of the position of each label in the previously mentioned voting scheme can also be expressed through G_x as $\hat{s}_x(i)=1+\sum_{j=1}^k\mathbb{1}\{(j,i)\in E_x\}$

Proof. Let $(x, \pi) \sim \mathcal{D}$. If all edges of G_x have the correct direction (with respect to the direction dictated by π), then G_x is a DAG, whose topological ordering yields the desired ranking. However, with probability at most $\sum_{1 \leq i < j \leq k} \mathbf{Pr}_{(x,\pi)\sim\mathcal{D}} [g_{ij}(x) \neq \pi_{ij}]$, there exist edges having wrong direction, which could cause G_x to have cycles. In particular, the expected number of edges having wrong direction (with respect to π) is

$$W = \sum_{1 \leq i < j \leq k} \mathop{\mathbf{Pr}}_{(x,\pi)\sim\mathcal{D}} [g_{ij}(x) \neq \pi_{ij}] \ .$$

Flipping those edges (which are of course unknown) would give a DAG, which we can obtain the desired ranking from as before. Instead, we will use the deterministic 3-approximation algorithm MFAS3 to make the graph acyclic at the cost of getting an upper bound on the expected KT distance, which is a multiple of W. Specifically, if we denote by M the expected minimum number of edges, whose removal (or flipping) would render the graph acyclic, we infer that $M \leq W$. Running MFAS3 on G_x, results in an acyclic graph that gives a ranking $\hat{\sigma}(x)$, corresponding to a hypothesis $\hat{\sigma} \colon \mathbb{R}^d \to \mathbb{S}_k$, such that

$$\mathop{\mathbf{E}}_{(x,\pi)\sim\mathcal{D}} [d_\tau (\hat{\sigma}(x), \pi)] \leq W + 3M \leq 4 \sum_{1 \leq i < j \leq k} \mathop{\mathbf{Pr}}_{(x,\pi)\sim\mathcal{D}} [g_{ij}(x) \neq \pi_{ij}] \ ,$$

which concludes the proof. □

Algorithm 8 KwikSort

Input: Tournament G
Output: Topological order of G's vertices

procedure KwikSort(G)
  $(V, E) \leftarrow G$
  if $V = \varnothing$ **then**
    return () ▷ empty tuple
  end if
  $V_L \leftarrow \varnothing$
  $V_R \leftarrow \varnothing$
  Select random pivot $v \in V$
  for $u \in V \setminus \{v\}$ **do**
    if $(u, v) \in E$ **then**
      $V_L \leftarrow V_L \cup \{u\}$ ▷ place u on the left of v
    else
      $V_R \leftarrow V_R \cup \{u\}$ ▷ place u on the right of v
    end if
  end for
  $G_L \leftarrow G[V_L]$ ▷ tournament induced in G by V_L
  $G_R \leftarrow G[V_R]$ ▷ tournament induced in G by V_L
  return (KwikSort(G_L), v, KwikSort(G_R)) ▷ concatenation of vertices
end procedure

Henceforth, we let PWT3(A) denote the pairwise decomposition algorithm combined the binary classification algorithm A and with the tournament aggregation method, with the latter using MFAS3 as MFAS algorithm. Lemma 4.4.3 has the following implications.

Lemma 4.4.4. *For every hypothesis class $\mathcal{H} \in \mathbb{S}_k^{\mathcal{X}}$, every family of structured probability distributions $\mathcal{F}$ on $\mathcal{X} \times \mathbb{S}_k$ and every binary classifier $A \in \mathcal{A}_{\mathcal{X},\{\pm 1\}}$, it holds*

1. $m^r_{\mathsf{PWT3}(A),\mathcal{H},d_\tau}(\epsilon, \delta) \leq \max_{1 \leq i < j \leq k} m^r_{A,\mathcal{H}_{ij},\ell_{0-1}} \left(\frac{\epsilon}{4\binom{k}{2}}, \frac{\delta}{\binom{k}{2}} \right)$
2. $m^{\text{LR}-\text{Massart}}_{\mathsf{PWT3}(A),\mathcal{H},\mathcal{F},d_\tau}(\epsilon, \delta, \eta) \leq \max_{1 \leq i < j \leq k} m^{\text{Massart}}_{A,\mathcal{H}_{ij},\mathcal{F}} \left(\frac{\epsilon}{4\binom{k}{2}}, \frac{\delta}{\binom{k}{2}}, \eta \right)$
3. $m^{\text{LR}-\text{Tsybakov}}_{\mathsf{PWT3}(A),\mathcal{H},\mathcal{F},d_\tau}(\epsilon, \delta, \alpha, B) \leq \max_{1 \leq i < j \leq k} m^{\text{Tsybakov}}_{A,\mathcal{H}_{ij},\mathcal{F}} \left(\frac{\epsilon}{4\binom{k}{2}}, \frac{\delta}{\binom{k}{2}}, \alpha, B \right)$

for all $\epsilon, \delta > 0$, $\eta \in [0, 1/2)$, $\alpha \in [0, 1)$ and $B \geq 1$.

Proof. Fix any $\epsilon, \delta > 0$ and let $\mathcal{D}$ be any probability distribution that is realizable by $\mathcal{H}$. For all $1 \leq i < j \leq k$, if A is given $m \geq m^r_{A,\mathcal{H}_{ij},\ell_{0-1}}(\epsilon, \delta)$ i.i.d. samples from $\mathcal{D}_{ij}$, it returns a hypothesis $g_{ij} \colon \mathcal{X} \to \{\pm 1\}$ such that, with probability at least $1 - \delta$, it holds

$$L_{\mathcal{D}_{ij},\ell_{0-1}}(g_{ij}) \leq \epsilon \,.$$

Then, from the union bound, we get that, with probability at least $1 - \delta\binom{k}{2}$, it holds

$$\max_{1 \leq i < j \leq k} L_{\mathcal{D}_{ij},\ell_{0-1}}(g_{ij}) \leq \epsilon \,.$$

Therefore, Lemma 4.4.3 implies that PWT3 outputs a hypothesis $\hat{\sigma}$ such that, with probability at least $1 - \delta\binom{k}{2}$, it holds

$$L_{\mathcal{D},d_\tau}(\hat{\sigma}) \leq 4\binom{k}{2}\epsilon \,,$$

which proves the first point. The proof for the rest points is nearly identical. □

4.4.2 Label Ranking by Labelwise Decomposition

We now present another the labelwise decomposition technique (Cheng et al. [2013], Cheng and Hüllermeier [2013]), whose main idea is to reduce the LR problem into multiple problems, each concerning a particular label. Specifically, we decompose the LR problem into k subproblems, one for each label i, where we aim to predict the position of i in the ranking independently.

Originally, Cheng et al. [2013] assumed that a probabilistic approach is used to train each labelwise predictor, in which the learner predicts a conditional probability distribution $\mathcal{M}_{i|x}$ on the set $[k]$ of possible ranks for each label $i \in [k]$. Namely, for each instance $x \in \mathcal{X}$ and for all $i \in [k]$, $\mathcal{M}_{i|x}$ gives the occupation probabilities of each position in x's ranking by label i. Cheng et al. [2013] showed that the minimization of the expected loss with respect to the aforementioned distributions, for any labelwise decomposable loss (such as the Spearman's footrule d_1 and the Spearman's distance d_2) can be reduced to an *assignment problem*, which can be solved by means of the Hungarian algorithm in $O(k^3)$ time.

Here, we focus on the more abstract regression approach of Fotakis et al. [2022a], where each subproblem yields a hypothesis, whose predictions lie in a real-valued output space.

Algorithm 9 Label Ranking by Labelwise Decomposition

Input: Training set $T \subseteq \mathcal{X} \times \mathbb{S}_k$, algorithm $A \in \mathcal{A}_{\mathcal{X},\mathbb{R}}$
Output: Hypothesis $h : \mathcal{X} \to \mathbb{S}_k$

```
procedure LabelwiseDecomposition(T, A)
    for 1 ≤ i ≤ k do
        T_i ← ∅
        for (x, π) ∈ T do
            T_i ← T_i ∪ (x, π(i))
        end for
        g_i ← A(T_i)
    end for
    return (g_1, ..., g_k)
end procedure
g ← LabelwiseDecomposition(T, A)
return argsort ∘ argsort ∘ g
```

Labelwise decomposition stage Let T be a training set with elements of the form (x, π), where $(x, \pi) \in \mathcal{X} \times \mathbb{S}_k$. For all $1 \leq i \leq k$, we assign the training set

$$T_i = \{(x, \pi(i)) : (x, \pi) \in T\}$$

to the subproblem concerning label i and run an algorithm $A \in \mathcal{A}_{\mathcal{X},\mathbb{R}}$[8] on T_i to obtain a predictor $g_i \colon \mathcal{X} \to \mathbb{R}$ that is hopefully able to yield a good estimate for the position of label i.

[8]Like in the pairwise decomposition method, we assume that the same binary classification algorithm is used for each subproblem, but one could use a different algorithm for each subproblem.

Aggregation stage As for the aggregation of the labels' position estimates into a single ranking, the most intuitive choice would be to rank the labels by increasing estimated position order, breaking ties arbitrarily. Namely, the final ranking $\hat{\sigma}(x)$ should satisfy that $g_i(x) < g_j(x) \implies \hat{\sigma}(x)(i) < \hat{\sigma}(x)(j)$ for all $i \neq j$.

Observe that the labelwise decomposition algorithm was defined assuming examples with complete only rankings. This is due to the fact that, unlike the pairwise decomposition case, the labelwise decomposition method should be expected to work only in the case where the training data consist of complete rankings. Indeed, the position of a label within a ranking provides a meaningful result only when all other labels are present in the ranking as well. To extend this labelwise approach to incomplete rankings, one would need to make additional assumptions about the way incomplete rankings are generated.

Lemma 4.4.5 (Fotakis et al. [2022a]). *Let $(g_i)_{i\in[k]}$ be any collection of predictors from $\mathcal{X}$ to $\mathbb{R}$ obtained by the labelwise decomposition algorithm. The hypothesis $\hat{\sigma}\colon \mathcal{X} \to \mathbb{S}_k$ returned by the said algorithm satisfies that:*

$$L_{\mathcal{D},d_2}(\hat{\sigma}) \in O\left(k \max_{i\in[k]} L_{\mathcal{D}_i,\ell_2}(g_i)\right)$$

The above lemma, which is implicitly stated in the work of Fotakis et al. [2022a], shows that to bound the expected Spearman's distance with respect to $\mathcal{D}$, it suffices to focus our attention on each label's subproblem and bound the expected squared loss with respect to $\mathcal{D}_i$ for all $i \in [k]$. As a corollary, we can obtain PAC results, similar to the ones that were previously mentioned in the pairwise decomposition section.

Chapter 5

Learning Linear Sorting Functions

In this chapter, we assume that the learner observes rankings that are produced by some target linear score function, which we aim to learn. Then, the underlying hypothesis class coincides with the class of *linear sorting functions*, introduced by Har-Peled et al. [2002]. This variant of LR is also referred to as linear LR (Fotakis et al. [2022b]). As we will see, the class of linear sorting functions is tightly connected to that of halfspaces, which will enable us to acquire similar results to those of halfspaces for linear LR, both in the noiseless and the noisy setting.

Definition 5.0.1 (Linear Sorting Function)**.** *A linear sorting function (LSF) in $\mathbb{R}^d$ with $k \geq 2$ labels is any function $\sigma_{\boldsymbol{W},\boldsymbol{b}} \colon \mathbb{R}^d \to \mathbb{S}_k$ parameterized by a matrix $\boldsymbol{W} \in \mathbb{R}^{k \times d}$ and a vector $\boldsymbol{b} \in \mathbb{R}^k$ that is defined as $\sigma_{\boldsymbol{W},\boldsymbol{b}}(\boldsymbol{x}) = \mathfrak{S}(\boldsymbol{W}\boldsymbol{x} + \boldsymbol{b})$.*

A linear sorting function $\sigma_{\boldsymbol{W},\boldsymbol{b}}$ is said to be homogeneous if $\boldsymbol{b} = \boldsymbol{0}$, in which case we simply denote it as $\sigma_{\boldsymbol{W}}$. We denote by $\mathcal{H}^{d,k}_{\mathrm{LSF}}$ (resp. $\mathcal{H}^{d,k}_{\mathrm{HLSF}}$) the class of LSFs (resp. homogeneous LSFs) in $\mathbb{R}^d$ with $k \geq 2$ labels.

Remark. *Like in the case of halfspaces, each LSF $\sigma_{\boldsymbol{W},\boldsymbol{b}}(\boldsymbol{x}) = \mathfrak{S}(\boldsymbol{W}\boldsymbol{x} + \boldsymbol{b})$ in $\mathbb{R}^d$ can be rewritten as $\sigma_{\boldsymbol{W}'}(\boldsymbol{x}') = \mathfrak{S}(\boldsymbol{W}'\boldsymbol{x}')$, where $\boldsymbol{W}' = [\boldsymbol{W}\ \boldsymbol{b}]$ and $\boldsymbol{x}' = (x_1, \ldots, x_d, 1)$. Namely, it can be expressed as a homogeneous LSF in $\mathbb{R}^{d+1}$ applied over the transformation that appends the constant 1 to each input vector.*

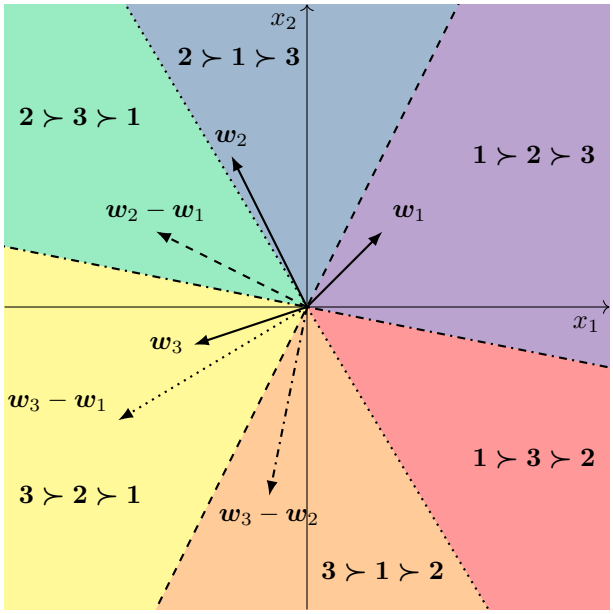


Figure 5.1: Visualization of a homogeneous linear sorting function in $\mathbb{R}^2$ with 3 labels

Notice that for $k = 2$, there is an equivalence between the concept class of LSFs and the concept class of halfspaces on $\mathbb{R}^d$. Moreover, for any $\sigma_{\boldsymbol{W},\boldsymbol{b}} \in \mathcal{H}^{d,k}_{\mathrm{LSF}}$, the function $(\sigma_{\boldsymbol{W},\boldsymbol{b}})_{ij}$ that is involved in the

pairwise problem concerning the label pair (i, j), can be written as

$$(\sigma_{\boldsymbol{W},\boldsymbol{b}})_{ij}(\boldsymbol{x}) = \operatorname{sign}\left(\sigma_{\boldsymbol{W},\boldsymbol{b}}(j)(\boldsymbol{x}) - \sigma_{\boldsymbol{W},\boldsymbol{b}}(i)(\boldsymbol{x})\right) = \operatorname{sign}\left(\langle \boldsymbol{w}_i - \boldsymbol{w}_j, \boldsymbol{x}\rangle + b_i - b_j\right) = h_{\boldsymbol{w}_i - \boldsymbol{w}_j, b_i - b_j}(\boldsymbol{x}),$$

for all $\boldsymbol{x} \in \mathbb{R}^d$ and $1 \leq i < j \leq k$. Namely, it is a halfspace with weight vector $\boldsymbol{w}_i - \boldsymbol{w}_j$ and bias term $b_i - b_j$. This means that the order between i and j in $\sigma(\boldsymbol{x})$ is determined by which side of the hyperplane corresponding to $h_{\boldsymbol{w}_i - \boldsymbol{w}_j, b_i - b_j}$ the point $\boldsymbol{x}$ lies in. This property is very crucial as it guides us towards a pairwise decomposition approach for the linear LR problem, concerning binary halfspace learning subproblems, whose learnability has been excessively studied both in the noiseless and the noisy setting.

5.1 Learning Linear Sorting Functions in the Noiseless Setting

Concerning the learnability of LSFs in the noiseless (realizable) setting, we begin by observing that Lemma 4.4.4 has the following implication.

Corollary 5.1.1. *For every binary classifier $A \in \mathcal{A}_{\mathcal{X},\{\pm 1\}}$, it holds*

$$m^r_{\mathsf{PWT3}(A),\mathcal{H}^{d,k}_{\mathrm{LSF}},d_\tau}(\epsilon,\delta) \leq m^r_{A,\mathcal{H}^d_{\mathrm{LTF}},\ell_{0-1}}\left(\frac{\epsilon}{4\binom{k}{2}}, \frac{\delta}{\binom{k}{2}}\right)$$

for all $\epsilon, \delta > 0$.

In Chapter 4, we have seen that the class of halfspaces is efficiently realizably PAC learnable using linear programming (Algorithm 1). Therefore, Corollary 5.1.1 implies that $\mathsf{PWT3}(A)$, choosing Algorithm 1 as the binary classification algorithm A, is an efficient realizable PAC learner for $\mathcal{H}^{d,k}_{\mathrm{LSF}}$ with respect to the KT distance.

A disadvantage of $\mathsf{PWT3}(A)$ is that it constitutes an improper learner, whose output hypothesis requires:

- $O(dk^2)$ memory to store the $\binom{k}{2}$ pairwise weight vectors,
- $O(dk^2 + k^3)$ runtime (evaluating $\binom{k}{2}$ inner products, constructing a preference graph and running $\mathsf{MFAS3}$ to break its cycles) to output a ranking for each fresh instance.

On the contrary, a proper learner outputs a hypothesis that requires:

- $O(dk)$ memory to store the matrix and vector that define an LSF,
- $O(dk + k \log k)$ steps (evaluating a score vector and sorting its elements) to output a ranking for each fresh instance.

In fact, by a more prudent application of linear programming, we can derive a proper and efficient realizable PAC learner for the class of LSFs with respect to the KT distance. The main idea is to leverage the fact that the linear programs used in each subproblem can be merged into a single linear program. In particular, let $S = \left(\left(\boldsymbol{x}^{(1)}, \pi^{(1)}\right), \ldots, \left(\boldsymbol{x}^{(m)}, \pi^{(m)}\right)\right)$ be any training set of samples in $\mathbb{R}^d \times \mathbb{S}_k$. Consider the following linear program, which we denote by $\mathsf{LP2}$[1]:

$$\begin{aligned} \text{Find} \quad & \boldsymbol{W} \in \mathbb{R}^{k \times d},\ \boldsymbol{b} \in \mathbb{R}^k \\ \text{subject to} \quad & \pi^{(t)}_{ij}\left(\left\langle \boldsymbol{w}_i - \boldsymbol{w}_j, \boldsymbol{x}^{(t)}\right\rangle + b_i - b_j\right) \geq 1 \quad \forall\ 1 \leq i < j \leq k,\ t \in [m] \end{aligned}$$

In Theorem 5.1.1, we show that $\mathsf{LP2}$ combined with the Ellipsoid method (Algorithm 10) yields a proper and efficient realizable PAC learner for the class of LSFs.

Theorem 5.1.1. *$\mathcal{H}^{d,k}_{\mathrm{LSF}}$ is properly realizably PAC learnable with Algorithm 10 with respect to the KT distance with sample complexity $O\left((d\log(k/\epsilon) + \log(k/\delta))k^2/\epsilon\right)$ and polynomial runtime in d, in k, in the number of samples and in the representation size of real numbers.*

[1] $\mathsf{LP2}$ can be typically formulated by arranging the unknown variables in a $k(d+1)$-dimensional vector.

Algorithm 10 Properly Learning Linear Sorting Functions with Linear Programming

Input: Training set $S \subset \mathbb{R}^d \times \mathbb{S}_k$
Output: LSF $\sigma_{\boldsymbol{W},\boldsymbol{b}} \colon \mathbb{R}^d \to \mathbb{S}_k$
1: Construct LP2 from S
2: $(\boldsymbol{W}, \boldsymbol{b}) \leftarrow$ ELLIPSOID(LP2) ▷ See Appendix B
3: **return** $\sigma_{\boldsymbol{W},\boldsymbol{b}}$

Proof. Fix any $\epsilon, \delta \in (0,1)$ and let $\mathcal{D}$ be any probability distribution on $\mathbb{R}^d \times \mathbb{S}_k$ that is realizable by some $\sigma^* \in \mathcal{H}^{d,k}_{\text{LSF}}$. Let $S = \left(\left(\boldsymbol{x}^{(1)}, \pi^{(1)}\right), \ldots, \left(\boldsymbol{x}^{(m)}, \pi^{(m)}\right)\right)$ be any training set of i.i.d. samples from $\mathcal{D}$. Since we are in the realizable case, the linear program LP2 constructed from S as above, is almost surely feasible (due to linear separability per pairwise subproblem) and any solution of it, obviously, corresponds to an ERM learner since it ensures zero empirical error on S.

For any $1 \leq i < j \leq k$, consider the set L_{ij} of linear constraints $\pi^{(t)}_{ij}\left(\left\langle \boldsymbol{w}_i - \boldsymbol{w}_j, \boldsymbol{x}^{(t)} \right\rangle + b_i - b_j\right) \geq 1$ for all $t \in [m]$. Then, LP2 consists of the constraints in $L = \bigcup_{i<j} L_{ij}$. Observe that for all $1 \leq i < j \leq k$, the set of constraints L_{ij} constitutes a linear programming ERM learner for a halfspace learning task with σ^*_{ij} being the target halfspace. In the previous chapter, we have seen that providing $O\left((d\log(1/\epsilon) + \log(1/\delta))/\epsilon\right)$ samples to this learner suffices to get the (ϵ, δ) realizable PAC learning guarantee. Thus, constructing L using $m \in O\left((d\log(1/\epsilon) + \log(1/\delta))/\epsilon\right)$ i.i.d. samples from $\mathcal{D}$ and solving the resultant linear program LP2, yields a matrix $\boldsymbol{W} \in \mathbb{R}^{k \times d}$ and a vector $\boldsymbol{b} \in \mathbb{R}^k$ such that for all $1 \leq i < j \leq k$

$$\Pr_{S \sim \mathcal{D}^m}\left[\Pr_{\boldsymbol{x} \sim \mathcal{D}_{\boldsymbol{x}}}\left[h_{\boldsymbol{w}_i - \boldsymbol{w}_j, b_i - b_j}(\boldsymbol{x}) \neq \sigma^*_{ij}(\boldsymbol{x})\right] > \epsilon\right] \leq \delta\,.$$

By application of the union bound, we get that with probability at least $1 - \binom{k}{2}\delta$, it holds

$$\begin{aligned}
\max_{1 \leq i < j \leq k} \Pr_{\boldsymbol{x} \sim \mathcal{D}_{\boldsymbol{x}}}\left[h_{\boldsymbol{w}_i - \boldsymbol{w}_j, b_i - b_j}(\boldsymbol{x}) \neq \sigma^*_{ij}(\boldsymbol{x})\right] \leq \epsilon \implies \\
\sum_{1 \leq i < j \leq k} \Pr_{\boldsymbol{x} \sim \mathcal{D}_{\boldsymbol{x}}}\left[h_{\boldsymbol{w}_i - \boldsymbol{w}_j, b_i - b_j}(\boldsymbol{x}) \neq \sigma^*_{ij}(\boldsymbol{x})\right] \leq \binom{k}{2}\epsilon \implies \\
\mathop{\mathbf{E}}_{\boldsymbol{x} \sim \mathcal{D}_{\boldsymbol{x}}}\left[\sum_{1 \leq i < j \leq k} \mathbb{1}\left\{\left(\sigma_{\boldsymbol{W},\boldsymbol{b}}(\boldsymbol{x})(i) - \sigma_{\boldsymbol{W},\boldsymbol{b}}(\boldsymbol{x})(j)\right)\left(\sigma^*(\boldsymbol{x})(i) - \sigma^*(\boldsymbol{x})(j)\right) < 0\right\} \leq\right] \binom{k}{2}\epsilon \implies \\
\mathop{\mathbf{E}}_{\boldsymbol{x} \sim \mathcal{D}_{\boldsymbol{x}}}\left[d_\tau\left(\sigma_{\boldsymbol{W},\boldsymbol{b}}(\boldsymbol{x}), \sigma^*(\boldsymbol{x})\right)\right] \leq \binom{k}{2}\epsilon
\end{aligned}$$

Therefore, constructing the aforementioned LP using $m \in O\left((d\log(k/\epsilon) + \log(k/\delta))k^2/\epsilon\right)$ i.i.d. samples suffices to get a matrix $\boldsymbol{W} \in \mathbb{R}^{k \times d}$ and a vector $\boldsymbol{b} \in \mathbb{R}^k$ such that

$$\Pr_{S \sim \mathcal{D}^m}\left[\mathop{\mathbf{E}}_{(\boldsymbol{x},\pi) \sim \mathcal{D}}\left[d_\tau\left(\sigma_{\boldsymbol{W},\boldsymbol{b}}(\boldsymbol{x}), \pi\right)\right] > \epsilon\right] \leq \delta\,.$$

Moreover, the linear program above can be efficiently solved, i.e. in time polynomial in m, d, k and in the representation size of real numbers using the Ellipsoid method. Finally, the properness comes trivially from the fact that LP2 returns a matrix and a vector, which define an LSF. These facts conclude the proof. □

5.2 Learning Homogeneous Linear Sorting Functions in the Noisy Setting

We now study the learnability of LSFs in the presence of noise. Initially, we observe that Lemma 4.4.4 has the following implication.

Corollary 5.2.1. *For every family of structured probability distributions $\mathcal{F}$ on $\mathcal{X} \times \mathbb{S}_k$ and every binary classifier $A \in \mathcal{A}_{\mathcal{X},\{\pm 1\}}$, it holds*

1. $m^{\text{LR}-\text{Massart}}_{\mathsf{PWT3}(A),\mathcal{H}^{d,k}_{\text{HLSF}},\mathcal{F},d_\tau}(\epsilon,\delta,\eta) \leq m^{\text{Massart}}_{A,\mathcal{H}^d_{\text{HLTF}},\mathcal{F}}\left(\frac{\epsilon}{4\binom{k}{2}}, \frac{\delta}{\binom{k}{2}}, \eta\right)$

2. $m^{\mathrm{LR-Tsybakov}}_{\mathsf{PWT3}(A),\mathcal{H}^{d,k}_{\mathrm{HLSF}},\mathcal{F},d_\tau}(\epsilon,\delta,\alpha,B) \leq m^{\mathrm{Tsybakov}}_{A,\mathcal{H}^{d}_{\mathrm{HLTF}},\mathcal{F}}\left(\frac{\epsilon}{4\binom{k}{2}},\frac{\delta}{\binom{k}{2}},\alpha,B\right)$

for all $\epsilon,\delta>0$, $\eta\in[0,1/2)$, $\alpha\in[0,1)$ *and* $B\geq 1$.

In Chapter 3, we saw that there exist efficient homogeneous halfspace learners in the distribution-dependent PAC model with Massart (e.g. the algorithm of Diakonikolas et al. [2020a] for isotropic log-concave marginals) and Tsybakov noise (e.g. the algorithm of Diakonikolas et al. [2021b] for isotropic log-concave marginals). Therefore, like in the noiseless case, Corollary 5.2.1, implies that $\mathsf{PWT3}(A)$, choosing A to be one of the aforementioned halfspace learners, is an improper efficient PAC learner for $\mathcal{H}^{d,k}_{\mathrm{HLSF}}$ with respect to the KT distance and $\mathcal{F}^d_{\mathrm{LC}}$[2], in the presence of LR-Massart and LR-Tsybakov noise.

As it has already been discussed though, such an improper learner has the downside of producing hypotheses with increased runtime and storage requirements, in comparison to a proper learner. In what follows, we show that the class of homogeneous LSFs in $\mathbb{R}^d$ is properly and efficiently PAC learnable, with respect to the KT distance and $\mathcal{F}^d_{\mathrm{LC}}$, in the presence of LR-Massart and LR-Tsybakov noise. Of course the proper Algorithm 10 fails to work in this case, since the linear separability property per halfspace subproblem does not hold anymore. Instead, we resort to the pairwise decomposition method and use a different aggregation method that ensures properness. Our approach extends the work of Fotakis et al. [2022b], which proves the aforementioned results for the specific case of the standard d-dimensional normal distribution $\mathcal{N}_d$, and is based on similar techniques.

Let $\boldsymbol{W}^*\in\mathbb{R}^{k\times d}$, where $\boldsymbol{w}^*_i\neq\boldsymbol{w}^*_j$ for all $1\leq i<j\leq k$, be the ground truth matrix associated with the target LSF $\sigma_{\boldsymbol{W}^*}$ and let $\mathcal{D}_{\boldsymbol{x}}\in\mathcal{F}^d_{\mathrm{LC}}$. Our algorithm begins by applying the pairwise decomposition method using a proper homogeneous halfspace learner A that satisfies the PAC learnability guarantee in the presence of Massart (resp. Tsybakov noise) with respect to $\mathcal{F}^d_{\mathrm{LC}}$ (such as Algorithm 4 for the Massart case). As a result, for all $\epsilon,\delta>0$, we can obtain $\binom{k}{2}$ vectors $\boldsymbol{v}_{ij}$ such that for all $1\leq i<j\leq k$, it holds

$$\mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{D}_{\boldsymbol{x}}}\left[\mathrm{sign}\left(\left\langle\boldsymbol{w}^*_i-\boldsymbol{w}^*_j,\boldsymbol{x}\right\rangle\right)\neq\mathrm{sign}\left(\left\langle\boldsymbol{v}_{ij},\boldsymbol{x}\right\rangle\right)\right]\leq\epsilon$$

with probability at least $1-\delta$. Moreover, without loss of generality, we assume that $\left\|\boldsymbol{v}_{ij}\right\|_2=1$ for all $1\leq i<j\leq k$.

Our goal is to exploit the information provided by the collection of pairwise vectors $\boldsymbol{v}_{ij}$, to obtain a matrix $\boldsymbol{W}\in\mathbb{R}^{k\times d}$, such that $L_{\mathcal{D}_{\boldsymbol{x}},\sigma_{\boldsymbol{W}^*},d_\tau}(\sigma_{\boldsymbol{W}})$ is small. Due to the form of the KT distance, as we will show afterwards, this can be achieved, if the angle between $\boldsymbol{w}_i-\boldsymbol{w}_j$ and $\boldsymbol{w}^*_i-\boldsymbol{w}^*_j$ is small for all $1\leq i<j\leq k$. Since $\boldsymbol{v}_{ij}$ is our only proxy for $\boldsymbol{w}^*_i-\boldsymbol{w}^*_j$, a natural approach is trying to ensure that the angle between $\boldsymbol{w}_i-\boldsymbol{w}_j$ and $\boldsymbol{v}_{ij}$ is small or, equivalently $\left\langle\boldsymbol{w}_i-\boldsymbol{w}_j,\boldsymbol{v}_{ij}\right\rangle\approx\left\|\boldsymbol{w}_i-\boldsymbol{w}_j\right\|_2$ for all $1\leq i<j\leq k$. The latter desideratum can be formalized through the second order conic constraint $\left\langle\boldsymbol{w}_i-\boldsymbol{w}_j,\boldsymbol{v}_{ij}\right\rangle\geq(1-\phi)\left\|\boldsymbol{w}_i-\boldsymbol{w}_j\right\|_2$ for $\phi\approx 0$. Additionally, it must hold that $\boldsymbol{w}_i\neq\boldsymbol{w}_j$, in accordance to the fact that $\boldsymbol{w}^*_i\neq\boldsymbol{w}^*_j$ for all $1\leq i<j\leq k$. Notice that, had there been a solution with $\boldsymbol{w}_i=\boldsymbol{w}_j$ for some $i\neq j$, it would yield a constant value for $\mathbf{Pr}_{\boldsymbol{x}\sim\mathcal{D}_{\boldsymbol{x}}}\left[\mathrm{sign}\left(\left\langle\boldsymbol{w}_i-\boldsymbol{w}_j,\boldsymbol{x}\right\rangle\right)\neq\mathrm{sign}\left(\left\langle\boldsymbol{w}^*_i-\boldsymbol{w}^*_j,\boldsymbol{x}\right\rangle\right)\right]$ and, consequently, we would be unable to make the expected loss arbitrarily small. The condition $\boldsymbol{w}_i\neq\boldsymbol{w}_j$ can be achieved by imposing the linear constraint $\left\langle\boldsymbol{w}_i-\boldsymbol{w}_j,\boldsymbol{v}_{ij}\right\rangle\geq\xi$ for some $\xi>0$.

The above facts lead us to the formulation of the following convex (second-order cone) program, which we denote by $\mathsf{CP1}$[3]:

$$\begin{aligned}
\text{Find}\quad & \boldsymbol{W}\in\mathbb{R}^{k\times d}\\
\text{subject to}\quad & \left\langle\boldsymbol{w}_i-\boldsymbol{w}_j,\boldsymbol{v}_{ij}\right\rangle\geq\max\left\{(1-\phi)\left\|\boldsymbol{w}_i-\boldsymbol{w}_j\right\|_2,\xi\right\}\quad\forall\,1\leq i<j\leq k\\
& \left\|\boldsymbol{W}\right\|_{\mathrm{F}}\leq 1
\end{aligned}$$

In the above formulation, the constants $\phi\in[0,1)$ and $\xi>0$ are to be determined. The additional constraint $\left\|\boldsymbol{W}\right\|_{\mathrm{F}}\leq 1$ bounds the feasible region of the convex program, which contributes into rendering the computation of $\boldsymbol{W}$ efficient, as we will later analyze more rigorously. Moreover, it is crucial that this constraint does not conflict with the previous constraints, since LSFs are invariant to multiplying their underlying matrix by a constant.

Having provided the intuition behind the formulation of the aforementioned convex program, we proceed by proving that it is indeed feasible, that any solution of it achieves arbitrarily small loss in

[2] $\mathcal{F}^d_{\mathrm{LC}}$ denotes the class of isotropic log-concave distributions on $\mathbb{R}^d$

[3] $\mathsf{CP1}$ can be typically formulated by arranging the unknown variables in a kd-dimensional vector.

Algorithm 11 Properly Learning Homogeneous LSFs with Noise

Input: Training set $S \subset \mathbb{R}^d \times \mathbb{S}_k$, proper homogeneous LTF learner A, $\phi \in (0,1)$ and $\xi > 0$
Output: Homogeneous LSF $\sigma_{\boldsymbol{W}} \colon \mathbb{R}^d \to \mathbb{S}_k$
1: $\left(h_{\boldsymbol{v}_{ij}}\right)_{1 \le i < j \le k} \leftarrow$ PairwiseDecomposition(S, A)
2: Construct $\mathsf{CP1}$ using ϕ, ξ and $(\boldsymbol{v}_{ij})_{1 \le i < j \le k}$
3: $\boldsymbol{W} \leftarrow$ Ellipsoid$(\mathsf{CP1})$ ▷ See Appendix B
4: **return** $\sigma_{\boldsymbol{W}}$

expectation and that it can be efficiently solved. We will need the following auxiliary lemma, whose proof can be found in Appendix C.

Lemma 5.2.1. *Let $\boldsymbol{U} \in \mathbb{R}^{k \times d}$, where $\boldsymbol{u}_i \neq \boldsymbol{u}_j$ for all $i \neq j$, and $\mathcal{D}_{\boldsymbol{x}}$ be an isotropic log-concave distribution on $\mathbb{R}^d$. There exists a polynomial $Q \colon \mathbb{R}^3 \to \mathbb{R}$ such that the following holds. For any $\epsilon > 0$, there exists a matrix $\boldsymbol{V} \in \mathbb{R}^{k \times d}$ with the properties:*

1. $\max_{1 \le i < j \le k} \mathbf{Pr}_{\boldsymbol{x} \sim \mathcal{D}_{\boldsymbol{x}}} \left[\operatorname{sign}\left(\langle \boldsymbol{u}_i - \boldsymbol{u}_j, \boldsymbol{x} \rangle\right) \neq \operatorname{sign}\left(\langle \boldsymbol{v}_i - \boldsymbol{v}_j, \boldsymbol{x} \rangle\right)\right] \le \epsilon$
2. $\min_{1 \le i < j \le k} \left\|\boldsymbol{v}_i - \boldsymbol{v}_j\right\|_2 \ge 2^{-Q(d,k,1/\epsilon)}$
3. $\left\|\boldsymbol{V}\right\|_{\mathrm{F}} \le 1$

The main idea of the above lemma is that for any matrix $\boldsymbol{U}$ with unequal rows, there exists a matrix $\boldsymbol{V}$ within the unit ball (with respect to the Frobenius norm), yielding an expected loss close to that of $\boldsymbol{U}$, while enjoying the auxiliary property that the norm of the difference between any pair of its rows is sufficiently large. This property plays a decisive role in the capability of rendering the solution of $\mathsf{CP1}$ by means of the Ellipsoid method efficient, as we will later explain. We now state and prove our main lemma.

Lemma 5.2.2. *There exist universal constants $\alpha, \beta > 0$ and a polynomial $P \colon \mathbb{R}^3 \to \mathbb{R}$ such that the following holds. For any $\epsilon > 0$, if $\max_{1 \le i < j \le k} L_{\mathcal{D}_{\boldsymbol{x}}, h_{\boldsymbol{w}_i^* - \boldsymbol{w}_j^*}, \ell_{0-1}}(h_{\boldsymbol{v}_{ij}}) \le \epsilon$, $\phi = \alpha\epsilon^2$ and $\xi = 2^{-P(d,k,1/\epsilon)}$, then the following properties hold.*

1. $\mathsf{CP1}$ *is feasible.*
2. *For any solution $\boldsymbol{W}$ of $\mathsf{CP1}$, it holds that $L_{\mathcal{D}_{\boldsymbol{x}}, \sigma_{\boldsymbol{W}^*}, d_\tau}(\sigma_{\boldsymbol{W}}) \le \beta\binom{k}{2}\epsilon$. If, additionally, $\mathcal{D}_{\boldsymbol{x}} = \mathcal{N}_d$, it holds that $L_{\mathcal{N}_d, \sigma_{\boldsymbol{W}^*}, d_{\mathrm{top}-r}}(\sigma_{\boldsymbol{W}}) \le \beta k r \sqrt{\log(kr)}\epsilon$ for all $r \in [k]$.*
3. *The feasible set of $\mathsf{CP1}$ contains a ball of radius $2^{-P(d,k,1/\epsilon)}$ and is contained in a ball of radius 1. Both balls are with respect to the Frobenius norm.*
4. $\mathsf{CP1}$ *can be solved in* $\mathrm{poly}(d, k, 1/\epsilon)$ *time using the Ellipsoid method.*

Proof. By Lemma A.0.4, the exist universal constants $c_1, c_2 > 0$ for which it holds

$$c_1^{-1}\theta(\boldsymbol{u}, \boldsymbol{v}) \le \mathop{\mathbf{Pr}}_{\boldsymbol{x} \sim \mathcal{D}_{\boldsymbol{x}}} \left[\operatorname{sign}\left(\langle \boldsymbol{u}, \boldsymbol{x} \rangle\right) \neq \operatorname{sign}\left(\langle \boldsymbol{v}, \boldsymbol{x} \rangle\right)\right] \le c_2\theta(\boldsymbol{u}, \boldsymbol{v})$$

for any $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^d \setminus \{\boldsymbol{0}\}$. Moreover, by Lemma 5.2.1, there exists a polynomial $Q \colon \mathbb{R}^3 \to \mathbb{R}$ such that the following holds. For any $\epsilon > 0$, there exists a matrix $\widetilde{\boldsymbol{W}}^* \in \mathbb{R}^{k \times d}$ with the properties:

1. $\max_{1 \le i < j \le k} \mathbf{Pr}_{\boldsymbol{x} \sim \mathcal{D}_{\boldsymbol{x}}} \left[\operatorname{sign}\left(\left\langle \boldsymbol{w}_i^* - \boldsymbol{w}_j^*, \boldsymbol{x} \right\rangle\right) \neq \operatorname{sign}\left(\left\langle \widetilde{\boldsymbol{w}}_i^* - \widetilde{\boldsymbol{w}}_j^*, \boldsymbol{x} \right\rangle\right)\right] \le \epsilon$
2. $\min_{1 \le i < j \le k} \left\|\widetilde{\boldsymbol{w}}_i^* - \widetilde{\boldsymbol{w}}_j^*\right\|_2 \ge 2^{-Q(d,k,1/\epsilon)}$
3. $\left\|\widetilde{\boldsymbol{W}}^*\right\|_{\mathrm{F}} \le 1$

Fix an arbitrary real constant $\gamma > 1$ and let $0 < \epsilon < 1/\left(c_1\sqrt{2\gamma}\right)$. We will show that by choosing

$$\phi^* = 2\gamma c_1^2 \epsilon^2$$
$$\xi^* = \frac{1 - 2\gamma c_1^2 \epsilon^2}{1 + (\gamma - 1) c_1^2 \epsilon^2} 2^{-Q(d,k,1/\epsilon)}$$

properties 1, 2, 3 and 4 are satisfied.

Proof of property 1 We will show that for $\phi = \phi^*$ and $\xi = \xi^*$, the matrix $\widetilde{\boldsymbol{W}}^*$ matrix belongs to the feasible region of the convex program. It holds that

$$\max_{1\leq i<j\leq k} \mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{D}_{\boldsymbol{x}}} \left[\operatorname{sign}\left(\left\langle \widetilde{\boldsymbol{w}}_i^* - \widetilde{\boldsymbol{w}}_j^*, \boldsymbol{x}\right\rangle\right) \neq \operatorname{sign}\left(\left\langle \boldsymbol{w}_i^* - \boldsymbol{w}_j^*, \boldsymbol{x}\right\rangle\right)\right] \leq \epsilon$$

and

$$\max_{1\leq i<j\leq k} \mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{D}_{\boldsymbol{x}}} \left[\operatorname{sign}\left(\left\langle \boldsymbol{w}_i^* - \boldsymbol{w}_j^*, \boldsymbol{x}\right\rangle\right) \neq \operatorname{sign}\left(\left\langle \boldsymbol{v}_{ij}, \boldsymbol{x}\right\rangle\right)\right] \leq \epsilon\,.$$

Therefore, using the triangle inequality, we get that

$$\max_{1\leq i<j\leq k} \mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{D}_{\boldsymbol{x}}} \left[\operatorname{sign}\left(\left\langle \widetilde{\boldsymbol{w}}_i^* - \widetilde{\boldsymbol{w}}_j^*, \boldsymbol{x}\right\rangle\right) \neq \operatorname{sign}\left(\left\langle \boldsymbol{v}_{ij}, \boldsymbol{x}\right\rangle\right)\right] \leq 2\epsilon \implies \max_{1\leq i<j\leq k} \theta\left(\widetilde{\boldsymbol{w}}_i^* - \widetilde{\boldsymbol{w}}_j^*, \boldsymbol{v}_{ij}\right) \leq 2c_1\epsilon\,.$$

Using the fact that $\cos(x) \geq 1 - x^2/2$ for all $x \in \mathbb{R}$, we get that

$$\cos\left(\theta\left(\widetilde{\boldsymbol{w}}_i^* - \widetilde{\boldsymbol{w}}_j^*, \boldsymbol{v}_{ij}\right)\right) \geq 1 - \frac{\theta\left(\widetilde{\boldsymbol{w}}_i^* - \widetilde{\boldsymbol{w}}_j^*, \boldsymbol{v}_{ij}\right)^2}{2} \geq 1 - 2c_1^2\epsilon^2\,,$$

which implies that

$$\left\langle \widetilde{\boldsymbol{w}}_i^* - \widetilde{\boldsymbol{w}}_j^*, \boldsymbol{v}_{ij}\right\rangle \geq \left(1 - 2c_1^2\epsilon^2\right)\left\|\widetilde{\boldsymbol{w}}_i^* - \widetilde{\boldsymbol{w}}_j^*\right\|_2 \geq \left(1 - 2c_1^2\epsilon^2\right)2^{-Q(d,k,1/\epsilon)}$$

for all $1 \leq i < j \leq k$. Therefore, for $0 < \epsilon < 1/\left(c_1\sqrt{2}\right)$, if we set $\phi_{\min}(\epsilon) \leq \phi < 1$ and $0 < \xi \leq (1 - \phi_{\min}(\epsilon))\,2^{-Q(d,k,1/\epsilon)}$, where $\phi_{\min}(\epsilon) = 2c_1^2\epsilon^2$, then the "well-conditioned" matrix $\widetilde{\boldsymbol{W}}^*$ satisfies every constraint of the convex program, so it belongs to its feasible region. These constraints for ϕ and ξ, are satisfied by ϕ^* and ξ^*, which concludes the proof.

Proof of property 2 Using the fact that $1 - x \geq \cos\left(\pi\sqrt{x}/2\right)$ for all $0 \leq x \leq 1$, we get that for $0 \leq \phi < 1$ and $\xi > 0$, any solution $\boldsymbol{W}$ of the convex program satisfies that $\boldsymbol{w}_i \neq \boldsymbol{w}_j$ and

$$\begin{aligned}
\left\langle \boldsymbol{w}_i - \boldsymbol{w}_j, \boldsymbol{v}_{ij}\right\rangle &\geq (1-\phi)\left\|\boldsymbol{w}_i - \boldsymbol{w}_j\right\|_2 \implies \\
\cos\left(\theta(\boldsymbol{w}_i - \boldsymbol{w}_j, \boldsymbol{v}_{ij})\right) &\geq 1 - \phi \implies \\
\cos\left(\theta(\boldsymbol{w}_i - \boldsymbol{w}_j, \boldsymbol{v}_{ij})\right) &\geq \cos\left(\frac{\pi\sqrt{\phi}}{2}\right) \implies \\
\theta\left(\boldsymbol{w}_i - \boldsymbol{w}_j, \boldsymbol{v}_{ij}\right) &\leq \frac{\pi\sqrt{\phi}}{2} \implies \\
\mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{D}_{\boldsymbol{x}}}\left[\operatorname{sign}\left(\left\langle \boldsymbol{w}_i - \boldsymbol{w}_j, \boldsymbol{x}\right\rangle\right) \neq \operatorname{sign}\left(\left\langle \boldsymbol{v}_{ij}, \boldsymbol{x}\right\rangle\right)\right] &\leq \frac{c_2\pi\sqrt{\phi}}{2}
\end{aligned}$$

for all $1 \leq i < j \leq k$. Combining the above with the fact that

$$\max_{1\leq i<j\leq k} \mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{D}_{\boldsymbol{x}}} \left[\operatorname{sign}\left(\left\langle \boldsymbol{w}_i^* - \boldsymbol{w}_j^*, \boldsymbol{x}\right\rangle\right) \neq \operatorname{sign}\left(\left\langle \boldsymbol{v}_{ij}, \boldsymbol{x}\right\rangle\right)\right] \leq \epsilon\,,$$

we get (using the triangle inequality) that

$$\max_{1\leq i<j\leq k} \mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{D}_{\boldsymbol{x}}} \left[\operatorname{sign}\left(\left\langle \boldsymbol{w}_i - \boldsymbol{w}_j, \boldsymbol{x}\right\rangle\right) \neq \operatorname{sign}\left(\left\langle \boldsymbol{w}_i^* - \boldsymbol{w}_j^*, \boldsymbol{x}\right\rangle\right)\right] \leq \epsilon + \frac{c_2\pi\sqrt{\phi}}{2}\,.$$

Therefore, it holds that

$$\begin{aligned}
\mathop{\mathbf{E}}_{\boldsymbol{x}\sim\mathcal{D}_{\boldsymbol{x}}}\left[d_\tau\left(\sigma_{\boldsymbol{W}}(\boldsymbol{x}), \sigma_{\boldsymbol{W}^*}(\boldsymbol{x})\right)\right] &= \sum_{i=1}^{k-1}\sum_{j=i+1}^{k} \mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{D}_{\boldsymbol{x}}}\left[\operatorname{sign}\left(\left\langle \boldsymbol{w}_i - \boldsymbol{w}_j, \boldsymbol{x}\right\rangle\right) \neq \operatorname{sign}\left(\left\langle \boldsymbol{w}_i^* - \boldsymbol{w}_j^*, \boldsymbol{x}\right\rangle\right)\right] \\
&\leq \binom{k}{2} \max_{1\leq i<j\leq k} \mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{D}_{\boldsymbol{x}}}\left[\operatorname{sign}\left(\left\langle \boldsymbol{w}_i - \boldsymbol{w}_j, \boldsymbol{x}\right\rangle\right) \neq \operatorname{sign}\left(\left\langle \boldsymbol{w}_i^* - \boldsymbol{w}_j^*, \boldsymbol{x}\right\rangle\right)\right] \\
&\leq \binom{k}{2}\left(\epsilon + \frac{c_2\pi\sqrt{\phi}}{2}\right)\,.
\end{aligned}$$

Lemma 5.2.3 (Fotakis et al. [2022b]). *There exists a universal constant $c > 0$ such that*

$$\mathop{\mathbf{E}}_{\boldsymbol{x}\sim\mathcal{N}_d}\left[d_{\text{top}-r}\left(\sigma_{\boldsymbol{U}}(\boldsymbol{x}),\sigma_{\boldsymbol{V}}(\boldsymbol{x})\right)\right] \le ckr\sqrt{\log(kr)}\max_{1\le i<j\le k}\mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{N}_d}\left[\text{sign}\left(\langle \boldsymbol{u}_i-\boldsymbol{u}_j,\boldsymbol{x}\rangle\right)\neq\text{sign}\left(\langle \boldsymbol{v}_i-\boldsymbol{v}_j,\boldsymbol{x}\rangle\right)\right]$$

for any $r\in[k]$ and $\boldsymbol{U},\boldsymbol{V}\in\mathbb{R}^{k\times d}$.

Additionally, if $\mathcal{D}_{\boldsymbol{x}}=\mathcal{N}_d$, Lemma 5.2.3 implies that

$$\mathop{\mathbf{E}}_{\boldsymbol{x}\sim\mathcal{N}_d}\left[d_{\text{top}-r}\left(\sigma_{\boldsymbol{W}}(\boldsymbol{x}),\sigma_{\boldsymbol{W}^*}(\boldsymbol{x})\right)\right] \le ckr\sqrt{\log(kr)}\left(\epsilon+\frac{c_2\pi\sqrt{\phi}}{2}\right)$$

for any $r\in[k]$, where c is the universal constant of Lemma 5.2.3. Choosing $\phi=\phi^*$ and $\xi=\xi^*$, we get the desired results.

Proof of property 3 The constraint $\|\boldsymbol{W}\|_{\text{F}}\le 1$ directly implies that the feasible region of the convex program is contained in a ball of radius 1. We will now show that the feasible set of the convex program contains a ball of radius $r=2^{-\text{poly}(d,k,1/\epsilon)}$ centered at the aforementioned "well-conditioned" matrix $\widetilde{\boldsymbol{W}}^*$. Namely, we will show that for $\phi=\phi^*$ and $\xi=\xi^*$, if $\left\|\boldsymbol{W}-\widetilde{\boldsymbol{W}}^*\right\|_{\text{F}}\le r$ for a sufficiently small radius r and of order $2^{-\text{poly}(d,k,1/\epsilon)}$, then $\boldsymbol{W}$ belongs to the feasible region of the convex program.

Initially, suppose that $\left\|\boldsymbol{W}-\widetilde{\boldsymbol{W}}^*\right\|_{\text{F}}\le r$, that is, $\boldsymbol{W}$ belongs to the ball (with respect to the Frobenius norm) of radius r centered at $\widetilde{\boldsymbol{W}}^*$. This implies that $\|\boldsymbol{w}_i-\widetilde{\boldsymbol{w}}_i^*\|_2\le r$ for all $i\in[k]$. Moreover, we have that

$$\begin{aligned}\left\langle \widetilde{\boldsymbol{w}}_i^*-\widetilde{\boldsymbol{w}}_j^*,\boldsymbol{v}_{ij}\right\rangle &= \left\langle \widetilde{\boldsymbol{w}}_i^*-\boldsymbol{w}_i,\boldsymbol{v}_{ij}\right\rangle+\left\langle \boldsymbol{w}_j-\widetilde{\boldsymbol{w}}_j^*,\boldsymbol{v}_{ij}\right\rangle+\left\langle \boldsymbol{w}_i-\boldsymbol{w}_j,\boldsymbol{v}_{ij}\right\rangle\\ &\le \|\boldsymbol{w}_i-\widetilde{\boldsymbol{w}}_i^*\|_2+\left\|\boldsymbol{w}_j-\widetilde{\boldsymbol{w}}_j^*\right\|_2+\left\langle \boldsymbol{w}_i-\boldsymbol{w}_j,\boldsymbol{v}_{ij}\right\rangle\\ &\le \left\langle \boldsymbol{w}_i-\boldsymbol{w}_j,\boldsymbol{v}_{ij}\right\rangle+2r\,,\end{aligned}$$

$$\begin{aligned}\|\boldsymbol{w}_i-\boldsymbol{w}_j\|_2 &= \left\|\boldsymbol{w}_i-\widetilde{\boldsymbol{w}}_i^*+\widetilde{\boldsymbol{w}}_i^*-\widetilde{\boldsymbol{w}}_j^*+\widetilde{\boldsymbol{w}}_j^*-\boldsymbol{w}_j\right\|_2\\ &\le \|\boldsymbol{w}_i-\widetilde{\boldsymbol{w}}_i^*\|_2+\left\|\widetilde{\boldsymbol{w}}_i^*-\widetilde{\boldsymbol{w}}_j^*\right\|_2+\le\left\|\boldsymbol{w}_j-\widetilde{\boldsymbol{w}}_j^*\right\|_2\\ &\le \left\|\widetilde{\boldsymbol{w}}_i^*-\widetilde{\boldsymbol{w}}_j^*\right\|_2+2r\end{aligned}$$

and, similarly, $\left\|\widetilde{\boldsymbol{w}}_i^*-\widetilde{\boldsymbol{w}}_j^*\right\|_2\le\|\boldsymbol{w}_i-\boldsymbol{w}_j\|_2+2r$ for all $1\le i<j\le k$. For $0<\epsilon<1/\left(c_1\sqrt{2}\right)$ and $\phi_{\min}(\epsilon)\le\phi<1$, we get that any $\boldsymbol{W}$ in the above ball satisfies that

$$\begin{aligned}\left\langle \boldsymbol{w}_i-\boldsymbol{w}_j,\boldsymbol{v}_{ij}\right\rangle &\ge \left\langle \widetilde{\boldsymbol{w}}_i^*-\widetilde{\boldsymbol{w}}_j^*,\boldsymbol{v}_{ij}\right\rangle-2r\\ &\ge (1-\phi)\left\|\widetilde{\boldsymbol{w}}_i^*-\widetilde{\boldsymbol{w}}_j^*\right\|_2-2r\\ &\ge (1-\phi)\left(\|\boldsymbol{w}_i-\boldsymbol{w}_j\|_2-2r\right)-2r\\ &\ge (1-\phi)\|\boldsymbol{w}_i-\boldsymbol{w}_j\|_2-4r\end{aligned}$$

for all $1\le i<j\le k$. Hence, we get that for any $\phi_{\min}(\epsilon)<\phi<1$

$$\begin{aligned}\left\langle \boldsymbol{w}_i-\boldsymbol{w}_j,\boldsymbol{v}_{ij}\right\rangle &\ge (1-\phi_{\min}(\epsilon))\|\boldsymbol{w}_i-\boldsymbol{w}_j\|_2-4r\\ &\ge (1-\phi)\|\boldsymbol{w}_i-\boldsymbol{w}_j\|_2+(\phi-\phi_{\min}(\epsilon))\|\boldsymbol{w}_i-\boldsymbol{w}_j\|_2-4r\\ &\ge (1-\phi)\|\boldsymbol{w}_i-\boldsymbol{w}_j\|_2+(\phi-\phi_{\min}(\epsilon))\left(\left\|\widetilde{\boldsymbol{w}}_i^*-\widetilde{\boldsymbol{w}}_j^*\right\|_2-2r\right)-4r\\ &\ge (1-\phi)\|\boldsymbol{w}_i-\boldsymbol{w}_j\|_2+(\phi-\phi_{\min}(\epsilon))\left(2^{-Q(d,k,1/\epsilon)}-2r\right)-4r\end{aligned}$$

for all $1\le i<j\le k$. Setting

$$r\le\frac{\phi-\phi_{\min}(\epsilon)}{4+2\left(\phi-\phi_{\min}(\epsilon)\right)}2^{-Q(d,k,1/\epsilon)}$$

the right hand side term $(\phi-\phi_{\min}(\epsilon))\left(2^{-Q(d,k,1/\epsilon)}-2r\right)-4r$ becomes nonnegative, implying that

$$\left\langle \boldsymbol{w}_i-\boldsymbol{w}_j,\boldsymbol{v}_{ij}\right\rangle\ge(1-\phi)\|\boldsymbol{w}_i-\boldsymbol{w}_j\|_2\ge\frac{2(1-\phi)}{2+\phi-\phi_{\min}(\epsilon)}2^{-Q(d,k,1/\epsilon)}\,.$$

Therefore, choosing $\phi = \phi^*$ and $\xi = \xi^*$, we get that $\boldsymbol{W}$ belongs to the feasible region of the convex program. Hence, the feasible region contains a ball of radius

$$r = \frac{\phi - \phi_{\min}(\epsilon)}{4 + 2\left(\phi - \phi_{\min}(\epsilon)\right)} 2^{-Q(d,k,1/\epsilon)} \in \Theta\left(2^{-\mathrm{poly}(d,k,1/\epsilon)}\right) ,$$

which concludes the proof.

Proof of property 4 Combining Lemma C.0.2 with property 3 implies that the Ellipsoid method will return a point within the region of our convex program in $\mathrm{poly}(d, k, 1/\epsilon)$ time[4].

□

The desired results on the learnability of LSFs are a direct consequence of Lemma 5.2.2 and are stated below.

Theorem 5.2.1. *Fix any $\eta \in [0, 1/2)$, $\epsilon, \delta > 0$ and $\boldsymbol{W}^* \in \mathbb{R}^{k\times d}$, where $\boldsymbol{w}_i^* \neq \boldsymbol{w}_j^*$ for all $1 \leq i < j \leq k$. Let $\mathcal{D}$ be any $(\eta, \sigma_{\boldsymbol{W}^*})$-LR distribution with Massart noise such that $\mathcal{D}_{\boldsymbol{x}} \in \mathcal{F}_{\mathrm{LC}}^d$. Let A be any proper PAC learner for $\mathcal{H}_{\mathrm{HLTF}}^d$ in the presence of Massart noise with respect to $\mathcal{F}_{\mathrm{LC}}^d$ that has polynomial runtime in d and in the number of training samples. Algorithm 11 has the following performance guarantee: If given as input A, $\phi \in \Theta(\epsilon^2)$, $\xi = 2^{-\mathrm{poly}(d,k,1/\epsilon)}$ and $m \geq m_{A,\mathcal{H}_{\mathrm{LTF}}^d,\mathcal{F}_{\mathrm{LC}}^d}^{\mathrm{Massart}}\left(\epsilon/O(k^2), \delta/\binom{k}{2}, \eta\right)$ i.i.d. samples drawn from $\mathcal{D}$, it runs in $\mathrm{poly}(m)$ time and outputs a matrix $\boldsymbol{W} \in \mathbb{R}^{k\times d}$ such that, with probability at least $1 - \delta$, it holds $L_{\mathcal{D}_{\boldsymbol{x}},\sigma_{\boldsymbol{W}^*},d_\tau}(\sigma_{\boldsymbol{W}}) \leq \epsilon$.*

Proof. By definition of A, we get that for all $1 \leq i < j \leq k$, if A is given $m \geq m_{A,\mathcal{H}_{\mathrm{LTF}}^d,\mathcal{F}_{\mathrm{LC}}^d}^{\mathrm{Massart}}(\epsilon, \delta, \eta)$ i.i.d. samples from $\mathcal{D}_{ij}$, it returns a unit vector $\boldsymbol{v}_{ij}$ such that, with probability at least $1 - \delta$, it holds

$$\Pr_{\boldsymbol{x}\sim\mathcal{D}_{\boldsymbol{x}}}\left[\mathrm{sign}\left(\langle \boldsymbol{w}_i^* - \boldsymbol{w}_j^*, \boldsymbol{x}\rangle\right) \neq \mathrm{sign}\left(\langle \boldsymbol{v}_{ij}, \boldsymbol{x}\rangle\right)\right] \leq \epsilon .$$

Then, from the union bound, we get that, with probability at least $1 - \delta\binom{k}{2}$, it holds

$$\max_{1\leq i<j\leq k} \Pr_{\boldsymbol{x}\sim\mathcal{D}_{\boldsymbol{x}}}\left[\mathrm{sign}\left(\langle \boldsymbol{w}_i^* - \boldsymbol{w}_j^*, \boldsymbol{x}\rangle\right) \neq \mathrm{sign}\left(\langle \boldsymbol{v}_{ij}, \boldsymbol{x}\rangle\right)\right] \leq \epsilon .$$

Therefore, Lemma 5.2.2, implies that Algorithm 11 outputs a matrix $\boldsymbol{W} \in \mathbb{R}^{k\times d}$ such that, with probability at least $1 - \delta\binom{k}{2}$, it holds

$$L_{\mathcal{D}_{\boldsymbol{x}},\sigma_{\boldsymbol{W}^*},d_\tau}(\sigma_{\boldsymbol{W}}) \leq \beta\binom{k}{2}\epsilon$$

for some universal constant $\beta > 0$. This concludes the proof. □

Theorem 5.2.2. *Fix any $\alpha \in [0, 1), B \geq 1$, $\epsilon, \delta > 0$ and $\boldsymbol{W}^* \in \mathbb{R}^{k\times d}$, where $\boldsymbol{w}_i^* \neq \boldsymbol{w}_j^*$ for all $1 \leq i < j \leq k$. Let $\mathcal{D}$ be any $(\alpha, B, \sigma_{\boldsymbol{W}^*})$-LR distribution with Tsybakov noise such that $\mathcal{D}_{\boldsymbol{x}} \in \mathcal{F}_{\mathrm{LC}}^d$. Let A be any proper PAC learner for $\mathcal{H}_{\mathrm{HLTF}}^d$ in the presence of Tsybakov noise with respect to $\mathcal{F}_{\mathrm{LC}}^d$ that has polynomial runtime in d and in the number of training samples. Algorithm 11 has the following performance guarantee: If given as input A, $\phi \in \Theta(\epsilon^2)$, $\xi = 2^{-\mathrm{poly}(d,k,1/\epsilon)}$ and $m \geq m_{A,\mathcal{H}_{\mathrm{LTF}}^d,\mathcal{F}_{\mathrm{LC}}^d}^{\mathrm{Tsybakov}}\left(\epsilon/O(k^2), \delta/\binom{k}{2}, \alpha, B\right)$ i.i.d. samples drawn from $\mathcal{D}$, it runs in $\mathrm{poly}(m)$ time and outputs a matrix $\boldsymbol{W} \in \mathbb{R}^{k\times d}$ such that, with probability at least $1 - \delta$, it holds $L_{\mathcal{D}_{\boldsymbol{x}},\sigma_{\boldsymbol{W}^*},d_\tau}(\sigma_{\boldsymbol{W}}) \leq \epsilon$.*

Proof. The proof is nearly identical to that of Theorem 5.2.1. □

Finally, we cite two similar theorems, based on the work of Fotakis et al. [2022b], that yield a better sample complexity bound for the case of $d_{\mathrm{top}-r}$ with respect to the multivariate standard normal distribution.

Theorem 5.2.3 (Fotakis et al. [2022b])**.** *Fix any $\eta \in [0, 1/2)$, $\epsilon, \delta > 0$ and $\boldsymbol{W}^* \in \mathbb{R}^{k\times d}$, where $\boldsymbol{w}_i^* \neq \boldsymbol{w}_j^*$ for all $1 \leq i < j \leq k$. Let $\mathcal{D}$ be any $(\eta, \sigma_{\boldsymbol{W}^*})$-LR distribution with Massart noise such that $\mathcal{D}_{\boldsymbol{x}} = \mathcal{N}_d$. Let A be any proper PAC learner for $\mathcal{H}_{\mathrm{HLTF}}^d$ in the presence of Massart noise with respect to $\mathcal{N}_d$ that has polynomial runtime in d and in the number of training samples. Algorithm 11 has the following performance guarantee: If given as input A, $\phi \in \Theta(\epsilon^2)$, $\xi = 2^{-\mathrm{poly}(d,k,1/\epsilon)}$ and $m \geq m_{A,\mathcal{H}_{\mathrm{LTF}}^d,\mathcal{N}_d}^{\mathrm{Massart}}\left(\epsilon/O\left(kr\sqrt{\log(kr)}\right), \delta/\binom{k}{2}, \eta\right)$ i.i.d. samples drawn from $\mathcal{D}$, it runs in $\mathrm{poly}(m)$ time and outputs a matrix $\boldsymbol{W} \in \mathbb{R}^{k\times d}$ such that, with probability at least $1-\delta$, it holds $L_{\mathcal{N}_d,\sigma_{\boldsymbol{W}^*},d_{\mathrm{top}-r}}(\sigma_{\boldsymbol{W}}) \leq \epsilon$.*

[4] We also have to assume that there exists a separation oracle for CP1 with runtime $T_{\mathrm{sep}} = O(\mathrm{poly}(d, 1/\phi, \log(1/\xi)))$.

Proof. The proof is nearly identical to that of Theorem 5.2.1. □

Theorem 5.2.4 (Fotakis et al. [2022b])**.** *Fix any $\alpha \in [0,1), B \geq 1$, $\epsilon, \delta > 0$ and $\boldsymbol{W}^* \in \mathbb{R}^{k \times d}$, where $\boldsymbol{w}_i^* \neq \boldsymbol{w}_j^*$ for all $1 \leq i < j \leq k$. Let $\mathcal{D}$ be any $(\alpha, B, \sigma_{\boldsymbol{W}^*})$-LR distribution with Tsybakov noise such that $\mathcal{D}_{\boldsymbol{x}} = \mathcal{N}_d$. Let A be any proper PAC learner for $\mathcal{H}^d_{\mathrm{HLTF}}$ in the presence of Tsybakov noise with respect to $\mathcal{N}_d$ that has polynomial runtime in d and in the number of training samples. Algorithm 11 has the following performance guarantee: If given as input A, $\phi \in \Theta(\epsilon^2)$, $\xi = 2^{-\mathrm{poly}(d,k,1/\epsilon)}$ and $m \geq m^{\mathrm{Tsybakov}}_{A, \mathcal{H}^d_{\mathrm{LTF}}, \mathcal{N}_d}\left(\epsilon / O\left(kr\sqrt{\log(kr)}\right), \delta / \binom{k}{2}, \alpha, B\right)$ i.i.d. samples drawn from $\mathcal{D}$, it runs in $\mathrm{poly}(m)$ time and outputs a matrix $\boldsymbol{W} \in \mathbb{R}^{k \times d}$ such that, with probability at least $1-\delta$, it holds $L_{\mathcal{N}_d, \sigma_{\boldsymbol{W}^*}, d_{\mathrm{top}-r}}(\sigma_{\boldsymbol{W}}) \leq \epsilon$.*

Proof. The proof is nearly identical to that of Theorem 5.2.1. □

Chapter 6

Experimental Results

In this chapter, we carry out an experimental evaluation of label ranking algorithms, based on the pairwise and labelwise decomposition frameworks that were discussed in Chapter 4 and focusing on the case of complete rankings. Our goal is to find out how algorithms adjusted to the linear LR setting perform against some of the state-of-the-art general-purpose LR algorithms based on decision trees and random forests. To this end, we evaluate our algorithms both on synthetic data sets, whose construction is based on the linear setting, and on standard LR data sets beyond the linear setting. Additionally, in the case of synthetic data sets, we aim to acquire an understanding of how different noise types, based on the LR noise models of Chapter 4, affect the performance of our algorithms.

6.1 Label Ranking Algorithms under Comparison

We compare six algorithms, all based on the pairwise and labelwise decomposition methods of Chapter 4. The difference of these algorithms lies in the choice of the binary classification (for the pairwise case) or regression (for the labelwise case) algorithm used for each subproblem. In particular, for each of these two decomposition methods, we implement three algorithms: an algorithm based on random forests, an algorithm based on decision trees and algorithm based on linear predictors.

Before citing explicitly our algorithms, we provide a very brief background on decision trees, random forests and linear regression, which were used in a black box manner, and include references for further details.

Linear Regression *Linear regression* constitutes a common statistical tool for modeling the relationship between "explanatory" variables and some real valued outcome. Cast as a learning problem, the instance space is $\mathcal{X} \subseteq \mathbb{R}^d$ and the output space is $\mathcal{Y} = \mathbb{R}$. Our goal is to learn a linear function $h \colon \mathbb{R}^d \to \mathbb{R}$, where $h(\boldsymbol{x}) = \langle \boldsymbol{w}, \boldsymbol{x} \rangle + b$ with $\boldsymbol{w} \in \mathbb{R}^d$, $b \in \mathbb{R}$, that reflects the relationship between our task's variables in the best possible way, which is usually formalized through the requirement that squared loss ℓ_2 or absolute loss ℓ_1 on the training data is minimized. It can be easily shown an efficient ERM learner for the hypothesis class of linear regression predictors with respect to ℓ_2 (resp. ℓ_1) can be obtained through the least squares method (resp. linear programming) (Shalev-Shwartz and Ben-David [2022]). We emphasize that, since linear regression is not a binary prediction task, its sample complexity cannot be analyzed using the VC dimension and the acquirement of generalization bounds becomes more challenging.

Decision Trees and Random Forests A *decision tree* is a predictor that maps each instance to a label by following a decision path from a root node of a tree to a leaf. At each node of the aforementioned path, the successor child is chosen on the basis of a splitting of the input space. Usually, the splitting is based on a specific feature of the input feature vector or on a predefined set of splitting rules. The end of a decision path is a leaf that always contains a specific label, which constitutes the output prediction. It can be shown that decision trees of arbitrary size can lead to overfitting. A way to mitigate this phenomenon is through the use of *random forests*. A random forest is a predictor consisting of a collection of decision trees. The prediction of a random forest is acquired by a majority vote over the predictions of the individual trees. In general, random forests tend to outperform decision trees. For a detailed exposition of decision trees and random forests, we refer to Breiman et al. [1984], Breiman [2001], Shalev-Shwartz and Ben-David [2022].

Analytically, the six algorithms implemented and compared in our experimental evaluation are as follows.

- The pairwise decomposition method using the homogeneous halfspace learning algorithm of Dikonikolas et al. [2020a] as binary classification algorithm and KwikSort for aggregation of the pairwise predictions. We denote this algorithm by PWHH.
- The pairwise decomposition method, using the `scikit-learn` implementation for decision tree classification (`sklearn.tree.DecisionTreeClassifier`) as binary classification algorithm and KwikSort for aggregation of the pairwise predictions. We denote this algorithm by PWDT.
- The pairwise decomposition method, using the `scikit-learn` implementation for random forest classification (`sklearn.ensemble.RandomForestClassifier`) as binary classification algorithm and KwikSort for aggregation of the pairwise predictions. We denote this algorithm by PWRF.
- The labelwise decomposition method, using the `scikit-learn` implementation for linear regression (`sklearn.linear_model.LinearRegression`) as labels' position predictor. We denote this algorithm by LWLR.
- The labelwise decomposition method, using the `scikit-learn` implementation for decision tree regression (`sklearn.tree.DecisionTreeRegressor`) as labels' position predictor. We denote this algorithm by LWDT.
- The labelwise decomposition method, using the `scikit-learn` implementation for random forest regression (`sklearn.ensemble.RandomForestRegressor`) as labels' position predictor. We denote this algorithm by LWRF.

The implementation of the aforementioned algorithms was in Python and the code to reproduce our results is available here.

6.2 Results on Synthetic Data Sets

Synthetic data sets For the experiments, two instance sets were constructed, each consisting of 10000 feature vectors, drawn independently from the standard Gaussian distribution of dimension $d = 10$ and $d = 100$ respectively. We refer to the two sets as SFN (Small Features Number) and LFN (Small Features Number) respectively. For each feature size ($d = 10$ and $d = 100$), a random matrix $\boldsymbol{W}^* \in \mathbb{R}^{k \times d}$, associated with the target score function $\boldsymbol{m}^*(\boldsymbol{x}) = \boldsymbol{W}^* \boldsymbol{x}$ and the target homogeneous LSF $\sigma^* = \sigma_{\boldsymbol{W}^*} = \mathfrak{S} \circ \boldsymbol{m}^*$, was constructed. Moreover, we chose the number of labels to be $k = 5$.

As for the addition of noise in the data, we corrupt the noiseless rankings (as they are produced by $\boldsymbol{m}^*$) by means of the previously defined *label ranking distribution with Mallows noise* (Definition 4.3.4) and *label ranking distribution with additive noise* (Definition 4.3.12) models (using σ^* and $\boldsymbol{m}^*$ as target functions respectively). For the Mallows-based model, we consider only the KT distance and create 50 noisy versions of the original noiseless dataset (the one labeled by σ^*), each with a different value for the spread parameter ϕ. Similarly, for the additive noise model, we create 50 noisy versions of the original noiseless dataset such that the noise vector $\boldsymbol{\xi}$ added to the score vector $\boldsymbol{m}^*(\boldsymbol{x})$ is sampled from a zero mean k-dimensional Gaussian distribution with a different variance value for each noisy dataset.

To quantify the presence of noise in the training data in a common and comprehensible way for the noise models in consideration, we adopt the following notions of distortion used in the experimental evaluation of Fotakis et al. [2022a].

Definition 6.2.1. *A training set $S \subset \mathbb{R}^d \times \mathbb{S}_k$ satisfies:*

- *the α-inconsistency property, if $\frac{1}{|S|} \sum_{(\boldsymbol{x},\pi) \in S} \mathbb{1}\{\sigma^*(\boldsymbol{x}) \neq \pi\} = \alpha \in [0,1]$ and*
- *the β-KT gap property, if $\frac{1}{|S|} \sum_{(\boldsymbol{x},\pi) \in S} \tau(\sigma^*(\boldsymbol{x}), \pi) = \beta \in [-1,1]$.*

The α-inconsistency property is an indication of the probability that the observed ranking is different from the ground truth one. As mentioned in the previous chapter, the $0-1$ loss fails to capture the structured nature of rankings, in the sense that there might exist unequal rankings, yet very similar. In this regard, we use supplementally the β-KT gap property so as to obtain a more rounded view of the existence of noise in the training data. Obviously, the cases $\alpha = 0$ and $\beta = 1$ correspond to the noiseless

setting. Finally, the performance of the algorithms is measured in terms of mean KT coefficient over a noiseless test set (labeled by σ^*). Namely, for each test example we calculate the KT coefficient between the trained model's output ranking and the example's ranking, and, then, we take the mean over all test examples.

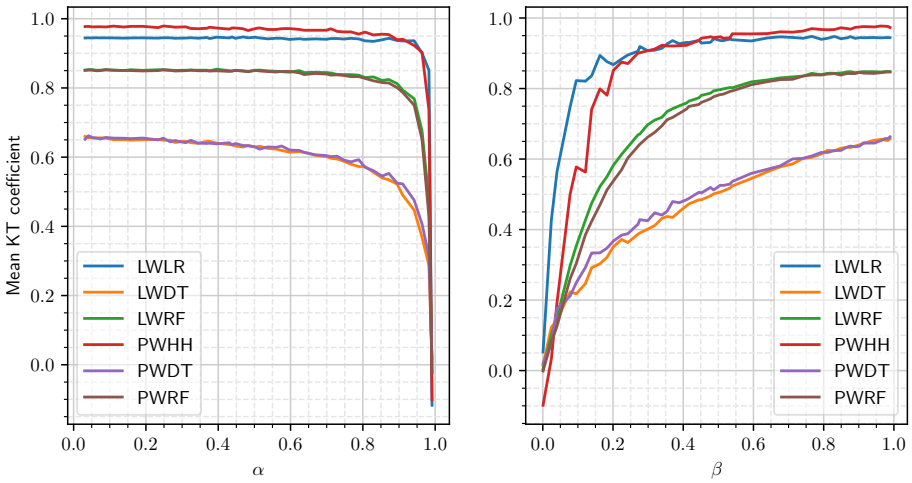


Figure 6.1: Evaluation in terms of mean KT coefficient on SFN datasets with Mallows noise

Concerning the results on SFN datasets with Mallows noise (Figure 6.1), we remark the following. First of all, we observe that PWHH and LWLR achieve the best performance among all algorithms, which is intuitively anticipated, due to them being customized to the linear nature of the data, as distinct from the four the tree-based algorithms. Moreover, the mean KT correlation for PWHH and LWLR manifests the lowest decay rate (up to a point), as the plot in terms of β indicates. The dominance of PWHH is in accordance with the fact that PWHH is the only algorithm to be supported by statistical guarantees in this specific experimental setting. Interestingly, though, as $\beta \to 0$, LWLR seems to outperform PWHH, in spite of not coming with any theoretical assurances. Among the tree-based algorithms, we can see that the random-forest-based algorithms (PWRF and LWRF) achieve a significantly higher performance than the decision-tree-based algorithms (PWDT and LWDT). Finally, it is remarkable that PWRF and LWRF (resp. PWDT and LWDT) achieve almost the same performance regardless of the decomposition method (pairwise or labelwise) used.

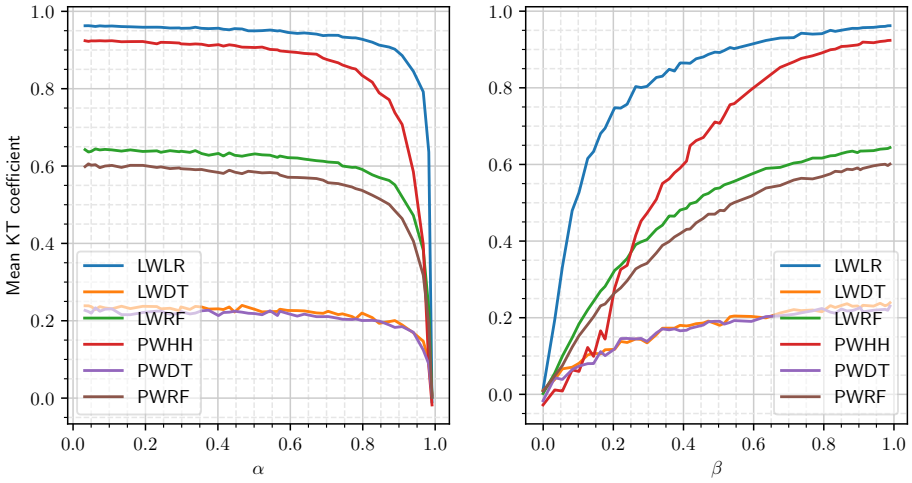


Figure 6.2: Evaluation in terms of mean KT coefficient on LFN datasets with Mallows noise

As for the results on LFN datasets with Mallows noise (Figure 6.2), a performance decay is observed

for all algorithms, with the four tree-based algorithms (PWRF, LWRF, PWDT and LWDT) experiencing the largest performance decay. Intuitively, this can be explained from the fact that the sample complexity is usually proportional to the dimensionality of the instances; the dimensionality in the LFN case is ten times larger than in the SFN case, while the number of samples remains constant. The best performance is achieved by LWLR for all values of α, β and its performance drop in comparison to the SFN case is negligible. Moreover, LWRF seems to be superior to PWRF, unlike the SFN case, where their performance difference is indiscernible.

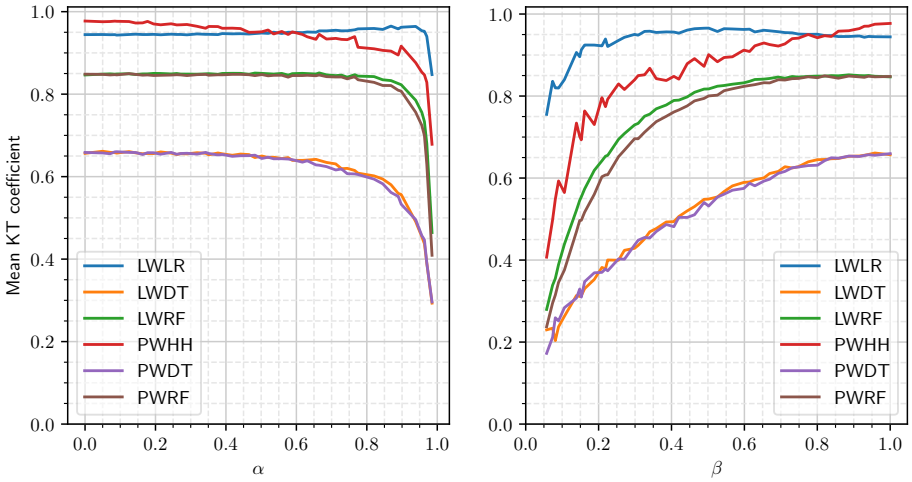


Figure 6.3: Evaluation in terms of mean KT coefficient on SFN datasets with Gaussian additive noise

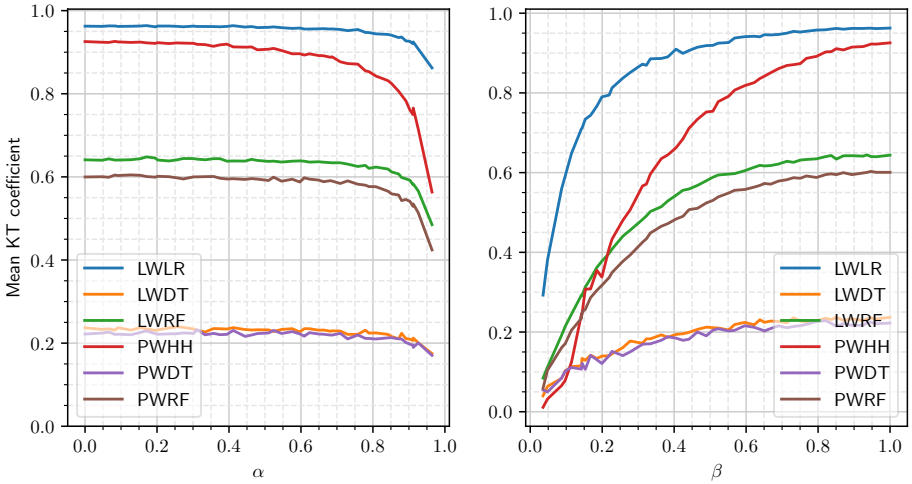


Figure 6.4: Evaluation in terms of mean KT coefficient on LFN datasets with Gaussian additive noise

Regarding the datasets with Gaussian additive noise (Figure 6.3 and Figure 6.4), we deduce roughly the same conclusions as in the case of Mallows noise. Namely, the way noise is added to the rankings seems to have a negligible effect on how the algorithms' performances are ranked.

Nevertheless, when it comes to examining the performance of each algorithm in an absolute manner, we observe that two data sets sharing the same value of α or β, but being corrupted by different types of noise, might yield a different mean KT correlation for the same algorithm. In particular, we observe that for a common noise rate in the training set, the performance is achieved by the majority of algorithms is slightly inferior in the Mallows case (see Figure 6.5, Figure 6.5, Figure 6.6, Figure 6.7, Figure 6.8, Figure 6.9 and Figure 6.10). The only exception is PWHH in the SFN case, where the opposite happens.

This indicates that neither α-inconsistency nor β-KT gap can be totally correlated to the algorithms' generalization capability, as observed by Fotakis et al. [2022a].

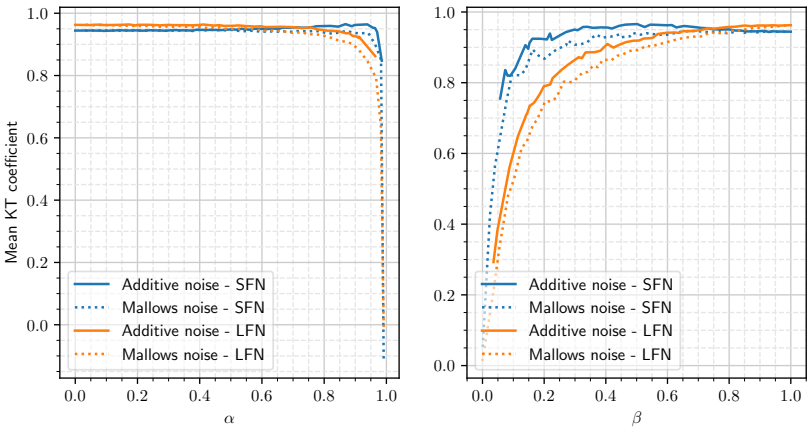


Figure 6.5: Evaluation of LWLR in terms of mean KT coefficient

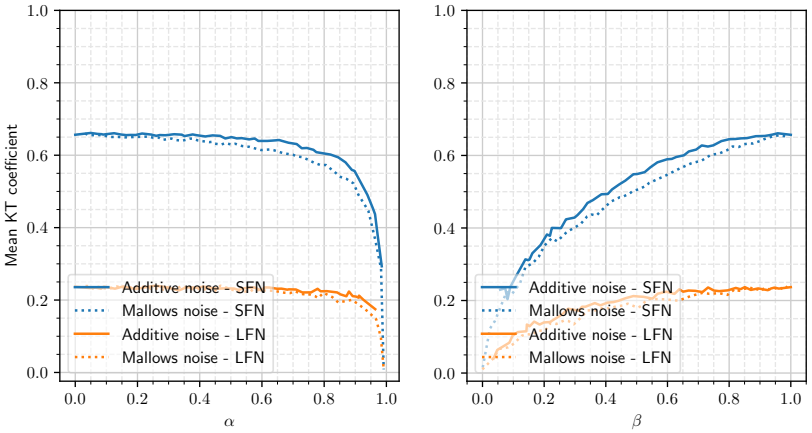


Figure 6.6: Evaluation of LWDT in terms of mean KT coefficient

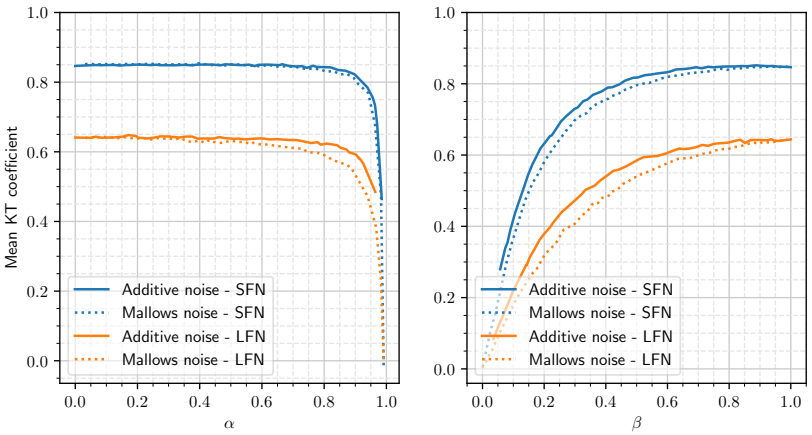


Figure 6.7: Evaluation of LWRF in terms of mean KT coefficient

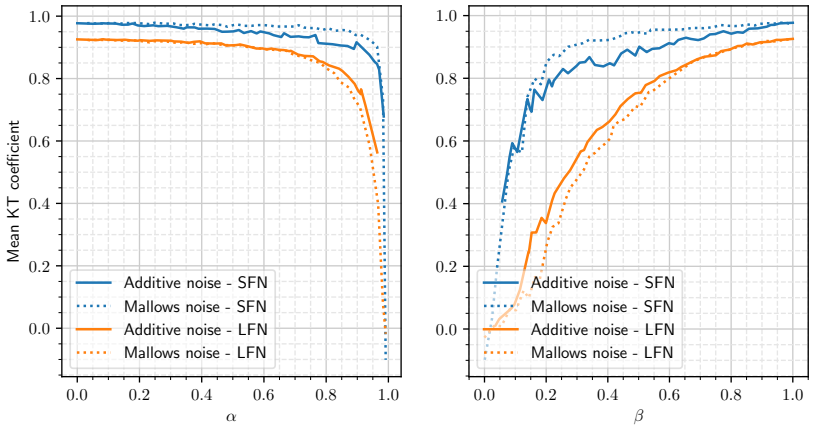


Figure 6.8: Evaluation of PWHH in terms of mean KT coefficient

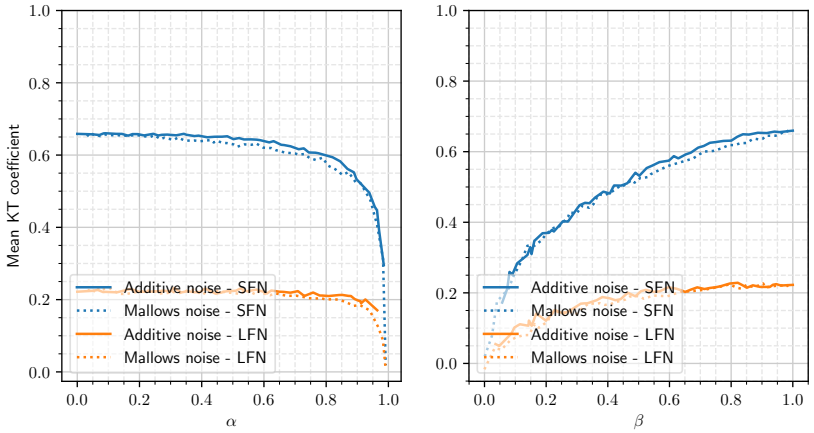


Figure 6.9: Evaluation of PWDT in terms of mean KT coefficient

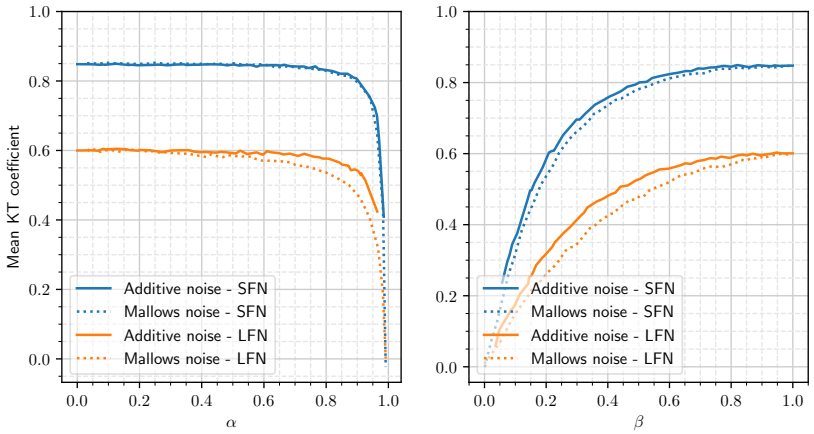


Figure 6.10: Evaluation of PWRF in terms of mean KT coefficient

6.3 Results on Semi-synthetic and Real-world Data Sets

In this section, we evaluate our algorithms on several standard LR data sets, for which, unlike the previous section, no underlying linear score function is assumed to exist. Thus, it is interesting to examine whether the bias incurred by the linear nature of PWHH and LWLR yields a performance drop in comparison to the rest of the algorithms, which are expected to be more flexible, since they are based on decision trees and random forests.

Semi-synthetic and real-world data sets The semi-synthetic and real-world data sets we used, were originally proposed in the works of Hüllermeier et al. [2008], Cheng et al. [2009] and are regarded as standard LR benchmarks ever since. The real-world datasets originate from bioinformatics fields, where ranking-related data can be quite frequently found. The semi-synthetic data sets were created from the transformation of multiclass (type A) and regression (type B) data sets from UCI repository and the Statlog collection into LR data (see Hüllermeier et al. [2008], Cheng et al. [2009, 2010, 2013], Fotakis et al. [2022a]). A summary of the aforementioned data sets and their characteristics is given in Table 6.1 and Table 6.2 respectively.

Dataset	Type	Number of examples	Number of features	Number of labels
authorship	A	841	70	4
bodyfat	B	4522	7	7
calhousing	B	37152	4	4
cpu-small	B	14744	6	5
elevators	B	29871	9	9
fried	B	73376	9	5
glass	A	214	9	6
housing	B	906	6	6
iris	A	150	4	3
pendigits	A	10992	16	10
segment	A	2310	18	7
stock	B	1710	5	5
vehicle	B	846	18	14
vowel	A	528	10	11
wine	A	178	13	3
wisconsin	B	346	16	16

Table 6.1: Semi-synthetic datasets

Dataset	Number of examples	Number of features	Number of labels
cold	2465	24	4
diau	2465	24	7
dtt	2465	24	4
heat	2465	24	6
spo	2465	24	11

Table 6.2: Real-world datasets

For each data set, we run five repetitions of a ten-fold cross-validation process, which is a common practice in many experimental LR works (Cheng and Hüllermeier [2008], Cheng et al. [2009, 2010, 2013], Fotakis et al. [2022a]). Namely, each data set is randomly divided into ten folds five times. For every split, we repeat the following process: every fold is used exactly one time as the validation set, while the rest are used as the training set (i.e. ten iterations for every repetition of the ten-fold cross-validation process). Finally, like before, we compute the mean and standard deviation of the KT correlation coefficient over every split's results.

As shown in Table 6.3 and Table 6.4, the random forest based algorithms LWRF and PWRF achieve the best performance in the overwhelming majority of the benchmarks. Moreover, we can see that LWRF and PWRF, which are both based on random forests, achieve almost the same performance for all benchmarks like in the case of synthetic datasets. The same holds for LWDT and PWDT, which

Dataset	LWRF	PWRF	LWDT	PWDT	LWLR	PWHH
authorship	0.926 ± 0.018	0.936 ± 0.014	0.871 ± 0.026	0.870 ± 0.022	$\mathbf{0.940 \pm 0.012}$	0.472 ± 0.060
bodyfat	0.191 ± 0.068	0.188 ± 0.053	0.108 ± 0.072	0.105 ± 0.066	$\mathbf{0.283 \pm 0.054}$	0.136 ± 0.073
calhousing	$\mathbf{0.491 \pm 0.010}$	0.486 ± 0.010	0.351 ± 0.011	0.356 ± 0.010	0.235 ± 0.009	0.169 ± 0.011
cpu-small	0.513 ± 0.011	$\mathbf{0.518 \pm 0.011}$	0.372 ± 0.014	0.369 ± 0.015	0.424 ± 0.012	0.402 ± 0.010
elevators	0.779 ± 0.005	$\mathbf{0.803 \pm 0.006}$	0.675 ± 0.007	0.694 ± 0.007	0.702 ± 0.005	0.576 ± 0.011
fried	0.985 ± 0.001	0.991 ± 0.001	0.964 ± 0.001	0.987 ± 0.001	$\mathbf{0.995 \pm 0.001}$	0.975 ± 0.001
glass	$\mathbf{0.906 \pm 0.032}$	0.905 ± 0.039	0.872 ± 0.043	0.865 ± 0.040	0.815 ± 0.052	0.759 ± 0.072
housing	$\mathbf{0.833 \pm 0.025}$	0.825 ± 0.028	0.783 ± 0.028	0.769 ± 0.030	0.583 ± 0.036	0.581 ± 0.035
iris	0.959 ± 0.043	$\mathbf{0.968 \pm 0.041}$	0.962 ± 0.047	0.946 ± 0.051	0.799 ± 0.085	0.476 ± 0.149
pendigits	$\mathbf{0.976 \pm 0.001}$	0.975 ± 0.001	0.957 ± 0.001	0.959 ± 0.002	0.855 ± 0.002	0.664 ± 0.007
segment	0.976 ± 0.004	$\mathbf{0.977 \pm 0.004}$	0.964 ± 0.005	0.968 ± 0.005	0.877 ± 0.008	0.847 ± 0.010
stock	0.922 ± 0.011	$\mathbf{0.924 \pm 0.011}$	0.902 ± 0.014	0.898 ± 0.015	0.685 ± 0.021	0.495 ± 0.027
vehicle	$\mathbf{0.886 \pm 0.017}$	0.884 ± 0.022	0.837 ± 0.030	0.824 ± 0.026	0.804 ± 0.031	0.745 ± 0.034
vowel	0.892 ± 0.014	$\mathbf{0.908 \pm 0.014}$	0.834 ± 0.020	0.831 ± 0.019	0.596 ± 0.026	0.484 ± 0.034
wine	0.925 ± 0.065	$\mathbf{0.956 \pm 0.042}$	0.888 ± 0.069	0.898 ± 0.065	0.950 ± 0.046	0.213 ± 0.133
wisconsin	0.552 ± 0.034	0.524 ± 0.036	0.415 ± 0.039	0.411 ± 0.045	$\mathbf{0.619 \pm 0.029}$	0.287 ± 0.061

Table 6.3: Evaluation in terms of mean KT coefficient on semi-synthetic datasets

Dataset	LWRF	PWRF	LWDT	PWDT	LWLR	PWHH
cold	0.105 ± 0.032	$\mathbf{0.108 \pm 0.038}$	0.050 ± 0.031	0.053 ± 0.038	0.085 ± 0.031	0.070 ± 0.042
diau	0.211 ± 0.026	0.209 ± 0.025	0.121 ± 0.025	0.107 ± 0.020	$\mathbf{0.220 \pm 0.026}$	0.191 ± 0.024
dtt	$\mathbf{0.141 \pm 0.028}$	0.130 ± 0.031	0.091 ± 0.034	0.080 ± 0.033	0.135 ± 0.029	0.111 ± 0.024
heat	$\mathbf{0.067 \pm 0.024}$	0.066 ± 0.021	0.036 ± 0.023	0.036 ± 0.022	0.049 ± 0.025	0.052 ± 0.024
spo	0.132 ± 0.018	0.127 ± 0.017	0.054 ± 0.012	0.056 ± 0.015	$\mathbf{0.137 \pm 0.016}$	0.121 ± 0.010

Table 6.4: Evaluation in terms of mean KT coefficient on real-world datasets

are both based on decision trees. Surprisingly, LWLR seems to achieve the best performance for several benchmarks, not being notably superior, though, than the two random forest algorithms. As for the halfspace-based algorithm PWHH, it seems to perform poorly and significantly worse than LWLR in the majority of the benchmarks, as distinct from the case of synthetic datasets, where their performance was comparable.

Bibliography

Johannes Fürnkranz and Eyke Hüllermeier. *Preference Learning*. Springer-Verlag, Berlin, Heidelberg, 1st edition, 2010. ISBN 3642141242.

Nemanja Djuric, Mihajlo Grbovic, Vladan Radosavljevic, Narayan Bhamidipati, and Slobodan Vucetic. Non-linear label ranking for large-scale prediction of long-term user interests. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, AAAI'14, page 1788–1794. AAAI Press, 2014.

Shankar Vembu and Thomas Gärtner. Label ranking algorithms: A survey. *Preference Learning*, 01 2011. doi: 10.1007/978-3-642-14125-6_3.

Yangming Zhou, Yangguang Liu, Jiangang Yang, Xiaoqi He, and Liangliang Liu. A taxonomy of label ranking algorithms. *J. Comput.*, 9:557–565, 2014a. URL https://api.semanticscholar.org/CorpusID:39329978.

Sariel Har-Peled, Dan Roth, and Dav Zimak. Constraint classification for multiclass classification and ranking. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems*, volume 15. MIT Press, 2002. URL https://proceedings.neurips.cc/paper_files/paper/2002/file/16026d60ff9b54410b3435b403afd226-Paper.pdf.

Ofer Dekel, Yoram Singer, and Christopher D Manning. Log-linear models for label ranking. In S. Thrun, L. Saul, and B. Schölkopf, editors, *Advances in Neural Information Processing Systems*, volume 16. MIT Press, 2003. URL https://proceedings.neurips.cc/paper_files/paper/2003/file/217c0e01c1828e7279051f1b6675745d-Paper.pdf.

J. Fürnkranz and E. Hüllermeier. Pairwise preference learning and ranking. *Lecture Notes in Artificial Intelligence (Subseries of Lecture Notes in Computer Science)*, 2837:145–156, 01 2003.

Eyke Hüllermeier, Johannes Fürnkranz, Weiwei Cheng, and Klaus Brinker. Label ranking by learning pairwise preferences. *Artif. Intell.*, 172(16–17):1897–1916, nov 2008. ISSN 0004-3702. doi: 10.1016/j.artint.2008.08.002. URL https://doi.org/10.1016/j.artint.2008.08.002.

Robin Vogel and Stéphan Clémençon. A multiclass classification approach to label ranking. In Silvia Chiappa and Roberto Calandra, editors, *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 1421–1430. PMLR, 26–28 Aug 2020. URL https://proceedings.mlr.press/v108/vogel20a.html.

Dimitris Fotakis, Alkis Kalavasis, and Eleni Psaroudaki. Label ranking through nonparametric regression. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 6622–6659. PMLR, 17–23 Jul 2022a. URL https://proceedings.mlr.press/v162/fotakis22a.html.

Dimitris Fotakis, Alkis Kalavasis, Vasilis Kontonis, and Christos Tzamos. Linear label ranking with bounded noise. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 15642–15656. Curran Associates, Inc., 2022b. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/64792f7bd5d400c9ac310c6fef97ef2d-Paper-Conference.pdf.

Massimo Gurrieri, Philippe Fortemps, and Xavier Siebert. Alternative decomposition techniques for label ranking. In Anne Laurent, Olivier Strauss, Bernadette Bouchon-Meunier, and Ronald R. Yager,

editors, *Information Processing and Management of Uncertainty in Knowledge-Based Systems - 15th International Conference, IPMU 2014, Montpellier, France, July 15-19, 2014, Proceedings, Part II*, volume 443 of *Communications in Computer and Information Science*, pages 464–474. Springer, 2014. doi: 10.1007/978-3-319-08855-6_47. URL https://doi.org/10.1007/978-3-319-08855-6_47.

Weiwei Cheng, Sascha Henzgen, and Eyke Hüllermeier. Labelwise versus pairwise decomposition in label ranking. In *LWA*, 2013. URL https://api.semanticscholar.org/CorpusID:14086581.

Weiwei Cheng and Eyke Hüllermeier. A nearest neighbor approach to label ranking based on generalized labelwise loss minimization. 2013. URL https://api.semanticscholar.org/CorpusID:4748938.

Weiwei Cheng and Eyke Hüllermeier. Instance-based label ranking using the mallows model. In *ECCBR Workshops*, 2008. URL https://api.semanticscholar.org/CorpusID:14482879.

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*, ICML '09, page 161–168, New York, NY, USA, 2009. Association for Computing Machinery. ISBN 9781605585161. doi: 10.1145/1553374.1553395. URL https://doi.org/10.1145/1553374.1553395.

Weiwei Cheng, Krzysztof Dembczyński, and Eyke Hüllermeier. Label ranking methods based on the plackett-luce model. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, page 215–222, Madison, WI, USA, 2010. Omnipress. ISBN 9781605589077.

Klaus Brinker and Eyke Hüllermeier. Case-based label ranking. volume 4212, pages 566–573, 09 2006. ISBN 978-3-540-45375-8. doi: 10.1007/11871842_53.

C. L. Mallows. Non-null ranking models. i. *Biometrika*, 44:114–130, 1957. URL https://api.semanticscholar.org/CorpusID:121527283.

R. L. Plackett. The analysis of permutations. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 24(2):193–202, 1975. doi: https://doi.org/10.2307/2346567. URL https://rss.onlinelibrary.wiley.com/doi/abs/10.2307/2346567.

Weiwei Cheng, Eyke Hüllermeier, Willem Waegeman, and Volkmar Welker. Label ranking with partial abstention based on thresholded probabilistic models. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012. URL https://proceedings.neurips.cc/paper_files/paper/2012/file/fe2d010308a6b3799a3d9c728ee74244-Paper.pdf.

Weiwei Cheng and Eyke Hüllermeier. Probability estimation for multi-class classification based on label ranking. In *ECML/PKDD*, 2012. URL https://api.semanticscholar.org/CorpusID:1483753.

Mihajlo Grbovic, Nemanja Djuric, and Slobodan Vucetic. Learning from pairwise preference data using gaussian mixture model. *Preference Learning: Problems and Applications in AI*, 33, 2012.

Yangming Zhou, Yangguang Liu, Xiao-Zhi Gao, and Guoping Qiu. A label ranking method based on gaussian mixture model. *Know.-Based Syst.*, 72(1):108–113, dec 2014b. ISSN 0950-7051. doi: 10.1016/j.knosys.2014.08.029. URL https://doi.org/10.1016/j.knosys.2014.08.029.

Yangming Zhou and Guoping Qiu. Random forest for label ranking. *Expert Syst. Appl.*, 112:99–109, 2016. URL https://api.semanticscholar.org/CorpusID:12169258.

Cláudio Rebelo de Sá, Carla Rebelo, Carlos Soares, and Arno J. Knobbe. Distance-based decision tree algorithms for label ranking. In *Portuguese Conference on Artificial Intelligence*, 2015. URL https://api.semanticscholar.org/CorpusID:7709302.

Cláudio Rebelo de Sá, Carlos Soares, Arno Knobbe, and Paulo Cortez. Label ranking forests. *Expert Systems*, 34(1):e12166, 2017. doi: https://doi.org/10.1111/exsy.12166. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/exsy.12166. e12166 EXSY-Nov-15-246.R1.

Juan A. Aledo, José A. Gámez, and David Molina. Tackling the supervised label ranking problem by bagging weak learners. *Inf. Fusion*, 35:38–50, 2017. doi: 10.1016/j.inffus.2016.09.002. URL https://doi.org/10.1016/j.inffus.2016.09.002.

Anna Korba, Alexandre Garcia, and Florence d'Alché-Buc. A structured prediction approach for label ranking. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/b3dd760eb02d2e669c604f6b2f1e803f-Paper.pdf.

Artur Aiguzhinov, Carlos Soares, and Ana Paula Serra. A similarity-based adaptation of naive bayes for label ranking: Application to the metalearning problem of algorithm recommendation. In *IFIP Working Conference on Database Semantics*, 2010. URL https://api.semanticscholar.org/CorpusID:41225546.

Cláudio Rebelo de Sá, Carlos Soares, Alípio Mário Jorge, Paulo J. Azevedo, and Joaquim Pinto da Costa. Mining association rules for label ranking. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 2011. URL https://api.semanticscholar.org/CorpusID:6295860.

Geraldina Ribeiro, Wouter Duivesteijn, Carlos Soares, and Arno J. Knobbe. Multilayer perceptron for label ranking. In *International Conference on Artificial Neural Networks*, 2012. URL https://api.semanticscholar.org/CorpusID:5649213.

Massimo Gurrieri, Xavier Siebert, Philippe Fortemps, Salvatore Greco, and Roman Słowiński. Label ranking: A new rule-based label ranking method. In *International Conference on Information Processing and Management of Uncertainty*, 2012. URL https://api.semanticscholar.org/CorpusID:9727346.

Mihajlo Grbovic, Nemanja Djuric, Shengbo Guo, and Slobodan Vucetic. Supervised clustering of label ranking data using label preference information. *Machine Learning*, 93:191–225, 2013. URL https://api.semanticscholar.org/CorpusID:7616589.

L. G. Valiant. A theory of the learnable. *Commun. ACM*, 27(11):1134–1142, nov 1984. ISSN 0001-0782. doi: 10.1145/1968.1972. URL https://doi.org/10.1145/1968.1972.

Dana Angluin and Philip Laird. Learning from noisy examples. *Mach. Learn.*, 2(4):343–370, apr 1988. ISSN 0885-6125. doi: 10.1023/A:1022873112823. URL https://doi.org/10.1023/A:1022873112823.

Pascal Massart and É lodie Nédélec. Risk bounds for statistical learning. *The Annals of Statistics*, 34(5), oct 2006. doi: 10.1214/009053606000000786. URL https://doi.org/10.1214%2F009053606000000786.

Enno Mammen and Alexandre B. Tsybakov. Smooth discrimination analysis. *The Annals of Statistics*, 27(6):1808 – 1829, 1999. doi: 10.1214/aos/1017939240. URL https://doi.org/10.1214/aos/1017939240.

Alexander B. Tsybakov. Optimal aggregation of classifiers in statistical learning. *The Annals of Statistics*, 32(1):135 – 166, 2004. doi: 10.1214/aos/1079120131. URL https://doi.org/10.1214/aos/1079120131.

Ilias Diakonikolas, Vasilis Kontonis, Christos Tzamos, and Nikos Zarifis. Learning halfspaces with massart noise under structured distributions. In Jacob Abernethy and Shivani Agarwal, editors, *Proceedings of Thirty Third Conference on Learning Theory*, volume 125 of *Proceedings of Machine Learning Research*, pages 1486–1513. PMLR, 09–12 Jul 2020a. URL https://proceedings.mlr.press/v125/diakonikolas20c.html.

Rajarajeswari Balasubramaniyan, Eyke Hüllermeier, Nils Weskamp, and Jörg Kämper. Clustering of gene expression data using a local shape-based similarity measure. *Bioinform.*, 21(7):1069–1077, 2005. doi: 10.1093/bioinformatics/bti095. URL https://doi.org/10.1093/bioinformatics/bti095.

Travis J. Hestilow, James Perez, and Yufei Huang. Clustering of gene expression data based on shape similarity. *EURASIP Journal on Bioinformatics and Systems Biology*, 2009:195712 – 195712, 2009. URL https://api.semanticscholar.org/CorpusID:16625176.

Pavel Brazdil, Carlos Soares, and Joaquim Pinto da Costa. Ranking learning algorithms: Using ibl and meta-learning on accuracy and time results. *Machine Learning*, 50:251–277, 2003. URL https://api.semanticscholar.org/CorpusID:2449067.

Qishen Wang, Ou Wu, Weiming Hu, Jinfeng Yang, and Wanqing Li. Ranking social emotions by learning listwise preference. In *First Asian Conference on Pattern Recognition, ACPR 2011, Beijing, China, 28-28 November, 2011*, pages 164–168. IEEE, 2011. doi: 10.1109/ACPR.2011.6166699. URL https://doi.org/10.1109/ACPR.2011.6166699.

Rajni Jindal and Taneja Shweta. Ranking in multi label classification of text documents using quantifiers. *2015 IEEE International Conference on Control System, Computing and Engineering (ICCSCE)*, pages 162–166, 2015. URL https://api.semanticscholar.org/CorpusID:7403280.

Anna Korba, Stéphan Clémençon, and Eric Sibony. A Learning Theory of Ranking Aggregation. In Aarti Singh and Jerry Zhu, editors, *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pages 1001–1010. PMLR, 20–22 Apr 2017. URL https://proceedings.mlr.press/v54/korba17a.html.

Stephan Clémençon, Anna Korba, and Eric Sibony. Ranking median regression: Learning to order through local consensus. In Firdaus Janoos, Mehryar Mohri, and Karthik Sridharan, editors, *Proceedings of Algorithmic Learning Theory*, volume 83 of *Proceedings of Machine Learning Research*, pages 212–245. PMLR, 07–09 Apr 2018. URL https://proceedings.mlr.press/v83/clemencon18a.html.

Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge University Press, 2022.

Amit Daniely, Sivan Sabato, Shai Ben-David, and Shai Shalev-Shwartz. Multiclass learnability and the erm principle, 2014.

Max Hopkins, Daniel M. Kane, Shachar Lovett, and Gaurav Mahajan. Realizable learning is all you need, 2023.

David Haussler. Decision theoretic generalizations of the pac model for neural net and other learning applications. *Inf. Comput.*, 100(1):78–150, sep 1992. ISSN 0890-5401. doi: 10.1016/0890-5401(92)90010-D. URL https://doi.org/10.1016/0890-5401(92)90010-D.

Michael J. Kearns, Robert E. Schapire, and Linda M. Sellie. Toward efficient agnostic learning. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, COLT '92, page 341–352, New York, NY, USA, 1992. Association for Computing Machinery. ISBN 089791497X. doi: 10.1145/130385.130424. URL https://doi.org/10.1145/130385.130424.

Michael J. Kearns and Umesh Vazirani. *An Introduction to Computational Learning Theory*. The MIT Press, 08 1994. ISBN 9780262276863. doi: 10.7551/mitpress/3897.001.0001. URL https://doi.org/10.7551/mitpress/3897.001.0001.

F. Rosenblatt. The perceptron - a perceiving and recognizing automaton. Technical Report 85-460-1, Cornell Aeronautical Laboratory, Ithaca, New York, January 1957.

F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386–408, 1958. doi: 10.1037/h0042519.

E. Cohen. Learning noisy perceptrons by a perceptron in polynomial time. In *Proceedings of the 38th Annual Symposium on Foundations of Computer Science*, FOCS '97, page 514, USA, 1997. IEEE Computer Society. ISBN 0818681977.

S. Vempala, R. Kannan, A. Blum, and A. Frieze. A polynomial-time algorithm for learning noisy linear threshold functions. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, page 330, Los Alamitos, CA, USA, oct 1996. IEEE Computer Society. doi: 10.1109/SFCS.1996.548492. URL https://doi.ieeecomputersociety.org/10.1109/SFCS.1996.548492.

Pranjal Awasthi, Maria-Florina Balcan, Nika Haghtalab, and Ruth Urner. Efficient learning of linear separators under bounded noise. In Peter Grünwald, Elad Hazan, and Satyen Kale, editors, *Proceedings of The 28th Conference on Learning Theory*, volume 40 of *Proceedings of Machine Learning Research*, pages 167–190, Paris, France, 03–06 Jul 2015. PMLR. URL https://proceedings.mlr.press/v40/Awasthi15b.html.

Pranjal Awasthi, Maria-Florina Balcan, Nika Haghtalab, and Hongyang Zhang. Learning and 1-bit compressed sensing under asymmetric noise. In Vitaly Feldman, Alexander Rakhlin, and Ohad Shamir, editors, *29th Annual Conference on Learning Theory*, volume 49 of *Proceedings of Machine Learning Research*, pages 152–192, Columbia University, New York, New York, USA, 23–26 Jun 2016. PMLR. URL https://proceedings.mlr.press/v49/awasthi16.html.

Songbai Yan and Chicheng Zhang. Revisiting perceptron: Efficient and label-optimal learning of halfspaces. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/556f391937dfd4398cbac35e050a2177-Paper.pdf.

Yuchen Zhang, Percy Liang, and Moses Charikar. A hitting time analysis of stochastic gradient langevin dynamics. In Satyen Kale and Ohad Shamir, editors, *Proceedings of the 2017 Conference on Learning Theory*, volume 65 of *Proceedings of Machine Learning Research*, pages 1980–2022. PMLR, 07–10 Jul 2017. URL https://proceedings.mlr.press/v65/zhang17b.html.

Chicheng Zhang, Jie Shen, and Pranjal Awasthi. Efficient active learning of sparse halfspaces with arbitrary bounded noise. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 7184–7197. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/5034a5d62f91942d2a7aeaf527dfe111-Paper.pdf.

Chicheng Zhang and Yinan Li. Improved algorithms for efficient active learning halfspaces with massart and tsybakov noise. In Mikhail Belkin and Samory Kpotufe, editors, *Proceedings of Thirty Fourth Conference on Learning Theory*, volume 134 of *Proceedings of Machine Learning Research*, pages 4526–4527. PMLR, 15–19 Aug 2021. URL https://proceedings.mlr.press/v134/zhang21a.html.

Ilias Diakonikolas, Themis Gouleakis, and Christos Tzamos. Distribution-independent pac learning of halfspaces with massart noise. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/358aee4cc897452c00244351e4d91f69-Paper.pdf.

Ilias Diakonikolas, Daniel M. Kane, and Christos Tzamos. Forster decomposition and learning halfspaces with noise. In *Neural Information Processing Systems*, 2021a. URL https://api.semanticscholar.org/CorpusID:235795575.

Ilias Diakonikolas, Daniel M. Kane, Vasilis Kontonis, Christos Tzamos, and Nikos Zarifis. Learning general halfspaces with general massart noise under the gaussian distribution. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2022, page 874–885, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450392648. doi: 10.1145/3519935.3519970. URL https://doi.org/10.1145/3519935.3519970.

Ilias Diakonikolas, Christos Tzamos, and Daniel M. Kane. A strongly polynomial algorithm for approximate forster transforms and its application to halfspace learning. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, STOC 2023, page 1741–1754, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9781450399135. doi: 10.1145/3564246.3585191. URL https://doi.org/10.1145/3564246.3585191.

Ilias Diakonikolas, Vasilis Kontonis, Christos Tzamos, and Nikos Zarifis. Learning halfspaces with tsybakov noise, 2020b.

Ilias Diakonikolas, Daniel M. Kane, Vasilis Kontonis, Christos Tzamos, and Nikos Zarifis. Efficiently learning halfspaces with tsybakov noise. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2021, page 88–101, New York, NY, USA, 2021b. Association for Computing Machinery. ISBN 9781450380539. doi: 10.1145/3406325.3450998. URL https://doi.org/10.1145/3406325.3450998.

Johannes Fürnkranz, Eyke Hüllermeier, Eneldo Loza Mencía, and Klaus Brinker. Multilabel classification via calibrated label ranking. *Machine Learning*, 73:133–153, 2008. URL https://api.semanticscholar.org/CorpusID:207211581.

Ravi Kumar and Sergei Vassilvitskii. Generalized distances between rankings. In *The Web Conference*, 2010. URL https://api.semanticscholar.org/CorpusID:2859966.

Trevor Hastie and Robert Tibshirani. Classification by pairwise coupling. In M. Jordan, M. Kearns, and S. Solla, editors, *Advances in Neural Information Processing Systems*, volume 10. MIT Press, 1997. URL https://proceedings.neurips.cc/paper_files/paper/1997/file/70feb62b69f16e0238f741fab228fec2-Paper.pdf.

Erin L. Allwein, Robert E. Schapire, and Yoram Singer. Reducing multiclass to binary: A unifying approach for margin classifiers. In *Proceedings of the Seventeenth International Conference on Machine Learning*, ICML '00, page 9–16, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc. ISBN 1558607072.

Johannes Fürnkranz. Round robin classification. *J. Mach. Learn. Res.*, 2:721–747, 2002. URL https://api.semanticscholar.org/CorpusID:5779282.

Olivier Bousquet, Stéphane Boucheron, and Gábor Lugosi. Introduction to statistical learning theory. In *Advanced Lectures on Machine Learning*, 2004. URL https://api.semanticscholar.org/CorpusID:669378.

Stéphane Boucheron, Olivier Bousquet, and Gábor Lugosi. Theory of classification : a survey of some recent advances. *Esaim: Probability and Statistics*, 9:323–375, 2005. URL https://api.semanticscholar.org/CorpusID:749141.

Nir Ailon, Moses Charikar, and Alantha Newman. Aggregating inconsistent information: Ranking and clustering. *J. ACM*, 55(5), nov 2008. ISSN 0004-5411. doi: 10.1145/1411509.1411513. URL https://doi.org/10.1145/1411509.1411513.

Anke van Zuylen, Rajneesh Hegde, Kamal Jain, and David P. Williamson. Deterministic pivoting algorithms for constrained ranking and clustering problems. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '07, page 405–414, USA, 2007. Society for Industrial and Applied Mathematics. ISBN 9780898716245.

L. Breiman, Jerome H. Friedman, Richard A. Olshen, and C. J. Stone. Classification and regression trees. *Biometrics*, 40:874, 1984. URL https://api.semanticscholar.org/CorpusID:29458883.

L. Breiman. Random forests. *Machine Learning*, 45:5–32, 2001. URL https://api.semanticscholar.org/CorpusID:89141.

Sudhakar Dharmadhikari and Kumar Joag-dev. *Unimodality, convexity, and applications*. Academic Press, 1988.

László Lovász and Santosh Vempala. The geometry of logconcave functions and sampling algorithms. *Random Struct. Algorithms*, 30(3):307–358, may 2007. ISSN 1042-9832.

Adam Klivans, Philip Long, and Alex Tang. Baum's algorithm learns intersections of halfspaces with respect to log-concave distributions. pages 588–600, 01 2009. ISBN 978-3-642-03684-2. doi: 10.1007/978-3-642-03685-9_44.

Maria-Florina Balcan and Phil Long. Active and passive learning of linear separators under log-concave distributions. In Shai Shalev-Shwartz and Ingo Steinwart, editors, *Proceedings of the 26th Annual Conference on Learning Theory*, volume 30 of *Proceedings of Machine Learning Research*, pages 288–316, Princeton, NJ, USA, 12–14 Jun 2013. PMLR. URL https://proceedings.mlr.press/v30/Balcan13.html.

Nisheeth K. Vishnoi. *Algorithms for Convex Optimization*. Cambridge University Press, 2021. doi: 10.1017/9781108699211.

Alexander Schrijver. Theory of linear and integer programming. In *Wiley-Interscience series in discrete mathematics and optimization*, 1986. URL https://api.semanticscholar.org/CorpusID:29180149.

Appendix A

Logarithmically Concave Probability Distributions

Definition A.0.1 (Logarithmically Concave Function)**.** *A nonnegative function $f\colon C \to \mathbb{R}_{\geq 0}$, where C is a convex set, is logarithmically concave (log-concave), if it satisfies the inequality*

$$f\left(\lambda x + (1-\lambda)y\right) \geq f(x)^{\lambda} f(y)^{1-\lambda}$$

for all $x, y \in C$ and $\lambda \in (0,1)$.

Definition A.0.2 (Isotropic Probability Distribution)**.** *A probability distribution $\mathcal{D}$ on $\mathbb{R}^d$ is isotropic, if it holds $\mathbf{E}_{\boldsymbol{x}\sim\mathcal{D}}[\boldsymbol{x}] = \mathbf{0}$ and $\mathbf{Var}_{\boldsymbol{x}\sim\mathcal{D}}[\boldsymbol{x}] = \boldsymbol{I}$, where $\boldsymbol{I}$ is the identity matrix.*

Definition A.0.3 (Logarithmically Concave Probability Distribution)**.** *A probability distribution $\mathcal{D}$ on $\mathbb{R}^d$ is log-concave, if it has a log-concave probability density function $f\colon \mathbb{R}^d \to \mathbb{R}_{\geq 0}$.*

It can be shown that the family of log-concave distributions captures many well-known parametric distribution families such as

- the uniform distribution over any convex set,
- the multivariate normal distribution,
- the exponential distribution,
- the logistic distribution,
- the chi distribution,
- the hyperbolic secant distribution,
- the Laplace distribution and
- the Gamma distribution $\Gamma(\alpha, \beta)$ with shape parameter $\alpha \geq 1$.

Lemma A.0.1 (Dharmadhikari and Joag-dev [1988])**.** *If a random vector $\boldsymbol{x}$ follows a log-concave probability distribution on $\mathbb{R}^d$, then, for any nonzero matrix $\boldsymbol{A} \in \mathbb{R}^{m\times d}$, $\boldsymbol{Ax}$ follows a log-concave probability distribution on $\mathbb{R}^m$.*

Lemma A.0.2 (Lovász and Vempala [2007], Klivans et al. [2009])**.** *Let $\mathcal{D}$ be an isotropic log-concave probability distribution on $\mathbb{R}^d$ with log-concave probability density function $f\colon \mathbb{R}^d \to \mathbb{R}_{\geq 0}$. The following properties hold.*

1. *For all $\boldsymbol{x} \in \mathbb{R}^d$ with $\|\boldsymbol{x}\|_2 \leq 1/9$, it holds $f(\boldsymbol{x}) \geq 2^{-7d} 2^{-9d\|\boldsymbol{x}\|_2}$ (anti-anti-concentration).*
2. *If $d = 1$, then for all $x \in \mathbb{R}$, it holds that $f(x) \leq 1$. If $d \geq 2$, then for all $\boldsymbol{x} \in \mathbb{R}^d$, it holds that $f(\boldsymbol{x}) \leq \beta_1(d) e^{-\beta_2(d)\|\boldsymbol{x}\|_2}$, where $\beta_1(d) \triangleq 2^{8d} d^{d/2} e$ and $\beta_2(d) \triangleq \frac{2^{-7d}}{2(d-1)(20(d-1))^{(d-1)/2}}$ (anti-concentration).*
3. *For any $R \geq 0$, it holds that $\mathbf{Pr}_{\boldsymbol{x}\sim\mathcal{D}}\left[\|\boldsymbol{x}\|_2 \geq R\right] \leq e^{1-R/\sqrt{d}}$ (concentration).*

Lemma A.0.3. *For any isotropic log-concave probability distribution $\mathcal{D}$ on $\mathbb{R}^d$, it holds*

$$2^{-7}\epsilon \leq \mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{D}}\left[|\langle \boldsymbol{v}, \boldsymbol{x}\rangle| \leq \epsilon\right] \leq 2\epsilon$$

for any $\boldsymbol{v} \in S^{d-1}$ and $\epsilon \geq 0$.

Proof. Let $\mathcal{D}'$ be the distribution of $\langle \boldsymbol{v}, \boldsymbol{x}\rangle = \boldsymbol{v}^\mathsf{T}\boldsymbol{x}$, where $\boldsymbol{x} \sim \mathcal{D}$. From Lemma A.0.1, we have that $\mathcal{D}'$ is log-concave on $\mathbb{R}$. Moreover, we have

$$\mathop{\mathbf{E}}_{x\sim\mathcal{D}'}[x] = \boldsymbol{v}^\mathsf{T} \mathop{\mathbf{E}}_{\boldsymbol{x}\sim\mathcal{D}}[\boldsymbol{x}] = 0$$

and

$$\mathbf{Var}_{x\sim\mathcal{D}'}[x] = \boldsymbol{v}^\mathsf{T}\mathbf{Var}_{\boldsymbol{x}\sim\mathcal{D}}[\boldsymbol{x}]\boldsymbol{v} = \boldsymbol{v}^\mathsf{T}\boldsymbol{v} = 1\,,$$

namely $\mathcal{D}'$ is isotropic. Let $f\colon \mathbb{R} \to \mathbb{R}_{\geq 0}$ be the isotropic log-concave probability density function corresponding to $\mathcal{D}'$. Therefore, we have that

$$\mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{D}}\left[|\langle \boldsymbol{v}, \boldsymbol{x}\rangle| \leq \epsilon\right] = \mathop{\mathbf{Pr}}_{x\sim\mathcal{D}'}\left[-\epsilon \leq x \leq \epsilon\right] = \int_{-\epsilon}^{\epsilon} f(x)\,\mathrm{d}x\,.$$

Using property 2 (anti-concentration) of Lemma A.0.2, we get that

$$\int_{-\epsilon}^{\epsilon} f(x)\,\mathrm{d}x \leq \int_{-\epsilon}^{\epsilon} \mathrm{d}x = 2\epsilon\,.$$

Using property 1 (anti-anti-concentration) of Lemma A.0.2, we get that

$$\int_{-\epsilon}^{\epsilon} f(x)\,\mathrm{d}x \geq \int_{-\min\{\epsilon,1/9\}}^{\min\{\epsilon,1/9\}} f(x)\,\mathrm{d}x \geq 2^{-7}2^{-9\min\{\epsilon,1/9\}} \int_{-\min\{\epsilon,1/9\}}^{\min\{\epsilon,1/9\}} \mathrm{d}x \geq 2^{-8}\int_{-\epsilon}^{\epsilon}\mathrm{d}x = 2^{-7}\epsilon\,.$$

The facts above conclude the proof. □

Lemma A.0.4. *For any isotropic log-concave probability distribution $\mathcal{D}$ on $\mathbb{R}^d$, it holds*

$$\frac{1}{81\cdot 2^{16}}\theta(\boldsymbol{u},\boldsymbol{v}) \leq \mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{D}}\left[\mathrm{sign}(\langle \boldsymbol{u}, \boldsymbol{x}\rangle) \neq \mathrm{sign}(\langle \boldsymbol{v}, \boldsymbol{x}\rangle)\right] \leq 5\cdot 2^{50} e\theta(\boldsymbol{u},\boldsymbol{v})$$

for all $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^d \setminus \{\boldsymbol{0}\}$.

Proof. First, assume that $\boldsymbol{u}$ and $\boldsymbol{v}$ are parallel (which is always the case when $d = 1$). Then, it holds that either $\mathbf{Pr}_{\boldsymbol{x}\sim\mathcal{D}}\left[\mathrm{sign}(\langle \boldsymbol{u}, \boldsymbol{x}\rangle) \neq \mathrm{sign}(\langle \boldsymbol{v}, \boldsymbol{x}\rangle)\right] = 0$ and $\theta(\boldsymbol{u},\boldsymbol{v}) = 0$ or $\mathbf{Pr}_{\boldsymbol{x}\sim\mathcal{D}}\left[\mathrm{sign}(\langle \boldsymbol{u}, \boldsymbol{x}\rangle) \neq \mathrm{sign}(\langle \boldsymbol{v}, \boldsymbol{x}\rangle)\right] = 1$ and $\theta(\boldsymbol{u},\boldsymbol{v}) = \pi$. In both cases, the inequality holds.

Now, assume that $\boldsymbol{u}$ and $\boldsymbol{v}$ are not parallel (so $d \geq 2$). Let V be the linear subspace of $\mathbb{R}^d$ spanned by $\boldsymbol{u}$ and $\boldsymbol{v}$ (we have $\dim(V) = 2$) and let $\boldsymbol{Q} \in \mathbb{R}^{d\times 2}$ be a matrix, whose columns are the vectors of an orthonormal basis of V. Let $\mathcal{D}'$ be the distribution of $\boldsymbol{Q}^\mathsf{T}\boldsymbol{x}$, where $\boldsymbol{x} \sim \mathcal{D}$. From Lemma A.0.1, we have that $\mathcal{D}'$ is log-concave on $\mathbb{R}^2$. Moreover, we have

$$\mathop{\mathbf{E}}_{\boldsymbol{x}\sim\mathcal{D}'}[\boldsymbol{x}] = \boldsymbol{Q}^\mathsf{T} \mathop{\mathbf{E}}_{\boldsymbol{x}\sim\mathcal{D}}[\boldsymbol{x}] = \boldsymbol{0}$$

and

$$\mathbf{Var}_{\boldsymbol{x}\sim\mathcal{D}'}[\boldsymbol{x}] = \boldsymbol{Q}^\mathsf{T}\mathbf{Var}_{\boldsymbol{x}\sim\mathcal{D}}[\boldsymbol{x}]\boldsymbol{Q} = \boldsymbol{Q}^\mathsf{T}\boldsymbol{Q} = \boldsymbol{I}$$

(since $\boldsymbol{Q}$ is semi-orthogonal), namely $\mathcal{D}'$ is isotropic. It can be easily shown that $\langle \boldsymbol{u}, \boldsymbol{x}\rangle = \langle \boldsymbol{Q}^\mathsf{T}\boldsymbol{u}, \boldsymbol{Q}^\mathsf{T}\boldsymbol{x}\rangle$ and $\langle \boldsymbol{v}, \boldsymbol{x}\rangle = \langle \boldsymbol{Q}^\mathsf{T}\boldsymbol{v}, \boldsymbol{Q}^\mathsf{T}\boldsymbol{x}\rangle$. Therefore, we have that

$$\mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{D}}\left[\mathrm{sign}(\langle \boldsymbol{u}, \boldsymbol{x}\rangle) \neq \mathrm{sign}(\langle \boldsymbol{v}, \boldsymbol{x}\rangle)\right] = \mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{D}'}\left[\mathcal{E}(\boldsymbol{x})\right]\,,$$

where $\mathcal{E}(\boldsymbol{x})$ denotes the event that $\mathrm{sign}(\langle \boldsymbol{Q}^\mathsf{T}\boldsymbol{u}, \boldsymbol{x}\rangle) \neq \mathrm{sign}(\langle \boldsymbol{Q}^\mathsf{T}\boldsymbol{v}, \boldsymbol{x}\rangle)$ for any $\boldsymbol{x} \in \mathbb{R}^2$. Furthermore, it can be shown that $\theta(\boldsymbol{Q}^\mathsf{T}\boldsymbol{u}, \boldsymbol{Q}^\mathsf{T}\boldsymbol{v}) = \theta(\boldsymbol{u},\boldsymbol{v})$. Let $f\colon \mathbb{R}^2 \to \mathbb{R}_{\geq 0}$ be the isotropic log-concave probability density function corresponding to $\mathcal{D}'$. For the following, we assume without loss of generality that $\boldsymbol{Q}^\mathsf{T}\boldsymbol{u} = (0, 1)$ and $\boldsymbol{Q}^\mathsf{T}\boldsymbol{v} = (-\sin(\theta(\boldsymbol{u},\boldsymbol{v})), \cos(\theta(\boldsymbol{u},\boldsymbol{v})))$.

We proceed by proving the left part of the inequality. Using property 1 (anti-anti-concentration) of Lemma A.0.2, we get that

$$\begin{aligned}\Pr_{\boldsymbol{x}\sim\mathcal{D}'}\left[\mathcal{E}(\boldsymbol{x})\right] &\geq \int_0^{2\pi}\int_0^{1/9} r\mathbb{1}\left\{\mathcal{E}\left(r\cos(\theta), r\sin(\theta)\right)\right\} f\left(r\cos(\theta), r\sin(\theta)\right)\, \mathrm{d}r\mathrm{d}\theta \\ &\geq \frac{1}{2^{16}}\left[\int_0^{\theta(\boldsymbol{u},\boldsymbol{v})}\int_0^{1/9} r\, \mathrm{d}r\mathrm{d}\theta + \int_{\pi}^{\pi+\theta(\boldsymbol{u},\boldsymbol{v})}\int_0^{1/9} r\, \mathrm{d}r\mathrm{d}\theta\right] \\ &= \frac{1}{81\cdot 2^{16}}\theta(\boldsymbol{u},\boldsymbol{v})\,,\end{aligned}$$

Next, we prove the left part of the inequality, applying a similar technique used in Theorem 4 of Balcan and Long [2013]. Using property 2 (anti-concentration) of Lemma A.0.2, we get that for any $\gamma > 0$, it holds that

$$\begin{aligned}\Pr_{\boldsymbol{x}\sim\mathcal{D}'}\left[\mathcal{E}(\boldsymbol{x})\right] &= \sum_{i=0}^{\infty}\int_{B^2((i+1)\gamma)\setminus B^2(i\gamma)} \mathbb{1}\left\{\mathcal{E}(x,y)\right\} f(x,y)\, \mathrm{d}x\mathrm{d}y \\ &= \sum_{i=0}^{\infty}\int_0^{2\pi}\int_{i\gamma}^{(i+1)\gamma} r\mathbb{1}\left\{\mathcal{E}\left(r\cos(\theta), r\sin(\theta)\right)\right\} f\left(r\cos(\theta), r\sin(\theta)\right)\, \mathrm{d}r\mathrm{d}\theta \\ &\leq \sum_{i=0}^{\infty}\beta_1(2)e^{-\beta_2(2)\gamma i}\left[\int_0^{\theta(\boldsymbol{u},\boldsymbol{v})}\int_{i\gamma}^{(i+1)\gamma} r\, \mathrm{d}r\mathrm{d}\theta + \int_{\pi}^{\pi+\theta(\boldsymbol{u},\boldsymbol{v})}\int_{i\gamma}^{(i+1)\gamma} r\, \mathrm{d}r\mathrm{d}\theta\right] \\ &= \sum_{i=0}^{\infty}\beta_1(2)e^{-\beta_2(2)\gamma i}\gamma^2(2i+1)\theta(\boldsymbol{u},\boldsymbol{v}) \\ &= \beta_1(2)\gamma^2\frac{1+e^{-\beta_2(2)\gamma}}{\left(1-e^{-\beta_2(2)\gamma}\right)^2}\theta(\boldsymbol{u},\boldsymbol{v})\,,\end{aligned}$$

where the functions β_1, β_2 are defined as in Lemma A.0.2. Therefore,

$$\begin{aligned}\Pr_{\boldsymbol{x}\sim\mathcal{D}'}\left[\mathcal{E}(\boldsymbol{x})\right] &\leq \beta_1(2)\theta(\boldsymbol{u},\boldsymbol{v})\inf_{\gamma>0}\gamma^2\frac{1+e^{-\beta_2(2)\gamma}}{\left(1-e^{-\beta_2(2)\gamma}\right)^2} \\ &= \beta_1(2)\theta(\boldsymbol{u},\boldsymbol{v})\lim_{\gamma\to 0}\gamma^2\frac{1+e^{-\beta_2(2)\gamma}}{\left(1-e^{-\beta_2(2)\gamma}\right)^2} \\ &= \frac{2\beta_1(2)}{\beta_2(2)^2}\theta(\boldsymbol{u},\boldsymbol{v}) \\ &= 5\cdot 2^{50}e\theta(\boldsymbol{u},\boldsymbol{v})\,.\end{aligned}$$

□

We now define a more general family of distributions, proposed by Diakonikolas et al. [2020a], that subsumes the family of isotropic log-concave distributions.

Definition A.0.4 (Bounded Probability Distribution)**.** *Let $U, R > 0$ and $t\colon \mathbb{R}_{>0} \to \mathbb{R}_{\geq 0}$. An isotropic probability distribution $\mathcal{D}$ on $\mathbb{R}^d$ is called (U, R, t)-bounded, if, for any projection $\mathcal{D}_V$ of $\mathcal{D}$ onto a 2-dimensional subspace V, $\mathcal{D}_V$ has a probability density function $\gamma_V\colon \mathbb{R}^2 \to \mathbb{R}_{\geq 0}$ that satisfies the following properties:*

1. *$\gamma_V(\boldsymbol{x}) \geq U^{-1}$ for all $\boldsymbol{x} \in V$ such that $\|\boldsymbol{x}\|_2 \leq R$ (anti-anti-concentration).*
2. *$\gamma_V(\boldsymbol{x}) \leq U$ for all $\boldsymbol{x} \in V$ (anti-concentration).*
3. *$\mathbf{Pr}_{\boldsymbol{x}\sim\mathcal{D}_V}\left[\|\boldsymbol{x}\|_2 \geq t(\epsilon)\right] \leq \epsilon$ for all $\epsilon \in (0,1)$ (concentration).*

Lemma A.0.5 (Diakonikolas et al. [2020a])**.** *Let $U, R > 0$, let $t\colon \mathbb{R}_{>0} \to \mathbb{R}_{\geq 0}$ and let $\mathcal{D}_{\boldsymbol{x}}$ be a (U, R, t)-bounded distribution on $\mathbb{R}^d$. It holds that*

$$R^2U^{-1}\theta(\boldsymbol{u},\boldsymbol{v}) \leq \Pr_{\boldsymbol{x}\sim\mathcal{D}_{\boldsymbol{x}}}\left[h_{\boldsymbol{u}}(\boldsymbol{x}) \neq h_{\boldsymbol{v}}(\boldsymbol{x})\right] \leq Ut(\epsilon)^2\theta(\boldsymbol{u},\boldsymbol{v}) + \epsilon$$

for any $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^d \setminus \{\mathbf{0}\}$ and $\epsilon > 0$.

Lemma A.0.6. *Any isotropic log-concave distribution on* $\mathbb{R}^d$ *is* (U, R, t)*-bounded with* $U = 2^{17}e$*,* $R = 1/9$ *and* $t(\epsilon) = \sqrt{d} + \sqrt{d}\ln(1/\epsilon)$ *for all* $\epsilon > 0$*.*

Proof. It is a direct implication of Lemma A.0.2. □

Appendix B

The Ellipsoid Method

We now make a brief reference to the Ellipsoid method that has been used as part of several algorithms in this thesis. The Ellipsoid method constitutes an iterative method for minimizing convex functions. Its main idea is to construct a sequence of decreasing volume ellipsoids that enclose a minimizer of a convex function, which is to be minimized. Here, we focus on the most basic version of the Ellipsoid method that concerns feasibility problems.

Definition B.0.1 (Positive Definite Matrix)**.** *A symmetric matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ is said to be positive definite, if $\boldsymbol{x}^\mathsf{T}\boldsymbol{A}\boldsymbol{x} > 0$ for all nonzero $\boldsymbol{x} \in \mathbb{R}^n$.*

Definition B.0.2 (Ellipsoid)**.** *A set of vectors of the form*

$$E(\boldsymbol{z}, \boldsymbol{D}) = \left\{\boldsymbol{x} \in \mathbb{R}^d : (\boldsymbol{x}-\boldsymbol{z})^\mathsf{T}\boldsymbol{D}^{-1}(\boldsymbol{x}-\boldsymbol{z})\right\} \leq 1\Big\} \ ,$$

where $\boldsymbol{z} \in \mathbb{R}^n$ and $\boldsymbol{D} \in \mathbb{R}^{n\times n}$ is a positive definite matrix, is said to be an ellipsoid in $\mathbb{R}^n$ with center $\boldsymbol{z}$.

Proposition B.0.1. *The volume of an ellipsoid $E(\boldsymbol{z}, \boldsymbol{D})$ in $\mathbb{R}^n$ is*

$$\operatorname{vol}\left(E(\boldsymbol{z}, \boldsymbol{D})\right) = \det\left(\boldsymbol{D}^{1/2}\right) \operatorname{vol}\left(B^n\right) \ .$$

Definition B.0.3 (Separation Oracle)**.** *A separation oracle for a convex set $P \subseteq \mathbb{R}^n$ works as follows. Given any point $\boldsymbol{z} \in \mathbb{R}^n$,*

- *if $\boldsymbol{z} \in P$, it answers YES,*
- *if $\boldsymbol{z} \notin P$, it answers NO and outputs a vector $\boldsymbol{a} \in \mathbb{R}^n$ such that $\langle \boldsymbol{a}, \boldsymbol{x}\rangle < \langle \boldsymbol{a}, \boldsymbol{z}\rangle$ for all $\boldsymbol{x} \in P$.*

Lemma B.0.1 (Vishnoi [2021])**.** *Let $E(\boldsymbol{z}, \boldsymbol{D})$ be any ellipsoid in $\mathbb{R}^n$ and let $\boldsymbol{a} \in \mathbb{R}^n$ be any nonzero vector. Consider the ellipsoid $E(\boldsymbol{z}', \boldsymbol{D}')$, where*

$$\boldsymbol{z}' = \boldsymbol{z} - \frac{1}{n+1}\frac{\boldsymbol{D}\boldsymbol{a}}{\sqrt{\boldsymbol{a}^\mathsf{T}\boldsymbol{D}\boldsymbol{a}}}$$
$$\boldsymbol{D}' = \frac{n^2}{n^2-1}\left(\boldsymbol{D} - \frac{2}{n+1}\frac{\boldsymbol{D}\boldsymbol{a}\boldsymbol{a}^\mathsf{T}\boldsymbol{D}}{\boldsymbol{a}^\mathsf{T}\boldsymbol{D}\boldsymbol{a}}\right) .$$

It holds that

- *$E(\boldsymbol{z}, \boldsymbol{D}) \cap \{\boldsymbol{x} \in \mathbb{R}^n : \langle \boldsymbol{a}, \boldsymbol{x}\rangle \leq \langle \boldsymbol{a}, \boldsymbol{z}\rangle\} \subset E(\boldsymbol{z}', \boldsymbol{D}')$ and*
- *$\operatorname{vol}\left(E(\boldsymbol{z}', \boldsymbol{D}')\right) < e^{-\frac{1}{2(n+1)}} \operatorname{vol}\left(E(\boldsymbol{z}, \boldsymbol{D})\right)$.*

Moreover, the minimum volume ellipsoid that contains $E(\boldsymbol{z}, \boldsymbol{D}) \cap \{\boldsymbol{x} \in \mathbb{R}^n : \langle \boldsymbol{a}, \boldsymbol{x}\rangle \leq \langle \boldsymbol{a}, \boldsymbol{z}\rangle\}$ is unique and given by $E(\boldsymbol{z}', \boldsymbol{D}')$.

Lemma B.0.2 (Vishnoi [2021])**.** *Let $P \subseteq \mathbb{R}^n$, where $n \geq 1$, be a convex set that is contained in a n-dimensional Euclidean ball of radius $R > 0$ and contains a n-dimensional Euclidean ball of radius $r > 0$. Then, the Ellipsoid method outputs a point $\boldsymbol{x} \in P$ after $O(n^2 \log(R/r))$ iterations. Moreover, every iteration can be implemented in $O(n^2 + T_{\text{sep}})$ time, where T_{sep} is the time required to answer a single query by the separation oracle.*

We remark that Lemma B.0.2 assumes that all calculations are executed in infinite precision and constant time, which is not the case in reality. This is, because the computation of the ellipsoids involves taking square roots, which cannot always be exactly performed. Therefore, one has to make some slight adjustments to the basic ellipsoid algorithm for the case of a predetermined representation size, so that its theoretical guarantees are not affected. For more details on this issue, we refer to Vishnoi [2021].

Algorithm 12 Ellipsoid Method

Input:

- A separation oracle for a convex set $P \subseteq \mathbb{R}^n$
- An ellipsoid $E(\boldsymbol{x}_0, \boldsymbol{D}_0)$ in $\mathbb{R}^n$ such that $P \subseteq E(\boldsymbol{x}_0, \boldsymbol{D}_0)$ and $V = \mathrm{vol}\left(E(\boldsymbol{x}_0, \boldsymbol{D}_0)\right)$.
- A parameter $v \in \mathbb{R}_{>0}$.

Output:

- YES and a point in P, which is always the case when $\mathrm{vol}(P) \geq v$.
- NO, in which case it is guaranteed that $\mathrm{vol}(P) < v$.

$T \leftarrow \lceil 2(n+1)\ln(V/v) \rceil$
for $t = 0, \ldots, T$ **do**
  Query the separation oracle for P on $\boldsymbol{x}_t$.
  if $\boldsymbol{x}_t \in P$ **then**
    return YES, $\boldsymbol{x}_t$
  else
    Let $\boldsymbol{a}_t$ be the vector returned by the separation oracle.
    Let $E(\boldsymbol{x}_{t+1}, \boldsymbol{D}_{t+1})$ be the minimum volume ellipsoid such that: ▷ See Lemma B.0.1

$$E(\boldsymbol{x}_t, \boldsymbol{D}_t) \cap \{\boldsymbol{x} \in \mathbb{R}^n : \langle \boldsymbol{a}_t, \boldsymbol{x} \rangle \leq \langle \boldsymbol{a}_t, \boldsymbol{x}_t \rangle\} \subseteq E(\boldsymbol{x}_{t+1}, \boldsymbol{D}_{t+1})$$

  end if
end for
return NO

Appendix C

Omitted Proofs

C.1 The Proof of Lemma 2.7.1

Lemma C.1.1. *Let $\mathcal{D}$ be an (η, f)-RCN distribution, where $\eta \in [0, 1/2)$ and $f \in \{\pm 1\}^{\mathcal{X}}$. For any $h \in \{\pm 1\}^{\mathcal{X}}$, it holds that*

$$L_{\mathcal{D}_x, f, \ell_{0-1}}(h) = \frac{L_{\mathcal{D}, \ell_{0-1}}(h) - \eta}{1 - 2\eta}$$

and f is a minimizer of $L_{\mathcal{D}, \ell_{0-1}}$.

Proof. We have that

$$\begin{aligned}
L_{\mathcal{D}, \ell_{0-1}}(h) &= \mathop{\mathbf{Pr}}_{y \sim \mathcal{D}_{y|x}}[h(x) \neq y] \\
&= \mathop{\mathbf{E}}_{x \sim \mathcal{D}_x}\left[\mathop{\mathbf{Pr}}_{y \sim \mathcal{D}_{y|x}}[h(x) \neq y \mid x]\right] \\
&= \mathop{\mathbf{E}}_{x \sim \mathcal{D}_x}\left[\mathbb{1}\{h(x) = f(x)\} \mathop{\mathbf{Pr}}_{y \sim \mathcal{D}_{y|x}}[y \neq f(x) \mid x] + \mathbb{1}\{h(x) \neq f(x)\} \mathop{\mathbf{Pr}}_{y \sim \mathcal{D}_{y|x}}[y = f(x) \mid x]\right] \\
&= \mathop{\mathbf{E}}_{x \sim \mathcal{D}_x}\left[\left(1 - 2\mathop{\mathbf{Pr}}_{y \sim \mathcal{D}_{y|x}}[y \neq f(x) \mid x]\right)\mathbb{1}\{h(x) \neq f(x)\}\right] + \mathop{\mathbf{E}}_{x \sim \mathcal{D}_x}\left[\mathop{\mathbf{Pr}}_{y \sim \mathcal{D}_{y|x}}[y \neq f(x) \mid x]\right] \\
&= \mathop{\mathbf{E}}_{x \sim \mathcal{D}_x}[(1 - 2\eta)\,\mathbb{1}\{h(x) \neq f(x)\}] + \mathop{\mathbf{Pr}}_{(x,y) \sim \mathcal{D}}[y \neq f(x)] \\
&= (1 - 2\eta) L_{\mathcal{D}_x, f, \ell_{0-1}}(h) + \eta\,,
\end{aligned}$$

and $L_{\mathcal{D}, \ell_{0-1}}(f) = \eta$, that is, f minimizes $L_{\mathcal{D}, \ell_{0-1}}$. □

C.2 The Proof of Lemma 2.7.2

Lemma C.2.1. *Let $\mathcal{D}$ be an (η, f)-Massart distribution, where $\eta \in [0, 1/2)$ and $f \in \{0, 1\}^{\mathcal{X}}$. For any $h \in \{0, 1\}^{\mathcal{X}}$, it holds that*

$$L_{\mathcal{D}_x, f, \ell_{0-1}}(h) \leq \frac{L_{\mathcal{D}, \ell_{0-1}}(h) - L_{\mathcal{D}, \ell_{0-1}}(f)}{1 - 2\eta}$$

and f is a minimizer of $L_{\mathcal{D}, \ell_{0-1}}$.

Proof. Following the same procedure as in Lemma C.1.1, we get that

$$\begin{aligned}
L_{\mathcal{D}, \ell_{0-1}}(h) &= \mathop{\mathbf{E}}_{x \sim \mathcal{D}_x}\left[\left(1 - 2\mathop{\mathbf{Pr}}_{y \sim \mathcal{D}_{y|x}}[y \neq f(x) \mid x]\right)\mathbb{1}\{h(x) \neq f(x)\}\right] + \mathop{\mathbf{E}}_{x \sim \mathcal{D}_x}\left[\mathop{\mathbf{Pr}}_{y \sim \mathcal{D}_{y|x}}[y \neq f(x) \mid x]\right] \\
&\geq \mathop{\mathbf{E}}_{x \sim \mathcal{D}_x}[(1 - 2\eta)\,\mathbb{1}\{h(x) \neq f(x)\}] + \mathop{\mathbf{Pr}}_{(x,y) \sim \mathcal{D}}[y \neq f(x)] \\
&= (1 - 2\eta) L_{\mathcal{D}_x, f, \ell_{0-1}}(h) + L_{\mathcal{D}, \ell_{0-1}}(f)\,.
\end{aligned}$$

and if $h = f$, the above holds with equality, that is, f minimizes $L_{\mathcal{D}, \ell_{0-1}}$. □

C.3 The Proof of Lemma 5.2.1

Lemma C.3.1. *Let $\boldsymbol{U} \in \mathbb{R}^{k\times d}$, where $\boldsymbol{u}_i \neq \boldsymbol{u}_j$ for all $i \neq j$, and $\mathcal{D}_{\boldsymbol{x}}$ be an isotropic log-concave distribution on $\mathbb{R}^d$. There exists a polynomial $P\colon \mathbb{R}^3 \to \mathbb{R}$ such that the following holds. For any $\epsilon > 0$, there exists a matrix $\boldsymbol{V} \in \mathbb{R}^{k\times d}$ with the properties:*

1. $\max_{1\leq i<j\leq k} \mathbf{Pr}_{\boldsymbol{x}\sim\mathcal{D}_{\boldsymbol{x}}} \left[\operatorname{sign}\left(\langle \boldsymbol{u}_i - \boldsymbol{u}_j, \boldsymbol{x}\rangle\right) \neq \operatorname{sign}\left(\langle \boldsymbol{v}_i - \boldsymbol{v}_j, \boldsymbol{x}\rangle\right)\right] \leq \epsilon$
2. $\min_{1\leq i<j\leq k} \left\|\boldsymbol{v}_i - \boldsymbol{v}_j\right\|_2 \geq 2^{-P(d,k,1/\epsilon)}$
3. $\left\|\boldsymbol{V}\right\|_{\mathrm{F}} \leq 1$

Proof. To prove the existence of a matrix with the aforementioned properties, we will construct a linear program, whose output satisfies the desired properties with positive probability.

Let $\mathcal{D}$ be any probability distribution on $\mathbb{R}^d \times \mathbb{S}_k$ that is realizable by $\sigma_{\boldsymbol{U}}$ and whose marginal on $\mathbb{R}^d$ is $\mathcal{D}_{\boldsymbol{x}}$. We have seen that LSFs are properly and efficiently PAC learnable in the noiseless setting with respect to the KT distance. Here, we will adopt the same linear programming approach, except that the samples drawn from $\mathcal{D}$ will now be rounded before constructing the LP's constraints. As we will shortly see, rounding the samples is necessary to bound the bit complexity of the LP's output, so that the second and third property of the lemma can simultaneously hold.

Let $\varepsilon, R > 0$. Let $\mathcal{C}$ be any finite subset of $B^d(R)$ with the property that

$$\max_{\boldsymbol{x}\in S^{d-1}(R)} \min_{\boldsymbol{y}\in\mathcal{C}} \|\boldsymbol{y} - \boldsymbol{x}\|_2 \leq \varepsilon\,. \tag{1}$$

We define the rounding function $\boldsymbol{r}\colon \mathbb{R}^d \setminus \{\boldsymbol{0}\} \to \mathcal{C}$, which takes a feature vector $\boldsymbol{x} \neq \boldsymbol{0}$ as input, projects it on $S^{d-1}(R)$, which corresponds to the point $R\boldsymbol{x}/\|\boldsymbol{x}\|_2$, and returns the closest point to $R\boldsymbol{x}/\|\boldsymbol{x}\|_2$ that is in $\mathcal{C}$. Namely,

$$\boldsymbol{r}(\boldsymbol{x}) \in \operatorname*{argmin}_{\boldsymbol{y}\in\mathcal{C}} \left\|\boldsymbol{y} - \frac{R\boldsymbol{x}}{\|\boldsymbol{x}\|_2}\right\|_2 .$$

Let $T = \left(\left(\boldsymbol{x}^{(t)}, \pi^{(t)}\right)\right)_{t\in[N]}$ be a training set of N independent samples drawn from $\mathcal{D}_{\boldsymbol{x}}$. Consider the following linear program, which we denote by LP3:

$$\begin{aligned} &\text{Find} && \boldsymbol{V} \in \mathbb{R}^{k\times d} \\ &\text{subject to} && \pi_{ij}^{(t)}\left(\left\langle \boldsymbol{v}_i - \boldsymbol{v}_j, \boldsymbol{r}\left(\boldsymbol{x}^{(t)}\right)\right\rangle\right) \geq 1 \quad \forall\, 1 \leq i < j \leq k,\ t \in [m] \end{aligned}$$

We previously showed that replacing the rounded feature vectors with the original ones in the above constraints, renders the linear program almost surely feasible with $\lambda\boldsymbol{U}$, for some sufficiently large λ, being a solution. We will now show that LP3 is also feasible with high probability. To this end, it suffices, to upper bound the probability of the event that there exist some i, j, t, such that the rounding procedure places $\boldsymbol{x}_t$ on the wrong (opposite) side of the hyperplane corresponding to $\boldsymbol{u}_i - \boldsymbol{u}_j$. We denote this event by $\mathcal{E}_1$.

By definition of $\boldsymbol{r}$, we infer that for any $\boldsymbol{a} \in S^{d-1}$ and $\boldsymbol{x} \neq \boldsymbol{0}$, if $\boldsymbol{x}$ and $\boldsymbol{r}(\boldsymbol{x})$ do not lie in the same side of the hyperplane defined by $\boldsymbol{a}$, then the distance from $R\boldsymbol{x}/\|\boldsymbol{x}\|_2$ to that hyperplane must be at most ε. This implies that

$$\mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{D}_{\boldsymbol{x}}} \left[\boldsymbol{x} \in \mathcal{R}(\boldsymbol{a})\right] \leq \mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{D}_{\boldsymbol{x}}} \left[|\langle \boldsymbol{a}, R\boldsymbol{x}/\|\boldsymbol{x}\|_2\rangle| \leq \varepsilon \mid \boldsymbol{x} \neq \boldsymbol{0}\right] ,$$

where $\mathcal{R}(\boldsymbol{a}) \triangleq \left\{\boldsymbol{x} \in \mathbb{R}^d : \operatorname{sign}(\langle \boldsymbol{a}, \boldsymbol{r}(\boldsymbol{x})\rangle) \neq \operatorname{sign}(\langle \boldsymbol{a}, \boldsymbol{x}\rangle)\right\}$. Using Lemma A.0.3 and property 3 of Lemma A.0.2 (concentration), we get that, for any $\boldsymbol{a} \in S^{d-1}$, it holds

$$\begin{aligned} \mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{D}_{\boldsymbol{x}}} \left[\boldsymbol{x} \in \mathcal{R}(\boldsymbol{a})\right] &\leq \mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{D}_{\boldsymbol{x}}} \left[|\langle \boldsymbol{a}, R\boldsymbol{x}/\|\boldsymbol{x}\|_2\rangle| \leq \varepsilon \mid \boldsymbol{x} \neq \boldsymbol{0}\right] \\ &\leq \mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{D}_{\boldsymbol{x}}} \left[|\langle \boldsymbol{a}, R\boldsymbol{x}/\|\boldsymbol{x}\|_2\rangle| \leq \varepsilon \wedge \|\boldsymbol{x}\|_2 \leq R \mid \boldsymbol{x} \neq \boldsymbol{0}\right] + \mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{D}_{\boldsymbol{x}}} \left[\|\boldsymbol{x}\|_2 \geq R \mid \boldsymbol{x} \neq \boldsymbol{0}\right] \\ &\leq \mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{D}_{\boldsymbol{x}}} \left[|\langle \boldsymbol{a}, \boldsymbol{x}\rangle| \leq \varepsilon\|\boldsymbol{x}\|_2/R \wedge \|\boldsymbol{x}\|_2/R \leq 1\right] + \mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{D}_{\boldsymbol{x}}} \left[\|\boldsymbol{x}\|_2 \geq R\right] \\ &\leq \mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{D}_{\boldsymbol{x}}} \left[|\langle \boldsymbol{a}, \boldsymbol{x}\rangle| \leq \varepsilon\right] + \mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{D}_{\boldsymbol{x}}} \left[\|\boldsymbol{x}\|_2 \geq R\right] \\ &\leq 2\varepsilon + e^{1-R/\sqrt{d}}\,. \end{aligned}$$

Using the above fact and applying the union bound, we get that

$$\mathbf{Pr}\left[\mathcal{E}_1\right] \leq \mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{D}_{\boldsymbol{x}}}\left[\boldsymbol{x} \in \bigcup_{i<j} \mathcal{R}(\boldsymbol{u}_i - \boldsymbol{u}_j)\right] \leq N\binom{k}{2}\left(2\varepsilon + e^{1-R/\sqrt{d}}\right).$$

Suppose that $\mathcal{E}_1$ does not occur. Then, the rounded points used to construct LP3 can be thought of independent samples drawn from the distribution $\mathcal{D}'_{\boldsymbol{x}}$, which is the conditional distribution of $\boldsymbol{r}(\boldsymbol{x})$, where $\boldsymbol{x} \sim \mathcal{D}_{\boldsymbol{x}}$, given that $\operatorname{sign}\left(\langle \boldsymbol{u}_i - \boldsymbol{u}_j, \boldsymbol{x}\rangle\right) = \operatorname{sign}\left(\langle \boldsymbol{u}_i - \boldsymbol{u}_j, \boldsymbol{r}(\boldsymbol{x})\rangle\right)$ for all $i \neq j$. The distribution $\mathcal{D}'_{\boldsymbol{x}}$ satisfies the property

$$\mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{D}'_{\boldsymbol{x}}}\left[\boldsymbol{x} \in S\right] = \mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{D}_{\boldsymbol{x}}}\left[\boldsymbol{r}(\boldsymbol{x}) \in S \;\middle|\; \boldsymbol{x} \notin \bigcup_{i<j} \mathcal{R}(\boldsymbol{u}_i - \boldsymbol{u}_j)\right] \tag{2}$$

for all measurable $S \subseteq \mathbb{R}^d$.

We fix any $\epsilon, \delta \in (0,1)$. Following the same procedure as in Section 5.1, we deduce that using $N \in \Theta\left(\left(d\log(k/\epsilon) + \log(k/\delta)\right)k^2/\epsilon\right)$ training samples and solving LP3, yields a matrix $\boldsymbol{V}$ such that, with probability at least $1-\delta$, it holds

$$\max_{1\leq i<j\leq k} \mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{D}'_{\boldsymbol{x}}}\left[\operatorname{sign}\left(\langle \boldsymbol{u}_i - \boldsymbol{u}_j, \boldsymbol{x}\rangle\right) \neq \operatorname{sign}\left(\langle \boldsymbol{v}_i - \boldsymbol{v}_j, \boldsymbol{x}\rangle\right)\right] \leq \epsilon\,.$$

From (2), we infer that, with probability at least $1-\delta$, it holds that

$$\mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{D}_{\boldsymbol{x}}}\left[\operatorname{sign}\left(\langle \boldsymbol{u}_i - \boldsymbol{u}_j, \boldsymbol{r}(\boldsymbol{x})\rangle\right) \neq \operatorname{sign}\left(\langle \boldsymbol{v}_i - \boldsymbol{v}_j, \boldsymbol{r}(\boldsymbol{x})\rangle\right) \;\middle|\; \boldsymbol{x} \notin \bigcup_{i<j} \mathcal{R}(\boldsymbol{u}_i - \boldsymbol{u}_j)\right] \leq \epsilon \implies$$

$$\mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{D}_{\boldsymbol{x}}}\left[\operatorname{sign}\left(\langle \boldsymbol{u}_i - \boldsymbol{u}_j, \boldsymbol{x}\rangle\right) \neq \operatorname{sign}\left(\langle \boldsymbol{v}_i - \boldsymbol{v}_j, \boldsymbol{r}(\boldsymbol{x})\rangle\right) \;\middle|\; \boldsymbol{x} \notin \bigcup_{i<j} \mathcal{R}(\boldsymbol{u}_i - \boldsymbol{u}_j)\right] \leq \epsilon \implies$$

$$\mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{D}_{\boldsymbol{x}}}\left[\operatorname{sign}\left(\langle \boldsymbol{u}_i - \boldsymbol{u}_j, \boldsymbol{x}\rangle\right) \neq \operatorname{sign}\left(\langle \boldsymbol{v}_i - \boldsymbol{v}_j, \boldsymbol{r}(\boldsymbol{x})\rangle\right)\right] \leq \epsilon + \mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{D}_{\boldsymbol{x}}}\left[\boldsymbol{x} \in \bigcup_{i<j} \mathcal{R}(\boldsymbol{u}_i - \boldsymbol{u}_j)\right]$$

for all $1 \leq i < j \leq k$. Moreover, we have that

$$\mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{D}_{\boldsymbol{x}}}\left[\operatorname{sign}\left(\langle \boldsymbol{v}_i - \boldsymbol{v}_j, \boldsymbol{x}\rangle\right) \neq \operatorname{sign}\left(\langle \boldsymbol{v}_i - \boldsymbol{v}_j, \boldsymbol{r}(\boldsymbol{x})\rangle\right)\right] = \mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{D}_{\boldsymbol{x}}}\left[\boldsymbol{x} \in \mathcal{R}(\boldsymbol{v}_i - \boldsymbol{v}_j)\right].$$

for all $1 \leq i < j \leq k$. Therefore, by application of the triangle inequality, we get that, with probability at least $1-\delta$, it holds

$$\max_{1\leq i<j\leq k} \mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{D}_{\boldsymbol{x}}}\left[\operatorname{sign}\left(\langle \boldsymbol{u}_i - \boldsymbol{u}_j, \boldsymbol{x}\rangle\right) \neq \operatorname{sign}\left(\langle \boldsymbol{v}_i - \boldsymbol{v}_j, \boldsymbol{x}\rangle\right)\right] \leq \epsilon + \left(\binom{k}{2} + 1\right)\left(2\varepsilon + e^{1-R/\sqrt{d}}\right)$$

We denote the complement of the above event by $\mathcal{E}_2$. Taking a union bound over $\mathcal{E}_1$ and $\mathcal{E}_1^c \cap \mathcal{E}_2$ we finally conclude that the above procedure, with probability at least

$$1 - \mathbf{Pr}[\mathcal{E}_1 \cup (\mathcal{E}_1^c \cap \mathcal{E}_2)] \geq 1 - \delta - N\binom{k}{2}\left(2\varepsilon + e^{1-R/\sqrt{d}}\right),$$

yields a matrix $\boldsymbol{V}$ such that

$$\max_{1\leq i<j\leq k} \mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{D}_{\boldsymbol{x}}}\left[\operatorname{sign}\left(\langle \boldsymbol{u}_i - \boldsymbol{u}_j, \boldsymbol{x}\rangle\right) \neq \operatorname{sign}\left(\langle \boldsymbol{v}_i - \boldsymbol{v}_j, \boldsymbol{x}\rangle\right)\right] \leq \epsilon + \left(\binom{k}{2} + 1\right)\left(2\varepsilon + e^{1-R/\sqrt{d}}\right).$$

Choosing $\delta < 1/2$ and setting

$$\varepsilon \leq \frac{\min\{\epsilon, \delta\}}{4N\binom{k}{2}} \tag{3}$$

and

$$R \geq \sqrt{d}\ln\frac{2eN\binom{k}{2}}{\min\{\epsilon, \delta\}} \tag{4}$$

we get that, with probability at least $1 - 2\delta > 0$, it holds

$$\max_{1 \le i < j \le k} \mathop{\mathbf{Pr}}_{\boldsymbol{x} \sim \mathcal{D}_{\boldsymbol{x}}} \left[\operatorname{sign} \left(\langle \boldsymbol{u}_i - \boldsymbol{u}_j, \boldsymbol{x} \rangle \right) \neq \operatorname{sign} \left(\langle \boldsymbol{v}_i - \boldsymbol{v}_j, \boldsymbol{x} \rangle \right) \right] \le 2\epsilon \,,$$

which proves property 1 of the lemma.

To prove properties 2 and 3, we leverage the fact that LP3 is constructed using rounded samples. Specifically, let $\mathcal{G}^d(\gamma) = \{\dots, -2\gamma, -\gamma, 0, \gamma, 2\gamma, \dots\}^d$. Choosing $\mathcal{C} = \mathcal{G}^d\left(\varepsilon/\sqrt{d}\right) \cap B^d(R)$, we get that $\mathcal{C}$ satisfies (1). Moreover, given the aforementioned constraints for ε and R (see (3) and (4)), we deduce that the elements of $\mathcal{C}$ can be represented using $\operatorname{poly}(d, k, 1/\epsilon)$ bits. Also, recall that $\pi_{ij}^{(t)} \in \{\pm 1\}$ for all $1 \le i < j \le k$ and $t \in [N]$. Hence, we infer that the constraints of LP3 can take the form $\boldsymbol{A}\boldsymbol{x} \le \boldsymbol{b}$, where $\boldsymbol{A} \in \mathbb{Z}^{N\binom{k}{2} \times kd}$, $\boldsymbol{b} \in \mathbb{Z}^{N\binom{k}{2}}$ and $\boldsymbol{x}$ contains the unknown variables[1].

Lemma C.3.2 (Schrijver [1986])**.** *Fix any $\boldsymbol{A} \in \mathbb{Z}^{m \times n}$, $\boldsymbol{b} \in \mathbb{Z}^m$ and $\mathbf{c} \in \mathbb{Z}^n$ and consider the linear program $\min\langle \boldsymbol{c}, \boldsymbol{x} \rangle$ subject to $\boldsymbol{A}\boldsymbol{x} \le \boldsymbol{b}$. Let U be the maximum size of a_{ij}, b_i, c_j, where $i \in [m]$ and $j \in [n]$. The output of the linear program has size $O(mnU + mn \log n)$ bits.*

Using the above facts and Lemma C.3.2, we get that any solution of LP3 is a matrix $\boldsymbol{V}$ with elements with minimum nonzero absolute value $2^{-\operatorname{poly}(d,k,1/\epsilon)}$ and maximum absolute value $2^{\operatorname{poly}(d,k,1/\epsilon)}$. Without loss of generality we assume that $\|\boldsymbol{V}\|_{\mathrm{F}} \le 1$ (we can always divide $\boldsymbol{V}$ by its Frobenius norm and the resultant matrix will still satisfy property 1 of the lemma and will still consist of elements with minimum nonzero absolute value $2^{-\operatorname{poly}(d,k,1/\epsilon)}$). Then, using the fact that $\boldsymbol{v}_i \neq \boldsymbol{v}_j$ (as the constraints of LP3 demand) for $i \neq j$, we get that $\min_{1 \le i < j \le k} \|\boldsymbol{v}_i - \boldsymbol{v}_j\|_2 \ge 2^{-\operatorname{poly}(d,k,1/\epsilon)}$. These facts conclude the proof. □

[1] Strictly speaking, the elements of $\boldsymbol{A}$, $\boldsymbol{b}$ and $\boldsymbol{c}$ are rational numbers, but can easily be converted to integers (by multiplying every constraint's coefficients with the least common multiple of the their denominators).