Linear Label Ranking with Bounded Noise

Dimitris Fotakis
NTUA
fotakis@cs.ntua.gr

Alkis Kalavasis
NTUA
kalavasisalkis@mail.ntua.gr

Vasilis Kontonis
UW Madison
kontonis@wisc.edu

Christos Tzamos
UW Madison
tzamos@wisc.edu

Abstract

Label Ranking (LR) is the supervised task of learning a sorting function that maps feature vectors $\boldsymbol{x} \in \mathbb{R}^d$ to rankings $\sigma(\boldsymbol{x}) \in \mathbb{S}_k$ over a finite set of k labels. We focus on the fundamental case of learning linear sorting functions (LSFs) under Gaussian marginals: $\boldsymbol{x}$ is sampled from the d-dimensional standard normal and the ground truth ranking $\sigma^\star(\boldsymbol{x})$ is the ordering induced by sorting the coordinates of the vector $\boldsymbol{W}^\star\boldsymbol{x}$, where $\boldsymbol{W}^\star \in \mathbb{R}^{k\times d}$ is unknown. We consider learning LSFs in the presence of bounded noise: assuming that a noiseless example is of the form $(\boldsymbol{x}, \sigma^\star(\boldsymbol{x}))$, we observe $(\boldsymbol{x}, \pi)$, where for any pair of elements $i \neq j$, the probability that the order of i, j is different in π than in $\sigma^\star(\boldsymbol{x})$ is *at most* $\eta < 1/2$. We design efficient non-proper and proper learning algorithms that learn hypotheses within normalized Kendall's Tau distance ϵ from the ground truth with $N = \widetilde{O}(d\log(k)/\epsilon)$ labeled examples and runtime $\mathrm{poly}(N, k)$. For the more challenging top-r disagreement loss, we give an efficient proper learning algorithm that achieves ϵ top-r disagreement with the ground truth with $N = \widetilde{O}(dkr/\epsilon)$ samples and $\mathrm{poly}(N)$ runtime.

1 Introduction

1.1 Background and Motivation

Label Ranking (LR) is the problem of learning a hypothesis that maps features to rankings over a finite set of labels. Given a feature vector $\boldsymbol{x} \in \mathbb{R}^d$, a sorting function $\sigma(\cdot)$ maps it to a ranking of k alternatives, i.e., $\sigma(\boldsymbol{x})$ is an element of the symmetric group with k elements, $\mathbb{S}_k$. Assuming access to a training dataset of features labeled with their corresponding rankings, i.e., pairs of the form $(\boldsymbol{x}, \pi) \in \mathbb{R}^d \times \mathbb{S}_k$, the goal of the learner is to find a sorting function $h(\boldsymbol{x})$ that generalizes well over a fresh sample. LR has received significant attention over the years [DSM03, SS07, HFCB08, CH08, FHMB08] due to the large number of applications. For example, ad targeting [DGR$^+$14] is an LR instance where for each user we want to use their feature vector to predict a ranking over ad categories and present them with the most relevant. The practical significance of LR has lead to the development of many techniques based on probabilistic models and instance-based methods [CH08, CDH10], [GDV12, ZLGQ14], decision trees [CHH09], entropy-based ranking trees [RdSRSK15], bagging [AGM17], and random forests [dSSKC17, ZQ18]. However, almost all of these works come without provable guarantees and/or fail to learn in the presence of noise in the observed rankings.

Linear Sorting Functions (LSFs). In this work, we focus on the fundamental concept class of Linear Sorting functions [HPRZ03]. A linear sorting function parameterized by a matrix $\boldsymbol{W} \in \mathbb{R}^{k\times d}$ with k rows $\boldsymbol{W}_1, \ldots, \boldsymbol{W}_k$ takes a feature $\boldsymbol{x} \in \mathbb{R}^d$, maps it to $\boldsymbol{W}\boldsymbol{x} = (\boldsymbol{W}_1 \cdot \boldsymbol{x}, \ldots, \boldsymbol{W}_k \cdot \boldsymbol{x}) \in \mathbb{R}^k$ and

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

then outputs an ordering $(i_1, \ldots, i_k)$ of the k alternatives such that $\boldsymbol{W}_{i_1} \cdot \boldsymbol{x} \geq \boldsymbol{W}_{i_2} \cdot \boldsymbol{x} \geq \ldots \geq \boldsymbol{W}_{i_k} \cdot \boldsymbol{x}$. In other words, a linear sorting function ranks the k alternatives (corresponding to rows of $\boldsymbol{W}$) with respect to how well they correlate with the feature $\boldsymbol{x}$. We denote a linear sorting function with parameter $\boldsymbol{W} \in \mathbb{R}^{k \times d}$ by $\sigma_{\boldsymbol{W}}(\boldsymbol{x}) \triangleq \mathrm{argsort}(\boldsymbol{W}\boldsymbol{x})$ where $\mathrm{argsort} : \mathbb{R}^k \to \mathbb{S}_k$ takes as input a vector $(v_1, \ldots, v_k) \in \mathbb{R}^k$, sorts it in decreasing order to obtain $v_{i_1} \geq v_{i_2} \geq \ldots \geq v_{i_k}$ and returns the ordering $(i_1, \ldots, i_k)$.

Noisy Ranking Distributions. Learning LSFs in the noiseless setting can be done efficiently by using linear programming. However, the common assumption both in theoretical and in applied works is that the observed rankings are noisy in the sense that they do not always correspond to the ground-truth ranking. We assume that the probability that the order of two elements i, j in the observed ranking π is different than their order in the ground-truth ranking $\sigma^\star$ is at most $\eta < 1/2$.

Definition 1 (Noisy Ranking Distribution)**.** *Fix $\eta \in [0, 1/2)$. An η-**noisy ranking distribution** $\mathcal{M}(\sigma^\star)$ with ground-truth ranking $\sigma^\star \in \mathbb{S}_k$ is a probability measure over $\mathbb{S}_k$ that, for any $i, j \in [k]$, with $i \neq j$, satisfies $\mathbf{Pr}_{\pi \sim \mathcal{M}(\sigma^\star)}[i \prec_\pi j \mid i \succ_{\sigma^\star} j] \leq \eta$.* [1]

Note that, when $\eta = 0$, we always observe the ground-truth permutation and, in the case of $\eta = 1/2$, we may observe a uniformly random permutation. We remark that most natural ranking distributions satisfy this bounded noise property, e.g., (i) the Mallows model, which is probably the most fundamental ranking distribution (see, e.g., [BM09, LB11, CPS13, ABSV14, BFFSZ19, FKS21, DOS18, LM18, MW20, LM21] for a small sample of this line of research) and (ii) the Bradley-Terry-Mallows model [Mal57], which corresponds to the ranking distribution analogue of the Bradley-Terry-Luce model [BT52, Luc12] (the most studied pairwise comparisons model; see, e.g., [Hun04, NOS17, APA18] and the references therein). For more details, see Appendix E.

We consider the fundamental setting where the feature vector $\boldsymbol{x} \in \mathbb{R}^d$ is generated by a standard normal distribution and the ground-truth ranking for each sample $\boldsymbol{x}$ is given by the LSF $\sigma_{\boldsymbol{W}^\star}(\boldsymbol{x})$ for some unknown parameter matrix $\boldsymbol{W}^\star \in \mathbb{R}^{k \times d}$. For a fixed $\boldsymbol{x}$, the ranking that we observe comes from an η-noisy ranking distribution with ground-truth ranking $\sigma_{\boldsymbol{W}^\star}(\boldsymbol{x})$.

Definition 2 (Noisy Linear Label Ranking Distribution)**.** *Fix $\eta \in [0, 1/2)$ and some ground-truth parameter matrix $\boldsymbol{W}^\star \in \mathbb{R}^{d \times k}$. We assume that the η-**noisy linear label ranking distribution** $\mathcal{D}$ over $\mathbb{R}^d \times \mathbb{S}_k$ satisfies the following:*

1. *The $\boldsymbol{x}$-marginal of $\mathcal{D}$ is the d-dimensional standard normal distribution.*
2. *For any $(\boldsymbol{x}, \pi) \sim \mathcal{D}$, the distribution of π conditional on $\boldsymbol{x}$ is an η-noisy ranking distribution with ground-truth ranking $\sigma_{\boldsymbol{W}^\star}(\boldsymbol{x})$.*

At first sight, the assumption that the underlying $\boldsymbol{x}$-marginal is the standard normal may look too strong. However, for $k = 2$, Definition 2 captures the problem of learning linear threshold functions with Massart noise. Without assumptions for the $\boldsymbol{x}$-marginal, it is known [DGT19, CKMY20, DK20, NT22] that optimal learning of halfspaces under Massart noise requires super-polynomial time (in the Statistical Query model of [Kea98]). On the other hand, a lot of recent works [BZ17, MV19, DKTZ20, ZSA20, ZL21] have obtained efficient algorithms for learning Massart halfspaces under Gaussian marginals. The goal of this work is to provide efficient algorithms for the more general problem of learning LSFs with bounded noise under Gaussian marginals.

1.2 Our Results

The main contributions of this paper are the first efficient algorithms for learning LSFs with bounded noise with respect to Kendall's Tau distance and top-r disagreement loss.

Learning in Kendall's Tau Distance. The most standard metric in rankings [SSBD14] is Kendall's Tau (KT) distance which, for two rankings $\pi, \tau \in \mathbb{S}_k$, measures the fraction of pairs (i, j) on which they disagree. That is, $\Delta_{\mathrm{KT}}(\pi, \tau) = \sum_{i \prec_\pi j} \mathbf{1}\{i \succ_\tau j\} / \binom{k}{2}$. Our first result is an efficient learning algorithm that, given samples from an η-noisy linear label ranking distribution $\mathcal{D}$, computes

[1] We use $i \succ_\pi j$ (resp. $i \prec_\pi j$) to denote that the element i is ranked higher (resp. lower) than j according to the ranking π.

a parameter matrix $\boldsymbol{W}$ that ranks the alternatives almost optimally with respect to the KT distance from the ground-truth ranking $\sigma_{\boldsymbol{W}^\star}(\cdot)$.

Theorem 1 (Learning LSFs in KT Distance)**.** *Fix $\eta \in [0, 1/2)$ and $\epsilon, \delta \in (0, 1)$. Let $\mathcal{D}$ be an η-noisy linear label ranking distribution satisfying the assumptions of Definition 2 with ground-truth LSF $\sigma_{\boldsymbol{W}^\star}(\cdot)$. There exists an algorithm that draws $N = \widetilde{O}\left(\frac{d}{\epsilon(1-2\eta)^6}\log(k/\delta)\right)$ samples from $\mathcal{D}$, runs in sample-polynomial time, and computes a matrix $\boldsymbol{W} \in \mathbb{R}^{k\times d}$ such that, with probability at least $1-\delta$,*

$$\mathop{\mathbf{E}}_{\boldsymbol{x}\sim\mathcal{N}_d}[\Delta_{\mathrm{KT}}(\sigma_{\boldsymbol{W}}(\boldsymbol{x}), \sigma_{\boldsymbol{W}^\star}(\boldsymbol{x}))] \leq \epsilon\,.$$

Theorem 1 gives the first efficient algorithm with provable guarantees for the supervised problem of learning noisy linear rankings. We remark that the sample complexity of our learning algorithm is qualitatively optimal (up to logarithmic factors) since, for $k = 2$, our problem subsumes learning a linear classifier with Massart noise [2] for which $\Omega(d/\epsilon)$ are known to be information theoretically necessary [MN06]. Moreover, our learning algorithm is *proper* in the sense that it computes a linear sorting function $\sigma_{\boldsymbol{W}}(\cdot)$. As opposed to improper learners (see also Section 1.3), a proper learning algorithm gives us a compact representation (storing $\boldsymbol{W}$ requires $O(kd)$ memory) of the sorting function that allows us to efficiently compute (with runtime $O(kd + k\log k)$) the ranking corresponding to a fresh datapoint $\boldsymbol{x} \in \mathbb{R}^d$.

Learning in top-r Disagreement. We next present our learning algorithm for the top-r metric formally defined as $\Delta_{\mathrm{top}-r}(\pi, \tau) = \mathbf{1}\{\pi_{1..r} \neq \tau_{1..r}\}$, where by $\pi_{1..r}$ we denote the ordering on the first r elements of the permutation π. The top-r metric is a disagreement metric in the sense that it takes binary values and for $r = 1$ captures the standard (multiclass) top-1 classification loss. We remark that, in contrast with the top-r classification loss, which only requires the predicted label to be in the top-r predictions of the model, the top-r ranking metric that we consider here requires that the model puts *the same elements in the same order* as the ground truth in the top-r positions. The top-r ranking is well-motivated as, for example, in ad targeting (discussed in Section 1.1) we want to be accurate on the top-r ad categories for a user so that we can diversify the content that they receive.

Theorem 2 (Learning LSFs in top-r Disagreement)**.** *Fix $\eta \in [0, 1/2)$, $r \in [k]$ and $\epsilon, \delta \in (0, 1)$. Let $\mathcal{D}$ be an η-noisy linear label ranking distribution satisfying the assumptions of Definition 2 with ground-truth LSF $\sigma_{\boldsymbol{W}^\star}(\cdot)$. There exists an algorithm that draws $N = \widetilde{O}\left(\frac{drk}{\epsilon(1-2\eta)^6}\log(1/\delta)\right)$ samples from $\mathcal{D}$, runs in sample-polynomial time and computes a matrix $\boldsymbol{W} \in \mathbb{R}^{k\times d}$ such that, with probability at least $1-\delta$,*

$$\mathop{\mathbf{E}}_{\boldsymbol{x}\sim\mathcal{N}_d}[\Delta_{\mathrm{top}-r}(\sigma_{\boldsymbol{W}}(\boldsymbol{x}), \sigma_{\boldsymbol{W}^\star}(\boldsymbol{x}))] \leq \epsilon\,.$$

As a direct corollary of our result, we obtain a proper algorithm for learning the top-1 element with respect to the standard 0-1 loss that uses $\widetilde{O}(kd)$ samples. In fact, for small values of r, i.e., $r = O(1)$, our sample complexity is essentially tight. It is known that $\Theta(kd)$ samples are information theoretically necessary [Nat89] for top-1 classification. [3] For the case $r = k$, i.e., when we want to learn the whole ranking with respect to the 0-1 loss, our sample complexity is $O(k^2 d)$. However, using arguments similar to [DSBDSS11], one can show that in fact $O(dk)$ ranking samples are sufficient in order to learn the whole ranking with respect to the 0-1 loss. In this case, it is unclear whether a better sample complexity can be achieved with an efficient algorithm and we leave this as an interesting open question for future work.

1.3 Our Techniques

Learning in Kendall's Tau distance. Our proper learning algorithm consists of two steps: an improper learning algorithm that decomposes the ranking problem to $O(k^2)$ binary linear classification problems and a convex (second order conic) program that "compresses" the k^2 linear classifiers

[2] Notice that in this case Kendall's Tau distance is simply the standard 0-1 binary loss.

[3] Strictly speaking, those lower bounds do not directly apply in our setting because our labels are whole rankings instead of just the top classes but, in the Appendix D, we show that we can adapt the lower bound technique of [DSBDSS11] to obtain the same sample complexity lower bound for our ranking setting.

to obtain a $k \times d$ matrix $\boldsymbol{W}$. Our improper learning algorithm splits the ranking learning problem into $O(k^2)$ binary, d-dimensional linear classification problems with Massart noise. In particular, for every pair of elements $i, j \in [k]$, each binary classification task asks whether element i is ranked higher than element j in the ground-truth permutation $\sigma_{\boldsymbol{W}^\star}(\boldsymbol{x})$. As we already discussed, we have that, under the Gaussian distribution, there exist efficient Massart learning algorithms [BZ17, MV19, DKTZ20, ZSA20, ZL21] that can recover linear classifiers $\mathrm{sgn}(\boldsymbol{v}_{ij} \cdot \boldsymbol{x})$ that correctly order the pair i, j for all $\boldsymbol{x}$ apart from a region of $O(\epsilon)$-Gaussian mass. However, we still need to aggregate the results of the *approximate* binary classifiers in order to obtain a ranking of the k alternatives for each $\boldsymbol{x}$. We first show that we can design a "voting scheme" that combines the results of the binary classifiers using an efficient constant factor approximation algorithm for the Minimum Feedback Arc Set (MFAS) problem [ACN08]. This gives us an efficient but improper algorithm for learning LSFs in Kendall's Tau distance. In order to obtain a proper learning algorithm, we further "compress" the $O(k^2)$ approximate linear classifiers with normal vectors $\boldsymbol{v}_{ij}$ and obtain a matrix $\boldsymbol{W} \in \mathbb{R}^{k \times d}$ with the property that the difference of every two rows $\boldsymbol{W}_i - \boldsymbol{W}_j$ is $O(\epsilon)$-close to the vector $\boldsymbol{v}_{ij}$. More precisely, we show that, given the linear classifiers $\boldsymbol{v}_{ij} \in \mathbb{R}^d$, we can efficiently compute a matrix $\boldsymbol{W} \in \mathbb{R}^{k \times d}$ such that the following angle distance with $\boldsymbol{W}^\star$ is small:

$$d_{\mathrm{angle}}(\boldsymbol{W}, \boldsymbol{W}^\star) \triangleq \max_{i,j} \theta(\boldsymbol{W}_i - \boldsymbol{W}_j, \boldsymbol{W}_i^\star - \boldsymbol{W}_j^\star) \leq O(\epsilon) \,. \tag{1}$$

It is not hard to show that, as long as the above angle metric is at most $O(\epsilon)$, then (in expectation over the standard Gaussian) Kendall's Tau distance between the LSFs is also $O(\epsilon)$. A key technical difficulty that we face in this reduction is bounding the "condition number" of the convex (second order conic) program that finds the matrix $\boldsymbol{W}$ given the vectors $\boldsymbol{v}_{ij}$, see Claim 2. Finally, we remark that the proper learning algorithm of Theorem 1 results in a compact and efficient sorting function that requires: (i) storing $O(k)$ weight vectors as opposed to the initial $O(k^2)$ vectors of the improper learner; and (ii) evaluating k inner products with $\boldsymbol{x}$ to find its ranking (instead of $O(k^2)$).

Learning in top-r Disagreement. We next turn our attention to the more challenging top-r ranking disagreement metric. In particular, suppose that we are interested in recovering only the top element of the ranking. One approach would be to directly use the improper learning algorithm for this task and ask for KT distance of order roughly ϵ/k^2. The resulting hypothesis would produce good predictions for the top element but the required sample complexity would be $O(dk^2)$. While it seems that training $O(k^2)$ d-dimensional binary classifiers inherently requires $O(dk^2)$ samples, we show that, using the proper KT distance learning algorithm of Theorem 1, we can also obtain improved sample complexity results for the top-r metric. Our main technical contribution here is a novel estimate of the top-r disagreement in terms of the angle metric. In general, one can show that the top-r disagreement is at most $O(k^2)\ d_{\mathrm{angle}}(\boldsymbol{W}, \boldsymbol{W}^\star)$. We significantly sharpen this estimate by showing the following lemma.

Lemma 1 (Top-r Disagreement via Parameter Distance)**.** *Consider two matrices $\boldsymbol{W}, \boldsymbol{W}^\star \in \mathbb{R}^{k \times d}$ and let $\mathcal{N}_d$ be the standard Gaussian in d dimensions. We have that*

$$\mathop{\mathbf{Pr}}_{\boldsymbol{x} \sim \mathcal{N}_d}[\sigma_{1..r}(\boldsymbol{W}\boldsymbol{x}) \neq \sigma_{1..r}(\boldsymbol{W}^\star \boldsymbol{x})] \leq \widetilde{O}(kr)\ d_{\mathrm{angle}}(\boldsymbol{W}, \boldsymbol{W}^\star) \,.$$

We remark that Lemma 1 is a general geometric tool that we believe will be useful in other distribution-specific multiclass learning settings. The proof of Lemma 1 mainly relies on geometric Gaussian surface area computations that we believe are of independent interest. For the details, we refer the reader to Section 4. An interesting question with a convex-geometric flavor is whether the sharp bound of Lemma 1 also holds under the more general class of isotropic log-concave distributions.

1.4 Related Work

Robust Supervised Learning. We start with a summary of prior work on PAC learning with Massart noise. The Massart noise model was formally defined in [MN06] but similar variants had been defined by Vapnik, Sloan and Rivest [Vap06, Slo88, Slo92, RS94, Slo96]. This model is a strict extension of the Random Classification Noise (RCN) model [AL88], where the label noise is uniform, i.e., context-independent and is a special case of the agnostic model [Hau18, KSS94], where the label noise is fully adversarial and computational barriers are known to exist [GR09, FGKP06, Dan16, DKZ20, GGK20, DKPZ21, HSSVG22]. Our work partially builds upon on the algorithmic task of

PAC learning halfspaces with Massart noise [BH20]. In the distribution-independent setting, known efficient algorithms [DGT19, CKMY20, DKT21] achieve error $\eta+\epsilon$ and the works of [DK20, NT22] indicate that this error bound is the best possible in the Statistical Query model [Kea98]. This lower bound motivates the study of the distribution-specific setting (which is also the case of our work). There is an extensive line of work in this direction: [ABHU15, ABHZ16, YZ17, ZLC17, BZ17, MV19, DKTZ20, ZSA20, ZL21] with the currently best algorithms succeeding for all $\eta < 1/2$ with a sample and computational complexity $\mathrm{poly}(d, 1/\epsilon, 1/(1-2\eta))$ under a class of distributions including isotropic log-concave distributions. For details, see [DKK+21]. In this work we focus on Gaussian marginals but some of our results extend to larger distribution classes.

Label Ranking. Our work lies in the area of Label Ranking, which has received significant attention over the years [SS07, HFCB08, CH08, HPRZ03, FHMB08, DSM03]. There are multiple approaches for tackling this problem (see [VG10], [ZLY+14]). Some of them are based on probabilistic models [CH08, CDH10, GDV12, ZLGQ14] or may be tree based, such as decision trees [CHH09], entropy based ranking trees and forests [RdSRSK15, dSSKC17], bagging techniques [AGM17] and random forests [ZQ18]. There are also works focusing on supervised clustering [GDGV13]. Finally, [CH08, CDH10, CHH09] adopt an instance-based approaches using nearest neighbors approaches. The above results are industrial. From a theoretical perspective, LR has been mainly studied from a statistical learning theory framework [CV20, CKS18, KGB18, KCS17]. [FKP21] provide some computational guarantees for the performance of decision trees in the noiseless case and some experimental results on the robustness of random forests to noise. The setting of [DGR+14] is close to ours but is investigated from an experimental standpoint. We remark that while reducing LR to multiple binary classification tasks has been used in prior literature [HFCB08, CH12, FKP21], standard reductions can not tolerate noise in rankings (nevertheless, from an experimental perspective, e.g., random forests seem robust to noise but lack formal theoretical guarantees). Our reduction crucially relies on the existence of efficient learning algorithms for binary linear classification with Massart noise.

2 Notation and Preliminaries

General Notation. We use $\widetilde{O}(\cdot)$ to omit poly-logarithmic factors. A learning algorithm has sample-polynomial runtime if it runs in time polynomial in the size of the description of the input training set. We denote vectors by boldface $\boldsymbol{x}$ (with elements x_i) and matrices with $\boldsymbol{W}$, where we let $\boldsymbol{W}_i \in \mathbb{R}^d$ denote the i-th row of $\boldsymbol{W} \in \mathbb{R}^{k\times d}$ and W_{ij} its elements. We denote $\boldsymbol{a}\cdot\boldsymbol{b}$ the inner product of two vectors and $\theta(\boldsymbol{a},\boldsymbol{b})$ their angle. Let $\mathcal{N}_d$ denote the d-dimensional standard normal and $\Gamma(\cdot)$ the Gaussian surface area.

Rankings. We let $\mathrm{argsort}_{i\in[k]}\boldsymbol{v}$ denote the ranking of $[k]$ in decreasing order according to the values of $\boldsymbol{v}$. For a ranking π, we let $\pi(i)$ denote the position of the i-th element. If $\pi = \pi(\boldsymbol{x})$, we may also write $\pi(\boldsymbol{x})(i)$ to denote the position of i. We often refer to the elements of a ranking as *alternatives*. For a ranking σ, we let $\sigma_{1..r}$ denote the top-r part of σ. When $\sigma = \sigma(\boldsymbol{x})$, we may also write $\sigma_{1..r}(\boldsymbol{x})$ and $\sigma_\ell(\boldsymbol{x})$ will be the alternative at the ℓ-th position. We let Δ_{KT} denote the (normalized) KT distance, i.e., $\Delta_{\mathrm{KT}}(\pi,\tau) = \sum_{i\prec_\pi j}\mathbf{1}\{i\succ_\tau j\}/\binom{k}{2}$ for $\pi,\tau\in\mathbb{S}_k$.

3 Learning in KT distance: Theorem 1

In this section, we present the main tools required to obtain our proper learning algorithm of Theorem 1. Our proper algorithm adopts a two-step approach: it first invokes an efficient *improper* algorithm which, instead of a linear sorting function (i.e., a matrix $\boldsymbol{W}\in\mathbb{R}^{k\times d}$), outputs a list of $O(k^2)$ linear classifiers. We then design a novel convex program in order to find the matrix $\boldsymbol{W}$ satisfying the guarantees of Theorem 1. Let us begin with the improper learner for LSFs with bounded noise with respect to the KT distance, whose description can be found in Algorithm 1.

3.1 Improper Learning Algorithm

Let us assume that the target function is $\sigma^\star(\boldsymbol{x}) = \sigma_{\boldsymbol{W}^\star}(\boldsymbol{x}) = \mathrm{argsort}(\boldsymbol{W}^\star\boldsymbol{x})$ for some $\boldsymbol{W}^\star\in\mathbb{R}^{k\times d}$.

Step 1: Binary decomposition and Noise Structure. For each drawn example $(\boldsymbol{x},\pi)$ from the η-noisy linear label ranking distribution $\mathcal{D}$ (see Definition 2), we create $\binom{k}{2}$ binary examples $(\boldsymbol{x}, y_{ij})$

Algorithm 1 Non-proper Learning Algorithm `ImproperLSF`

`Input`: Training set $T = \{(\boldsymbol{x}^t, \pi^t)\}_{t\in[N]}, \epsilon, \delta \in (0,1), \eta \in [0, 1/2)$
`Output`: Sorting function $h : \mathbb{R}^d \to \mathbb{S}_k$

For any $1 \le i < j \le k$, create $T_{ij} = \{(\boldsymbol{x}^t, \mathrm{sgn}(\pi^t(i) - \pi^t(j)))\}$
For any $1 \le i < j \le k$, compute $\boldsymbol{v}_{ij} = \texttt{MassartLTF}(T_{ij}, \frac{\epsilon}{4}, \frac{\delta}{10k^2}, \eta)$ ▷ See Appendix A.1
`Ranking Phase`: Given $\boldsymbol{x} \in \mathbb{R}^d$:
(a) Construct directed graph G with $V(G) = [k]$ and edges $e_{i\to j}$ only if $\boldsymbol{v}_{ij} \cdot \boldsymbol{x} > 0 \; \forall i \neq j$
(b) Output $h(\boldsymbol{x}) = \texttt{MFAS}(G)$ ▷ See Appendix A.1

with $y_{ij} = \mathrm{sgn}(\pi(i) - \pi(j))$ for any $1 \le i < j \le k$. We have that

$$\mathop{\mathbf{Pr}}_{(\boldsymbol{x},\pi)\sim\mathcal{D}} \left[y_{ij} \cdot \mathrm{sgn}((\boldsymbol{W}_i^\star - \boldsymbol{W}_j^\star) \cdot \boldsymbol{x}) < 0 \mid \boldsymbol{x} \right] = \mathop{\mathbf{Pr}}_{\pi\sim\mathcal{M}(\sigma^\star(\boldsymbol{x}))} \left[\pi(i) < \pi(j) \mid \boldsymbol{W}_i^\star \cdot \boldsymbol{x} < \boldsymbol{W}_j^\star \cdot \boldsymbol{x} \right] .$$

Since $\mathcal{M}(\sigma^\star(\boldsymbol{x}))$ is an η-noisy ranking distribution (see Definition 1), we get that the above quantity is at most $\eta < 1/2$. Therefore, each sample $(\boldsymbol{x}, y_{ij})$ can be viewed as a sample from a distribution $\mathcal{D}_{ij}$ with Gaussian $\boldsymbol{x}$-marginal, optimal linear classifier $\mathrm{sgn}((\boldsymbol{W}_i^\star - \boldsymbol{W}_j^\star) \cdot \boldsymbol{x})$, and Massart noise η. Hence, we have reduced the task of learning noisy LSFs to a number of $\binom{k}{2}$ sub-problems concerning the learnability of halfspaces in the presence of bounded (Massart) noise.

Step 2: Solving Binary Sub-problems. We can now apply the algorithm `MassartLTF` for LTFs with Massart noise under standard Gaussian marginals [ZSA20] (for details, see Appendix A.1): for all the pairs of alternatives $1 \le i < j \le k$ with accuracy parameter ϵ', confidence $\delta' = O(\delta/k^2)$, and a total number of $N = \widetilde{\Omega}\left(\frac{d}{\epsilon'(1-2\eta)^6} \log(k/\delta)\right)$ i.i.d. samples from $\mathcal{D}$, we can obtain a collection of linear classifiers with normal vectors $\boldsymbol{v}_{ij}$ for any $i < j$. We remark that each one of these halfspaces $\boldsymbol{v}_{ij}$ achieves ϵ disagreement with the ground-truth halfspaces $\boldsymbol{W}_i^\star - \boldsymbol{W}_j^\star$ with high probability, i.e.,

$$\mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{N}_d} [\mathrm{sgn}(\boldsymbol{v}_{ij} \cdot \boldsymbol{x}) \neq \mathrm{sgn}((\boldsymbol{W}_i^\star - \boldsymbol{W}_j^\star) \cdot \boldsymbol{x})] \le \epsilon' .$$

Step 3: Ranking Phase. We now have to aggregate the linear classifiers and compute a single sorting function $h : \mathbb{R}^d \to \mathbb{S}_k$. Given an example $\boldsymbol{x}$, we create the tournament graph G with k nodes that contains a directed edge $e_{i\to j}$ if $\boldsymbol{v}_{ij} \cdot \boldsymbol{x} > 0$. If G is acyclic, we output the induced permutation; otherwise, the graph contains cycles which should be eliminated. In order to output a ranking, we remove cycles from G with an efficient, 3-approximation algorithm for MFAS [ACN08, VZW09]. Hence, the output $h(\boldsymbol{x})$ and the true target $\sigma^\star(\boldsymbol{x})$ will have $\mathbf{E}_{\boldsymbol{x}\sim\mathcal{N}_d}[\Delta_{\mathrm{KT}}(h(\boldsymbol{x}), \sigma^\star(\boldsymbol{x}))] \le \epsilon' + 3\epsilon' = 4\epsilon'$. This last equation indicates why a constant factor approximation algorithm suffices for our purposes – we can always pick $\epsilon' = \epsilon/4$ and complete the proof. For details, see Appendix A.1.

3.2 Proper Learning Algorithm: Theorem 1

Having obtained the improper learning algorithm, we can now describe our proper Algorithm 2. Initially, the algorithm starts similarly with the improper learner and obtains a collection of binary linear classifiers. The crucial idea is the next step: the design of an appropriate convex program which will efficiently give the matrix $\boldsymbol{W}$. We proceed with the details. For the proof, see Appendix A.2.

Algorithm 2 Proper Learning Algorithm `ProperLSF`

`Input`: Training set $T = \{(\boldsymbol{x}^t, \pi^t)\}_{t\in[N]}, \epsilon, \delta \in (0,1), \eta \in [0, 1/2)$
`Output`: Linear Sorting function $h : \mathbb{R}^d \to \mathbb{S}_k$, i.e., $h(\cdot) = \sigma_{\boldsymbol{W}}(\cdot)$ for some matrix $\boldsymbol{W} \in \mathbb{R}^{k\times d}$

Compute $(\boldsymbol{v}_{ij})_{1\le i<j\le k} = \texttt{ImproperLSF}(T, \epsilon, \delta, \eta)$ ▷ See Algorithm 1
Setup the CP 1 and compute $\boldsymbol{W} = \texttt{Ellipsoid}(\text{CP})$ ▷ See Appendix A.2
`Ranking Phase`: Given $\boldsymbol{x} \in \mathbb{R}^d$, output $h(\boldsymbol{x}) = \mathrm{argsort}(\boldsymbol{W}\boldsymbol{x})$

Step 1: Calling Non-proper Learners. As a first step, the algorithm calls Algorithm 1 with parameters ϵ, δ and $\eta \in [0, 1/2)$ and obtains a list of linear classifiers with normal vectors $\boldsymbol{v}_{ij}$ for $i < j$. Without loss of generality, assume that $\|\boldsymbol{v}_{ij}\|_2 = 1$.

Step 2: Designing and Solving the CP 1. Our main goal is to find a matrix $\boldsymbol{W}$ whose LSF is close to the true target in KT distance. We show the following lemma that connects the KT distance between two LSFs with the angle metric $d_{\mathrm{angle}}(\cdot,\cdot)$ defined in Eq. (1). The proof can be found in the Appendix A.2.

Lemma 2. *For $\boldsymbol{W}, \boldsymbol{W}^\star \in \mathbb{R}^{k\times d}$, it holds $\mathbf{E}_{\boldsymbol{x}\sim\mathcal{N}_d}[\Delta_{\mathrm{KT}}(\sigma_{\boldsymbol{W}}(\boldsymbol{x}), \sigma_{\boldsymbol{W}^\star}(\boldsymbol{x}))] \leq d_{\mathrm{angle}}(\boldsymbol{W}, \boldsymbol{W}^\star)$.*

The above lemma states that, for our purposes, it suffices to control the d_{angle} metric between the guess $\boldsymbol{W}$ and the true matrix $\boldsymbol{W}^\star$. It turns out that, given the binary classifiers $\boldsymbol{v}_{ij}$, we can design a convex program whose solution will satisfy this property. Thinking of the binary classifier $\boldsymbol{v}_{ij}$ as a proxy for $\boldsymbol{W}_i^\star - \boldsymbol{W}_j^\star$, we want each difference $\boldsymbol{W}_i - \boldsymbol{W}_j$ to have small angle with $\boldsymbol{v}_{ij}$ or equivalently to have large correlation with it, i.e., $(\boldsymbol{W}_i - \boldsymbol{W}_j)\cdot \boldsymbol{v}_{ij} \approx \|\boldsymbol{W}_i - \boldsymbol{W}_j\|_2$. To enforce this condition, we can therefore use the second order conic constraint $(\boldsymbol{W}_i - \boldsymbol{W}_j)\cdot \boldsymbol{v}_{ij} \geq (1-\phi)\|\boldsymbol{W}_i - \boldsymbol{W}_j\|_2$. We formulate the following convex program 1 with variable the matrix $\boldsymbol{W}$:

$$\begin{aligned} &\text{Find} && \boldsymbol{W} \in \mathbb{R}^{k\times d}, \quad \|\boldsymbol{W}\|_F \leq 1, \\ &\text{such that} && (\boldsymbol{W}_i - \boldsymbol{W}_j)\cdot \boldsymbol{v}_{ij} \geq (1-\phi)\cdot \|\boldsymbol{W}_i - \boldsymbol{W}_j\|_2 \quad \text{for any } 1 \leq i < j \leq k, \end{aligned} \tag{1}$$

for some $\phi \in (0,1)$ to be decided. Intuitively, since any $\boldsymbol{v}_{ij}$ has good correlation with $\boldsymbol{W}_i^\star - \boldsymbol{W}_j^\star$ (by the guarantees of the improper learning algorithm) and the CP 1 requires that its solution $\boldsymbol{W}$ similarly correlates well with $\boldsymbol{v}_{ij}$, we expect that $d_{\mathrm{angle}}(\boldsymbol{W}, \boldsymbol{W}^\star)$ will be small. We show that:

Claim 1. *The convex program 1 is feasible and any solution $\boldsymbol{W}$ of 1 satisfies $d_{\mathrm{angle}}(\boldsymbol{W}, \boldsymbol{W}^\star) \leq \epsilon$.*

To see this, note that any solution of CP 1 is a matrix $\boldsymbol{W}$ whose angle metric (see Eq. (1)) with the true matrix is small by an application of the triangle inequality between the angles of $(\boldsymbol{v}_{ij}, \boldsymbol{W}_i - \boldsymbol{W}_j)$ and $(\boldsymbol{v}_{ij}, \boldsymbol{W}_i^\star - \boldsymbol{W}_j^\star)$ for any $i \neq j$. We next have to deal with the feasibility of CP 1. Our goal is to determine the value of ϕ that makes the CP 1 feasible. For the pair $1 \leq i < j \leq k$, the guess $\boldsymbol{v}_{ij}$ and the true normal vector $\boldsymbol{W}_i^\star - \boldsymbol{W}_j^\star$ satisfy, with high probability,

$$\Pr_{\boldsymbol{x}\sim\mathcal{D}_x}[\mathrm{sgn}(\boldsymbol{v}_{ij}\cdot\boldsymbol{x}) \neq \mathrm{sgn}((\boldsymbol{W}_i^\star - \boldsymbol{W}_j^\star)\cdot\boldsymbol{x})] \leq \epsilon\,. \tag{2}$$

Under the Gaussian distribution (which is rotationally symmetric), it is well known that the angle $\theta(\boldsymbol{u},\boldsymbol{v})$ between two vectors $\boldsymbol{u},\boldsymbol{v} \in \mathbb{R}^d$ is equal to $\pi \cdot \mathbf{Pr}_{\boldsymbol{x}\sim\mathcal{N}_d}[\mathrm{sgn}(\boldsymbol{u}\cdot\boldsymbol{x}) \neq \mathrm{sgn}(\boldsymbol{v}\cdot\boldsymbol{x})]$. Hence, using Eq. (2), we get that the angle between the guess $\boldsymbol{v}_{ij}$ and the true normal vector $\boldsymbol{W}_i^\star - \boldsymbol{W}_j^\star$ is $\theta(\boldsymbol{W}_i^\star - \boldsymbol{W}_j^\star, \boldsymbol{v}_{ij}) \leq c\epsilon$. For sufficiently small ϵ, this bound implies that the cosine of the above angle is of order $1-(c\epsilon)^2$ and so the following inequality will hold (since $\boldsymbol{v}_{ij}$ is unit):

$$(\boldsymbol{W}_i^\star - \boldsymbol{W}_j^\star)\cdot\boldsymbol{v}_{ij} \geq (1-2(c\epsilon)^2)\cdot\|\boldsymbol{W}_i^\star - \boldsymbol{W}_j^\star\|_2\,.$$

Hence, by setting $\phi = 2(c\epsilon)^2$, the convex program 1 with variables $\boldsymbol{W} \in \mathbb{R}^{k\times d}$ will be feasible; since $\|\boldsymbol{W}^\star\|_F \leq 1$ comes without loss of generality, $\boldsymbol{W}^\star$ will be a solution with probability $1-\delta$.

Next, we have to control the volume of the feasible region. This is crucial in order to apply the ellipsoid algorithm (for details, see in Appendix A.2.1) and, hence, solve the convex program. We show the following claim (see Appendix A.2.1 for the proof):

Claim 2. *There exists $r \geq 2^{-\mathrm{poly}(d,k,1/\epsilon,\log(1/\delta))}$ so that the feasible set of CP 1 with $\phi = O(\epsilon^2)$ contains a ball (with respect to the Frobenius norm) of radius r.*

Critically, the runtime of the ellipsoid algorithm is *logarithmic* in $1/r$. So, the ellipsoid runs in time polynomial in the parameters of the problem and outputs the desired matrix $\boldsymbol{W}$.

4 Learning in top-r Disagreement: Theorem 2

In this section we show that the proper learning algorithm of Section 3.2 learns noisy LSFs in the top-r disagreement metric. We have seen that, with $\widetilde{O}(d\log(k)/\epsilon)$ samples, Algorithm 2 of Section 3.2 computes a matrix $\boldsymbol{W}$ such that $d_{\mathrm{angle}}(\boldsymbol{W}, \boldsymbol{W}^\star) \leq \epsilon$, see Claim 1. Let us be more specific. Lemma 2 relates the expected KT distance with the angle metric of the two matrices (see also Equation (1)). Our Algorithm 2 essentially gives an upper bound on this angle metric. When we shift our objective and our goal is to control the top-r disagreement, we can still apply Algorithm 2 which essentially controls the angle metric. The crucial ingredient that is missing is the relation between the loss we

have to control, i.e., the expected top-r disagreement and the angle metric of Equation 1. This relation is presented right after and essentially says that the expected top-r disagreement is at most $O(kr)$ times this angle metric. Hence, in order to get top-r disagreement of order ϵ, it suffices to apply our Algorithm 2 with $\epsilon' = O(\epsilon/(kr))$.

We continue with our main contribution which is the following lemma that connects the top-r disagreement metric with the geometric distance $d_{\text{angle}}(\cdot,\cdot)$, recall Lemma 1. To keep this sketch simple we shall present a sketch of the proof of Lemma 1 for the special case of top-1 classification, which we restate below. The proof of the top-1 case can be found at the Appendix B. The detailed proof of the general case $(r > 1)$ can be found in the Appendix C.

Lemma 3 (Top-1 Disagreement Loss via $d_{\text{angle}}(\cdot,\cdot)$)**.** *Consider two matrices $\boldsymbol{U}, \boldsymbol{V} \in \mathbb{R}^{k\times d}$ and let $\mathcal{N}_d$ be the standard Gaussian in d dimensions. We have that*

$$\mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{N}_d}[\sigma_1(\boldsymbol{U}\boldsymbol{x}) \neq \sigma_1(\boldsymbol{V}\boldsymbol{x})] \leq O\left(k\sqrt{\log k}\right)\, d_{\text{angle}}(\boldsymbol{U},\boldsymbol{V})\,.$$

We observe that

$$\mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{N}_d}[\sigma_1(\boldsymbol{U}\boldsymbol{x}) \neq \sigma_1(\boldsymbol{V}\boldsymbol{x})] = \sum_{i\in[k]} \mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{N}_d}[\sigma_1(\boldsymbol{U}\boldsymbol{x}) = i, \sigma_1(\boldsymbol{V}\boldsymbol{x}) \neq i]\,. \tag{1}$$

We denote by $\mathcal{C}_{\boldsymbol{U}}^{(i)} \triangleq \mathbf{1}\{\boldsymbol{x} : \sigma_1(\boldsymbol{U}\boldsymbol{x}) = i\} = \prod_{j\neq i}\mathbf{1}\{(\boldsymbol{U}_i - \boldsymbol{U}_j)\cdot\boldsymbol{x} \geq 0\}$, i.e., this is the set where the ranking corresponding to $\boldsymbol{U}$ picks i as the top element. Note that $\mathcal{C}_{\boldsymbol{U}}^{(i)}$ is the indicator of a homogeneous polyhedral cone since it can be written as the intersection of homogeneous halfspaces. Using these cones we can rewrite the top-1 disagreement of Eq. (1) as

$$\mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{N}_d}[\sigma_1(\boldsymbol{U}\boldsymbol{x}) \neq \sigma_1(\boldsymbol{V}\boldsymbol{x})] = \sum_{i\in[k]} \mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{N}_d}[C_{\boldsymbol{U}}^{(i)}(\boldsymbol{x}) = 1, C_{\boldsymbol{V}}^{(i)}(\boldsymbol{x}) = 0]\,. \tag{2}$$

Hence, our task is to control the mass of the disagreement region of two cones. The next Lemma 4 achieves this task and, combined with Eq. (2) directly gives the conclusion of Lemma 3.

Next we work with two general homogeneous polyhedral cones with set indicators C_1, C_2:

Lemma 4 (Cone Disagreement)**.** *Let $C_1, C_2 : \mathbb{R}^d \mapsto \{0,1\}$ be homogeneous polyhedral cones defined by the k unit vectors $\boldsymbol{v}_1,\ldots,\boldsymbol{v}_k$ and $\boldsymbol{u}_1,\ldots,\boldsymbol{u}_k$ respectively. For some universal constant $c > 0$, it holds that $\mathbf{Pr}_{\boldsymbol{x}\sim\mathcal{N}_d}[C_1(\boldsymbol{x}) \neq C_2(\boldsymbol{x})] \leq c\sqrt{\log k}\,\max_{i\in[k]}\theta(\boldsymbol{v}_i,\boldsymbol{u}_i)$.*

Roadmap of the Proof of Lemma 4: Assume that we rotate one face of the polyhedral cone C_1 by a very small angle θ to obtain the perturbed cone C_2. At a high-level, we expect the probability of the disagreement region between the new cone C_2 and C_1 to be roughly (this is an underestimation) equal to the size of the perturbation θ times the (Gaussian) surface area of the face of the convex cone that we perturbed. The Gaussian Surface Area (GSA) of a convex set $A \subset \mathbb{R}^d$, is defined as $\Gamma(A) \triangleq \int_{\partial A}\phi_d(\boldsymbol{x})d\mu(\boldsymbol{x})$, where $d\mu(\boldsymbol{x})$ is the standard surface measure in $\mathbb{R}^d$ and $\phi_d(\boldsymbol{x}) = (2\pi)^{-d/2}\cdot\exp(-\|\boldsymbol{x}\|_2^2/2)$. In fact, in Claim 3 below, we show that the probability of the disagreement between C_1 and C_2 is roughly $O(\theta)\Gamma(F_1)\sqrt{\log(1/\Gamma(F_1)+1)}$, where F_1 is the face of cone C_1 that we rotated. Now, when we perturb all the faces by small angles (all perturbations are at most θ), we can show (via a sequence of triangle inequalities) that the total probability of the disagreement region is bounded above by the perturbation size θ times the sum of the Gaussian surface area of every face (times a logarithmic blow-up factor):

$$\mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{N}_d}[C_1(\boldsymbol{x}) \neq C_2(\boldsymbol{x})] \leq O(\theta)\,\sum_{i=1}^{k}\Gamma(F_i)\sqrt{\log(1/\Gamma(F_i)+1)}\,.$$

Surprisingly, for homogeneous convex cones, the above sum cannot grow very fast with k. In fact, we show that it can be at most $O(\sqrt{\log k})$. To prove this, we crucially rely on the following convex geometry result showing that the Gaussian surface area of a homogeneous convex cone is $O(1)$ regardless of the number of its faces k.

Lemma 5 ([Naz03])**.** *Let C be a homogeneous polyhedral cone with k faces $F_1,\ldots,F_k$. Then C has Gaussian surface area $\Gamma(C) = \sum_{i=1}^{k}\Gamma(F_i) \leq 1$.*

Using an inequality similar to the fact that the maximum entropy of a discrete distribution on k elements is at most $\log k$, and, since, from Lemma 5, it holds that $\sum_{i=1}^{k} \Gamma(F_i) \leq 1$, we can show that $\sum_{i=1}^{k} \Gamma(F_i)\sqrt{\log(1/\Gamma(F_i)+1)} = O(\sqrt{\log k})$. Therefore, with the above lemma we conclude that, if the maximum angle perturbation that we perform on C_1 is θ, then the probability of the disagreement region is $O(\theta)$. We next give the formal proof resulting in the upper bound of $O(\sqrt{\log k}\,\theta)$ for the disagreement.

Single Face Perturbation Bound: Claim 3: We will use the following notation for the positive orthant indicator $R(\boldsymbol{z}) = \prod_{i=1}^{k} \mathbf{1}\{\boldsymbol{z}_i \geq 0\}$. Notice that the homogeneous polyhedral cone C_1 can be written as $C_1(\boldsymbol{x}) = R(\boldsymbol{V}\boldsymbol{x}) = R(\boldsymbol{v}_1 \cdot \boldsymbol{x}, \ldots, \boldsymbol{v}_k \cdot \boldsymbol{x})$. Claim 3 below shows that the disagreement of two cones that differ on a single normal vector is bounded by above by the Gaussian surface area of a particular face F_1 times a logarithmic blow-up factor $\sqrt{\log(1/\Gamma(F_1)+1)}$.

Claim 3. *Let $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_k \in \mathbb{R}^d$ and $\boldsymbol{r} \in \mathbb{R}^d$ with $\theta(\boldsymbol{v}_1, \boldsymbol{r}) \leq \theta$ for some sufficiently small $\theta \in (0, \pi/2)$. Let F_1 be the face with $\boldsymbol{v}_1 \cdot \boldsymbol{x} = 0$ of the cone $R(\boldsymbol{V}\boldsymbol{x})$ and $c > 0$ be some universal constant. Then,*

$$\mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{N}_d}\left[R(\boldsymbol{v}_1 \cdot \boldsymbol{x}, \ldots, \boldsymbol{v}_k \cdot \boldsymbol{x}) \neq R(\boldsymbol{r} \cdot \boldsymbol{x}, \boldsymbol{v}_2 \cdot \boldsymbol{x}, \ldots, \boldsymbol{v}_k \cdot \boldsymbol{x})\right] \leq c \cdot \theta \cdot \Gamma(F_1)\sqrt{\log\left(\frac{1}{\Gamma(F_1)} + 1\right)}\,.$$

Proof Sketch of Claim 3. Since the constraints $\boldsymbol{v}_2 \cdot \boldsymbol{x} \geq 0, \ldots, \boldsymbol{v}_k \cdot \boldsymbol{x} \geq 0$ are common in the two cones, we have that $R(\boldsymbol{v}_1 \cdot \boldsymbol{x}, \ldots, \boldsymbol{v}_k \cdot \boldsymbol{x}) \neq R(\boldsymbol{r} \cdot \boldsymbol{x}, \boldsymbol{v}_2 \cdot \boldsymbol{x}, \ldots, \boldsymbol{v}_k \cdot \boldsymbol{x})$ only when the first "halfspaces" disagree, i.e., when $(\boldsymbol{v}_1 \cdot \boldsymbol{x})(\boldsymbol{r} \cdot \boldsymbol{x}) < 0$. Thus, we have that the LHS probability of Claim 3 is equal to

$$\mathop{\mathbf{E}}_{\boldsymbol{x}\sim\mathcal{N}_d}\left[R(\boldsymbol{v}_2 \cdot \boldsymbol{x}, \ldots, \boldsymbol{v}_k \cdot \boldsymbol{x}) \cdot \mathbf{1}\{(\boldsymbol{v}_1 \cdot \boldsymbol{x})(\boldsymbol{r} \cdot \boldsymbol{x}) < 0\}\right]\,. \tag{3}$$

This expectation contains two terms: the term $R(\boldsymbol{v}_2 \cdot \boldsymbol{x}, \ldots \boldsymbol{v}_k \cdot \boldsymbol{x})$ that contains the last $k-1$ common constrains of the two cones and the region where the first two halfspaces disagree, i.e., the set $\{\boldsymbol{x} : (\boldsymbol{v}_1 \cdot \boldsymbol{x})(\boldsymbol{r} \cdot \boldsymbol{x}) < 0\}$. In order to upper bound this integral in terms of the angle θ, we observe that (for θ sufficiently small) it is not hard to show (see Appendix B) that the disagreement region, which is itself a (non-convex) cone, is a subset of the region $\{\boldsymbol{x} : |\boldsymbol{v}_1 \cdot \boldsymbol{x}| \leq 2\theta|\boldsymbol{q} \cdot \boldsymbol{x}|\}$, where $\boldsymbol{q}$ the normalized projection of $\boldsymbol{r}$ onto the orthogonal complement of $\boldsymbol{v}_1$, i.e., $\boldsymbol{q} = \mathrm{proj}_{\boldsymbol{v}_1^\perp}\boldsymbol{r}/\|\mathrm{proj}_{\boldsymbol{v}_1^\perp}\boldsymbol{r}\|_2$. Therefore, we have that the integral of Eq. (3) is at most

$$\mathop{\mathbf{E}}_{\boldsymbol{x}\sim\mathcal{N}_d}\left[R(\boldsymbol{v}_2 \cdot \boldsymbol{x}, \ldots, \boldsymbol{v}_k \cdot \boldsymbol{x})\ \mathbf{1}\{|\boldsymbol{v}_1 \cdot \boldsymbol{x}| \leq 2\theta|\boldsymbol{q} \cdot \boldsymbol{x}|\}\right]\,.$$

This is where the definition of the Gaussian surface area appears. In fact, we have to compute the derivative of the above expression (which is a function of θ) with respect to θ and evaluate it at $\theta = 0$. The idea behind this computation is that we can upper bound probability mass of the cone disagreement, i.e., the term $\mathbf{Pr}_{\boldsymbol{x}\sim\mathcal{N}_d}\left[R(\boldsymbol{v}_1 \cdot \boldsymbol{x}, \ldots, \boldsymbol{v}_k \cdot \boldsymbol{x}) \neq R(\boldsymbol{r} \cdot \boldsymbol{x}, \boldsymbol{v}_2 \cdot \boldsymbol{x}, \ldots, \boldsymbol{v}_k \cdot \boldsymbol{x})\right]$ by its derivative with respect to θ (evaluated at 0) times θ by introducing $o(\theta)$ error. Hence, it suffices to upper bound the value of this derivative at 0, which is:

$$2\mathop{\mathbf{E}}_{\boldsymbol{x}\sim\mathcal{N}_d}\left[R(\boldsymbol{v}_2 \cdot \boldsymbol{x}, \ldots, \boldsymbol{v}_k \cdot \boldsymbol{x})\ |\boldsymbol{q} \cdot \boldsymbol{x}|\ \delta(|\boldsymbol{v}_1 \cdot \boldsymbol{x}|)\right]\,,$$

where δ is the Dirac delta function. Notice that, if we did not have the term $|\boldsymbol{q} \cdot \boldsymbol{x}|$, the above expression would be exactly equal to two times the Gaussian surface area of the face with $\boldsymbol{v}_1 \cdot \boldsymbol{x} = 0$, i.e., it would be equal to $2\Gamma(F_1)$. We now show that this extra term of $|\boldsymbol{q} \cdot \boldsymbol{x}|$ can only increase the above surface integral by at most a logarithmic factor. For some ξ to be decided, we have that

$$\begin{aligned}
&\mathop{\mathbf{E}}_{\boldsymbol{x}\sim\mathcal{N}_d}\left[R(\boldsymbol{v}_2 \cdot \boldsymbol{x}, \ldots, \boldsymbol{v}_k \cdot \boldsymbol{x})\ |\boldsymbol{q} \cdot \boldsymbol{x}|\ \delta(|\boldsymbol{v}_1 \cdot \boldsymbol{x}|)\right] = \int_{\boldsymbol{x}\in F_1} \phi_d(\boldsymbol{x})|\boldsymbol{q} \cdot \boldsymbol{x}| d\mu(\boldsymbol{x}) \\
&\leq \int_{\boldsymbol{x}\in F_1} \phi_d(\boldsymbol{x})|\boldsymbol{q} \cdot \boldsymbol{x}|\mathbf{1}\{|\boldsymbol{q} \cdot \boldsymbol{x}| \leq \xi\} d\mu(\boldsymbol{x}) + \int_{\boldsymbol{x}\in F_1} \phi_d(\boldsymbol{x})|\boldsymbol{q} \cdot \boldsymbol{x}|\mathbf{1}\{|\boldsymbol{q} \cdot \boldsymbol{x}| \geq \xi\} d\mu(\boldsymbol{x}) \\
&\leq \xi \int_{\boldsymbol{x}\in F_1} \phi_d(\boldsymbol{x}) d\mu(\boldsymbol{x}) + \int_{\boldsymbol{x}\in F_1} \phi_d(\boldsymbol{x})|\boldsymbol{q} \cdot \boldsymbol{x}|\mathbf{1}\{|\boldsymbol{q} \cdot \boldsymbol{x}| \geq \xi\} d\mu(\boldsymbol{x})\,,
\end{aligned}$$

where $d\mu(\boldsymbol{x})$ is the standard surface measure in $\mathbb{R}^d$. The first integral above is exactly equal to the Gaussian surface area of the face F_1. To bound from above the second term we can use the next claim showing that not a lot of mass of the face F_1 can concentrate on the region where $|\boldsymbol{q} \cdot \boldsymbol{x}|$ is very large. Its proof relies on standard Gaussian concentration arguments, and is provided in Appendix B.

Claim 4. *It holds that* $\int_{\boldsymbol{x}\in F_1} \phi_d(\boldsymbol{x})|\boldsymbol{q}\cdot\boldsymbol{x}|\mathbf{1}\{|\boldsymbol{q}\cdot\boldsymbol{x}|\geq\xi\}d\mu(\boldsymbol{x}) \leq O(\exp(-\xi^2/2))\,.$

Using the above result, we get that

$$\frac{d}{d\theta}\Big(\mathop{\mathbf{E}}_{\boldsymbol{x}\sim\mathcal{N}_d}\left[R(\boldsymbol{v}_2\cdot\boldsymbol{x},\ldots,\boldsymbol{v}_k\cdot\boldsymbol{x})\;\mathbf{1}\{|\boldsymbol{v}_1\cdot\boldsymbol{x}|\leq 2\theta|\boldsymbol{q}\cdot\boldsymbol{x}|\}\right]\Big)\Big|_{\theta=0} \leq O(\xi)\,\Gamma(F_1)+O(\exp(-\xi^2/2))\,.$$

By picking $\xi=\Theta(\sqrt{\log(1+1/\Gamma(F_1))})$, the result follows since, up to introducing $o(\theta)$ error, we can bound the term $\mathbf{Pr}_{\boldsymbol{x}\sim\mathcal{N}_d}\left[R(\boldsymbol{v}_1\cdot\boldsymbol{x},\ldots,\boldsymbol{v}_k\cdot\boldsymbol{x})\neq R(\boldsymbol{r}\cdot\boldsymbol{x},\boldsymbol{v}_2\cdot\boldsymbol{x},\ldots,\boldsymbol{v}_k\cdot\boldsymbol{x})\right]$ by its derivative with respect to θ, evaluated at 0, times θ. □

Conclusion. Our work presents the first theoretical guarantees for (linear) LR with noise and settles interesting directions for future work, as mentioned in Section 1. This paper is theoretical and does not have any negative social impact.

Acknowledgments and Disclosure of Funding

Dimitris Fotakis and Alkis Kalavasis were supported by the Hellenic Foundation for Research and Innovation (H.F.R.I.) under the "First Call for H.F.R.I. Research Projects to support Faculty members and Researchers and the procurement of high-cost research equipment grant", project BALSAM, HFRIFM17-1424.

References

[ABHU15] Pranjal Awasthi, Maria-Florina Balcan, Nika Haghtalab, and Ruth Urner. Efficient learning of linear separators under bounded noise. In *Conference on Learning Theory*, pages 167–190. PMLR, 2015.

[ABHZ16] Pranjal Awasthi, Maria-Florina Balcan, Nika Haghtalab, and Hongyang Zhang. Learning and 1-bit compressed sensing under asymmetric noise. In *Conference on Learning Theory*, pages 152–192. PMLR, 2016.

[ABSV14] Pranjal Awasthi, Avrim Blum, Or Sheffet, and Aravindan Vijayaraghavan. Learning mixtures of ranking models. *arXiv preprint arXiv:1410.8750*, 2014.

[ACN08] Nir Ailon, Moses Charikar, and Alantha Newman. Aggregating inconsistent information: ranking and clustering. *Journal of the ACM (JACM)*, 55(5):1–27, 2008.

[AGM17] Juan A Aledo, José A Gámez, and David Molina. Tackling the supervised label ranking problem by bagging weak learners. *Information Fusion*, 35:38–50, 2017.

[AL88] Dana Angluin and Philip Laird. Learning from noisy examples. *Machine Learning*, 2(4):343–370, 1988.

[APA18] Arpit Agarwal, Prathamesh Patil, and Shivani Agarwal. Accelerated spectral ranking. In *International Conference on Machine Learning*, pages 70–79. PMLR, 2018.

[BDCBL92] Shai Ben-David, Nicolò Cesa-Bianchi, and Philip M Long. Characterizations of learnability for classes of {O,..., n}-valued functions. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 333–340, 1992.

[BFFSZ19] Róbert Busa-Fekete, Dimitris Fotakis, Balázs Szörényi, and Manolis Zampetakis. Optimal learning of mallows block model. In *Conference on Learning Theory*, pages 529–532. PMLR, 2019.

[BH20] Maria-Florina Balcan and Nika Haghtalab. Noise in classification., 2020.

[BM09] Mark Braverman and Elchanan Mossel. Sorting from noisy information. *arXiv preprint arXiv:0910.1191*, 2009.

[BT52] Ralph Allan Bradley and Milton E Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

[BZ17] Maria-Florina F Balcan and Hongyang Zhang. Sample and computationally efficient learning algorithms under s-concave distributions. *Advances in Neural Information Processing Systems*, 30, 2017.

[CDH10] Weiwei Cheng, Krzysztof Dembczynski, and Eyke Hüllermeier. Label ranking methods based on the plackett-luce model. In *ICML*, 2010.

[CH08] Weiwei Cheng and Eyke Hüllermeier. Instance-based label ranking using the mallows model. In *ECCBR Workshops*, pages 143–157, 2008.

[CH12] Weiwei Cheng and Eyke Hüllermeier. Probability estimation for multi-class classification based on label ranking. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 83–98. Springer, 2012.

[CHH09] Weiwei Cheng, Jens Hühn, and Eyke Hüllermeier. Decision tree and instance-based learning for label ranking. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 161–168, 2009.

[CKMY20] Sitan Chen, Frederic Koehler, Ankur Moitra, and Morris Yau. Classification under misspecification: Halfspaces, generalized linear models, and evolvability. *Advances in Neural Information Processing Systems*, 33:8391–8403, 2020.

[CKS18] Stephan Clémençon, Anna Korba, and Eric Sibony. Ranking median regression: Learning to order through local consensus. In *Algorithmic Learning Theory*, pages 212–245. PMLR, 2018.

[CPS13] Ioannis Caragiannis, Ariel D Procaccia, and Nisarg Shah. When do noisy votes reveal the truth? In *Proceedings of the fourteenth ACM conference on Electronic commerce*, pages 143–160, 2013.

[CV20] Stéphan Clémençon and Robin Vogel. A multiclass classification approach to label ranking. In *International Conference on Artificial Intelligence and Statistics*, pages 1421–1430. PMLR, 2020.

[Dan16] Amit Daniely. Complexity theoretic limitations on learning halfspaces. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pages 105–117, 2016.

[DGR+14] 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 *Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014.

[DGT19] Ilias Diakonikolas, Themis Gouleakis, and Christos Tzamos. Distribution-independent pac learning of halfspaces with massart noise. *Advances in Neural Information Processing Systems*, 32, 2019.

[DK20] Ilias Diakonikolas and Daniel M Kane. Hardness of learning halfspaces with massart noise. *arXiv preprint arXiv:2012.09720*, 2020.

[DKK+21] Ilias Diakonikolas, Daniel M Kane, Vasilis Kontonis, Christos Tzamos, and Nikos Zarifis. Learning general halfspaces with general massart noise under the gaussian distribution. *arXiv preprint arXiv:2108.08767*, 2021.

[DKM05] Sanjoy Dasgupta, Adam Tauman Kalai, and Claire Monteleoni. Analysis of perceptron-based active learning. In *International conference on computational learning theory*, pages 249–263. Springer, 2005.

[DKPZ21] Ilias Diakonikolas, Daniel M Kane, Thanasis Pittas, and Nikos Zarifis. The optimality of polynomial regression for agnostic learning under gaussian marginals. *arXiv preprint arXiv:2102.04401*, 2021.

[DKT21] Ilias Diakonikolas, Daniel Kane, and Christos Tzamos. Forster decomposition and learning halfspaces with noise. *Advances in Neural Information Processing Systems*, 34, 2021.

[DKTZ20] Ilias Diakonikolas, Vasilis Kontonis, Christos Tzamos, and Nikos Zarifis. Learning halfspaces with massart noise under structured distributions. In *Conference on Learning Theory*, pages 1486–1513. PMLR, 2020.

[DKZ20] Ilias Diakonikolas, Daniel Kane, and Nikos Zarifis. Near-optimal sq lower bounds for agnostically learning halfspaces and relus under gaussian marginals. *Advances in Neural Information Processing Systems*, 33:13586–13596, 2020.

[DOS18] Anindya De, Ryan O'Donnell, and Rocco Servedio. Learning sparse mixtures of rankings from noisy information. *arXiv preprint arXiv:1811.01216*, 2018.

[DSBDSS11] Amit Daniely, Sivan Sabato, Shai Ben-David, and Shai Shalev-Shwartz. Multiclass learnability and the erm principle. In *Proceedings of the 24th Annual Conference on Learning Theory*, pages 207–232. JMLR Workshop and Conference Proceedings, 2011.

[DSM03] Ofer Dekel, Yoram Singer, and Christopher D Manning. Log-linear models for label ranking. *Advances in neural information processing systems*, 16:497–504, 2003.

[DSS14] Amit Daniely and Shai Shalev-Shwartz. Optimal learners for multiclass problems. In *Conference on Learning Theory*, pages 287–316. PMLR, 2014.

[dSSKC17] Cláudio Rebelo de Sá, Carlos Soares, Arno Knobbe, and Paulo Cortez. Label ranking forests. *Expert systems*, 34(1):e12166, 2017.

[FGKP06] Vitaly Feldman, Parikshit Gopalan, Subhash Khot, and Ashok Kumar Ponnuswami. New results for learning noisy parities and halfspaces. In *2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*, pages 563–574. IEEE, 2006.

[FHMB08] Johannes Fürnkranz, Eyke Hüllermeier, Eneldo Loza Mencía, and Klaus Brinker. Multilabel classification via calibrated label ranking. *Machine learning*, 73(2):133–153, 2008.

[FKP21] Dimitris Fotakis, Alkis Kalavasis, and Eleni Psaroudaki. Label ranking through nonparametric regression. *arXiv preprint arXiv:2111.02749*, 2021.

[FKS21] Dimitris Fotakis, Alkis Kalavasis, and Konstantinos Stavropoulos. Aggregating incomplete and noisy rankings. In *International Conference on Artificial Intelligence and Statistics*, pages 2278–2286. PMLR, 2021.

[GDGV13] Mihajlo Grbovic, Nemanja Djuric, Shengbo Guo, and Slobodan Vucetic. Supervised clustering of label ranking data using label preference information. *Machine learning*, 93(2-3):191–225, 2013.

[GDV12] 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.

[GGK20] Surbhi Goel, Aravind Gollakota, and Adam Klivans. Statistical-query lower bounds via functional gradients. *Advances in Neural Information Processing Systems*, 33:2147–2158, 2020.

[GR09] Venkatesan Guruswami and Prasad Raghavendra. Hardness of learning halfspaces with noise. *SIAM Journal on Computing*, 39(2):742–765, 2009.

[Hau18] David Haussler. Decision theoretic generalizations of the pac model for neural net and other learning applications. In *The Mathematics of Generalization*, pages 37–116. CRC Press, 2018.

[HFCB08] Eyke Hüllermeier, Johannes Fürnkranz, Weiwei Cheng, and Klaus Brinker. Label ranking by learning pairwise preferences. *Artificial Intelligence*, 172(16):1897–1916, 2008.

[HPRZ03] Sariel Har-Peled, Dan Roth, and Dav Zimak. Constraint classification for multiclass classification and ranking. *Advances in neural information processing systems*, pages 809–816, 2003. URL: https://proceedings.neurips.cc/paper/2002/file/16026d60ff9b54410b3435b403afd226-Paper.pdf.

[HSSVG22] Daniel Hsu, Clayton Sanford, Rocco Servedio, and Emmanouil-Vasileios Vlatakis-Gkaragkounis. Near-optimal statistical query lower bounds for agnostically learning intersections of halfspaces with gaussian marginals. *arXiv preprint arXiv:2202.05096*, 2022.

[Hun04] David R Hunter. Mm algorithms for generalized bradley-terry models. *The annals of statistics*, 32(1):384–406, 2004.

[KCS17] Anna Korba, Stephan Clémençon, and Eric Sibony. A learning theory of ranking aggregation. In *Artificial Intelligence and Statistics*, pages 1001–1010. PMLR, 2017.

[Kea98] Michael Kearns. Efficient noise-tolerant learning from statistical queries. *Journal of the ACM (JACM)*, 45(6):983–1006, 1998.

[KGB18] Anna Korba, Alexandre Garcia, and Florence d'Alché Buc. A structured prediction approach for label ranking. *arXiv preprint arXiv:1807.02374*, 2018.

[KMS06] Claire Kenyon-Mathieu and Warren Schudy. How to rank with few errors–a ptas for weighted feedback arc set on tournaments. In *ELECTRONIC COLLOQUIUM ON COMPUTATIONAL COMPLEXITY, REPORT NO. 144 (2006)*. Citeseer, 2006.

[KSS94] Michael J Kearns, Robert E Schapire, and Linda M Sellie. Toward efficient agnostic learning. *Machine Learning*, 17(2):115–141, 1994.

[LB11] Tyler Lu and Craig Boutilier. Learning mallows models with pairwise preferences. In *ICML*, 2011.

[LM18] Allen Liu and Ankur Moitra. Efficiently learning mixtures of mallows models. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 627–638. IEEE, 2018.

[LM21] Allen Liu and Ankur Moitra. Robust voting rules from algorithmic robust statistics. *arXiv preprint arXiv:2112.06380*, 2021.

[Luc12] R Duncan Luce. *Individual choice behavior: A theoretical analysis*. Courier Corporation, 2012.

[LV07] László Lovász and Santosh Vempala. The geometry of logconcave functions and sampling algorithms. *Random Structures & Algorithms*, 30(3):307–358, 2007.

[Mal57] Colin L Mallows. Non-null ranking models. i. *Biometrika*, 44(1/2):114–130, 1957.

[MN06] Pascal Massart and Élodie Nédélec. Risk bounds for statistical learning. *The Annals of Statistics*, 34(5):2326–2366, 2006.

[MV19] Oren Mangoubi and Nisheeth K Vishnoi. Nonconvex sampling with the metropolis-adjusted langevin algorithm. In *Conference on Learning Theory*, pages 2259–2293. PMLR, 2019.

[MW20] Cheng Mao and Yihong Wu. Learning mixtures of permutations: Groups of pairwise comparisons and combinatorial method of moments. *arXiv preprint arXiv:2009.06784*, 2020.

[Nat89] Balas K Natarajan. On learning sets and functions. *Machine Learning*, 4(1):67–97, 1989.

[Naz03] Fedor Nazarov. On the maximal perimeter of a convex set in $\mathbb{R}^n$ with respect to a gaussian measure. In *Geometric aspects of functional analysis*, pages 169–187. Springer, 2003.

[NOS17] Sahand Negahban, Sewoong Oh, and Devavrat Shah. Rank centrality: Ranking from pairwise comparisons. *Operations Research*, 65(1):266–287, 2017.

[NT22] Rajai Nasser and Stefan Tiegel. Optimal sq lower bounds for learning halfspaces with massart noise. *arXiv preprint arXiv:2201.09818*, 2022.

[Pap81] Christos H Papadimitriou. On the complexity of integer programming. *Journal of the ACM (JACM)*, 28(4):765–768, 1981.

[RdSRSK15] Cláudio Rebelo de Sá, Carla Rebelo, Carlos Soares, and Arno Knobbe. Distance-based decision tree algorithms for label ranking. In *Portuguese Conference on Artificial Intelligence*, pages 525–534. Springer, 2015.

[RS94] Ronald L Rivest and Robert Sloan. A formal model of hierarchical concept-learning. *Information and Computation*, 114(1):88–114, 1994.

[Sch98] Alexander Schrijver. *Theory of linear and integer programming*. John Wiley & Sons, 1998.

[Slo88] Robert Sloan. Types of noise in data for concept learning. In *Proceedings of the first annual Workshop on Computational Learning Theory*, pages 91–96, 1988.

[Slo92] Robert H Sloan. Corrigendum to types of noise in data for concept learning. In *Proceedings of the fifth annual workshop on Computational learning theory*, page 450, 1992.

[Slo96] Robert H Sloan. Pac learning, noise, and geometry. In *Learning and Geometry: Computational Approaches*, pages 21–41. Springer, 1996.

[SS07] Shai Shalev-Shwartz. *Online learning: Theory, algorithms, and applications*. PhD thesis, The Hebrew University of Jerusalem, 2007.

[SSBD14] Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

[Vap06] Vladimir Vapnik. *Estimation of dependences based on empirical data*. Springer Science & Business Media, 2006.

[VG10] Shankar Vembu and Thomas Gärtner. Label ranking algorithms: A survey. In *Preference learning*, pages 45–64. Springer, 2010.

[Vis21] Nisheeth K Vishnoi. *Algorithms for convex optimization*. Cambridge University Press, 2021.

[VZW09] Anke Van Zuylen and David P Williamson. Deterministic pivoting algorithms for constrained ranking and clustering problems. *Mathematics of Operations Research*, 34(3):594–620, 2009.

[YZ17] Songbai Yan and Chicheng Zhang. Revisiting perceptron: Efficient and label-optimal learning of halfspaces. *Advances in Neural Information Processing Systems*, 30, 2017.

[ZL21] 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.

[ZLC17] Yuchen Zhang, Percy Liang, and Moses Charikar. A hitting time analysis of stochastic gradient langevin dynamics. In *Conference on Learning Theory*, pages 1980–2022. PMLR, 2017.

[ZLGQ14] Yangming Zhou, Yangguang Liu, Xiao-Zhi Gao, and Guoping Qiu. A label ranking method based on gaussian mixture model. *Knowledge-Based Systems*, 72:108–113, 2014.

[ZLY+14] Yangming Zhou, Yangguang Liu, Jiangang Yang, Xiaoqi He, and Liangliang Liu. A taxonomy of label ranking algorithms. *J. Comput.*, 9(3):557–565, 2014.

[ZQ18] Yangming Zhou and Guoping Qiu. Random forest for label ranking. *Expert Systems with Applications*, 112:99–109, 2018.

[ZSA20] 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.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] We have presented the assumptions of our models and we have explained future research directions in Section 1.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] We have discussed that in the Conclusion paragraph at the end of the main body.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] We have specified the theoretical models we are working on, see e.g., Section 1.1.
   (b) Did you include complete proofs of all theoretical results? [Yes] Due to space constraints we have included the full proofs of the results in the Supp. Material.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [N/A]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [N/A]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [N/A]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [N/A]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Learning LSFs with Bounded Noise in Kendall's Tau distance

A.1 Improperly Learning LSFs with Bounded Noise

We provide an improper learner for LSFs in the presence of bounded noise. We first restate the main result of this section, whose proof relies on a connection between noisy linear label ranking distributions and the Massart noise model.

Theorem 3 (Non-Proper Learning Algorithm)**.** *Fix $\eta \in [0, 1/2)$ and $\epsilon, \delta \in (0, 1)$. Let $\mathcal{D}$ be an η-noisy linear label ranking distribution satisfying the assumptions of Definition 2.* `ImproperLSF` *(Algorithm 1) draws $N = \widetilde{O}\left(\frac{d}{\epsilon(1-2\eta)^6} \log(k/\delta)\right)$ samples from $\mathcal{D}$, runs in $\mathrm{poly}(d, k, 1/\epsilon, \log(1/\delta))$ time and, with probability at least $1 - \delta$, outputs a hypothesis $h : \mathbb{R}^d \to \mathbb{S}_k$ that is ϵ-close in KT distance to the target.*

Proof. Assume that the target function is $\sigma^\star(\boldsymbol{x}) = \sigma_{\boldsymbol{W}^\star}(\boldsymbol{x}) = \mathrm{argsort}(\boldsymbol{W}^\star \boldsymbol{x})$ for some unknown matrix $\boldsymbol{W}^\star \in \mathbb{R}^{k \times d}$. Consider a collection of N i.i.d. samples from an η-noisy linear label ranking distribution $\mathcal{D}$ (see Definition 2) and let T be the associated training set. For each example $(\boldsymbol{x}, \pi) \in T$, we create a list of $\binom{k}{2}$ binary examples $(\boldsymbol{x}, y_{ij})$ with $y_{ij} = \mathrm{sgn}(\pi(i) - \pi(j))$ for any $1 \le i < j \le k$, where $\pi(i)$ denotes the position of the element i. Hence, we create the datasets T_{ij} consisting of the binary labeled examples $(\boldsymbol{x}, y_{ij})$. We have that

$$\mathop{\mathbf{Pr}}_{(\boldsymbol{x},\pi)\sim\mathcal{D}}\left[y_{ij} \cdot \mathrm{sgn}((\boldsymbol{W}_i^\star - \boldsymbol{W}_j^\star) \cdot \boldsymbol{x}) < 0 \mid \boldsymbol{x}\right] = \mathop{\mathbf{Pr}}_{\pi\sim\mathcal{M}(\sigma^\star(\boldsymbol{x}))}\left[\pi(i) < \pi(j) \mid \boldsymbol{W}_i^\star \cdot \boldsymbol{x} < \boldsymbol{W}_j^\star \cdot \boldsymbol{x}\right] .$$

Since $\mathcal{M}(\sigma^\star(\boldsymbol{x}))$ is an η-bounded noise ranking distribution (see Definition 1), we get that

$$\mathop{\mathbf{Pr}}_{\pi\sim\mathcal{M}(\sigma^\star(\boldsymbol{x}))}\left[\pi(i) < \pi(j) \mid \sigma^\star(\boldsymbol{x})(i) > \sigma^\star(\boldsymbol{x})(j)\right] \le \eta < 1/2 ,$$

where $\sigma^\star(\boldsymbol{x})(i)$ denotes the position of the element i in the ranking $\sigma^\star(\boldsymbol{x})$. Focusing on the training set T_{ij}, we have that the sign y_{ij} is flipped with probability at most η. So, we have reduced the problem to $\binom{k}{2}$ sub-problems concerning the learnability of halfspaces in the presence of Massart noise. The Massart noise model is a special case of Definition 2 where $k = 2$. Note also that for each training set T_{ij}, the features $\boldsymbol{x}$ have the same distribution. We can now apply the following result for LTFs with Massart noise for the standard Gaussian distribution. Recall that the concept class of homogeneous halfspaces (or linear threshold functions) is $\mathcal{C}_{\mathrm{LTF}} = \{h_{\boldsymbol{w}}(\boldsymbol{x}) = \mathrm{sgn}(\boldsymbol{w} \cdot \boldsymbol{x}) : \boldsymbol{w} \in \mathbb{R}^d\}$.

Lemma 6 (Learning Halfspaces with Massart noise [ZSA20])**.** *Fix $\eta \in [0, 1/2)$ and let $\epsilon, \delta \in (0, 1)$. Let $\mathcal{D}$ be an η-noisy linear label ranking distribution satisfying the assumptions of Definition 2 with $k = 2$ (where $\mathcal{C}_{\mathrm{LSF}} = \mathcal{C}_{\mathrm{LTF}}$). There is a computationally efficient algorithm* `MassartLTF` *that draws $m = O(\frac{d\ \mathrm{polylog}(d)}{\epsilon(1-2\eta)^6} \cdot \log(1/\delta))$ samples from $\mathcal{D}$, runs in $\mathrm{poly}(m)$ time and outputs a linear threshold function h that is ϵ-close to the target linear threshold function $h^\star$ with probability at least $1 - \delta$, i.e., it holds $\mathbf{Pr}_{\boldsymbol{x}\sim\mathcal{N}_d}[h(\boldsymbol{x}) \neq h^\star(\boldsymbol{x})] \le \epsilon$.*

We can invoke the algorithm of Lemma 6 for any alternatives $1 \le i < j \le k$ with accuracy $\epsilon' = O(\epsilon)$, $\delta' = O(\delta/k^2)$ and error rate $\eta < 1/2$[4]. We remark that Lemma 6 returns a halfspace. Each one of the $\binom{k}{2}$ calls will provide a vector $\boldsymbol{v}_{ij} \in \mathbb{R}^d$ such that, with probability at least $1 - \delta'$, it satisfies

$$\mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{N}_d}[\mathrm{sgn}(\boldsymbol{v}_{ij} \cdot \boldsymbol{x}) \neq \mathrm{sgn}((\boldsymbol{W}_i^\star - \boldsymbol{W}_j^\star) \cdot \boldsymbol{x})] \le \epsilon' ,$$

where the true target halfspace has normal vector $\boldsymbol{W}_i^\star - \boldsymbol{W}_j^\star$. Moreover, for any $i < j$, the algorithm requires that the training set T_{ij} is of size

$$|T_{ij}| = \Omega\left(\frac{d}{\epsilon'} \cdot \frac{1}{(1-2\eta)^6} \cdot \log(1/\delta')\right) ,$$

and, so, a total number of

$$N = \Omega\left(\frac{d}{\epsilon} \cdot \frac{1}{(1-2\eta)^6} \cdot \log(k/\delta)\right) ,$$

[4] We can assume that η is known without loss of generality.

samples $(\boldsymbol{x}, \pi)$ is required from the distribution $\mathcal{D}$. Given a collection of linear classifiers with normal vectors $\boldsymbol{v}_{ij}$ for any $i < j$, it remains to aggregate them and compute a sorting function $h : \mathbb{R}^d \to \mathbb{S}_k$. To this end, the estimator h, given an example $\boldsymbol{x}$, creates the directed complete graph G with k nodes with directed edge $i \to j$ if $\boldsymbol{v}_{ij} \cdot \boldsymbol{x} > 0$. If all the linear classifiers are correct (which occurs with probability $1 - O(\epsilon k^2)$ over $\mathcal{D}_x$ due to the union bound), the graph G is acyclic (since it will match the true directions induced by $\boldsymbol{W}^\star$) and the estimator h outputs the induced permutation. Observe that the KT distance is

$$\frac{1}{\binom{k}{2}} \cdot \mathop{\mathbf{E}}_{\boldsymbol{x} \sim \mathcal{N}_d} \left[\sum_{1 \le i < j \le k} \mathbf{1}\{\mathrm{sgn}(\boldsymbol{v}_{ij} \cdot \boldsymbol{x}) \neq \mathrm{sgn}((\boldsymbol{W}_i^\star - \boldsymbol{W}_j^\star) \cdot \boldsymbol{x})\} \right] \le \epsilon' .$$

Otherwise, the classifiers are inconsistent and G contains cycles. So, the expected number of mistakes in the graph G is ϵk^2. The estimator in order to output a ranking uses a deterministic constant approximation algorithm for the minimum Feedback Arc Set [ACN08] in order to remove the cycles. For an overview of this fundamental line of research, we refer to [ACN08, VZW09, KMS06].

Lemma 7 (3-Approximation Algorithm for mimimum FAS (see [VZW09, ACN08]))**.** *There is a deterministic algorithm* `MFAS` *for the minimum Feedback Arc Set on unweighted tournaments with k vertices that outputs orderings with cost less than $3 \cdot \mathrm{OPT}$. The running time is* $\mathrm{poly}(k)$.

In the above, OPT is the minimum number of flips the algorithm should perform. With input the cyclic directed graph G induced by the estimated linear classifiers, the algorithm of Lemma 7 computes, in $\mathrm{poly}(k)$ time, a 3-approximation of the optimal solution (i.e., instead of correcting ϵ_0 directed edges, the algorithm will provide a directed acyclic graph with $3\epsilon_0$ changed edges). Hence, for the hypothesis $h : \mathbb{R}^d \to \mathbb{S}_k$, where $h(\boldsymbol{x})$ is the output of the minimum FAS approximation algorithm with input G (G depends on the input $\boldsymbol{x}$, the randomness of the samples and the internal randomness of the $\binom{k}{2}$ calls of the Massart linear classifiers), and the target function $\sigma^\star(\boldsymbol{x})$, we have that

$$\mathop{\mathbf{E}}_{\boldsymbol{x} \sim \mathcal{N}_d} [\Delta_{KT}(h(\boldsymbol{x}), \sigma^\star(\boldsymbol{x}))] \le (\epsilon' + 3\epsilon') = 4\epsilon' ,$$

which completes the proof, by setting $\epsilon' = \epsilon/4$. □

Remark 1. *Consider the following variant of the above procedure: compute the $O(k^2)$ linear classifiers with accuracy $\epsilon' = \epsilon/k^2$: If the induced directed graph is acyclic, output the ranking; otherwise, output a random permutation. With probability ϵ, the KT distance will be of order k^2. Hence, one has to draw in total $O(k^4 d/\epsilon)$ samples to make the expected KT distance roughly $O(\epsilon)$. The algorithm of Theorem 3 improves on this approach.*

A.2 The Proof of Theorem 1: Properly Learning LSFs with Bounded Noise

We first restate the main result of this section.

Theorem 4 (Proper Learning Algorithm)**.** *Fix $\eta \in [0, 1/2)$ and $\epsilon, \delta \in (0, 1)$. Let $\mathcal{D}$ be an η-noisy linear label ranking distribution satisfying the assumptions of Definition 2.* `ProperLSF` *(Algorithm 2) draws $N = \widetilde{O}\left(\frac{d}{\epsilon(1-2\eta)^6} \log(k/\delta)\right)$ samples from $\mathcal{D}$, runs in* $\mathrm{poly}(d, k, 1/\epsilon, \log(1/\delta))$ *time and, with probability at least $1 - \delta$, outputs a Linear Sorting function $h : \mathbb{R}^d \to \mathbb{S}_k$ that is ϵ-close in KT distance to the target.*

We are now ready to provide the proof of our efficient proper learning algorithm for the class of Linear Sorting functions in the presence of bounded noise with respect to the standard Gaussian probability measure.

Proof. As a first step, the algorithm calls the improper learning algorithm `ImproperLSF` (Algorithm 1) with parameters ϵ, δ and $\eta < 1/2$ and obtains a list of linear classifiers with normal vectors $\boldsymbol{v}_{ij}$ for $i < j$. The utility of this step implies that, with probability at least $1 - \delta$, each one of the classifiers ϵ-learns the associated true halfspace, i.e., it holds

$$\mathop{\mathbf{Pr}}_{\boldsymbol{x} \sim \mathcal{N}_d} [\mathrm{sgn}(\boldsymbol{v}_{ij} \cdot \boldsymbol{x}) \neq \mathrm{sgn}((\boldsymbol{W}_i^\star - \boldsymbol{W}_j^\star) \cdot \boldsymbol{x})] \le \epsilon ,$$

where $\boldsymbol{W}^\star$ is the matrix of the target Linear Sorting function. Without loss of generality, assume that $\|\boldsymbol{v}_{ij}\|_2 = 1$. In order to make the learner proper, it suffices to solve the following convex program on $\boldsymbol{W}$:

$$\begin{aligned} &\text{Find} \quad \boldsymbol{W} \in \mathbb{R}^{k\times d}, && (1)\\ &\text{such that} \quad (\boldsymbol{W}_i - \boldsymbol{W}_j)\cdot \boldsymbol{v}_{ij} \geq (1-\phi)\cdot \|\boldsymbol{W}_i - \boldsymbol{W}_j\|_2 \quad \text{for any } 1 \leq i < j \leq k\,, \qquad (\texttt{CP}) && (2)\\ &\qquad \|\boldsymbol{W}\|_F \leq 1\,, && (3) \end{aligned}$$

for some $\phi \in (0,1)$ to be decided. The main key ideas are summarized in the next claim.

Claim 5. *The following properties hold true for $\phi = O(\epsilon^2)$ with probability at least $1-\delta$.*

1. *The convex program 1 is feasible.*
2. *Any solution of the convex program 1 induces an LSF that is ϵ-close in KT distance to the true target $\sigma_{\boldsymbol{W}^\star}(\cdot)$.*
3. *The feasible set of the convex program 1 contains a ball of radius $r = 2^{-\mathrm{poly}(d,k,1/\epsilon,\log(1/\delta))}$ and is contained in a ball of radius 1. Both balls are with respect to the Frobenius norm.*
4. *The convex program 1 can be solved in time $\mathrm{poly}(d,k,1/\epsilon,\log(1/\delta))$ using the ellipsoid algorithm.*

Proof of Item 1. First, we can choose the error ϕ so that this convex program is feasible. Let us set $\boldsymbol{W} = \boldsymbol{W}^\star$, where $\boldsymbol{W}^\star$ is the underlying matrix of the target Linear Sorting function $\sigma^\star$ with $\sigma^\star(\boldsymbol{x}) = \mathrm{argsort}(\boldsymbol{W}^\star\boldsymbol{x})$. Recall that, by the guarantees of the improper learning algorithm, for the pair $1 \leq i < j \leq k$, it holds

$$\mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{N}_d}[\mathrm{sgn}(\boldsymbol{v}_{ij}\cdot\boldsymbol{x}) \neq \mathrm{sgn}((\boldsymbol{W}_i^\star - \boldsymbol{W}_j^\star)\cdot\boldsymbol{x})] \leq \epsilon\,. \tag{4}$$

Since the standard Gaussian is rotationally symmetric, the angle $\theta(\boldsymbol{u},\boldsymbol{v})$ between two vectors $\boldsymbol{u},\boldsymbol{v} \in \mathbb{R}^d$ is equal to $\pi\cdot\mathbf{Pr}_{\boldsymbol{x}\sim\mathcal{N}_d}[\mathrm{sgn}(\boldsymbol{u}\cdot\boldsymbol{x}) \neq \mathrm{sgn}(\boldsymbol{v}\cdot\boldsymbol{x})]$. Hence, using this observation and Equation (4), we get that the angle between the guess vector $\boldsymbol{v}_{ij}$ and the true normal vector $\boldsymbol{W}_i^\star - \boldsymbol{W}_j^\star$ is

$$\theta(\boldsymbol{W}_i^\star - \boldsymbol{W}_j^\star, \boldsymbol{v}_{ij}) \leq c\cdot\epsilon\,,$$

for some constant $c > 0$. For sufficiently small ϵ, this bound implies that the cosine of the above angle is of order $1-(c\epsilon)^2$ and so the following inequality will hold

$$(\boldsymbol{W}_i^\star - \boldsymbol{W}_j^\star)\cdot\boldsymbol{v}_{ij} \geq (1-2(c\epsilon)^2)\cdot\|\boldsymbol{W}_i^\star - \boldsymbol{W}_j^\star\|_2\,,$$

since $\boldsymbol{v}_{ij}$ is unit. Hence, by setting $\phi = 2(c\epsilon)^2$, the convex program with variables $\boldsymbol{W} \in \mathbb{R}^{k\times d}$ will be feasible; $\boldsymbol{W}^\star$ will be a solution with probability $1-\delta$, where the randomness is over the output of the algorithm dealing with the Massart linear classifiers. Note that we can assume that $\|\boldsymbol{W}^\star\|_F \leq 1$ without loss of generality, since we can divide each row with the Frobenius norm.

Proof of Item 2. Let $\widetilde{\boldsymbol{W}}$ be a solution of the convex program.We will make use of the observation that the angle between two vectors is equal to the disagreement of the associated linear threshold functions with respect to the standard normal times π. Observe that any solution $\widetilde{\boldsymbol{W}}$ to the convex program will satisfy that

$$(\forall i,j) \quad \theta(\boldsymbol{v}_{ij}, \widetilde{\boldsymbol{W}}_i - \widetilde{\boldsymbol{W}}_j) \leq O(\sqrt{\phi}) = c\epsilon\,.$$

and

$$(\forall i,j) \quad \theta(\boldsymbol{W}_i^\star - \boldsymbol{W}_j^\star, \boldsymbol{v}_{ij}) \leq \epsilon\,.$$

This implies that

$$d_{\mathrm{angle}}(\boldsymbol{W}^\star, \widetilde{\boldsymbol{W}}) \leq c'\,\epsilon$$

Claim 6. *For the matrices $\boldsymbol{W}, \boldsymbol{W}^\star \in \mathbb{R}^{k\times d}$, it holds that*

$$\mathop{\mathbf{E}}_{\boldsymbol{x}\sim\mathcal{N}_d}[\Delta_{\mathrm{KT}}(\sigma_{\boldsymbol{W}}(\boldsymbol{x}), \sigma_{\boldsymbol{W}^\star}(\boldsymbol{x}))] \leq d_{\mathrm{angle}}(\boldsymbol{W}, \boldsymbol{W}^\star)\,.$$

Proof. We have that

$$\begin{aligned}
\mathop{\mathbf{E}}_{\boldsymbol{x}\sim\mathcal{N}_d}[\Delta_{\mathrm{KT}}(\sigma_{\boldsymbol{W}}(\boldsymbol{x}),\sigma_{\boldsymbol{W}^\star}(\boldsymbol{x}))] &= \frac{1}{\binom{k}{2}}\cdot \mathop{\mathbf{E}}_{\boldsymbol{x}\sim\mathcal{N}_d}[\sum_{1\le i<j\le k}\mathbf{1}\{((\boldsymbol{W}_i-\boldsymbol{W}_j)\cdot\boldsymbol{x})\,((\boldsymbol{W}_i^\star-\boldsymbol{W}_j^\star)\cdot\boldsymbol{x})<0\} \\
&= \frac{1}{\binom{k}{2}}\cdot \sum_{1\le i<j\le k}\mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{N}_d}[\mathrm{sgn}(\boldsymbol{W}_i-\boldsymbol{W}_j)\cdot\boldsymbol{x})\neq \mathrm{sgn}((\boldsymbol{W}_i^\star-\boldsymbol{W}_j^\star)\cdot\boldsymbol{x})] \\
&= \frac{1}{\pi}\max_{i,j}\theta(\boldsymbol{W}_i-\boldsymbol{W}_j,\boldsymbol{W}_i^\star-\boldsymbol{W}_j^\star) \\
&\le d_{\mathrm{angle}}(\boldsymbol{W},\boldsymbol{W}^\star)\,.
\end{aligned}$$

□

Using the above claim, we get an expected KT distance bound of order $O(\epsilon)$. This gives the desired result.

Proof of Item 3. We will make use of the next lemma.

Lemma 8. *Fix $\epsilon,\delta\in(0,1)$. Let $\boldsymbol{W}^\star\in\mathbb{R}^{k\times d}$ be the true parameter matrix. There exists a matrix $\widetilde{\boldsymbol{W}}^\star\in\mathbb{R}^{k\times d}$ such that, with probability at least $1-\delta$:*

- $\mathbf{Pr}_{\boldsymbol{x}\sim\mathcal{N}_d}[\mathrm{sgn}((\boldsymbol{W}_i^\star-\boldsymbol{W}_j^\star)\cdot\boldsymbol{x})\neq\mathrm{sgn}((\widetilde{\boldsymbol{W}}_i^\star-\widetilde{\boldsymbol{W}}_j^\star)\cdot\boldsymbol{x})]\le\epsilon$ *for all $i\neq j$, and,*
- $\|\widetilde{\boldsymbol{W}}_i^\star-\widetilde{\boldsymbol{W}}_j^\star\|_2\ge 2^{-\mathrm{poly}(d,k,1/\epsilon,\log(1/\delta))}$ *for any $i\neq j$.*

Proof of Lemma 8. The above lemma is a result of the next Appendix A.2.1. In particular, it is a direct implication of Lemma 10 and Corollary 1. □

Note that the above lemma implies that

$$(\forall i,j)\quad \mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{N}_d}[\mathrm{sgn}(\boldsymbol{v}_{ij}\cdot\boldsymbol{x})\neq\mathrm{sgn}((\widetilde{\boldsymbol{W}}_i^\star-\widetilde{\boldsymbol{W}}_j^\star)\cdot\boldsymbol{x})]\le 2\epsilon\,,$$

with probability at least $1-2\delta$. Hence, up to constants, the analysis concerning the feasibility of the true matrix $\boldsymbol{W}^\star$ (see Item 1) will still hold for $\widetilde{\boldsymbol{W}}^\star$. From now on we can work with this matrix $\widetilde{\boldsymbol{W}}^\star$ which enjoys the "well-conditionedness" property of the second item of the lemma.

We will use the above lemma in order to prove Item 3 which controls the volume of the feasible region: it states that there exist $0<r<R$ so that the feasible region of the convex program contains a ball of radius r and is contained in a ball of radius R (where the balls are with respect to the Frobenius norm). Moreover, $r=2^{-\mathrm{poly}(d,k,1/\epsilon,\log(1/\delta))}$ and $R=1$.

For the chosen $\phi\in(0,1)$, the feasible set contains matrices $\boldsymbol{W}\in\mathbb{R}^{k\times d}$ that satisfy $\|\boldsymbol{W}-\widetilde{\boldsymbol{W}}^\star\|_F\le 2r$, r to be decided. For any $i\neq j$, we have that the following properties hold:

1. $\|\widetilde{\boldsymbol{W}}_i^\star-\widetilde{\boldsymbol{W}}_j^\star\|_2\ge 2^{-\mathrm{poly}(d,k,1/\epsilon,\log(1/\delta))}$ (well-conditionedness).
2. $(\widetilde{\boldsymbol{W}}_i^\star-\widetilde{\boldsymbol{W}}_j^\star)\cdot\boldsymbol{v}_{ij}\ge(1-\phi)\,\|\widetilde{\boldsymbol{W}}_i^\star-\widetilde{\boldsymbol{W}}_j^\star\|_2$ (feasibility).
3. $\|\boldsymbol{W}-\widetilde{\boldsymbol{W}}^\star\|_F\le 2r$ which implies that $\|\boldsymbol{W}_i-\widetilde{\boldsymbol{W}}_i^\star\|_2\le 2r$ for any $i\in[k]$ (ball around feasible point).
4. $\|\boldsymbol{v}_{ij}\|_2=1$.

Our goal is to prove that for a matrix in the above ball it holds $(\boldsymbol{W}_i-\boldsymbol{W}_j)\cdot\boldsymbol{v}_{ij}\ge(1-\phi)\,\|\boldsymbol{W}_i-\boldsymbol{W}_j\|_2$.

We have that

$$\begin{aligned}
(\widetilde{\boldsymbol{W}}_i^\star-\widetilde{\boldsymbol{W}}_j^\star)\cdot\boldsymbol{v}_{ij} &= (\widetilde{\boldsymbol{W}}_i^\star-\boldsymbol{W}_i)\cdot\boldsymbol{v}_{ij}+(\boldsymbol{W}_j-\widetilde{\boldsymbol{W}}_j^\star)\cdot\boldsymbol{v}_{ij}+(\boldsymbol{W}_i-\boldsymbol{W}_j)\cdot\boldsymbol{v}_{ij} \\
&\le \|\widetilde{\boldsymbol{W}}_i^\star-\boldsymbol{W}_i\|_2+\|\boldsymbol{W}_j-\widetilde{\boldsymbol{W}}_j^\star\|_2+(\boldsymbol{W}_i-\boldsymbol{W}_j)\cdot\boldsymbol{v}_{ij} \\
&\le 4r+(\boldsymbol{W}_i-\boldsymbol{W}_j)\cdot\boldsymbol{v}_{ij}\,.
\end{aligned}$$

More to that

$$\begin{aligned}\|\boldsymbol{W}_i - \boldsymbol{W}_j\|_2 &= \|\boldsymbol{W}_i - \widetilde{\boldsymbol{W}}_i^\star + \widetilde{\boldsymbol{W}}_i^\star - \widetilde{\boldsymbol{W}}_j^\star + \widetilde{\boldsymbol{W}}_j^\star - \boldsymbol{W}_j\|_2 \\ &\leq \|\boldsymbol{W}_i - \widetilde{\boldsymbol{W}}_i^\star\|_2 + \|\widetilde{\boldsymbol{W}}_i^\star - \widetilde{\boldsymbol{W}}_j^\star\|_2 + \|\widetilde{\boldsymbol{W}}_j^\star - \boldsymbol{W}_j\|_2 \\ &\leq 4r + \|\widetilde{\boldsymbol{W}}_i^\star - \widetilde{\boldsymbol{W}}_j^\star\|_2 \,,\end{aligned}$$

and similarly: $\|\boldsymbol{W}_i - \boldsymbol{W}_j\|_2 \geq \|\widetilde{\boldsymbol{W}}_i^\star - \widetilde{\boldsymbol{W}}_j^\star\|_2 - 4r$.

Combining the above inequalities, we get that

$$\begin{aligned}(\boldsymbol{W}_i - \boldsymbol{W}_j) \cdot \boldsymbol{v}_{ij} &\geq (\widetilde{\boldsymbol{W}}_i^\star - \widetilde{\boldsymbol{W}}_j^\star) \cdot \boldsymbol{v}_{ij} - 4r \\ &\geq (1-\phi)\, \|\widetilde{\boldsymbol{W}}_i^\star - \widetilde{\boldsymbol{W}}_j^\star\|_2 - 4r \\ &\geq (1-\phi)\, (\|\boldsymbol{W}_i - \boldsymbol{W}_j\|_2 - 4r) - 4r \\ &= (1-\phi)\, \|\boldsymbol{W}_i - \boldsymbol{W}_j\|_2 - 8r \,.\end{aligned}$$

We pick r sufficiently small and of order $2^{-\mathrm{poly}(d,k,1/\epsilon,\log(1/\delta))}$ and get that $\boldsymbol{W}$ is a feasible solution of the convex program. Moreover, we can select $R = 1$ since $\|\widetilde{\boldsymbol{W}}^\star\|_F = 1$ without loss of generality, since we can normalize the row differences of $\widetilde{\boldsymbol{W}}^\star$ with the norm $\|\widetilde{\boldsymbol{W}}^\star\|_F$.

Proof of Item 4. We apply the ellipsoid algorithm in order to solve the convex program 1 and compute a matrix $\widetilde{\boldsymbol{W}} \in \mathbb{R}^{k\times d}$. The algorithm `ProperLSF` outputs the linear sorting function $h(\cdot) = \sigma_{\widetilde{\boldsymbol{W}}}(\cdot)$.

Lemma 9 (Efficiency of the Ellipsoid Algorithm [Vis21])**.** *Suppose that $P \subseteq \mathbb{R}^d$ is a full-dimensional polytope that is contained in a d-dimensional Euclidean ball of radius $R > 0$ and contains a d-dimensional Euclidean ball of radius $r > 0$. Then, the ellipsoid method outputs a point $\widetilde{\boldsymbol{x}} \in P$ after $O(d^2 \log(R/r))$ iterations. Moreover, every iteration can be implemented in $O(d^2 + T_{\mathrm{sep}})$ time, where T_{sep} is the time required to answer a single query by the separation oracle.*

Assume that Item 3 holds true. Then the algorithm can be used with $r = 2^{-\mathrm{poly}(d,k,1/\epsilon,\log(1/\delta))}$ and $R = 1$. Hence, the ellipsoid algorithm will provide in time $\mathrm{poly}(d, k, 1/\epsilon, \log(1/\delta))$ a point $\widetilde{\boldsymbol{W}}$ that lies in the feasible region of the convex program 1[5].

□

Remark 2. *We remark that both the improper (Algorithm 1) and the proper (Algorithm 2) learning algorithms hold for the more general case where the $\boldsymbol{x}$-marginal lies in the class of isotropic log-concave distributions [LV07]: A distribution $\mathcal{D}_x$ lies inside the class of isotropic log-concave distributions $\mathcal{F}_{\mathrm{LC}}$ over $\mathbb{R}^d$ if $\mathcal{D}_x$ has a probability density function f over $\mathbb{R}^d$ such that $\log f$ is concave, its mean is zero, and its covariance is identity, i.e., $\mathbf{E}_{\boldsymbol{x}\sim\mathcal{D}_x}[\boldsymbol{x}\boldsymbol{x}^\top] = \boldsymbol{I}$.*

A.2.1 The proof of Lemma 8

We provide the following result.

Lemma 10. *Fix $\epsilon, \delta \in (0,1)$. Let $\boldsymbol{W}^\star \in \mathbb{R}^{k\times d}$ be the true parameter matrix. There exists a matrix $\boldsymbol{W} \in \mathbb{R}^{k\times d}$ such that, with probability at least $1-\delta$:*

- $\mathbf{Pr}_{\boldsymbol{x}\sim\mathcal{N}_d}[\mathrm{sgn}((\boldsymbol{W}_i^\star - \boldsymbol{W}_j^\star)\cdot\boldsymbol{x}) \neq \mathrm{sgn}((\boldsymbol{W}_i - \boldsymbol{W}_j)\cdot\boldsymbol{x})] \leq \epsilon$ *for all $i \neq j$, and,*
- *The bit complexity of $\boldsymbol{W}$ is* $\mathrm{poly}(k, d, 1/\epsilon, \log(1/\delta))$

Proof. The matrix $\boldsymbol{W}$ will be the output of a linear program that can be used to learn the LSF $\sigma_{\boldsymbol{W}^\star}(\cdot)$ in the noiseless setting.

[5]We remark that the runtime will also depend on the time required to answer a single query by the separation oracle. We assume that this time is polynomial in the parameters of our problem and we opt not to track these details in this work.

Consider the unit sphere $\mathcal{S}^{d-1}$ and a δ_0-cover of the unit sphere with parameter $\delta_0 > 0$ to be decided. For any sample $(\boldsymbol{x}, \pi) \sim \mathcal{D}$ of the 0-noisy linear label ranking distribution, i.e., $\boldsymbol{x} \sim \mathcal{N}_d$ and $\pi = \sigma_{\boldsymbol{W}^\star}(\boldsymbol{x})$, we consider the rounded sample $(\widetilde{\boldsymbol{x}}, \pi)$ where $\widetilde{\boldsymbol{x}}$ is obtained by first projecting $\boldsymbol{x} \in \mathbb{R}^d$ to $\mathcal{S}^{d-1}$ and then by obtaining the closest point of $\widehat{x}$ in the cover. The cover's size is $O(1/\delta_0)^d$.

Let us fix $1 \leq i < j \leq k$ and set $y_{ij} = \mathrm{sgn}(\pi(i) - \pi(j))$. For a training set $\{(\boldsymbol{x}^{(t)}, \pi^{(t)})\}_{t \in [N]}$ of size N, we create the following linear system L_{ij} with variables $\boldsymbol{W} \in \mathbb{R}^{k \times d}$:

$$y_{ij}^{(t)} \, (\boldsymbol{W}_i - \boldsymbol{W}_j) \cdot \widetilde{\boldsymbol{x}}^{(t)} \geq 0 \,, \; t \in [N] \qquad (\mathrm{L}_{ij}) \,.$$

Consider the concatenation of the linear systems $\mathrm{L} = \cup_{i<j} \mathrm{L}_{ij}$. The number of equations in the linear system of equations L is $N \cdot \binom{k}{2}$.

We first have to show that, with high probability, the system L is feasible, i.e., there exists $\boldsymbol{W}$ that satisfies the system's equations. Note that if we replace $\widetilde{\boldsymbol{x}}^{(t)}$ with the original points $\boldsymbol{x}^{(t)}$, the true matrix $\boldsymbol{W}^\star$ is a solution to the system. We now have to study the rounded linear system.

Claim 7. *The (rounded) linear system* L *is feasible with high probability.*

Proof. In order to show the feasibility of L, we will use the anti-concentration properties of the Gaussian.

Fact 1 ([DKM05])**.** *Let $\mathcal{P}$ be the standard normal distribution over $\mathbb{R}^d$. For any fixed unit vector $\boldsymbol{a} \in \mathbb{R}^d$ and any $\gamma \leq 1$,*

$$\gamma/4 \leq \mathop{\mathbf{Pr}}_{\boldsymbol{x} \sim \mathcal{P}} \left[|\boldsymbol{a} \cdot \frac{\boldsymbol{x}}{\|\boldsymbol{x}\|_2}| \leq \frac{\gamma}{\sqrt{d}} \right] \leq \gamma \,.$$

Let us focus on the pair $1 \leq i < j \leq k$. We first observe that scaling all samples to lie on the unit sphere does not affect the feasibility of the system. It suffices to focus on that single halfspace with normal vector $\boldsymbol{v}_{ij} = \boldsymbol{W}_i^\star - \boldsymbol{W}_j^\star \in \mathbb{R}^d$ and consider the probability of the event that the collection of the N rounded points $\{\widetilde{\boldsymbol{x}}^{(t)}\}_t$ with labels $\{y_{ij}^{(t)}\}_t$, that come from N Gaussian vectors $\{\boldsymbol{x}^{(t)}\}_t$ which are linearly separable (with labels $\{y_{ij}^{(t)}\}_t$), becomes non-linearly separable. For this it suffices to control the probability that the rounding procedure flips the label of the data point. Using the union bound, we have that, if the rounding has accuracy δ_0, the described bad event has probability

$$\mathop{\mathbf{Pr}}_{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(N)} \sim \mathcal{N}_d} [\exists t \in [N] : \mathrm{sgn}(\boldsymbol{v}_{ij} \cdot \widetilde{\boldsymbol{x}}^{(t)}) \neq \mathrm{sgn}(\boldsymbol{v}_{ij} \cdot \boldsymbol{x}^{(t)})] \leq N \cdot \mathop{\mathbf{Pr}}_{\boldsymbol{x} \sim \mathcal{N}_d} [|\boldsymbol{v}_{ij} \cdot \boldsymbol{x}/\|\boldsymbol{x}\|_2| \leq 2\delta_0] \leq N \cdot O(\delta_0 \sqrt{d}) \,,$$

where we remark that the first event is scale invariant and so we can assume that the normal vector is unit, the first inequality follows from the fact that it suffices to control the mass assigned to a strip of width $2\delta_0$ (due to the discretization) and the second inequality follows from Fact 1. We now have to select the discretization. Let $\delta \in (0, 1)$. By choosing $\delta_0 = O(\frac{\delta}{N\sqrt{d}k^2})$, the bad event for all the pairs $i < j$ occurs with probability at most δ, i.e., with probability at least $1 - \delta$, each one of the N drawn i.i.d. samples does not fall in any one of the $\binom{k}{2}$ "bad" strips. □

We can now consider the case that the system L is feasible (with the target matrix $\boldsymbol{W}^\star$ being a feasible point) that occurs with probability $1 - \delta$. The class of homogenous halfspaces in d dimensions has VC dimension d; therefore, the sample complexity of learning halfspaces using ERM is $O((d + \log(1/\delta))/\epsilon)$. Moreover, in the realizable case, we can implement the ERM using e.g., linear programming and find a solution in $\mathrm{poly}(d, 1/\epsilon, \log(1/\delta))$ time. We next focus on the quality of the solution which will give the desired sample complexity.

Claim 8. *Assume that the algorithm draws $N = \widetilde{O}(\frac{d + \log(k/\delta)}{\epsilon})$ i.i.d. samples of the form $(\boldsymbol{x}, \pi)$ with $\boldsymbol{x} \sim \mathcal{N}_d$ and $\pi = \sigma_{\boldsymbol{W}^\star}(\boldsymbol{x})$. For any $i \neq j$ and with probability at least $1 - 2\delta$, the solution $\boldsymbol{W}$ of the linear system* L *satisfies*

$$\mathop{\mathbf{Pr}}_{\boldsymbol{x} \sim \mathcal{N}_d} [\mathrm{sgn}((\boldsymbol{W}_i^\star - \boldsymbol{W}_j^\star) \cdot \boldsymbol{x}) \neq \mathrm{sgn}((\boldsymbol{W}_i - \boldsymbol{W}_j) \cdot \boldsymbol{x})] \leq \epsilon \,.$$

Proof. Since the matrix $\boldsymbol{W}$ satisfies the sub-system L_{ij}, the result follows using a union bound on the events that (i) the linear system is feasible and (ii) the ERM is a successful PAC learner. □

Claim 9. *Consider the solution $\boldsymbol{W}$ of the linear system. Then, $\boldsymbol{W}$ has bounded bit complexity of order* $\mathrm{poly}(d, k, 1/\epsilon, \log(1/\delta))$.

Proof. We will make use of the following result that relates the size of the input and the output of a linear program using Cramer's rule.

Lemma 11 ([Sch98, Pap81])**.** *Let $\boldsymbol{A} \in \mathbb{Z}^{m\times n}, \boldsymbol{b} \in \mathbb{Z}^m, \boldsymbol{c} \in \mathbb{Z}^n$. Consider a linear program $\min \boldsymbol{c}\cdot\boldsymbol{x}$ subject to $\boldsymbol{A}\boldsymbol{x} \le \boldsymbol{b}$ and $\boldsymbol{x} \ge \boldsymbol{0}$. Let U be the maximum size of A_{ij}, b_i, c_j. The output of the linear program has size $O(m(nU + n\log(n)))$ bits.*

We will apply the above lemma (which holds even by dropping the constraint $\boldsymbol{x} \ge \boldsymbol{0}$) to our setting where $\boldsymbol{A}\boldsymbol{w} \ge 0$ where $\boldsymbol{w} = (\boldsymbol{W}_i)_{i\in[k]} \in \mathbb{Q}^{kd}$, i.e., $\boldsymbol{w}$ is the vectorization of the matrix $\boldsymbol{W}$. Moreover, $\boldsymbol{A}$ is the matrix containing the N (rounded) Gaussian samples $\widetilde{\boldsymbol{x}}^{(t)}$. We have that the matrix $\boldsymbol{A}$ has dimension $N\binom{k}{2} \times kd$ and each entry A_{ij} is an integer and has size at most $U = \mathrm{poly}(d, k)$ (since the samples are rounded on the δ_0-cover of the sphere. Recall that the labels $y_{ij}^{(t)} \in \{-1, +1\}$ and $\widetilde{\boldsymbol{x}}^{(t)}$ lie in the unit sphere. In particular, each row of the matrix $\boldsymbol{A}$ has $2d$ non-zero entries and is associated with a tuple (i, j, t) for $1 \le i < j \le k$ and $t \in [N]$. Then, it holds that the output has size at most $O(Nk^2(dU + dk\log(dk)))$ bits. So, we get that the output $\boldsymbol{W}$ can be described using at most $\mathrm{poly}(d, k, 1/\epsilon, U, \log(1/\delta)) = \mathrm{poly}(d, k, 1/\epsilon, \log(1/\delta))$ bits (due to the size of the entries of the matrix $\boldsymbol{A}$). □

Combining the above claims, we conclude the proof. □

As a corollary of the bounded bit complexity, we obtain the following key result.

Corollary 1. *Let $\epsilon > 0$. Assume that $\boldsymbol{W} \in \mathbb{R}^{k\times d}$ has bit complexity at most* $\mathrm{poly}(d, k, 1/\epsilon, \log(1/\delta))$. *Then, for any $i, j \in [k]$ with $i \ne j$, it holds that $\|\boldsymbol{W}_i - \boldsymbol{W}_j\|_2 > 2^{-\mathrm{poly}(d,k,1/\epsilon,\log(1/\delta))}$.*

Proof. First, we can assume that $\boldsymbol{W}_i \ne \boldsymbol{W}_j$ for any $i \ne j$; in case of equal rows, we obtain a low-dimensional instance. Then, since any vector $\boldsymbol{W}_i$ has bounded bit complexity, we have that the difference of any two such vectors, provided that it is non-zero, has a lower bound in its norm, i.e., $\|\boldsymbol{W}_i - \boldsymbol{W}_j\|_2 > 2^{-\mathrm{poly}(d,k,1/\epsilon,\log(1/\delta))}$ for any $i, j \in [k]$. □

B Learning in Top-1 Disagreement from Label Rankings

Let us set $\sigma_1(\boldsymbol{W}\boldsymbol{x}) = \mathrm{argmax}_{i\in[k]} \boldsymbol{W}_i \cdot \boldsymbol{x}$ for $\boldsymbol{x} \in \mathbb{R}^d$. The main result of this section follows.

Theorem 5 (Proper Top-1 Learning Algorithm)**.** *Fix $\eta \in [0, 1/2)$ and $\epsilon, \delta \in (0, 1)$. Let $\mathcal{D}$ be an η-noisy linear label ranking distribution satisfying the assumptions of Definition 2. There exists an algorithm that draws $N = O\left(\frac{dk\sqrt{\log k}}{\epsilon(1-2\eta)^6}\log(k/\delta)\right)$ samples from $\mathcal{D}$, runs in* $\mathrm{poly}(N)$ *time and, with probability at least $1 - \delta$, outputs a Linear Sorting function $h : \mathbb{R}^d \to \mathbb{S}_k$ that is ϵ-close in top-1 disagreement to the target.*

Proof. Note that the `MassartLTF` algorithm (see Lemma 6) has the guarantee that it returns a vector $\boldsymbol{w}$ so that

$$\mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{N}_d}[\mathrm{sgn}(\boldsymbol{w}\cdot\boldsymbol{x}) \ne \mathrm{sgn}(\boldsymbol{w}^\star\cdot\boldsymbol{x})] \le \epsilon\,,$$

with probability $1 - \delta$, where $\boldsymbol{w}^\star$ is the target normal vector. Since the above misclassification probability with respect to $\mathcal{N}_d$ is directly connected with the angle $\theta(\boldsymbol{w}, \boldsymbol{w}^\star)$, we get that we can control the angle between $\boldsymbol{w}$ and $\boldsymbol{w}^\star$ efficiently. Moreover, in our setting, for a matrix $\boldsymbol{W} \in \mathbb{R}^{k\times d}$, there exist $\binom{k}{2}$ homogeneous halfspaces with normal vectors $\boldsymbol{W}_i - \boldsymbol{W}_j$ and so we can control the angles $\theta(\boldsymbol{W}_i - \boldsymbol{W}_j, \boldsymbol{W}_i^\star - \boldsymbol{W}_j^\star)$. In order to deduce the sample complexity bound of Theorem 5, we show the next lemma which essentially bounds the top-1 misclassification error using the angles of these $O(k^2)$ halfspaces. We apply Lemma 12 with $\boldsymbol{U} = \boldsymbol{W}$ and $\boldsymbol{V} = \boldsymbol{W}^\star$ and so we can take $\epsilon' = \epsilon/(k\sqrt{\log k})$ and invoke the proper learning algorithm of Algorithm 2. This completes the proof. □

We continue with the proof of our key lemma.

Lemma 12 (Misclassification Error)**.** *Consider two matrices $\boldsymbol{U}, \boldsymbol{V} \in \mathbb{R}^{k\times d}$ and let $\mathcal{N}_d$ be the standard Gaussian in d dimensions. We have that*

$$\Pr_{\boldsymbol{x}\sim\mathcal{N}_d}[\sigma_1(\boldsymbol{U}\boldsymbol{x}) \neq \sigma_1(\boldsymbol{V}\boldsymbol{x})] \leq c\cdot k\cdot\sqrt{\log k}\cdot \max_{i\neq j}\theta(\boldsymbol{U}_i-\boldsymbol{U}_j, \boldsymbol{V}_i-\boldsymbol{V}_j)\,,$$

where $c>0$ is some universal constant.

Proof. We have that

$$\Pr_{\boldsymbol{x}\sim\mathcal{N}_d}[\sigma_1(\boldsymbol{U}\boldsymbol{x}) \neq \sigma_1(\boldsymbol{V}\boldsymbol{x})] = \sum_{i\in[k]}\Pr_{\boldsymbol{x}\sim\mathcal{N}_d}[\sigma_1(\boldsymbol{U}\boldsymbol{x}) = i, \sigma_1(\boldsymbol{V}\boldsymbol{x}) \neq i]\,.$$

We have that $\mathcal{C}_{\boldsymbol{U}}^{(i)} = \mathbf{1}\{\boldsymbol{x} : \sigma_1(\boldsymbol{U}\boldsymbol{x}) = i\} = \prod_{j\neq i}\mathbf{1}\{(\boldsymbol{U}_i-\boldsymbol{U}_j)\cdot\boldsymbol{x}\geq 0\}$ is the set indicator of a homogeneous polyhedral cone as the intersection of $k-1$ homogeneous halfspaces. Similarly, we consider the cone $\mathcal{C}_{\boldsymbol{V}}^{(i)} = \{\boldsymbol{x} : \sigma_1(\boldsymbol{V}\boldsymbol{x}) = i\}$. Hence, we have that $\{\boldsymbol{x} : \sigma_1(\boldsymbol{V}\boldsymbol{x}) \neq i\}$ is the complement of a homogeneous polyhedral cone. Let us define $C_{\boldsymbol{U}}^{(i)} : \mathbb{R}^d \mapsto \{0,1\}$ and $C_{\boldsymbol{V}}^{(i)} : \mathbb{R}^d \mapsto \{0,1\}$ be the associated indicator functions of the two cones. We have that

$$\Pr_{\boldsymbol{x}\sim\mathcal{N}_d}[\sigma_1(\boldsymbol{U}\boldsymbol{x}) = i, \sigma_1(\boldsymbol{V}\boldsymbol{x}) \neq i] = \Pr_{\boldsymbol{x}\sim\mathcal{N}_d}[C_{\boldsymbol{U}}^{(i)}(\boldsymbol{x}) = 1, C_{\boldsymbol{V}}^{(i)}(\boldsymbol{x}) = 0]\,.$$

Finally, we have that

$$\mathcal{C}_{\boldsymbol{U}}^{(i)} \cap \left(\mathcal{C}_{\boldsymbol{V}}^{(i)}\right)^c = \mathcal{C}_{\boldsymbol{U}}^{(i)} \setminus \mathcal{C}_{\boldsymbol{V}}^{(i)} \subseteq \mathcal{C}_{\boldsymbol{U}}^{(i)} \setminus \mathcal{C}_{\boldsymbol{V}}^{(i)} \cup \mathcal{C}_{\boldsymbol{V}}^{(i)} \setminus \mathcal{C}_{\boldsymbol{U}}^{(i)}\,.$$

We can hence apply Lemma 13 for the cones $\mathcal{C}_{\boldsymbol{U}}^{(i)}, \mathcal{C}_{\boldsymbol{V}}^{(i)}$ for each $i\in[k]$. □

Lemma 13 (Cone Disagreement)**.** *Let $C_1 : \mathbb{R}^d \mapsto \{0,1\}$ be the indicator function of the homogeneous polyhedral cone defined by the k unit vectors $\boldsymbol{v}_1,\ldots,\boldsymbol{v}_k \in \mathbb{R}^d$, i.e., $C_1(\boldsymbol{x}) = \prod_{i=1}^k \mathbf{1}\{\boldsymbol{v}_i\cdot\boldsymbol{x}\geq 0\}$. Similarly, define $C_2 : \mathbb{R}^d \mapsto \{0,1\}$ to be the homogeneous polyhedral cone with normal vectors $\boldsymbol{u}_1,\ldots,\boldsymbol{u}_k$. It holds that*

$$\Pr_{\boldsymbol{x}\sim\mathcal{N}_d}[C_1(\boldsymbol{x}) \neq C_2(\boldsymbol{x})] \leq c\sqrt{\log(k)}\,\max_{i\in[k]}\theta(\boldsymbol{v}_i,\boldsymbol{u}_i)\,,$$

where $c>0$ is some universal constant.

Proof. To simplify notation, denote $\theta = \max_{i\in[k]}\theta(\boldsymbol{v}_i,\boldsymbol{u}_i)$. We first observe that it suffices to prove the upper bound on the probability of $C_1(\boldsymbol{x}) \neq C_2(\boldsymbol{x})$ for sufficiently small values of θ. Indeed, if we have that the bound is true for θ smaller than some θ_0 we can then form a path of sufficiently large length N (in particular we need $\theta/N \leq \theta_0$) starting from the vectors $\boldsymbol{v}_1,\ldots,\boldsymbol{v}_k$ to the final vectors $\boldsymbol{u}_1,\ldots,\boldsymbol{u}_k$, where at each step we only rotate the vectors by at most $\theta/N \leq \theta_0$. By the triangle inequality, we immediately obtain that the probability that $C_1(\boldsymbol{x}) \neq C_2(\boldsymbol{x})$ is at most equal to the sum of the probabilities of the intermediate steps which is at most $\sum_{i=1}^N c\sqrt{\log(k)}\frac{\theta}{N} = c\sqrt{\log(k)}\theta$. Notice in the above argument the constant θ_0 can be arbitrarily small and may also depend on k and d.

We define the indicator of the positive orthant in k dimensions to be $R(\boldsymbol{t}) = \prod_{i=1}^k \mathbf{1}\{\boldsymbol{t}_i \geq 0\}$. Using this notation, we have that the cone indicator can be written as $C_1(\boldsymbol{x}) = R(\boldsymbol{v}_1\cdot\boldsymbol{x},\ldots,\boldsymbol{v}_k\cdot\boldsymbol{x}) = R(\boldsymbol{V}\boldsymbol{x})$, where $\boldsymbol{V}$ is the $k\times d$ matrix whose i-th row is the vector $\boldsymbol{v}_i$. Moreover, we define the i-th face of the cone $R(\boldsymbol{V}\boldsymbol{x})$ to be

$$F_i(\boldsymbol{V}\boldsymbol{x}) = R(\boldsymbol{V}\boldsymbol{x})\,\mathbf{1}\{\boldsymbol{v}_i\cdot\boldsymbol{x} = 0\}\,.$$

We will first handle the case where only one of the normal vectors $\boldsymbol{v}_i$ changes. We show the following claim.

Claim 10. *Let $\boldsymbol{v}_1,\ldots,\boldsymbol{v}_k \in \mathbb{R}^d$ and $\boldsymbol{r}\in\mathbb{R}^d$ with $\theta(\boldsymbol{v}_1,\boldsymbol{r}) \leq \theta$ for some sufficiently small $\theta \in (0,\pi/2)$. It holds that*

$$\Pr_{\boldsymbol{x}\sim\mathcal{N}_d}[R(\boldsymbol{v}_1\cdot\boldsymbol{x},\ldots,\boldsymbol{v}_k\cdot\boldsymbol{x}) \neq R(\boldsymbol{r}\cdot\boldsymbol{x},\boldsymbol{v}_2\cdot\boldsymbol{x},\ldots,\boldsymbol{v}_k\cdot\boldsymbol{x})] \leq c\cdot\theta\cdot\Gamma(F_1)\sqrt{\log\left(\frac{1}{\Gamma(F_1)}+1\right)}\,,$$

where F_1 is the face with $\boldsymbol{v}_1\cdot\boldsymbol{x} = 0$ of the cone $R(\boldsymbol{V}\boldsymbol{x})$ and c is some universal constant.

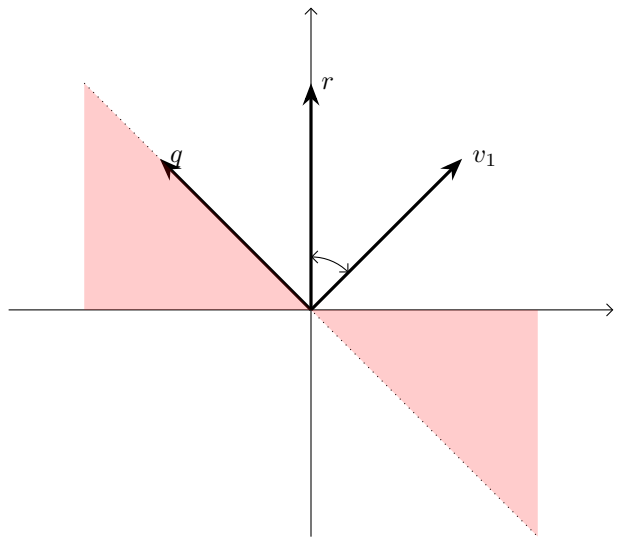

Figure 1: The vectors $\boldsymbol{r}, \boldsymbol{v}_1$ and $\boldsymbol{q}$ and the disagreement region of the halfspaces with normal vectors $\boldsymbol{r}$ and $\boldsymbol{v}_1$.

Proof. We have

$$\begin{aligned}
&\Pr_{\boldsymbol{x}\sim\mathcal{N}_d}\left[R(\boldsymbol{v}_1\cdot\boldsymbol{x},\ldots,\boldsymbol{v}_k\cdot\boldsymbol{x})\neq R(\boldsymbol{r}\cdot\boldsymbol{x},\boldsymbol{v}_2\cdot\boldsymbol{x},\ldots,\boldsymbol{v}_k\cdot\boldsymbol{x})\right]\\
&\quad=\mathop{\mathbf{E}}_{\boldsymbol{x}\sim\mathcal{N}_d}\left[\left|R(\boldsymbol{v}_1\cdot\boldsymbol{x},\ldots,\boldsymbol{v}_k\cdot\boldsymbol{x})-R(\boldsymbol{r}\cdot\boldsymbol{x},\boldsymbol{v}_2\cdot\boldsymbol{x},\ldots,\boldsymbol{v}_k\cdot\boldsymbol{x})\right|\right]\\
&\quad=\mathop{\mathbf{E}}_{\boldsymbol{x}\sim\mathcal{N}_d}\left[R(\boldsymbol{v}_2\cdot\boldsymbol{x},\ldots,\boldsymbol{v}_k\cdot\boldsymbol{x})\left|\mathbf{1}\{\boldsymbol{v}_1\cdot\boldsymbol{x}\geq 0\}-\mathbf{1}\{\boldsymbol{r}\cdot\boldsymbol{x}\geq 0\}\right|\right].
\end{aligned}$$

We have that $|\mathbf{1}\{\boldsymbol{v}_1\cdot\boldsymbol{x}\geq 0\}-\mathbf{1}\{\boldsymbol{r}\cdot\boldsymbol{x}\geq 0\}|=\mathbf{1}\{(\boldsymbol{v}_1\cdot\boldsymbol{x})(\boldsymbol{r}\cdot\boldsymbol{x})<0\}$, i.e., this is the event that the halfspaces $\mathbf{1}\{\boldsymbol{v}_1\cdot\boldsymbol{x}\geq 0\}$ and $\mathbf{1}\{\boldsymbol{r}\cdot\boldsymbol{x}\geq 0\}$ disagree. Let $\boldsymbol{q}$ be the normalized projection of $\boldsymbol{r}$ onto the orthogonal complement of $\boldsymbol{v}_1$, i.e., $\boldsymbol{q}=\mathrm{proj}_{\boldsymbol{v}_1^\perp}\boldsymbol{r}/\|\mathrm{proj}_{\boldsymbol{v}_1^\perp}\boldsymbol{r}\|_2$. We have that $\boldsymbol{v}_1$ and $\boldsymbol{q}$ is an orthonormal basis of the subspace spanned by the vectors $\boldsymbol{v}_1$ and $\boldsymbol{r}$. We have that $\boldsymbol{r}=\cos\theta(\boldsymbol{v}_1,\boldsymbol{r})\boldsymbol{v}_1+\sin\theta(\boldsymbol{v}_1,\boldsymbol{r})\boldsymbol{q}$. Moreover, we have that the region $(\boldsymbol{v}_1\cdot\boldsymbol{x})(\boldsymbol{r}\cdot\boldsymbol{x})<0$ is equal to

$$\{0<\boldsymbol{v}_1\cdot\boldsymbol{x}<-(\boldsymbol{q}\cdot\boldsymbol{x})\tan\theta(\boldsymbol{v}_1,\boldsymbol{r})\}\cup\{-(\boldsymbol{q}\cdot\boldsymbol{x})\tan\theta(\boldsymbol{v}_1,\boldsymbol{r})<\boldsymbol{v}_1\cdot\boldsymbol{x}<0\}.$$

Thus, we have that the disagreement region $(\boldsymbol{v}_1\cdot\boldsymbol{x})(\boldsymbol{r}\cdot\boldsymbol{x})<0$ is a subset of the region $\{|\boldsymbol{v}_1\cdot\boldsymbol{x}|\leq|\boldsymbol{q}\cdot\boldsymbol{x}|\tan\theta(\boldsymbol{v}_1,\boldsymbol{r})\}$. Since $\tan\theta(\boldsymbol{v}_1,\boldsymbol{r})\leq\theta$ and we have that θ is sufficiently small we can also replace the above region by the larger region: $\{|\boldsymbol{v}_1\cdot\boldsymbol{x}|\leq 2\theta|\boldsymbol{q}\cdot\boldsymbol{x}|\}$. Therefore, we have

$$\begin{aligned}
&\mathop{\mathbf{E}}_{\boldsymbol{x}\sim\mathcal{N}_d}\left[R(\boldsymbol{v}_2\cdot\boldsymbol{x},\ldots,\boldsymbol{v}_k\cdot\boldsymbol{x})\,\mathbf{1}\{(\boldsymbol{v}_1\cdot\boldsymbol{x})(\boldsymbol{r}\cdot\boldsymbol{x})<0\}\}\right]\\
&\qquad\leq\mathop{\mathbf{E}}_{\boldsymbol{x}\sim\mathcal{N}_d}\left[R(\boldsymbol{v}_2\cdot\boldsymbol{x},\ldots,\boldsymbol{v}_k\cdot\boldsymbol{x})\,\mathbf{1}\{|\boldsymbol{v}_1\cdot\boldsymbol{x}|\leq 2\theta|\boldsymbol{q}\cdot\boldsymbol{x}|\}\right].
\end{aligned}$$

The derivative of the above expression with respect to θ is equal to

$$\mathop{\mathbf{E}}_{\boldsymbol{x}\sim\mathcal{N}_d}\left[R(\boldsymbol{v}_2\cdot\boldsymbol{x},\ldots,\boldsymbol{v}_k\cdot\boldsymbol{x})\,\delta\left(\frac{|\boldsymbol{v}_1\cdot\boldsymbol{x}|}{2|\boldsymbol{q}\cdot\boldsymbol{x}|}-\theta\right)\right],$$

where $\delta(t)$ is the Dirac delta function. At $\theta=0$ and using the property that $\delta(t/a)=a\delta(t)$, we have that the above derivative is equal to

$$2\mathop{\mathbf{E}}_{\boldsymbol{x}\sim\mathcal{N}_d}\left[R(\boldsymbol{v}_2\cdot\boldsymbol{x},\ldots,\boldsymbol{v}_k\cdot\boldsymbol{x})\,|\boldsymbol{q}\cdot\boldsymbol{x}|\,\delta(|\boldsymbol{v}_1\cdot\boldsymbol{x}|)\right].$$

Notice that, if we did not have the term $|\boldsymbol{q}\cdot\boldsymbol{x}|$, the above expression would be exactly equal to two times the Gaussian surface area of the face with $\boldsymbol{v}_1\cdot\boldsymbol{x}=0$, i.e., it would be equal to $2\Gamma(F_1)$. We now show that this extra term of $|\boldsymbol{q}\cdot\boldsymbol{x}|$ can only increase the above surface integral by at most a

logarithmic factor. We have that

$$\mathop{\mathbf{E}}_{\boldsymbol{x}\sim\mathcal{N}_d}\left[R(\boldsymbol{v}_2\cdot\boldsymbol{x},\ldots,\boldsymbol{v}_k\cdot\boldsymbol{x})\,|\boldsymbol{q}\cdot\boldsymbol{x}|\,\delta(|\boldsymbol{v}_1\cdot\boldsymbol{x}|)\right]=\int_{\boldsymbol{x}\in F_1}\phi_d(\boldsymbol{x})|\boldsymbol{q}\cdot\boldsymbol{x}|d\mu(\boldsymbol{x})$$
$$\leq\int_{\boldsymbol{x}\in F_1}\phi_d(\boldsymbol{x})|\boldsymbol{q}\cdot\boldsymbol{x}|\mathbf{1}\{|\boldsymbol{q}\cdot\boldsymbol{x}|\leq\xi\}d\mu(\boldsymbol{x})+\int_{\boldsymbol{x}\in F_1}\phi_d(\boldsymbol{x})|\boldsymbol{q}\cdot\boldsymbol{x}|\mathbf{1}\{|\boldsymbol{q}\cdot\boldsymbol{x}|\geq\xi\}d\mu(\boldsymbol{x})$$
$$\leq\xi\int_{\boldsymbol{x}\in F_1}\phi_d(\boldsymbol{x})d\mu(\boldsymbol{x})+\int_{\boldsymbol{x}\in F_1}\phi_d(\boldsymbol{x})|\boldsymbol{q}\cdot\boldsymbol{x}|\mathbf{1}\{|\boldsymbol{q}\cdot\boldsymbol{x}|\geq\xi\}d\mu(\boldsymbol{x})\,,$$

where $d\mu(\boldsymbol{x})$ is the standard surface measure in $\mathbb{R}^d$. The first term above is exactly equal to the Gaussian surface area of the face F_1. To bound from above the second term we can use the fact that the face F_1 is a subset of the hyperplane $\boldsymbol{v}_1\cdot\boldsymbol{x}=0$, i.e., it holds that $F_1\subseteq\{\boldsymbol{x}:|\boldsymbol{v}_1\cdot\boldsymbol{x}|=0\}$. To simplify notation we may assume that $\boldsymbol{v}_1=\boldsymbol{e}_1$ and $\boldsymbol{q}=\boldsymbol{e}_2$ (recall that $\boldsymbol{v}_1$ and $\boldsymbol{q}$ are orthogonal unit vectors), and in this case we obtain

$$\begin{aligned}\int_{\boldsymbol{x}\in F_1}\phi_d(\boldsymbol{x})|\boldsymbol{q}\cdot\boldsymbol{x}|\mathbf{1}\{|\boldsymbol{q}\cdot\boldsymbol{x}|\geq\xi\}d\mu(\boldsymbol{x})&\leq\int_{x_1=0}\phi_d(\boldsymbol{x})|x_2|\mathbf{1}\{|x_2|\geq\xi\}d\mu(\boldsymbol{x})\\&=\frac{1}{\sqrt{2\pi}}\int_{-\infty}^{+\infty}|x_2|\mathbf{1}\{|x_2|\geq\xi\}\frac{e^{-x_2^2/2}}{\sqrt{2\pi}}dx_2\\&=\frac{1}{\pi}e^{-\xi^2/2}\,.\end{aligned}$$

Combining the above bounds we obtain that the derivative with respect to θ of the expression $\mathbf{E}_{\boldsymbol{x}\sim\mathcal{N}_d}\left[R(\boldsymbol{v}_2\cdot\boldsymbol{x},\ldots,\boldsymbol{v}_k\cdot\boldsymbol{x})\,\mathbf{1}\{|\boldsymbol{v}_1\cdot\boldsymbol{x}|\leq 2\theta|\boldsymbol{q}\cdot\boldsymbol{x}|\}\right]$ is equal to

$$\frac{d}{d\theta}\Big(\mathop{\mathbf{E}}_{\boldsymbol{x}\sim\mathcal{N}_d}\left[R(\boldsymbol{v}_2\cdot\boldsymbol{x},\ldots,\boldsymbol{v}_k\cdot\boldsymbol{x})\,\mathbf{1}\{|\boldsymbol{v}_1\cdot\boldsymbol{x}|\leq 2\theta|\boldsymbol{q}\cdot\boldsymbol{x}|\}\right]\Big)\Big|_{\theta=0}\leq 2\xi\Gamma(F_1)+\frac{2e^{-\xi^2/2}}{\pi}\,.$$

By picking $\xi=\sqrt{2\log(1+1/\Gamma(F_1))}$, the result follows since up to introducing $o(\theta)$ error we can bound the term $\mathbf{Pr}_{\boldsymbol{x}\sim\mathcal{N}_d}\left[R(\boldsymbol{v}_1\cdot\boldsymbol{x},\ldots,\boldsymbol{v}_k\cdot\boldsymbol{x})\neq R(\boldsymbol{r}\cdot\boldsymbol{x},\boldsymbol{v}_2\cdot\boldsymbol{x},\ldots,\boldsymbol{v}_k\cdot\boldsymbol{x})\right]$ by its derivative with respect to θ (evaluated at 0) times θ. □

We can complete the proof of Lemma 13 using Claim 10. In order to bound the disagreement of the cones C_1 and C_2 we can start from C_1 and change one of its vectors at a time so that we can use Claim 10 that can handle this case. For example, at the first step, we can swap $\boldsymbol{v}_1$ for $\boldsymbol{u}_1$ and use the triangle inequality to obtain that

$$\begin{aligned}\mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{N}_d}[C_1(\boldsymbol{x})\neq C_2(\boldsymbol{x})]&\leq\mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{N}_d}[R(\boldsymbol{v}_1\cdot\boldsymbol{x},\ldots,\boldsymbol{v}_k\cdot\boldsymbol{x})\neq R(\boldsymbol{u}_1\cdot\boldsymbol{x},\boldsymbol{v}_2\cdot\boldsymbol{x}\ldots,\boldsymbol{v}_k\cdot\boldsymbol{x})]\\&+\mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{N}_d}[R(\boldsymbol{u}_1\cdot\boldsymbol{x},\boldsymbol{v}_2\cdot\boldsymbol{x},\ldots,\boldsymbol{v}_k\cdot\boldsymbol{x})\neq R(\boldsymbol{u}_1\cdot\boldsymbol{x},\boldsymbol{u}_2\cdot\boldsymbol{x}\ldots,\boldsymbol{u}_k\cdot\boldsymbol{x})]\\&\leq c\cdot\theta\,\Gamma(F_1)\sqrt{\log(1/\Gamma(F_1)+1)}\\&+\mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{N}_d}[R(\boldsymbol{u}_1\cdot\boldsymbol{x},\boldsymbol{v}_2\cdot\boldsymbol{x},\ldots,\boldsymbol{v}_k\cdot\boldsymbol{x})\neq R(\boldsymbol{u}_1\cdot\boldsymbol{x},\boldsymbol{u}_2\cdot\boldsymbol{x}\ldots,\boldsymbol{u}_k\cdot\boldsymbol{x})]\,,\end{aligned}$$

where $F_1=F_1(\boldsymbol{V}\boldsymbol{x})$ is the face with $\boldsymbol{v}_1\cdot\boldsymbol{x}=0$ of the cone C_1. Notice that we have replaced $\boldsymbol{v}_1$ by $\boldsymbol{u}_1$ in the above bound. Our plan is to use the triangle inequality and continue replacing the vectors of C_1 by the vectors of C_2 sequentially. To make this formal we define the matrix $\boldsymbol{A}^{(i)}\in\mathbb{R}^{k\times d}$ whose first $i-1$ rows are the vectors $\boldsymbol{u}_1,\ldots,\boldsymbol{u}_{i-1}$ and its last $k-i+1$ rows are the vectors $\boldsymbol{v}_i,\ldots,\boldsymbol{v}_k$, i.e.,

$$\boldsymbol{A}_j^{(i)}=\begin{cases}\boldsymbol{u}_j & \text{if}\quad 1\leq j\leq i-1,\\ \boldsymbol{v}_j & \text{if}\quad i\leq j\leq k\,.\end{cases}$$

Notice that $\boldsymbol{A}^{(1)}=\boldsymbol{V}$ and $\boldsymbol{A}^{(k+1)}=\boldsymbol{U}$. Using the triangle inequality we obtain that

$$\mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{N}_d}[C_1(\boldsymbol{x})\neq C_2(\boldsymbol{x})]\leq\sum_{i=1}^{k}\mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{N}_d}[R(\boldsymbol{A}^{(i)}\boldsymbol{x})\neq R(\boldsymbol{A}^{(i+1)}\boldsymbol{x})].$$

Since the matrices $\boldsymbol{A}^{(i)}$ and $\boldsymbol{A}^{(i+1)}$ only differ on one row, we can use Claim 10 to obtain the following bound:

$$\mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{N}_d}[C_1(\boldsymbol{x}) \neq C_2(\boldsymbol{x})] \leq c \cdot \theta \cdot \sum_{i=1}^{k} \Gamma(F_i(\boldsymbol{A}^{(i)}\boldsymbol{x}))\sqrt{\frac{1}{\Gamma(F_i(\boldsymbol{A}^{(i)}\boldsymbol{x}))} + 1}\,.$$

We now observe that the Gaussian surface area $\Gamma(F_i(\boldsymbol{A}^{(i)}\boldsymbol{x}))$ is a continuous function of the matrix $\boldsymbol{A}^{(i)}$. By flattening the matrix $\boldsymbol{A}^{(i)}$ (since it is isomorphic to a vector $\boldsymbol{z} \in \mathbb{R}^{n^2}$) and letting $S_{\boldsymbol{z}}$ be the induced surface $\{\boldsymbol{x} : R(\boldsymbol{A}^{(i)}\boldsymbol{x}) = 1 \wedge \boldsymbol{v}_i \cdot \boldsymbol{x} = 0\}$, it suffices to show that

$$\lim_{\boldsymbol{w}\to\boldsymbol{z}} \int \phi_n(\boldsymbol{x})\mathbf{1}\{\boldsymbol{x} \in S_{\boldsymbol{w}}\}d\mu(\boldsymbol{x}) = \int \phi_n(\boldsymbol{x})\mathbf{1}\{\boldsymbol{x} \in S_{\boldsymbol{z}}\}d\mu(\boldsymbol{x})\,,$$

by the smoothness of the surface $S_{\boldsymbol{z}}$. Consider a sequence of functions (g_m) and vectors $(\boldsymbol{w}_m)$ so that $g_m(\boldsymbol{x}) = \phi_n(\boldsymbol{x})\mathbf{1}\{\boldsymbol{x} \in S_{\boldsymbol{w}_m}\}$ and $\lim_{m\to\infty} \boldsymbol{w}_m = \boldsymbol{z}$. Note that $|g_m(\boldsymbol{x})| \leq 1$ everywhere. Hence, by the dominated convergence theorem, we have that

$$\lim_{m\to\infty} \int g_m(\boldsymbol{x})d\mu(\boldsymbol{x}) = \int \lim_{m\to\infty} g_m(\boldsymbol{x})d\mu(\boldsymbol{x}) = \int \phi_n(\boldsymbol{x}) \lim_{m\to\infty} \mathbf{1}\{\boldsymbol{x} \in S_{\boldsymbol{w}_m}\}d\mu(\boldsymbol{x})\,.$$

Since the sequence consists of smooth surfaces, we have that $\lim_{m\to\infty} \mathbf{1}\{\boldsymbol{x} \in S_{\boldsymbol{w}_m}\} = \mathbf{1}\{\boldsymbol{x} \in S_{\boldsymbol{z}}\}$ and so the Gaussian surface area is continuous with respect to the matrix $\boldsymbol{A}^{(i)}$ for any $i \in [k]$.

Also, as $\theta \to 0$, we have that $\boldsymbol{A}^{(i)} \to \boldsymbol{V}$. This is because the sequence of matrices $\boldsymbol{A}^{(i)}$ depends only on the vectors $\boldsymbol{u}_j$ and $\boldsymbol{v}_j$ for $j \in [k]$ and the following two properties hold true: $\theta = \max_{j\in[k]} \theta(\boldsymbol{v}_j, \boldsymbol{u}_j)$ and all the vectors are unit. Hence, as θ tends to zero, they tend to become the same vectors and so any matrix $\boldsymbol{A}^{(i)}$ tends to become $\boldsymbol{V}$. Therefore, taking this limit we obtain that for $\theta \to 0$ it holds that

$$\lim_{\theta\to 0} \frac{\mathbf{Pr}_{\boldsymbol{x}\sim\mathcal{N}_d}[C_1(\boldsymbol{x}) \neq C_2(\boldsymbol{x})]}{\theta} \leq c \cdot \sum_{i=1}^{k} \Gamma(F_i(\boldsymbol{V}\boldsymbol{x}))\sqrt{\log\left(1/\Gamma(F_i(\boldsymbol{V}\boldsymbol{x})) + 1\right)}\,. \tag{1}$$

We will now use the following lemma that shows that the surface area of any homogeneous polyhedral cone is independent of the number of faces k and in fact is at most 1 for all k.

Lemma 14 (Gaussian Surface Area of Homogeneous Cones [Naz03])**.** *Let C be a cone with apex at the origin (i.e., an intersection of arbitrarily many halfspaces all of whose boundaries contain the origin). Then C has Gaussian surface area $\Gamma(C)$ at most* 1.

Using Lemma 14 we obtain that $\sum_{i=1}^{k} \Gamma(F_i(\boldsymbol{V}\boldsymbol{x})) \leq 1$. Next, we observe that, when the positive numbers $a_1, \ldots, a_k$ satisfy $\sum_{i=1}^{k} a_i \leq 1$, it holds that $\sum_{i=1}^{k} a_i\sqrt{\log(1/a_i)} \leq \sqrt{\sum_{i=1}^{k} a_i \log(1/a_i)} \leq \sqrt{\log(k)}$ (using the fact that the uniform distribution maximizes the entropy). Using this fact and Equation (1), we obtain

$$\lim_{\theta\to 0} \frac{\mathbf{Pr}_{\boldsymbol{x}\sim\mathcal{N}_d}[C_1(\boldsymbol{x}) \neq C_2(\boldsymbol{x})]}{\theta} \leq c\sqrt{\log(k)}\,.$$

Thus, we have shown that, for sufficiently small θ, it holds that $\mathbf{Pr}_{\boldsymbol{x}\sim\mathcal{N}_d}[C_1(\boldsymbol{x}) \neq C_2(\boldsymbol{x})] \leq c\sqrt{\log(k)}\theta$, but, as we discussed in the start of the proof, the general bound follows directly from the bound for sufficiently small values of $\theta > 0$. □

C Learning in Top-r Disagreement from Label Rankings

We prove the next result which corresponds to a proper learning algorithm for LSF in the presence of bounded noise with respect to the top-r disagreement.

Theorem 6 (Proper Top-r Learning Algorithm)**.** *Fix $\eta \in [0, 1/2)$, $r \in [k]$ and $\epsilon, \delta \in (0, 1)$. Let $\mathcal{D}$ be an η-noisy linear label ranking distribution satisfying the assumptions of Definition 2. There exists an algorithm that draws $N = \widetilde{O}\left(\frac{d\,rk}{\epsilon(1-2\eta)^6}\log(1/\delta)\right)$ samples from $\mathcal{D}$, runs in* poly(N) *time and, with probability at least $1-\delta$, outputs a Linear Sorting function $h : \mathbb{R}^d \to \mathbb{S}_k$ that is ϵ-close in top-r disagreement to the target.*

The main result of this section is the next lemma, which directly implies the above theorem (using the same steps as the proof of Theorem 5).

Lemma 15 (Top-r Misclassification)**.** *Let $r \in [k]$. Consider two matrices $\boldsymbol{U}, \boldsymbol{V} \in \mathbb{R}^{k\times d}$ and let $\mathcal{N}_d$ be the standard Gaussian in d dimensions. We have that*

$$\mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{N}_d}[\sigma_{1..r}(\boldsymbol{U}\boldsymbol{x}) \neq \sigma_{1..r}(\boldsymbol{V}\boldsymbol{x})] \leq c\cdot k\cdot r\cdot\sqrt{\log(kr)}\cdot\max_{i\neq j}\theta(\boldsymbol{U}_i-\boldsymbol{U}_j,\boldsymbol{V}_i-\boldsymbol{V}_j)\,,$$

where $c > 0$ is some universal constant.

Proof. Let us set $\sigma_{1..r}(\boldsymbol{W}\boldsymbol{x})$ denote the ordering of the top-r alternatives in the ranking $\sigma(\boldsymbol{W}\boldsymbol{x})$. Moreover, recall that $\sigma_\ell(\boldsymbol{W}\boldsymbol{x})$ denotes the alternative in the ℓ-th position of the ranking $\sigma(\boldsymbol{W}\boldsymbol{x})$. For two matrices $\boldsymbol{U}, \boldsymbol{V} \in \mathbb{R}^{k\times d}$, we have that

$$\mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{N}_d}[\sigma_{1..r}(\boldsymbol{U}\boldsymbol{x}) \neq \sigma_{1..r}(\boldsymbol{V}\boldsymbol{x})] = \sum_{j=1}^{k}\mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{N}_d}\left[\bigcup_{\ell=1}^{r}\{j=\sigma_\ell(\boldsymbol{U}\boldsymbol{x}), j\neq\sigma_\ell(\boldsymbol{V}\boldsymbol{x})\}\right].$$

The first step is to understand the geometry of the set $\bigcup_{\ell=1}^{r}\{\boldsymbol{x} : j = \sigma_\ell(\boldsymbol{U}\boldsymbol{x})\} = \{\boldsymbol{x} : j \in \sigma_{1..r}(\boldsymbol{U}\boldsymbol{x})\}$ for $j \in [k]$. We have that this set is equal to

$$\mathcal{T}_{\boldsymbol{U}}^{(j)} = \bigcup_{S\subseteq[k]:|S|\leq r-1}\bigcap_{i\in S}\{\boldsymbol{x} : (\boldsymbol{U}_i-\boldsymbol{U}_j)\cdot\boldsymbol{x}\geq 0\}\cap\bigcap_{i\notin S}\{\boldsymbol{x} : (\boldsymbol{U}_i-\boldsymbol{U}_j)\cdot\boldsymbol{x}\leq 0\}\,.$$

In words, $\mathcal{T}_{\boldsymbol{U}}^{(j)}$ iterates over any possible collection of alternatives that can win the element j (they lie in the set of top elements S) and the remaining elements lose when compared with j (they lie in the complement set $[k]\setminus S$). Overloading the notation, let us define the mapping $T(\boldsymbol{t}) = T(t_1,...,t_k) = \sum_{S\subseteq[k]:|S|\leq r-1}\prod_{i\in S}\mathbf{1}\{t_i \geq 0\}\prod_{i\notin S}\mathbf{1}\{t_i \leq 0\}$. Using this mapping, we can define the indicator of the set $T_{\boldsymbol{U}}^{(j)}$ as $T((\boldsymbol{U}_1-\boldsymbol{U}_j)\cdot\boldsymbol{x},\ldots,(\boldsymbol{U}_k-\boldsymbol{U}_j)\cdot\boldsymbol{x})$. The top-$r$ disagreement $\mathbf{Pr}_{\boldsymbol{x}\sim\mathcal{N}_d}[j\in\sigma_{1..r}(\boldsymbol{U}\boldsymbol{x}), j\notin\sigma_{1..r}(\boldsymbol{V}\boldsymbol{x})]$ is equal to:

$$\mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{N}_d}[T((\boldsymbol{U}_1-\boldsymbol{U}_j)\cdot\boldsymbol{x},...,(\boldsymbol{U}_k-\boldsymbol{U}_j)\cdot\boldsymbol{x})=1, T((\boldsymbol{V}_1-\boldsymbol{V}_j)\cdot\boldsymbol{x},...,(\boldsymbol{V}_k-\boldsymbol{V}_j)\cdot\boldsymbol{x})=0]\,.$$

So we have that

$$\mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{N}_d}[\sigma_{1..r}(\boldsymbol{U}\boldsymbol{x}) \neq \sigma_{1..r}(\boldsymbol{V}\boldsymbol{x})] = \sum_{j=1}^{k}\mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{N}_d}[T_j(\boldsymbol{U}\boldsymbol{x})=1, T_j(\boldsymbol{V}\boldsymbol{x})=0] \leq \sum_{j=1}^{k}\mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{N}_d}[T_j(\boldsymbol{U}\boldsymbol{x})\neq T_j(\boldsymbol{V}\boldsymbol{x})]\,.$$

In order to show the desired bound, it suffices to prove the following two lemmas.

Lemma 16 (Disagreement Region)**.** *Consider a positive integer $r \leq k$. Fix $j \in [k]$ and let $\theta = \max_{i\in[k]}\theta(\boldsymbol{U}_i-\boldsymbol{U}_j,\boldsymbol{V}_i-\boldsymbol{V}_j)$. Then it holds that*

$$\lim_{\theta\to 0}\frac{\mathbf{Pr}_{\boldsymbol{x}\sim\mathcal{N}_d}[T_j(\boldsymbol{U}\boldsymbol{x})\neq T_j(\boldsymbol{V}\boldsymbol{x})]}{\theta} \leq c\cdot\sum_{i\in[k]}\Gamma(F_i^j)\sqrt{\log\left(\frac{1}{\Gamma(F_i^j)}+1\right)}\,,$$

where $c > 0$ is some constant and F_i^j is the surface $\{\boldsymbol{x} : j\in\sigma_{1..r}(\boldsymbol{V}\boldsymbol{x})\}\cap\{\boldsymbol{x} : \boldsymbol{V}_i\cdot\boldsymbol{x} = \boldsymbol{V}_j\cdot\boldsymbol{x}\}$ for the matrix $\boldsymbol{V}\in\mathbb{R}^{k\times d}$.

and,

Lemma 17. *Let F_i^j, r, k as in the previous lemma. It holds that*

$$\sum_{i\in[k]}\sum_{j\in[k]}\Gamma(F_i^j)\leq 2kr\,.$$

Applying these two lemmas with $\theta = \max_{i\neq j}\theta(\boldsymbol{U}_i-\boldsymbol{U}_j,\boldsymbol{V}_i-\boldsymbol{V}_j)$, we get that

$$Z := \lim_{\theta\to 0}\frac{\sum_{j\in[k]}\mathbf{Pr}_{\boldsymbol{x}\sim\mathcal{N}_d}[T_j(\boldsymbol{U}\boldsymbol{x})\neq T_j(\boldsymbol{V}\boldsymbol{x})]}{\theta} \leq c\cdot\sum_{j\in[k]}\sum_{i\in[k]}\Gamma(F_i^j)\sqrt{\log\left(\frac{1}{\Gamma(F_i^j)}+1\right)}\,.$$

Let us set $\Gamma'(F_i^j) = \Gamma(F_i^j)/(2kr)$. Then we have that

$$Z \leq 2ckr \cdot \sum_{j\in[k]} \sum_{i\in[k]} \Gamma'(F_i^j) \sqrt{\log\left(\frac{1}{2kr \cdot \Gamma'(F_i^j)} + 1\right)}.$$

It suffices to bound the quantity

$$\sum_{j\in[k]} \sum_{i\in[k]} \Gamma'(F_i^j) \sqrt{\log\left(\frac{1}{\Gamma'(F_i^j)} + 1\right)} = O\left(kr\sqrt{\log(kr)}\right),$$

where we used a similar "entropy-like" inequality as we did in the top-1 case. This yields (by recalling that it is sufficient to consider only the case of arbitrarily small angles, as in the top-1 case) that

$$\Pr_{\boldsymbol{x}\sim\mathcal{N}_d}[\sigma_{1..r}(\boldsymbol{U}\boldsymbol{x}) \neq \sigma_{1..r}(\boldsymbol{V}\boldsymbol{x})] \leq c\, rk\, \sqrt{\log(kr)} \cdot \max_{i\neq j} \theta(\boldsymbol{U}_i - \boldsymbol{U}_j, \boldsymbol{V}_i - \boldsymbol{V}_j),$$

for some universal constant c. □

C.1 The proof of Lemma 16

We proceed with the proof of the key lemma concerning the disagreement region. We first show the following claim where we only change a single vector. Recall that

$$T(\boldsymbol{V}\boldsymbol{x}) = \sum_{S:|S|\leq r-1} \prod_{i\in S} \mathbf{1}\{\boldsymbol{v}_i \cdot \boldsymbol{x} \geq 0\} \prod_{i\notin S} \mathbf{1}\{\boldsymbol{v}_i \cdot \boldsymbol{x} \leq 0\}.$$

We will be interested in the surface $F_1 := F_1(\boldsymbol{V}\boldsymbol{x}) = T(\boldsymbol{V}\boldsymbol{x})\mathbf{1}\{\boldsymbol{v}_1 \cdot \boldsymbol{x} = 0\}$.

Claim 11. *Let $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_k \in \mathbb{R}^d$ and $\boldsymbol{r} \in \mathbb{R}^d$ with $\theta(\boldsymbol{v}_1, \boldsymbol{r}) \leq \theta$ for some sufficiently small $\theta \in (0, \pi/2)$. It holds that*

$$\Pr_{\boldsymbol{x}\sim\mathcal{N}_d}[T(\boldsymbol{v}_1 \cdot \boldsymbol{x}, \ldots, \boldsymbol{v}_k \cdot \boldsymbol{x}) \neq T(\boldsymbol{r} \cdot \boldsymbol{x}, \boldsymbol{v}_2 \cdot \boldsymbol{x}, \ldots, \boldsymbol{v}_k \cdot \boldsymbol{x})] \leq c \cdot \theta \cdot \Gamma(F_1) \sqrt{\log\left(\frac{1}{\Gamma(F_1)} + 1\right)},$$

where F_1 is the surface $T(\boldsymbol{V}\boldsymbol{x}) \cap \{\boldsymbol{x} : \boldsymbol{v}_1 \cdot \boldsymbol{x} = 0\}$ and c is some universal constant.

Proof. We first decompose the sum of $T(\boldsymbol{V}\boldsymbol{x})$ depending on whether $1 \in S$ or not. Hence, we have that $T(\boldsymbol{v}_1 \cdot \boldsymbol{x}, \ldots, \boldsymbol{v}_k \cdot \boldsymbol{x}) = T^+(\boldsymbol{v}_1 \cdot \boldsymbol{x}, \ldots, \boldsymbol{v}_k \cdot \boldsymbol{x}) + T^-(\boldsymbol{v}_1 \cdot \boldsymbol{x}, \ldots, \boldsymbol{v}_k \cdot \boldsymbol{x})$ where

$$\begin{aligned}
T^+(\boldsymbol{v}_1 \cdot \boldsymbol{x}, \ldots, \boldsymbol{v}_k \cdot \boldsymbol{x}) &= \sum_{S\subseteq[k]:|S|\leq r-1, 1\in S} \prod_{i\in S} \mathbf{1}\{\boldsymbol{v}_i \cdot \boldsymbol{x} \geq 0\} \prod_{i\notin S} \mathbf{1}\{\boldsymbol{v}_i \cdot \boldsymbol{x} \leq 0\} \\
&= \sum_{S\subseteq[k]:|S|\leq r-1, 1\in S} \mathbf{1}\{\boldsymbol{v}_1 \cdot \boldsymbol{x} \geq 0\} \cdot \prod_{i\in S\setminus\{1\}} \mathbf{1}\{\boldsymbol{v}_i \cdot \boldsymbol{x} \geq 0\} \prod_{i\notin S} \mathbf{1}\{\boldsymbol{v}_i \cdot \boldsymbol{x} \leq 0\} \\
&= \mathbf{1}\{\boldsymbol{v}_1 \cdot \boldsymbol{x} \geq 0\} \cdot \sum_{S\subseteq[k]:|S|\leq r-1, 1\in S} \prod_{i\in S\setminus\{1\}} \mathbf{1}\{\boldsymbol{v}_i \cdot \boldsymbol{x} \geq 0\} \prod_{i\notin S} \mathbf{1}\{\boldsymbol{v}_i \cdot \boldsymbol{x} \leq 0\} \\
&=: \mathbf{1}\{\boldsymbol{v}_1 \cdot \boldsymbol{x} \geq 0\} \cdot G^+(\boldsymbol{v}_2 \cdot \boldsymbol{x}, \ldots, \boldsymbol{v}_k \cdot \boldsymbol{x}),
\end{aligned}$$

and similarly

$$\begin{aligned}
T^-(\boldsymbol{v}_1 \cdot \boldsymbol{x}, \ldots, \boldsymbol{v}_k \cdot \boldsymbol{x}) &= \mathbf{1}\{\boldsymbol{v}_1 \cdot \boldsymbol{x} \leq 0\} \cdot \sum_{S\subseteq[k]:|S|\leq r-1, 1\notin S} \prod_{i\in S} \mathbf{1}\{\boldsymbol{v}_i \cdot \boldsymbol{x} \geq 0\} \prod_{i\notin S\setminus\{1\}} \mathbf{1}\{\boldsymbol{v}_i \cdot \boldsymbol{x} \leq 0\} \\
&=: \mathbf{1}\{\boldsymbol{v}_1 \cdot \boldsymbol{x} \leq 0\} \cdot G^-(\boldsymbol{v}_2 \cdot \boldsymbol{x}, \ldots, \boldsymbol{v}_k \cdot \boldsymbol{x}).
\end{aligned}$$

Notice that the indicator G^s does not depend on the alternative 1 for $s \in \{-, +\}$. Since $T : \mathbb{R}^k \to \{0, 1\}$, we have that

$$\begin{aligned}
&\Pr_{\boldsymbol{x}\sim\mathcal{N}_d}[T(\boldsymbol{v}_1 \cdot \boldsymbol{x}, \ldots, \boldsymbol{v}_k \cdot \boldsymbol{x}) \neq T(\boldsymbol{r} \cdot \boldsymbol{x}, \boldsymbol{v}_2 \cdot \boldsymbol{x}, \ldots, \boldsymbol{v}_k \cdot \boldsymbol{x})] \\
&= \mathop{\mathbf{E}}_{\boldsymbol{x}\sim\mathcal{N}_d}[|T(\boldsymbol{v}_1 \cdot \boldsymbol{x}, \ldots, \boldsymbol{v}_k \cdot \boldsymbol{x}) - T(\boldsymbol{r} \cdot \boldsymbol{x}, \boldsymbol{v}_2 \cdot \boldsymbol{x}, \ldots, \boldsymbol{v}_k \cdot \boldsymbol{x})|] \\
&\leq \sum_{s\in\{-,+\}} \mathop{\mathbf{E}}_{\boldsymbol{x}\sim\mathcal{N}_d}[|T^s(\boldsymbol{v}_1 \cdot \boldsymbol{x}, \ldots, \boldsymbol{v}_k \cdot \boldsymbol{x}) - T^s(\boldsymbol{r} \cdot \boldsymbol{x}, \boldsymbol{v}_2 \cdot \boldsymbol{x}, \ldots, \boldsymbol{v}_k \cdot \boldsymbol{x})|] \\
&= \sum_{s\in\{-,+\}} \mathop{\mathbf{E}}_{\boldsymbol{x}\sim\mathcal{N}_d}[G^s(\boldsymbol{v}_2 \cdot \boldsymbol{x}, \ldots, \boldsymbol{v}_k \cdot \boldsymbol{x}) \cdot |\mathbf{1}\{s \cdot \boldsymbol{v}_1 \cdot \boldsymbol{x} \geq 0\} - \mathbf{1}\{s \cdot \boldsymbol{r} \cdot \boldsymbol{x} \geq 0\}|].
\end{aligned}$$

Let us focus on the case $s = +$. The difference between the two indicators in the last line of the above equation corresponds to the event that the halfspaces $\mathbf{1}\{\boldsymbol{v}_1 \cdot \boldsymbol{x} \geq 0\}$ and $\mathbf{1}\{\boldsymbol{r} \cdot \boldsymbol{x} \geq 0\}$ disagree. Hence, we have that $|\mathbf{1}\{\boldsymbol{v}_1 \cdot \boldsymbol{x} \geq 0\} - \mathbf{1}\{\boldsymbol{r} \cdot \boldsymbol{x} \geq 0\}| = \mathbf{1}\{(\boldsymbol{v}_1 \cdot \boldsymbol{x})(\boldsymbol{r} \cdot \boldsymbol{x}) < 0\}$. Note that the above indicator depends on both $\boldsymbol{v}_1$ and $\boldsymbol{r}$. We would like to work only with one of these two vectors. To this end, let us introduce $\boldsymbol{q}$, the normalized projection of $\boldsymbol{r}$ onto the orthogonal complement of $\boldsymbol{v}_1$, i.e., $\boldsymbol{q} = \mathrm{proj}_{\boldsymbol{v}_1^\perp} \boldsymbol{r} / \|\mathrm{proj}_{\boldsymbol{v}_1^\perp} \boldsymbol{r}\|_2$. We have that $\boldsymbol{v}_1$ and $\boldsymbol{q}$ is an orthonormal basis of the subspace spanned by the vectors $\boldsymbol{v}_1$ and $\boldsymbol{r}$. Notice that $\boldsymbol{r} = \cos\theta(\boldsymbol{v}_1, \boldsymbol{r})\boldsymbol{v}_1 + \sin\theta(\boldsymbol{v}_1, \boldsymbol{r})\boldsymbol{q}$, by the construction of $\boldsymbol{q}$. Our goal is to understand the structure of the region $(\boldsymbol{v}_1 \cdot \boldsymbol{x})(\boldsymbol{r} \cdot \boldsymbol{x}) < 0$. This set is equal to

$$\{0 < \boldsymbol{v}_1 \cdot \boldsymbol{x} < -(\boldsymbol{q} \cdot \boldsymbol{x}) \tan\theta(\boldsymbol{v}_1, \boldsymbol{r})\} \cup \{-(\boldsymbol{q} \cdot \boldsymbol{x}) \tan\theta(\boldsymbol{v}_1, \boldsymbol{r}) < \boldsymbol{v}_1 \cdot \boldsymbol{x} < 0\}\,.$$

To see this, we have that $(\boldsymbol{v}_1 \cdot \boldsymbol{x})(\boldsymbol{r} \cdot \boldsymbol{x}) = (\boldsymbol{v}_1 \cdot \boldsymbol{x})(\cos\theta(\boldsymbol{v}_1, \boldsymbol{r})\boldsymbol{v}_1 \cdot \boldsymbol{x} + \sin\theta(\boldsymbol{v}_1, \boldsymbol{r})\boldsymbol{q} \cdot \boldsymbol{x})$. This quantity must be negative. The left-hand set considers the case where $\boldsymbol{v}_1 \cdot \boldsymbol{x} > 0$ and so $\tan\theta(\boldsymbol{v}_1, \boldsymbol{r})(\boldsymbol{q} \cdot \boldsymbol{x}) < -\boldsymbol{v}_1 \cdot \boldsymbol{x}$. We obtain the right-hand set in a similar way. Thus, we have that the disagreement region $(\boldsymbol{v}_1 \cdot \boldsymbol{x})(\boldsymbol{r} \cdot \boldsymbol{x}) < 0$ is a subset of the region $\{|\boldsymbol{v}_1 \cdot \boldsymbol{x}| \leq |\boldsymbol{q} \cdot \boldsymbol{x}| \tan\theta(\boldsymbol{v}_1, \boldsymbol{r})\}$. Since $\tan\theta(\boldsymbol{v}_1, \boldsymbol{r}) \leq \theta$ and we have that θ is sufficiently small we can also replace the above region by the larger region: $\{|\boldsymbol{v}_1 \cdot \boldsymbol{x}| \leq 2\theta|\boldsymbol{q} \cdot \boldsymbol{x}|\}$. Therefore, we have

$$\begin{aligned}
&\mathop{\mathbf{E}}_{\boldsymbol{x}\sim\mathcal{N}_d}\left[G^+(\boldsymbol{v}_2 \cdot \boldsymbol{x}, \ldots, \boldsymbol{v}_k \cdot \boldsymbol{x})\, \mathbf{1}\{(\boldsymbol{v}_1 \cdot \boldsymbol{x})(\boldsymbol{r} \cdot \boldsymbol{x}) < 0\}\}\right] \\
&\qquad \leq \mathop{\mathbf{E}}_{\boldsymbol{x}\sim\mathcal{N}_d}\left[G^+(\boldsymbol{v}_2 \cdot \boldsymbol{x}, \ldots, \boldsymbol{v}_k \cdot \boldsymbol{x})\, \mathbf{1}\{|\boldsymbol{v}_1 \cdot \boldsymbol{x}| \leq 2\theta|\boldsymbol{q} \cdot \boldsymbol{x}|\}\right]\,.
\end{aligned}$$

From this point, the proof goes as in the top-1 case. In total, we will get that

$$\begin{aligned}
&\mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{N}_d}[T(\boldsymbol{v}_1 \cdot \boldsymbol{x}, \ldots, \boldsymbol{v}_k \cdot \boldsymbol{x}) \neq T(\boldsymbol{r} \cdot \boldsymbol{x}, \boldsymbol{v}_2 \cdot \boldsymbol{x}, \ldots, \boldsymbol{v}_k \cdot \boldsymbol{x})] \\
&= \mathop{\mathbf{E}}_{\boldsymbol{x}\sim\mathcal{N}_d}\left[(G^+(\boldsymbol{v}_2 \cdot \boldsymbol{x}, \ldots, \boldsymbol{v}_k \cdot \boldsymbol{x}) + G^-(\boldsymbol{v}_2 \cdot \boldsymbol{x}, \ldots, \boldsymbol{v}_k \cdot \boldsymbol{x}))\, |\boldsymbol{q} \cdot \boldsymbol{x}|\, \delta(|\boldsymbol{v}_1 \cdot \boldsymbol{x}|)\right] \\
&\leq 2\int_{\boldsymbol{x}\in F_1} \phi_d(\boldsymbol{x})|\boldsymbol{q} \cdot \boldsymbol{x}| d\mu(\boldsymbol{x}) \\
&\leq 2\int_{\boldsymbol{x}\in F_1} \phi_d(\boldsymbol{x})|\boldsymbol{q} \cdot \boldsymbol{x}| \mathbf{1}\{|\boldsymbol{q} \cdot \boldsymbol{x}| \leq \xi\} d\mu(\boldsymbol{x}) + 2\int_{\boldsymbol{x}\in F_1} \phi_d(\boldsymbol{x})|\boldsymbol{q} \cdot \boldsymbol{x}| \mathbf{1}\{|\boldsymbol{q} \cdot \boldsymbol{x}| \geq \xi\} d\mu(\boldsymbol{x}) \\
&\leq 2\xi\int_{\boldsymbol{x}\in F_1} \phi_d(\boldsymbol{x}) d\mu(\boldsymbol{x}) + 2\int_{\boldsymbol{x}\in F_1} \phi_d(\boldsymbol{x})|\boldsymbol{q} \cdot \boldsymbol{x}| \mathbf{1}\{|\boldsymbol{q} \cdot \boldsymbol{x}| \geq \xi\} d\mu(\boldsymbol{x})\,,
\end{aligned}$$

where $d\mu(\boldsymbol{x})$ is the standard surface measure in $\mathbb{R}^d$. Let us explain the first inequality above. Note that the space induced by $G^-(\boldsymbol{v}_2 \cdot \boldsymbol{x}, \ldots, \boldsymbol{v}_k \cdot \boldsymbol{x})$ contains the space induced by $G^+(\boldsymbol{v}_2 \cdot \boldsymbol{x}, \ldots, \boldsymbol{v}_k \cdot \boldsymbol{x})$. Hence, in the integration, we can integrate over the surface $F_1 = T(\boldsymbol{V}\boldsymbol{x}) \cap \mathbf{1}\{\boldsymbol{x} : \boldsymbol{v}_1 \cdot \boldsymbol{x} = 0\}$ twice. Essentially, this surface corresponds to $\mathbf{1}\{\boldsymbol{v}_1 \cdot \boldsymbol{x} = 0\} \cdot \sum_{S\subseteq[k]\setminus\{1\}:|S|\leq r-1} \prod_{i\in S} \mathbf{1}\{\boldsymbol{v}_i \cdot \boldsymbol{x} \geq 0\} \prod_{i\notin S} \mathbf{1}\{\boldsymbol{v}_i \cdot \boldsymbol{x} \leq 0\}$. Applying the steps of the top-1 case, we can obtain the desired bound in terms of the Gaussian surface area of F_1. □

Next, for fixed $j \in [k]$, we can apply the above claim sequentially (as we did in the end of the top-1 case) to get

$$\lim_{\theta\to 0} \frac{\mathbf{Pr}_{\boldsymbol{x}\sim\mathcal{N}_d}[T_j(\boldsymbol{U}\boldsymbol{x}) \neq T_j(\boldsymbol{V}\boldsymbol{x})]}{\theta} \leq c \cdot \sum_{i\in[k]} \Gamma(F_i^j)\sqrt{\log\left(\frac{1}{\Gamma(F_i^j)} + 1\right)}\,,$$

for some small constant $c > 0$.

C.2 The proof of Lemma 17

Using the above result, we get that it suffices to control the value $\Gamma(F_i^j)$, where F_i^j is the surface of $T_j(\boldsymbol{V}\boldsymbol{x}) \cap \{\boldsymbol{x} : \boldsymbol{V}_i \cdot \boldsymbol{x} = \boldsymbol{V}_j \cdot \boldsymbol{x}\}$ for the matrix $\boldsymbol{V}$ and $i, j \in [k]$. We next have to control the Gaussian surface area of the induced shape, i.e., the quantity

$$\Gamma(\{\boldsymbol{x} : j \in \sigma_{1..r}(\boldsymbol{V}\boldsymbol{x})\} \cap \{\boldsymbol{x} : \boldsymbol{V}_i \cdot \boldsymbol{x} = \boldsymbol{V}_j \cdot \boldsymbol{x}\})\,.$$

To this end, we give the next lemma.

Lemma 18. *Let $r \le k$ with $r, k \in \mathbb{N}$. For any matrix $\boldsymbol{V} \in \mathbb{R}^{k\times d}$ and $i, j \in [k]$, there exists a matrix $\boldsymbol{Q} = \boldsymbol{Q}^{(i)} \in \mathbb{R}^{k\times d}$ which depends only on i such that*

$$\Gamma(F_i^j) := \Gamma(\{\boldsymbol{x} : j \in \sigma_{1..r}(\boldsymbol{V}\boldsymbol{x})\} \cap \{\boldsymbol{x} : \boldsymbol{V}_i \cdot \boldsymbol{x} = \boldsymbol{V}_j \cdot \boldsymbol{x}\}) \le 2 \cdot \mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{N}_d}[j \in \sigma_{1..r}(\boldsymbol{Q}\boldsymbol{x})] \,.$$

Before proving this result, let us see how to apply it in order to get Lemma 17. We will have that

$$\begin{aligned}
\sum_{i\in[k]}\sum_{j\in[k]} \Gamma(F_i^j) &= \sum_{i\in[k]}\sum_{j\in[k]} \Gamma(\{\boldsymbol{x} : j \in \sigma_{1..r}(\boldsymbol{V}\boldsymbol{x})\} \cap \{\boldsymbol{x} : \boldsymbol{V}_i \cdot \boldsymbol{x} = \boldsymbol{V}_j \cdot \boldsymbol{x}\}) \\
&\le 2 \sum_{i\in[k]}\sum_{j\in[k]} \mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{N}_d}[j \in \sigma_{1..r}(\boldsymbol{Q}^{(i)}\boldsymbol{x})] \\
&= 2 \sum_{i\in[k]} \mathop{\mathbf{E}}_{\boldsymbol{x}\sim\mathcal{N}_d}[|\sigma_{1..r}(\boldsymbol{Q}^{(i)}\boldsymbol{x})|] \\
&= 2 \sum_{i\in[k]} r \\
&= 2kr \,.
\end{aligned}$$

Proof of Lemma 18. For this proof, we fix $i, j \in [k]$. The first step is to design the matrix $\boldsymbol{Q}$. As a first observation, we can subtract the vector $\boldsymbol{V}_i$ from each weight vector and do not affect the resulting orderings. Second, we can assume that the weight vectors that correspond to indices which j beats are unit. Let us be more specific Assume that initially we have that

$$(\boldsymbol{V}_j - \boldsymbol{V}_\ell) \cdot \boldsymbol{x} \ge 0 \,.$$

The first observation gives that

$$(\boldsymbol{V}_j - \boldsymbol{V}_i) \cdot \boldsymbol{x} \ge (\boldsymbol{V}_\ell - \boldsymbol{V}_i) \cdot \boldsymbol{x} \,.$$

Let us set $\widetilde{\boldsymbol{Q}}$ the intermediate matrix with rows $\boldsymbol{V}_j - \boldsymbol{V}_i$. The second observation states that the inequalities where j beats some index ℓ are not affected by normalization. Note that $\widetilde{\boldsymbol{Q}}_j \cdot \boldsymbol{x} = 0$ and hence $\widetilde{\boldsymbol{Q}}_\ell \cdot \boldsymbol{x} \le 0$. Hence, dividing with non-negative numbers will not affect the order of these two values, i.e.,

$$\frac{\widetilde{\boldsymbol{Q}}_j \cdot \boldsymbol{x}}{\|\widetilde{\boldsymbol{Q}}_j\|_2} \ge \frac{\widetilde{\boldsymbol{Q}}_\ell \cdot \boldsymbol{x}}{\|\widetilde{\boldsymbol{Q}}_\ell\|_2} \,.$$

Note that the above ordering is $\boldsymbol{x}$-dependent, since the indices that j beats depend on $\boldsymbol{x}$. However, we can normalize any row of $\widetilde{\boldsymbol{Q}}$ without affecting the fact that the element j is top-r (since the sign of the inner products is not affected by normalization). This transformation yields a matrix $\boldsymbol{Q} = \boldsymbol{Q}^{(i)}$ and depends only on i (crucially, it is independent of j). For simplicity, we will omit the index i in what follows. For this matrix, we have that

$$\{\boldsymbol{x} : j \in \sigma_{1..r}(\boldsymbol{Q}\boldsymbol{x}), \boldsymbol{Q}_j \cdot \boldsymbol{x} = 0\} = \{\boldsymbol{x} : j \in \sigma_{1..r}(\boldsymbol{V}\boldsymbol{x}), \boldsymbol{V}_i \cdot \boldsymbol{x} = \boldsymbol{V}_j \cdot \boldsymbol{x}\} \,.$$

We will now prove that

$$\mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{N}_d}[j \in \sigma_{1..r}(\boldsymbol{Q}\boldsymbol{x})] \ge \frac{\Gamma(F_i^j)}{2} \,.$$

Let us fix some $\boldsymbol{x}$ and set $\boldsymbol{x}^{\parallel} = \mathrm{proj}_{\boldsymbol{Q}_j}\boldsymbol{x}$ and $x^{\perp} = \mathrm{proj}_{\boldsymbol{Q}_j^{\perp}}\boldsymbol{x}$. We assume that $\boldsymbol{x}$ lies in the set $\{\boldsymbol{x} : j \in \sigma_{1..r}(\boldsymbol{Q}\boldsymbol{x})\}$. This implies that there exist an index set I of size at least $k - r$ so that if $\ell \in I$ then

$$\boldsymbol{Q}_j \cdot \boldsymbol{x}^{\parallel} + \boldsymbol{Q}_j \cdot \boldsymbol{x}^{\perp} \ge \boldsymbol{Q}_\ell \cdot \boldsymbol{x}^{\parallel} + \boldsymbol{Q}_\ell \cdot \boldsymbol{x}^{\perp} \,.$$

Let us condition on the event

$$\boldsymbol{Q}_j \cdot \boldsymbol{x}^{\perp} \ge \boldsymbol{Q}_\ell \cdot \boldsymbol{x}^{\perp} \,.$$

We hence get that

$$\boldsymbol{Q}_j \cdot \boldsymbol{x}^{\parallel} = (\boldsymbol{Q}_j \cdot \boldsymbol{Q}_j) \cdot (\boldsymbol{Q}_j \cdot \boldsymbol{x}) \ge \boldsymbol{Q}_\ell \cdot \boldsymbol{x}^{\parallel} = (\boldsymbol{Q}_\ell \cdot \boldsymbol{Q}_j) \cdot (\boldsymbol{Q}_j \cdot \boldsymbol{x})$$

Using that $\boldsymbol{Q}_j$ is unit, that the inner product between $\boldsymbol{Q}_\ell$ and $\boldsymbol{Q}_j$ is at most one and that $\boldsymbol{Q}_j \cdot \boldsymbol{x}$ is a univariate Gaussian, we get that

$$\mathop{\mathbf{Pr}}_{z\sim\mathcal{N}(0,1)}[z \cdot (1 - \boldsymbol{Q}_\ell \cdot \boldsymbol{Q}_j) \geq 0] = 1/2\,.$$

The above discussion implies that

$$\mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{N}_d}[j \in \sigma_{1..r}(\boldsymbol{Q}\boldsymbol{x})] = \mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{N}_d}[(\forall \ell \in I)\; \boldsymbol{Q}_j \cdot \boldsymbol{x}^{\parallel} + \boldsymbol{Q}_j \cdot \boldsymbol{x}^{\perp} \geq \boldsymbol{Q}_\ell \cdot \boldsymbol{x}^{\parallel} + \boldsymbol{Q}_\ell \cdot \boldsymbol{x}^{\perp}]$$

and so $\mathbf{Pr}_{\boldsymbol{x}\sim\mathcal{N}_d}[j \in \sigma_{1..r}(\boldsymbol{Q}\boldsymbol{x})]$ equals to

$$\mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{N}_d}[(\forall \ell \in I)\; \boldsymbol{Q}_j \cdot \boldsymbol{x}^{\parallel} \geq \boldsymbol{Q}_j \cdot \boldsymbol{x}^{\parallel} \mid (\forall \ell \in I)\; \boldsymbol{Q}_j \cdot \boldsymbol{x}^{\perp} \geq \boldsymbol{Q}_\ell \cdot \boldsymbol{x}^{\perp}] \cdot \mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{N}_d}[(\forall \ell \in I)\; \boldsymbol{Q}_j \cdot \boldsymbol{x}^{\perp} \geq \boldsymbol{Q}_\ell \cdot \boldsymbol{x}^{\perp}]\,.$$

However, in the above product, we have that the first term is $1/2$ and the second term is the probability that $j \in \sigma_{1..r}(\boldsymbol{Q}\boldsymbol{x}^{\perp})$, i.e.,

$$\mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{N}_d}[j \in \sigma_{1..r}(\boldsymbol{Q}\boldsymbol{x})] \geq \frac{\mathbf{Pr}[j \in \sigma_{1..r}(\boldsymbol{Q}\boldsymbol{x}^{\perp})]}{2} = \Gamma(F_i^j)/2\,,$$

since the space in the RHS is low-dimensional and corresponds to the desired surface. □

D Distribution-Free Lower Bounds for Top-1 Disagreement Error

We begin with some definitions concerning the PAC Label Ranking setting. Let $\mathcal{X}$ be an instance space and $\mathcal{Y} = \mathbb{S}_k$ be the space of labels, which are rankings over k elements. A sorting function or hypothesis is a mapping $h : \mathcal{X} \to \mathbb{S}_k$. We denote by $h_1(x)$ the top-1 element of the ranking $h(x)$. A hypothesis class is a set of classifiers $\mathcal{H} \subset \mathbb{S}_k^{\mathcal{X}}$.

Top-1 Disagreement Error. The top-1 disagreement error with respect to a joint distribution $\mathcal{D}$ over $\mathcal{X} \times \mathbb{S}_k$ equals to the probability $\mathbf{Pr}_{(x,\sigma)\sim\mathcal{D}}[h_1(x) \neq \sigma^{-1}(1)]$. We mainly consider learning in the **realizable** case, which means that there is $h^\star \in \mathcal{H}$ which has (almost surely) zero error. Therefore, we can focus on the marginal distribution $\mathcal{D}_x$ over $\mathcal{X}$ and denote the top-1 disagreement error of a sorting function h with respect to the true hypothesis $h^\star$ by $\mathrm{Err}_{\mathcal{D}_x,h^\star}(h) := \mathbf{Pr}_{x\sim\mathcal{D}_x}[h_1(x) \neq h_1^\star(x)]$.

A learning algorithm is a function $\mathcal{A}$ that receives a training set of m instances, $S \in \mathcal{X}^m$, together with their labels according to $h^\star$. We denote the restriction of $h^\star$ to the instances in S by $h^\star|_S$. The output of the algorithm $\mathcal{A}$, denoted $\mathcal{A}(S, h^\star|_S)$ is a sorting function. A learning algorithm is proper if it always outputs a hypothesis from $\mathcal{H}$.

The top-1 PAC Label Ranking sample complexity of a learning algorithm $\mathcal{A}$ is the function $m^{(1)}_{\mathcal{A},\mathcal{H}}$ defined as follows: for every $\epsilon, \delta > 0$, $m^{(1)}_{\mathcal{A},\mathcal{H}}(\epsilon, \delta)$ is the minimal integer such that for every $m \geq m^{(1)}_{\mathcal{A},\mathcal{H}}(\epsilon, \delta)$, every distribution $\mathcal{D}_x$ on $\mathcal{X}$, and every target hypothesis $h^\star \in \mathcal{H}$, $\mathbf{Pr}_{S\sim\mathcal{D}_x^m}[\mathrm{Err}_{\mathcal{D}_x,h^\star}(\mathcal{A}(S, h^\star|_S)) > \epsilon] \leq \delta$. In this case, we say that the learning algorithm (ϵ, δ)-learns the class of sorting functions $\mathcal{H}$ with respect to the top-1 disagreement error. If no integer satisfies the inequality above, define $m^{(1)}_{\mathcal{A}}(\epsilon, \delta) = \infty$. $\mathcal{H}$ is learnable with $\mathcal{A}$ if for all ϵ and δ the sample complexity is finite. The **top-1 PAC Label Ranking sample complexity** of a class $\mathcal{H}$ is $m^{(1)}_{\mathrm{PAC},\mathcal{H}}(\epsilon, \delta) = \inf_{\mathcal{A}} m^{(1)}_{\mathcal{A},\mathcal{H}}(\epsilon, \delta)$, where the infimum is taken over all learning algorithms. Clearly, the above top-1 definition can be extended to the top-r setting.

In this section, we show the next result. We denote by $\mathcal{L}_{d,k}$ the class of Linear Sorting functions in d dimensions with k labels.

Theorem 7. *In the realizable PAC Label Ranking setting, any algorithm that (ϵ, δ)-learns the class $\mathcal{L}_{d,k}$ with respect to the top-1 disagreement error requires at least $\Omega((dk + \log(1/\delta))/\epsilon)$ samples.*

D.1 Top-1 Ranking Natarajan Dimension

In order to establish the above result, we introduce a variant of the standard Natarajan dimension [Nat89, BDCBL92, DSBDSS11, DSS14]. For a ranking π, we will also let $L_1(\pi)$ its top-1 element and $L_{3..k}(\pi)$ the ranking after deleting its top-2 part.

Definition 3 (Top-1 Ranking Natarajan Dimension)**.** *Let $\mathcal{H} \subseteq \mathbb{S}_k^{\mathcal{X}}$ be a hypothesis class of sorting functions and let $S \subseteq \mathcal{X}$. We say that $\mathcal{H}$ N-shatters S if there exist two mappings $f_1, f_2 : S \to \mathbb{S}_k$ such that for every $y \in S$, $L_1(f_1(y)) \neq L_1(f_2(y))$ and $L_{3..k}(f_1(y)) = L_{3..k}(f_2(y))$ and for every $T \subseteq S$, there exists a sorting function $g \in \mathcal{H}$ such that*

$$(i)\ \forall x \in T,\ \ g(x) = f_1(x),\ \text{and}\ (ii)\ \forall x \in S \setminus T,\ \ g(x) = f_2(x)\,.$$

*The **top-1 Ranking Natarajan dimension** of $\mathcal{H}$, denoted $d_N^{(1)}(\mathcal{H})$ is the maximal cardinality of a set that is N-shattered by $\mathcal{H}$.*

First, we connect PAC Label Ranking learnability to the top-1 disagreement error with the notion of top-1 Ranking Natarajan dimension.

Theorem 8 (Top-1-Natarajan Lower Bounds Sample Complexity)**.** *In the realizable PAC Label Ranking setting, we have for every hypothesis class $\mathcal{H} \subseteq \mathbb{S}_k^{\mathcal{X}}$*

$$m^{(1)}_{\mathrm{PAC},\mathcal{H}}(\epsilon, \delta) = \Omega\left(\frac{d_N^{(1)}(\mathcal{H}) + \ln(1/\delta)}{\epsilon}\right)\,.$$

Proof. Let $\mathcal{H} \subseteq \mathbb{S}_k^{\mathcal{X}}$ be a class of sorting functions of top-1-Natarajan dimension $d_N^{(1)} = d_N$. Consider the binary hypothesis class $\mathcal{H}_{\mathrm{bin}} = \{0,1\}^{[d_N]}$ which contains all the classifiers from $[d_N] = \{1, ..., d_N\}$ to $\{0,1\}$. It suffices to show the following.

Claim 12. *It holds that $m^{(1)}_{\mathrm{PAC},\mathcal{H}}(\epsilon, \delta) \geq m_{\mathrm{PAC},\mathcal{H}_{\mathrm{bin}}}(\epsilon, \delta)$.*

This is sufficient since we have that $m_{\mathrm{PAC},\mathcal{H}_{\mathrm{bin}}}(\epsilon, \delta) = \Omega\left(\frac{\mathrm{VC}(\mathcal{H}_{\mathrm{bin}}) + \ln(1/\delta)}{\epsilon}\right)$ and $\mathrm{VC}(\mathcal{H}_{\mathrm{bin}}) = d_N$. Let us now prove the claim.

We assume that the instance space is the set $\mathcal{X}$. Assume that A is a learning algorithm for the hypothesis class $\mathcal{H} \subseteq \mathbb{S}_k^{\mathcal{X}}$ and A_{bin} is a learning algorithm for the associated binary class $\mathcal{H}_{\mathrm{bin}}$. It suffices to show that A requires at least as many samples as A_{bin}. In fact, we will show that whenever A_{bin} errs, so does A. Let $S = \{s_1, ..., s_{d_N}\}$, f_0, f_1 be the set and the two functions that witness that the top-1-Natarajan dimension of $\mathcal{H}$ is d_N. Given a training set $(x_i, y_i)_{i\in[m]} \in ([d_N] \times \{0,1\})^m$, we set $g : \mathcal{X} \to \mathbb{S}_k$ be equal to the output of the algorithm A with input $(s_{x_i}, f_{y_i}(x_i))_{i\in[m]} \in (S \times \mathbb{S}_k)^m$. We also set f be the output of the algorithm A_{bin} with input $(x_i, y_i)_{i\in[m]}$ by setting $f(i) = 1$ if and only if $L_1(g(s_i)) = L_1(f_1(s_i))$. We will show that whenever A_{bin} errs, so does A. Fix $(x_i, y_i) \in S \times \{0,1\}$. Assume that $A_{\mathrm{bin}}(x_i) \neq y_i$ and say $y_i = 0$. Then $f(i) = 1$ and so $L_1(g(s_i)) = L_1(f_1(s_i)) \neq L_1(f_0(s_i))$. This implies that A errs. The case $y_i = 1$ is similar. □

D.2 Lower Bound for top-1 disagreement error for LSFs

Theorem 9 (Top-1 Natarajan Dimension of LSFs)**.** *Consider the hypothesis class $\mathcal{L}_{d,k} = \{\sigma_{\boldsymbol{W}} : \mathbb{R}^d \to \mathbb{S}_k : \sigma_{\boldsymbol{W}}(\boldsymbol{x}) = \mathrm{argsort}(\boldsymbol{W}\boldsymbol{x}), \boldsymbol{W} \in \mathbb{R}^{k\times d}\}$. Then, $d_N^{(1)}(\mathcal{L}_{d,k}) = \Omega(dk)$.*

Proof. Fix $k \in \mathbb{N}$. Let us consider the case $d = 2$ that will correspond as the building block for the general case $d > 2$. Let us first choose the set of points: Set P be the collection of pairs $P = \{(2i-1, 2i)\}_{i\in[b]}$ for any $i \in [b]$ with $b = \lfloor k/2 \rfloor$ and $S = \{\boldsymbol{x}_m\}_{m\in P}$ where these points correspond to $|P|$ equidistributed points on the unit sphere in $\mathbb{R}^2$. This set of points has size $|P| = \Theta(k)$ and we are going to N-shatter it using $\mathcal{L}_{2,k}$.

Consider the matrix $\boldsymbol{W} \in \mathbb{R}^{k\times 2}$ so that $\{\boldsymbol{W}_i\}_{i\in[k]}$ correspond to the rows of $\boldsymbol{W}$. The structure of the problem relies on the hyperplanes with normal vectors $(\boldsymbol{W}_i - \boldsymbol{W}_j)_{i\neq j}$ and our choice of $\boldsymbol{W}$ will rely on these hyperplanes. For any $m = (2i-1, 2i)$, we set $\boldsymbol{W}_{2i-1}, \boldsymbol{W}_{2i}$ on the unit sphere so that $\boldsymbol{W}_{2i-1} \cdot \boldsymbol{W}_{2i} = 1 - \phi$ with $\phi \in (0,1)$ sufficiently small (set $\arccos(1-\phi) = 2\pi/(100k)$) and let C_m be the cone generated by these two vectors with axis I_m. We place $\boldsymbol{W}_{2i-1}$ so that the distance between $\boldsymbol{x}_m$ and the hyperplane I_m is sufficiently small (say that the angle between $\boldsymbol{x}_m$ and I_m is $\arccos(1-\phi)/100$). Note that the normal vector of I_m is $\boldsymbol{W}_{2i-1} - \boldsymbol{W}_{2i}$ and we place $\boldsymbol{x}_m$ so that it has positive correlation with this vector. This uniquely identifies the location of $\boldsymbol{W}_{2i}$. Crucially, each vector $\boldsymbol{x}_m$ has the following properties: (i) $\boldsymbol{x}_m$ is very close to the boundary of the hyperplane

with normal vector $(\boldsymbol{W}_{2i-1} - \boldsymbol{W}_{2i})$, (ii) $\boldsymbol{W}_{2i-1} \cdot \boldsymbol{x}_m > \boldsymbol{W}_{2i} \cdot \boldsymbol{x} > \boldsymbol{W}_j \cdot \boldsymbol{x}_m$ for any $j \notin m$ and (iii) $\boldsymbol{x}_m$ is far from any boundary induced by hyperplanes with normal vectors $\boldsymbol{W}_j - \boldsymbol{W}_{j'}$ for any $(j, j') \neq m$.

Since the points are well-separated on the unit sphere, for any $m = (2i-1, 2i) \in P$, we have $\boldsymbol{W}_{2i-1} \cdot \boldsymbol{W}_{2i} = 1 - \phi \approx 1$ and for any other pair of indices $(i, j) \notin P$, there exists $c = c(k) \in (0, 1)$, $|\langle \boldsymbol{W}_i, \boldsymbol{W}_j \rangle| \leq c$.

For any $m = (2i-1, 2i) \in P$, we set $\boldsymbol{W}'_{2i-1} - \boldsymbol{W}'_{2i} = \boldsymbol{R}_\theta(\boldsymbol{W}_{2i-1} - \boldsymbol{W}_{2i})$ for some θ to be chosen, where $\boldsymbol{R}_\theta$ is the 2×2 rotation matrix. We choose θ so that each point $\boldsymbol{x}_m$ for $m = (2i-1, 2i) \in P$ with $(\boldsymbol{W}_{2i-1} - \boldsymbol{W}_{2i}) \cdot \boldsymbol{x}_m > 0$ satisfies $(\boldsymbol{W}'_{2i-1} - \boldsymbol{W}'_{2i}) \cdot \boldsymbol{x}_m < 0$. The main idea is that since $\boldsymbol{x}_m$ has the properties (i)-(iii) described above, the rankings induced by the vectors $\boldsymbol{W}\boldsymbol{x}_m$ and $\boldsymbol{W}'\boldsymbol{x}_m$ will be different in the first two positions but the same in the rest.

Given the training set $\{\boldsymbol{x}_m\}_{m \in P}$, we have to construct f_0, f_1 and verify that they satisfy the top-1 Ranking Natarajan conditions. For $m = (2i-1, 2i)$, we have that $f_0(\boldsymbol{x}_m) = (2i-1, 2i, \pi)$ and $f_1(\boldsymbol{x}_m) = (2i, 2i-1, \pi)$ for some ranking π of size $k-2$ that depends on m. Specifically, we will set $f_0(\boldsymbol{x}) = \sigma(\boldsymbol{W}\boldsymbol{x})$ and $f_1(\boldsymbol{x}) = \sigma(\boldsymbol{W}'\boldsymbol{x})$, where σ gives the decreasing ordering of the elements of the input vector. By the choice of the set S and $\boldsymbol{W}, \boldsymbol{W}'$, it remains to show that the $k-2$ last elements of the rankings $f_0(\boldsymbol{x}_m)$ (say π_0) and of $f_1(\boldsymbol{x}_m)$ (say π_1) are in the same order, i.e., $L_{3..k}(f_0(\boldsymbol{x}_m)) = L_{3..k}(f_1(\boldsymbol{x}_m))$. Assume that $u \succ v$ in π_0. It suffices to show that $(\boldsymbol{W}'_u - \boldsymbol{W}'_v) \cdot \boldsymbol{x}_m \geq 0$, i.e., the order of u and v is preserved when transforming $\boldsymbol{W}$ to $\boldsymbol{W}'$. We have that $(\boldsymbol{W}_u - \boldsymbol{W}_v) \cdot \boldsymbol{x}_m > c_1$ for some constant $c_1 > 0$ (c_1 is the minimum over $(u, v) \neq m = (2i-1, 2i)$). Hence, we can pick θ small enough so that $(\boldsymbol{W}'_u - \boldsymbol{W}'_v) \cdot \boldsymbol{x}_m > c_2$ and this can be done for any pair u, v that does not correspond to m. This implies that $\pi_0 = \pi_1 = \pi$. In particular, we have that

$$(\boldsymbol{W}'_u - \boldsymbol{W}'_v) \cdot \boldsymbol{x}_m = \cos(\theta) \cdot (\boldsymbol{W}_u - \boldsymbol{W}_v) \cdot \boldsymbol{x}_m + \sin(\theta) \cdot (W^{(1)}_{uv} x^{(2)}_m - W^{(2)}_{uv} x^{(1)}_m) > c_2 > 0$$

for some θ sufficiently small, where $W^{(t)}_{uv}$ is the t-th entry of the vector $\boldsymbol{W}_u - \boldsymbol{W}_v$ for $t \in \{1, 2\}$ and $\boldsymbol{x}_m, \boldsymbol{W}_u, \boldsymbol{W}_v$ are unit vectors.

For any subset T of S, it remains to choose a linear classifier in $\mathcal{L}_{2,k}$ (which is allowed to depend on T). For any $T \subseteq S = \{\boldsymbol{x}_m\}_{m \in P}$, we consider the matrix $\overline{\boldsymbol{W}} \in \mathbb{R}^{k \times 2}$ so that for the i-th row $\overline{W}_i = \boldsymbol{W}_i 1\{i \in m \in T\} + \boldsymbol{W}'_i 1\{i \in m \in S \setminus T\}$ for any $i \in [k]$. This is valid since the pairs $m \in P$ partition $[k]$. We have to show the following two properties: (i) $\sigma(\overline{\boldsymbol{W}}\boldsymbol{x}) = f_0(\boldsymbol{x})$ for $x \in T$ and (ii) $\sigma(\overline{\boldsymbol{W}}\boldsymbol{x}) = f_1(\boldsymbol{x})$ for $x \in S \setminus T$.

Assume that $m = (2i-1, 2i)$ and $\boldsymbol{x}_m \in T$. We have that $f_0(\boldsymbol{x}_m) = (2i-1, 2i, \pi)$ and $\overline{\boldsymbol{W}}_{2i-1} - \overline{\boldsymbol{W}}_{2i} = \boldsymbol{W}_{2i-1} - \boldsymbol{W}_{2i}$ and so $2i-1 \succ 2i$ in the ranking $\sigma(\overline{\boldsymbol{W}}\boldsymbol{x}_m)$. It remains to show that the remaining $\binom{k}{2} - 1$ pairwise comparisons are the same in the two rankings. Let us consider a pair of points $u \neq v$ so that $u \succ v$ in $f_0(\boldsymbol{x}_m)$. It suffices to show that $u \succ v$ in $\sigma(\overline{\boldsymbol{W}}\boldsymbol{x}_m)$.

1. If u, v are so that $\overline{\boldsymbol{W}}_u - \overline{\boldsymbol{W}}_v = \boldsymbol{W}_u - \boldsymbol{W}_v$, the result holds.
2. If u, v are so that $\overline{\boldsymbol{W}}_u - \overline{\boldsymbol{W}}_v = \boldsymbol{W}_u - \boldsymbol{W}'_v$: In this case, u and v lie in a different pair of P and this implies that the correct direction is preserved if θ is appropriately chosen. For θ as above, it holds that $(\boldsymbol{W}_u - \boldsymbol{R}_\theta \boldsymbol{W}_v) \cdot \boldsymbol{x}_m$ has the same sign as $(\boldsymbol{W}_u - \boldsymbol{W}_v) \cdot \boldsymbol{x}_m$. In particular,

   $$\boldsymbol{W}_u \cdot \boldsymbol{x}_m - \boldsymbol{R}_\theta \boldsymbol{W}_v \cdot \boldsymbol{x}_m = \boldsymbol{W}_u \cdot \boldsymbol{x}_m - (\cos(\theta) W^{(1)}_v - \sin(\theta) W^{(2)}_v) x^{(1)}_m - (\sin(\theta) W^{(1)}_v + \cos(\theta) W^{(2)}_v) x^{(2)}_m \,,$$

   and so

   $$(\boldsymbol{W}_u - \boldsymbol{W}'_v) \cdot \boldsymbol{x}_m = \cos(\theta) \cdot (\boldsymbol{W}_u - \boldsymbol{W}_v) \cdot \boldsymbol{x}_m + \sin(\theta)(W^{(2)}_v x^{(1)}_m - W^{(1)}_v x^{(2)}_m) > 0 \,.$$

3. If u, v are so that $\overline{\boldsymbol{W}}_u - \overline{\boldsymbol{W}}_v = \boldsymbol{W}'_u - \boldsymbol{W}'_v$, the analysis for the inner product with $\boldsymbol{x}_m$ will be similar.

We now have to extend this proof for $d > 2$. We will "tensorize" the above construction as follows. Let $S = \{\boldsymbol{y}_{mj}\}_{m \in [b], j \in [d/2]}$ with $|S| = \lfloor k/2 \rfloor \cdot \lfloor d/2 \rfloor$. We first define the points of S: For $s \in [d]$,

set $y_{mj}[s] = x_m[1]1\{s = 2j-1\} + x_m[2]1\{s = 2j\}$ with $\boldsymbol{y}_{mj} \in \mathbb{R}^d$, i.e., $\boldsymbol{y}_{mj}$ has the values of $\boldsymbol{x}_m$ at the consecutive entries indicated by $m = (2i-1, 2i) \in P$ and zeros at the other positions.

We have to show that the set S is N-shattered. Given $T \subseteq S$, we are going to create the matrix $\overline{\boldsymbol{W}} \in \mathbb{R}^{k \times d}$. For illustration, think of each row of the matrix as having $d/2$ blocks of size two. If $\boldsymbol{y}_{mj} \in T$ with $m = (2i-1, 2i)$, set the two associated rows (indicated by m) of $\overline{\boldsymbol{W}}$ with $\boldsymbol{W}_{2i-1}, \boldsymbol{W}_{2i}$ at the j-th block and with $\boldsymbol{W}'_{2i-1}, \boldsymbol{W}'_{2i}$ otherwise. We will have that $\sigma(\overline{\boldsymbol{W}}\boldsymbol{y}) = f_0(\boldsymbol{y})$ if $y \in T$ and $\sigma(\overline{\boldsymbol{W}}\boldsymbol{y}) = f_1(\boldsymbol{y})$ otherwise and the analysis is the same as the $d = 2$ case. □

E Examples of Noisy Ranking Distributions

Definition 4 (Mallows model [Mal57])**.** *Consider k alternatives and let $\pi \in \mathbb{S}_k, \phi \in [0,1]$. The Mallows distribution $\mathcal{M}_{\mathrm{Mal}}(\pi, \phi)$ with central ranking π and spread parameter ϕ is a probability measure over $\mathbb{S}_k$ with density $\mathbf{Pr}_{\sigma \sim \mathcal{M}_{\mathrm{Mal}}(\pi,\phi)}[\sigma]$ that is proportional to $\phi^{d(\sigma,\pi)}$, where d is a ranking distance.*

We focus on Mallows models accociated with the Kendall's Tau distance $d = d_{KT}$ (the standard distance, not the normalized one), which measures the number of discordant pairs.

Fact 2. *When $\phi < 1$, the Mallows model $\mathcal{M}_{\mathrm{Mal}}(\pi, \phi)$ is a ranking distribution with bounded noise at most $\frac{1+\phi}{4} < 1/2$.*

Proof. The following property holds [Mal57]

$$\mathop{\mathbf{Pr}}_{\sigma \sim \mathcal{M}_{\mathrm{Mal}}(\pi,\phi)}[\sigma(i) < \sigma(j) | \pi(i) < \pi(j)] = \frac{\pi(j) - \pi(i) + 1}{1 - \phi^{\pi(j)-\pi(i)+1}} - \frac{\pi(j) - \pi(i)}{1 - \phi^{\pi(j)-\pi(i)}} \geq \frac{1}{2} + \frac{1-\phi}{4}\,.$$

□

The Bradley-Terry-Luce model [BT52, Luc12] is the most studied pairwise comparisons model. In his seminal paper, Mallows [Mal57] also studied the following natural ranking distribution:

Definition 5 (Bradley-Terry-Mallows [Mal57])**.** *Consider a score vector $\boldsymbol{w} \in \mathbb{R}_+^k$ with k distinct entries and let π be the ranking induced by the values of $\boldsymbol{w}$ in decreasing order. The Bradley-Terry-Mallows distribution $\mathcal{M}_{\mathrm{BTM}}(\boldsymbol{w})$ with central ranking π is a probability measure over $\mathbb{S}_k$ with density $\mathbf{Pr}_{\sigma \sim \mathcal{M}_{\mathrm{BTM}}(\boldsymbol{w})}[\sigma]$ that is proportional to $\prod_{i \succ_\sigma j} \frac{w_i}{w_i + w_j}$.*

Lemma 19. *There exists a real number $0 < \eta < 1/2$ so that the Bradley-Terry-Mallows distribution $\mathcal{M}_{\mathrm{BTM}}(\boldsymbol{w})$ is a ranking distribution with bounded noise at most η.*

Proof. In the standard Bradley-Terry-Luce model, the pairwise comparison between the alternatives i, j is a Bernoulli random variable with $\mathbf{Pr}[i \succ j] = w_i/(w_i + w_j)$. The Bradley-Terry-Mallows distribution can be considered as the Bradley-Terry-Luce model conditioned on the event that all the pairwise comparisons are consistent to a ranking. Hence, we have that

$$\mathop{\mathbf{Pr}}_{\sigma \sim \mathcal{M}_{\mathrm{BTM}}(\boldsymbol{w})}[\sigma] = \frac{1}{Z(k, \boldsymbol{w})} \prod_{i \succ_\sigma j} \frac{w_i}{w_i + w_j}\,.$$

Let us set $\mathcal{A}_{i \succ j} = \{\sigma \in \mathbb{S}_k : \sigma(i) < \sigma(j)\}$. We are interested in the following probability

$$\mathop{\mathbf{Pr}}_{\sigma \sim \mathcal{M}_{\mathrm{BTM}}(\boldsymbol{w})}[i \succ_\sigma j | w_i > w_j] = \mathop{\mathbf{Pr}}_{\sigma \sim \mathcal{M}_{\mathrm{BTM}}(\boldsymbol{w})}[\sigma(i) < \sigma(j) | w_i > w_j] = \frac{1}{Z(k, \boldsymbol{w})} \sum_{\sigma \in \mathcal{A}_{i \succ j}} \prod_{p \succ_\sigma q} \frac{w_p}{w_p + w_q}\,.$$

Note that in order to show the desired property, it suffices to show that

$$\sum_{\sigma \in \mathcal{A}_{i \succ j}} \prod_{p \succ_\sigma q} \frac{w_p}{w_p + w_q} > \sum_{\sigma \in \mathcal{A}_{i \prec j}} \prod_{p \succ_\sigma q} \frac{w_p}{w_p + w_q}\,.$$

First, observe that there exists a correspondence mapping $\sigma \in \mathcal{A}_{i \succ j}$ to $\mathcal{A}_{i \prec j}$, where one flips the elements i and j. Hence, it suffices to show that the mass of the ranking $(u_a)i(u_b)j(u_c)$ is larger than the one of the ranking $(u_a)j(u_b)i(u_c)$, where u_a, u_b, u_c are permutations of length between 0 and

$k-2$ with elements in $[k] \setminus \{i, j\}$. For the two above rankings, the only terms of the product that are not identical are the following

$$\frac{w_i}{w_i + w_j} \prod_{x \in u_b} \frac{w_i}{w_i + w_x} \frac{w_x}{w_x + w_j} > \frac{w_j}{w_i + w_j} \prod_{x \in u_b} \frac{w_j}{w_j + w_x} \frac{w_x}{w_x + w_i} ,$$

since $w_i > w_j$ and so the result follows. $\square$