Journal of Machine Learning Research 13 (2012) 137-164 Submitted 5/11; Revised 11/11; Published 1/12

An Active Learning Algorithm for Ranking from Pairwise Preferences with an Almost Optimal Query Complexity

Nir Ailon[*] NAILON@CS.TECHNION.AC.IL
Department of Computer Science
Taub Building
Technion Haifa 32000, Israel

Editor: Sanjoy Dasgupta

Abstract

Given a set V of n elements we wish to linearly order them given pairwise preference labels which may be non-transitive (due to irrationality or arbitrary noise).

The goal is to linearly order the elements while disagreeing with as few pairwise preference labels as possible. Our performance is measured by two parameters: The number of disagreements (loss) and the query complexity (number of pairwise preference labels). Our algorithm adaptively queries at most $O(\varepsilon^{-6} n \log^5 n)$ preference labels for a regret of ε times the optimal loss. As a function of n, this is asymptotically better than standard (non-adaptive) learning bounds achievable for the same problem.

Our main result takes us a step closer toward settling an open problem posed by learning-to-rank (from pairwise information) theoreticians and practitioners: What is a provably correct way to sample preference labels? To further show the power and practicality of our solution, we analyze a typical test case in which a large margin linear relaxation is used for efficiently solving the simpler learning problems in our decomposition.

Keywords: statistical learning theory, active learning, ranking, pairwise ranking, preferences

1. Introduction

We study the problem of learning to rank from pairwise preferences, and solve a long-standing open problem that has led to development of many heuristics but no provable results.

The setting is as follows: We are given a set V of n elements from some universe, and we wish to linearly order them given pairwise preference labels. given two elements $u, v \in V$, a pairwise preference label is obtained as a response, typically from a human, to the question *which if preferred, u or v?* We assume no abstention, hence, either u is preferred to v (denoted $u \prec v$) or the other way around.

The goal is to linearly order the elements from the most preferred to the least preferred, while disagreeing with as few pairwise preference labels as possible. Our performance is measured by two parameters: The loss (number of pairwise preference labels we disagree with) and the query complexity (number of pairwise preference labels we obtain). This is a typical learning problem, with the exception that the sample space is finite, consisting of $\binom{n}{2}$ possibilities only.

The loss minimization problem given the entire $n \times n$ preference matrix is a well known NP-hard problem called MFAST (minimum feedback arc-set in tournaments) (Alon, 2006). Recently,

∗. Supported by a Marie Curie International Reintegration Grant PIRG07-GA-2010-268403

Kenyon-Mathieu and Schudy (2007) have devised a PTAS for it, namely, a polynomial (in n) -time algorithm computing a solution with loss at most $(1+\varepsilon)$ the optimal, for and $\varepsilon > 0$ (the degree of the polynomial may depend on ε). In our case each edge from the input graph is given for a unit cost. Our main algorithm is derived from Kenyon et al's algorithm. Our output, however, is not a solution to MFAST, but rather a reduction of the original learning problem to a different, simpler one. The reduced problem can be solved using any general ERM (empirical risk minimization) black-box. The sampling of preference labels from the original problem is adaptive, hence the combination of our algorithm and any ERM blackbox is an active learning one. We give examples with an SVM based ERM black-box toward the end.

1.1 Our Setting vs. The Usual "Learning to Rank" Problem

Our setting defers from much of the *learning to rank* (LTR) literature. Usually, the labels used in LTR problems are responses to individual elements, and not to pairs of elements. A typical example is the $1..5$ scale rating for restaurants, or $0,1$ rating (irrelevant/relevant) for candidate documents retrieved for a query (known as the *binary ranking* problem). The goal there is, as in ours, to order the elements while disagreeing with as little pairwise relations as possible, where a pairwise relation is derived from any two elements rated differently. Note that the underlying preference graph there is transitive, hence no combinatorial problem due to nontransitivity. In fact, some view the rating setting as an ordinal regression problem and not a ranking problem. Here the preference graph may contain cycles, and is hence agnostic with respect to the concept class we are allowed to output from, namely, permutations. We note that some LTR literature does consider the pairwise preference label approach, and there is much justification to it (see Carterette et al. 2008; Hüllermeier et al. 2008 and reference therein). As far as we know, our work provides a sound solution to a problem addressed by machine learning practitioners (e.g., Carterette et al. 2008) who use pairwise preferences as labels for the task of learning to rank items, but wish to avoid obtaining labels for the quadratically many preference pairs, without compromising low error bounds. We also show that the *problem of quadraticity* found in much work dealing with pairwise preference based learning to rank (e.g., from Crammer and Singer 2001 *the [pairwise] approach is time consuming since it requires increasing the sample size ... to* $O(n^2)$) can be alleviated in the light of new advances in combinatorial optimization (Ailon et al., 2008a; Kenyon-Mathieu and Schudy, 2007).

1.2 Using Kenyon and Schudy's PTAS as a Starting Point

As mentioned above, our main algorithm is derived from the PTAS of Kenyon-Mathieu and Schudy (2007), but it is important to note a significant difference between our work and theirs. A good way to explain this is to compare two learners, Larry and Linda. On the first day, Larry queries all $\binom{n}{2}$ pairwise preference labels and sends them to a perfect solver for MFAST. Linda uses our work to query only $O(n\,\mathrm{poly}(\log n, \varepsilon^{-1}))$ preference labels and obtains a decomposition of the original input V into an *ordered list* of sub-problems $V_1, \ldots, V_k$ where each V_i is contained in V. Using the same perfect solver for the induced subproblems corresponding to each part and concatenating the individual output permutations, Linda will incur a loss of at most $(1+\varepsilon)$ that of Larry. If the decomposition is nontrivial, then Linda enjoys reduced query complexity for a small regret compared to Larry. The next day, both Larry and Linda realize that the perfect MFAST solver cannot deal with large inputs (the problem is NP Hard). They cannot use the PTAS of Kenyon-Mathieu and Schudy (2007) because they seek a multiplicative regret of $(1+\varepsilon)$ with respect to the

optimal solution (we also say a *relative regret* of ε), and the sought ε makes this infeasible.[1] To remedy this, Larry takes advantage of the fact that the set V does not merely consist of abstract elements, but rather each $u \in V$ is endowed with a feature vector $\varphi(u)$ and hence each pair of points u,v is endowed with the combined feature vector $(\varphi(u),\varphi(v))$. As in typical learning, he posits that the order relation between u,v can be deduced from a linear function of $(\varphi(u),\varphi(v))$, and invokes an optimizer (e.g., SVM) on the relaxed problem, with all pairs as input. Note that Larry may try to sample pairs uniformly to reduce the query complexity (and, perhaps, the running time of the relaxed solver), but as we show below, he will be discouraged from doing so because in certain realistic cases a relative regret of ε may entail sampling the entire pairwise preference space. Linda uses the same relaxed optimizer, say, SVM. The labels she sends to the solver consist of a uniform sample of pairs from each block V_i, together with all pairs u,v residing in separate blocks from her aforementioned construction decomposition. From the former label type she would need only $O(n\,\mathrm{poly}(\log n,\varepsilon^{-1}))$ many, because (per our decomposition design) within the blocks the cost of any solution is high, and hence a *relative* error is tantamount to an absolute error of similar magnitude, for which careful arguments allow low query complexity. From the latter label type, she would generate a label for all pairs u,v in distinct V_i,V_j, using a "made up" label corresponding to the order of V_i,V_j (recall that the decomposition is ordered).

As the above story suggests, we do not run the PTAS of Kenyon-Mathieu and Schudy (2007) verbatim, but use it only to obtain a certain decomposition of the input. Among other changes, a key change to their algorithm is required by replacing a highly sensitive greedy improvement step into a robust approximate one, by careful sampling. The main difficulty stems from the fact that after a single greedy improvement step, the sample becomes stale and requires refreshing. We show a query efficient refreshing technique that allows iterated approximate greedy improvement steps. Interestingly, the original analysis is amenable to this change. It is also interesting to note that the sampling scheme used for identifying greedy improvement steps for a current solution are similar to ideas used by Ailon et al. (2007, 2008b) and Halevy and Kushilevitz (2007) in the context of property testing and reconstruction, where elements are sampled from exponentially growing intervals in a linear order.

The 3-approximation algorithm for MFAST using QuickSort by Ailon et al. (2008a) is used in Kenyon-Mathieu and Schudy (2007) as well as here as an initialization step. Note that this is a sub-linear algorithm. In fact, it samples only $O(n\log n)$ pairs from the $\binom{n}{2}$ possible, on expectation. Note also that the pairs from which we query the preference relation in QuickSort are chosen adaptively.

1.3 Our Work in the Context of Machine Learning Reductions

Our main algorithm reduces a given instance to smaller subproblems decomposing it. We compare the machine learning reduction approach to two other works, that of Balcan et al. (2008) and that of Ailon and Mohri (2010). The former also considers a reduction of the problem of learning to rank, but in the *bipartite ranking* (see Section 1.1) setting where, in the first place, it is assumed that individual elements are endowed with unknown labels on a scale (of size two). The output is a permutation, and the loss function is the number of pairwise inversions. Their work shows that the problem of minimizing the regrets of the underlying binary classification and ranking problems is, up to a constant, the same thing. Their work, in fact, questions the justification for the so-called

1. The running time of the PTAS is exponential in ε^{-1}. We note here, for the sake of comparison, that our sampling scheme has complexity polynomial in ε^{-1}.

binary ranking problem. The latter (Ailon and Mohri, 2010) considers the same setting as here, and shows a query efficient algorithm that reduces the original instance, which may contain cycles, to a binary classification problem over an adaptively chosen set of $O(n\log n)$ pairs on expectation. The results there guarantee a total regret of at most twice that of the optimal.[2] Here we obtain at most $1+\varepsilon$ that of the optimal using $O(n\,\mathrm{poly}(\log n, \varepsilon^{-1}))$ pairwise queries.

1.4 Our Work in the Context of Active Learning

Active learning is an important field of statistical learning theory and practice (El-Yaniv and Wiener, 2010; Balcan et al., 2010; Hanneke, 2007; Dasgupta, 2005; Culotta and McCallum, 2005; Roth and Small, 2006; Dasgupta et al., 2007; Atlas et al., 1994; Freund et al., 1997; Lindenbaum et al., 2004; Begleiter et al., 2008; Balcan et al., 2009; Angluin, 2004; Dasgupta et al., 2009; Fine et al., 2002; Baram et al., 2004; Atlas et al., 1994; Friedman, 2009; Atlas et al., 1990; Yu et al., 2006). In the most general setting, one wishes to improve on standard query complexity bounds (using, for example, VC or Rademacher complexity) by actively choosing which instances to obtain labels for. Many heuristics have been developed, while algorithms with provable bounds (especially in the agnostic case) are known for few problems. Balcan et al. (2010) show that any learning algorithm for a finite VC dimensional space admits an active learning algorithm which asymptotically beats, in query complexity, that of a passive learning algorithm. Their guarantees are, however, unverifiable in the sense that the learner does not know when to stop querying in order to achieve a certain error. Also, their scheme still requires a considerable amount of work in order to be applicable for individual problems. It is an interesting open question to apply it to the problem at hand and compare the results with our algorithms' guarantees. Also, Balcan et al. (2009) proposed an active learning algorithm called A2. A useful measure of complexity which was later defined by Hanneke (2007) is key in analysis of A2. He defined a disagreement coefficient for a concept space and showed how this measure could be used for active learning in certain cases. We show in Appendix B why this measure does not help here.

1.5 Our Work in the Context of Noisy Sorting

There is much literature in theoretical computer science on sorting noisy data. For example, Braverman and Mossel (2008) present an algorithm with an $O(n\log n)$ query complexity for exact order reconstruction when the input is Bayesian with certain natural priors. Feige et al. (2002) consider a scenario in which the input preference graph is transitive, but queries may result in noisy comparisons which may be inconsistent with previous information (hence, querying the same pair multiple times would result in difference independent responses). Ajtai et al. (2009) consider a setting in which each element has a latent value, and comparisons of two elements with similar value may result in errors. In this work the input is not Bayesian, query responses are fixed and elements do not have a latent value.

1.6 Paper Organization

In Section 2 we present basic definitions and lemmata, and in particular define what a good decomposition is and how it can be used in learning permutations from pairwise preferences. Section 3 presents our main active learning algorithm which is, in fact, an algorithm for producing a good

2. Additionally, they consider the so called binary ranking, which is not the problem here.

decomposition query efficiently. The main result is presented in Theorem 7. Section 4 discusses our main results as a preconditioner for a standard SVM relaxation for the hard combinatorial problems underlying the problem of minimum feedback-arcset in sparse graphs.

2. Notation and Basic Lemmata

We start by introducing basic notations and definitions of the problem, together with results from statistical learning theory which we will later improve.

2.1 The Learning Theoretical Problem

Let V denote a finite set that we wish to rank. In a more general setting we are given a sequence $V^1, V^2, \ldots$ of sets, but there is enough structure and interest in the single set case, which we focus on in this work. Denote by n the cardinality of V. We assume there is an underlying preference function W on pairs of elements in V, which is unknown to us. For any ordered pair $u, v \in V$, the preference value $W(u,v)$ takes the value of 1 if u is deemed preferred over v, and 0 otherwise. We enforce $W(u,v) + W(v,u) = 1$, hence, (V,W) is a tournament. We assume that W is *agnostic* in the sense that it does not necessarily encode a transitive preference function, and may contain errors and inconsistencies. For convenience, for any two real numbers a, b we will let $[a,b]$ denote the interval $\{x : a \leq x \leq b\}$ if $a \leq b$ and $\{x : b \leq x \leq a\}$ otherwise.

Assume now that we wish to predict W using a hypothesis h from some concept class $\mathcal{H}$. The hypothesis h will take an ordered pair $(u,v) \in V$ as input, and will output label of 1 to assert that u *precedes* v and 0 otherwise. We want $\mathcal{H}$ to contain only consistent hypotheses, satisfying transitivity (i.e., if $h(u,v) = h(v,w) = 1$ then $h(u,w) = 1$). A typical way to do this is using a linear score function: Each $u \in V$ is endowed with a feature vector $\varphi(u)$ in some RKHS H, a weight vector $w \in H$ is used for parametrizing each $h_w \in \mathcal{H}$, and the prediction is as follows:[3]

$$h_w(u,v) = \begin{cases} 1 & \langle w, \varphi(u) \rangle > \langle w, \varphi(v) \rangle \\ 0 & \langle w, \varphi(u) \rangle < \langle w, \varphi(v) \rangle \\ \mathbf{1}_{u<v} & \text{otherwise} \end{cases} .$$

Our work is relevant, however, to nonlinear hypothesis classes as well. We denote by $\Pi(V)$ the set permutations on the set V, hence we always assume $\mathcal{H} \subseteq \Pi(V)$. (Permutations π are naturally viewed as binary classifiers of pairs of elements via the preference predicate: The notation is, $\pi(u,v) = 1$ if and only if $u \prec_\pi v$, namely, if u precedes v in π. Slightly abusing notation, we also view permutations as injective functions from $[n]$ to V, so that the element $\pi(1) \in V$ is in the first, most preferred position and $\pi(n)$ is the least preferred one. We also define the function ρ_π inverse to π as the unique function satisfying $\pi(\rho_\pi(v)) = v$ for all $v \in V$. Hence, $u \prec_\pi v$ is equivalent to $\rho_\pi(u) < \rho_\pi(v)$.)

As in standard ERM setting, we assume a non-negative risk function $C_{u,v}$ penalizing the error of h with respect to the pair u, v, namely,

$$C_{u,v}(h, V, W) = \mathbf{1}_{h(u,v) \neq W(u,v)} .$$

3. We assume that $V$is endowed with an arbitrary linear order relation, so we can formally write $u < v$ to arbitrarily yet consistently break ties.

The total loss, $C(h,V,W)$ is defined as $C_{u,v}$ summed over all unordered $u,v \in V$. Our goal is to devise an active learning algorithm for the purpose of minimizing this loss.

In this paper we find an almost optimal solution to the problem using important breakthroughs in combinatorial optimization of a related problem called *minimum feedback arc-set in tournaments* (MFAST). The relation between this NP-Hard problem and our learning problem has been noted before (Cohen et al., 1998), but no provable almost optimal active learning has been devised, as far as we know.

2.2 The Combinatorial Optimization Counterpart

MFAST is defined as follows: Assume we are given V and W and its entirety, in other words, we pay no price for reading W. The goal is to order the elements of V in a full linear order, while minimizing the total pairwise violation. More precisely, we wish to find a permutation π on the elements of V such that the total backward cost:

$$C(\pi,V,W) = \sum_{u \prec_\pi v} W(v,u) \tag{1}$$

is minimized. The expression in (1) will be referred to as the *MFAST cost* henceforth.

When W is given as input, this problem is known as the minimum feedback arc-set in tournaments (MFAST). A PTAS has been discovered for this NP-Hard very recently (Kenyon-Mathieu and Schudy, 2007). Though a major theoretical achievement from a combinatorial optimization point of view, the PTAS is not useful for the purpose of *learning to rank from pairwise preferences* because it is not query efficient. Indeed, it may require in some cases to read all quadratically many entries in W. In this work we fix this drawback, while using their main ideas for the purpose of machine learning to rank. We are not interested in MFAST per se, but use the algorithm by Kenyon-Mathieu and Schudy (2007) to obtain a certain useful decomposition of the input (V,W) from which our main active learning result easily follows.

Definition 1 *Given a set V of size n, an ordered decomposition is a list of pairwise disjoint subsets $V_1,\ldots,V_k \subseteq V$ such that $\cup_{i=1}^k V_i = V$. For a given decomposition, we let $W|_{V_i}$ denote the restriction of W to $V_i \times V_i$ for $i = 1,\ldots,k$. Similarly, for a permutation $\pi \in \Pi(v)$ we let $\pi|_{V_i}$ denote the restriction of the permutation to the elements of V_i (hence, $\pi|_{V_i} \in \Pi(V_i)$). We say that $\pi \in \Pi(V)$ respects $V_1,\ldots,V_k$ if for all $u \in V_i, v \in V_j, i < j$, $u \prec_\pi v$. We denote the set of permutations $\pi \in \Pi(V)$ respecting the decomposition $V_1,\ldots,V_k$ by $\Pi(V_1,\ldots,V_k)$. We say that a subset U of V is* small in V *if $|U| \leq \log n / \log\log n$, otherwise we say that U* is big in V. *A decomposition $V_1,\ldots,V_k$ is* ε-good *with respect to W if:*[4]

- *Local chaos:*

$$\min_{\pi \in \Pi(V)} \sum_{i:V_i \text{ big in } V} C(\pi_{|V_i}, V_i, W_{|V_i}) \geq \varepsilon^2 \sum_{i:V_i \text{ big in } V} \binom{n_i}{2} . \tag{2}$$

- *Approximate optimality:*

$$\min_{\sigma \in \Pi(V_1,\ldots,V_k)} C(\sigma,V,W) \leq (1+\varepsilon) \min_{\pi \in \Pi(V)} C(\pi,V,W) . \tag{3}$$

4. We will just say ε-good if W is clear from the context.

Intuitively, an ε-good decomposition identifies a block-ranking of the data that is difficult to rank in accordance with W internally on average among big blocks (*local chaos*), yet possible to rank almost optimally while respecting the decomposition (*approximate optimality*). We show how to take advantage of an ε-good decomposition for learning in Section 2.3. The ultimate goal will be to find an ε-good decomposition of the input set V using $O(\mathrm{poly}(\log n, \varepsilon^{-1}))$ queries into W.

2.3 Basic Results from Statistical Learning Theory

In statistical learning theory, one seeks to find a classifier minimizing an expected cost incurred on a random input by minimizing the empirical cost on a sample thereof. If we view pairs of elements in V as data points, then the MFAST cost can be cast, up to normalization, as an expected cost over a random draw of a data point. The distribution space is finite, hence we may view this as a transductive learning algorithm. Recall our notation of $\pi(u,v)$ denoting the indicator function for the predicate $u \prec_\pi v$. Thus π is viewed as a binary hypothesis function over $\binom{V}{2}$, and $\Pi(V)$ can be viewed as the concept class of all binary hypotheses satisfying transitivity: $\pi(u,v)+\pi(v,y) \geq \pi(u,y)$ for all u,v,y.

A sample E of unordered pairs gives rise to a *partial cost*, C_E defined as follows:

Definition 2 *Let (V,E) denote an undirected graph over V, which may contain parallel edges (E is a multi-set). The partial MFAST cost $C_E(\pi)$ is defined as*

$$C_E(\pi,V,W) = \binom{n}{2}|E|^{-1} \sum_{\substack{(u,v)\in E \\ u<_\pi v}} W(v,u) \ .$$

(The accounting of parallel edges in E is clear.) The function $C_E(\cdot,\cdot,\cdot)$ can be viewed as an *empirical unbiased estimator* of $C(\pi,V,W)$ if $E \subseteq \binom{V}{2}$ is chosen uniformly at random among all (multi)subsets of a given size.

The basic question in statistical learning theory is, how good is the minimizer π of C_E, in terms of C? The notion of VC dimension by Vapnik and Chervonenkis (1971) gives us a nontrivial bound which is, albeit suboptimal (as we shall soon see), a good start for our purpose.

Lemma 3 *The VC dimension of the set of permutations on V, viewed as binary classifiers on pairs of elements, is $n-1$.*

It is easy to show that the VC dimension is at most $O(n\log n)$. Indeed, the number of permutations is at most $n!$, and the VC dimension is always bounded by the log of the concept class cardinality. That the bound is linear was proven by Ailon and Radinsky (2011). We present the proof here in Appendix A for completeness. The implications of the VC bound are as follows.

Proposition 4 *Assume E is chosen uniformly at random (with repetitions) as a sample of m elements from $\binom{V}{2}$, where $m > n$. Then with probability at least $1-\delta$ over the sample, all permutations π satisfy:*

$$|C_E(\pi,V,W) - C(\pi,V,W)| = n^2 O\left(\sqrt{\frac{n\log m + \log(1/\delta)}{m}}\right) \ .$$

The consequence of Proposition 4 are as follows: If we want to minimize $C(\pi,V,W)$ over π to within an additive error of μn^2, and succeed in doing so with probability at least $1-\delta$, it is enough

to choose a sample E of $O(\mu^{-2}(n\log n+\log\delta^{-1}))$ elements from $\binom{V}{2}$ uniformly at random (with repetitions), and optimize $C_E(\pi,V,W)$. Assume from now on that δ is at least e^{-n}, so that we get a more manageable sample bound of $O(\mu^{-2}n\log n)$. Before turning to optimizing $C_E(\pi,V,W)$, a hard problem in its own right (Karp, 1972; Dinur and Safra, 2002), we should first understand whether this bound is at all good for various scenarios. We need some basic notions of distance between permutations. For two permutations π,σ, the Kendall-Tau distance $d_\tau(\pi,\sigma)$ is defined as

$$d_\tau(\pi,\sigma)=\sum_{u\neq v}\mathbf{1}[(u\prec_\pi v)\wedge(v\prec_\sigma u)]\ .$$

The Spearman Footrule distance $d_{\text{foot}}(\pi,\sigma)$ is defined as

$$d_{\text{foot}}(\pi,\sigma)=\sum_u|\rho_\pi(u)-\rho_\sigma(u)|\ .$$

The following is a well known inequality due to Diaconis and Graham (1977) relating the two distance measures for all π,σ:

$$d_\tau(\pi,\sigma)\leq d_{\text{foot}}(\pi,\sigma)\leq 2d_\tau(\pi,\sigma)\ . \tag{4}$$

Clearly d_τ and d_{foot} are metrics. It is also clear that $C(\cdot,V,\cdot)$ is an extension of $d_\tau(\cdot,\cdot)$ to distances between permutations and binary tournaments, with the triangle inequality of the form $d_\tau(\pi,\sigma)\leq C(\pi,V,W)+C(\sigma,V,W)$ satisfied for all W and $\pi,\sigma\in\Pi(V)$.

Assume now that we are able, using Proposition 4 and the ensuing comment, to find a solution π for MFAST, with an additive regret of $O(\mu n^2)$ with respect to an optimal solution π^* for some $\mu>0$. The triangle inequality implies that the distance $d_\tau(\pi,\pi^*)$ between our solution and the true optimal is $\Omega(\mu n^2)$. By (4), this means that $d_{\text{foot}}(\pi,\pi^*)=\Omega(\mu n^2)$. By the definition of d_{foot}, this means that the average element $v\in V$ is translated $\Omega(\mu n)$ positions away from its position in π^*. In a real life application (e.g., in information retrieval), one may want elements to be at most a constant γ positions away from their position in a correct permutation. This translates to a sought regret of $O(\gamma n)$ in $C(\pi,V,W)$, or, using the above notation, to $\mu=\gamma/n$. Clearly, Proposition 4 cannot guarantee less than a quadratic sample size for such a regret, which is tantamount to querying W in its entirety. We can do better: In this work, for any $\varepsilon>0$ we will achieve a regret of $O(\varepsilon C(\pi^*,V,W))$ using $O(\varepsilon^{-6}n\log^5 n)$ queries into W, regardless of how small the optimal cost $C(\pi^*,V,W)$ is. Hence, our regret is relative to the optimal loss. This is clearly not achievable using Proposition 4. Let us outline another practical case of interest. Assume a scenario in which a ground truth permutation $\pi\in\Pi(V)$ exists, and the noisy preference matrix W is generated by a human responder who errs on a pair u,v with probability $f(|\rho_\pi(u)-\rho_\pi(v)|)$, where f is some monotonically decreasing function. Intuitively, this scenario posits that people confuse the order of two elements the "closer" they are to each other. If, say, $f(x)=px^{-\nu}$ for some $\nu>0$ and $p>0$, then the cost of the optimal solution π would be $\Theta(pn^{2-\nu})$ with high probability.[5] Proposition 4 tells us that if we wanted to find a permutation with *relative* error of ε, namely, of absolute error $\Theta(\varepsilon pn^{2-\nu})$, then we would need $O(\varepsilon^{-2}p^{-2}n^{1+2\nu}\log n)$ queries. Our result achieves the same error with an almost linear dependence on n (albeit a worse dependence on ε).

One may argue that the VC bound measures the merits of uniform, non-adaptive sampling too pessimistically. This isn't the case. Consider the extreme case in which the optimal cost is zero. We

5. We are assuming stochastic noise for the sake of the example, although this work deals with adversarial noise.

argue that a uniform sample of pairs *requires* $\Omega(n^2)$ query complexity. Indeed, if the optimal cost is zero then unless one queries all $n-1$ consecutive pairs in the unique optimal permutation, one cannot reveal it. It is now easy to see that sampling $o(n^2)$ pairs uniformly (either with or without repetition) would succeed in doing so with exponentially (in n) small probability. A relative ε approximation cannot be thus achieved. But we know that an adaptive sample of $O(n\log n)$ pairs on expectation (QuickSort) does better. It is folklore that $\Omega(n\log n)$ is also a lower bound in the perfect (zero cost) case. Hence, one cannot hope to get rid of the $\log n$ factor in our main result, Theorem 7 below.

Before continuing, we need need a slight generalization of Proposition 4.

Proposition 5 *Let $V_1,\ldots,V_k$ be an ordered decomposition of V. Let $\mathcal{B}$ denote the set of indices $i\in[k]$ such that V_i is big in V. Assume E is chosen uniformly at random (with repetitions) as a sample of m elements from $\bigcup_{i\in\mathcal{B}}\binom{V_i}{2}$, where $m>n$. For each $i=1,\ldots,k$, let $E_i=E\cap\binom{V_i}{2}$. Define $C_E(\pi,\{V_1,\ldots,V_k\},W)$ to be*

$$C_E(\pi,\{V_1,\ldots,V_k\},W)=\left(\sum_{i\in\mathcal{B}}\binom{n_i}{2}\right)|E|^{-1}\sum_{i\in\mathcal{B}}\binom{n_i}{2}^{-1}|E_i|C_{E_i}(\pi_{|V_i},V_i,W_{|V_i})\;. \tag{5}$$

(The normalization is defined so that the expression is an unbiased estimator of $\sum_{i\in\mathcal{B}}C(\pi_{|V_i},V_i,W_{|V_i})$. If $|E_i|=0$ for some i, formally define $\binom{n_i}{2}^{-1}|E_i|C_{E_i}(\pi_{|V_i},V_i,W_{|V_i})=0$.) Then with probability at least $1-e^{-n}$ over the sample, all permutations $\pi\in\Pi(V)$ satisfy:

$$\left|C_E(\pi,\{V_1,\ldots,V_k\},W)-\sum_{i\in\mathcal{B}}C(\pi|_{V_i},V_i,W|_{V_i})\right|=\sum_{i\in\mathcal{B}}\binom{n_i}{2}O\left(\sqrt{\frac{n\log m+\log(1/\delta)}{m}}\right)\;.$$

Proof Consider the set of binary functions $\prod_{i\in\mathcal{B}}\Pi(V_i)$ on the domain $\bigcup_{i\in\mathcal{B}}V_i\times V_i$, defined as follows: If $u,v\in V_j\times V_j$ for some $j\in\mathcal{B}$, then

$$\left((\pi_i)_{i\in\mathcal{B}}\right)(u,v)=\pi_j(u,v)\;.$$

It is clear that the VC dimension of this function set is at most the sum of the VC dimensions of $\{\Pi(V_i)\}_{i\in\mathcal{B}}$, hence by Lemma 3 at most n. The result follows. ■

2.4 Using an ε-Good Partition

The following lemma explains why an ε-good partition is good for our purpose.

Lemma 6 *Fix $\varepsilon>0$ and assume we have an ε-good partition (Definition 1) $V_1,\ldots,V_k$ of V. Let $\mathcal{B}$ denote the set of $i\in[k]$ such that V_i is big in V, and let $\bar{\mathcal{B}}=[k]\setminus\mathcal{B}$. Let $n_i=|V_i|$ for $i=1,\ldots,n$, and let E denote a random sample of $O(\varepsilon^{-6}n\log n)$ elements from $\bigcup_{i\in\mathcal{B}}\binom{V_i}{2}$, each element chosen uniformly at random with repetitions. Let E_i denote $E\cap\binom{V_i}{2}$. Let $C_E(\pi,\{V_1,\ldots,V_k\},W)$ be defined as in (5). For any $\pi\in\Pi(V_1,\ldots,V_k)$ define:*

$$\tilde{C}(\pi):=C_E(\pi,\{V_1,\ldots,V_k\},W)+\sum_{i\in\bar{\mathcal{B}}}C(\pi_{|V_i},V_i,W_{|V_i})+\sum_{1\le i<j\le k}\sum_{(u,v)\in V_i\times V_j}\mathbf{1}_{v\prec_\pi u}\;.$$

Then the following event occurs with probability at least $1-e^{-n}$*: For all* $\sigma \in \Pi(V_1,\ldots,V_k)$*,*

$$\left|\tilde{C}(\sigma) - C(\sigma,V,W)\right| \leq \varepsilon \min_{\pi\in\Pi(V)} C(\pi,V,W)\ . \tag{6}$$

Also, if σ^* *is any minimizer of* $\tilde{C}(\cdot)$ *over* $\Pi(V_1,\ldots,V_k)$*, then*

$$C(\sigma^*,V,W) \leq (1+2\varepsilon) \min_{\pi\in\Pi(V)} C(\pi,V,W)\ . \tag{7}$$

Before we prove the lemma, let us discuss its consequences: Given an ε-good decomposition $V_1,\ldots,V_k$ of V, the theorem implies that if we could optimize $\tilde{C}(\sigma)$ over $\sigma \in \Pi(V_1,\ldots,V_k)$, we would obtain a permutation π with a *relative regret* of 2ε with respect to the optimizer of $C(\cdot,V,W)$ over $\Pi(V)$. Optimizing $\sum_{i\in\hat{\mathcal{B}}} C(\pi_{|V_i},V_i,W_{|V_i})$ is easy: Each V_i is of size at most $\log n/\log\log n$, hence exhaustively searching its corresponding permutation space can be done in polynomial time. In order to compute the cost of each permutation inside the small sets V_i, we would need to query $W_{|V_i}$ in its entirety. This incurs a query cost of at most $\sum_{i\in\bar{\mathcal{B}}}\binom{n_i}{2} = O(n\log n/\log\log n)$, which is dominated by the cost of obtaining the ε-good partition in the first place (see next section). Optimizing $C_E(\pi,\{V_1,\ldots,V_k\},W)$ given E is a tougher nut to crack, is known as the minimum feedback arc-set (MFAS) problem and is computationally much harder than than MFAST (Karp, 1972; Dinur and Safra, 2002). For now we focus on query and not computational complexity, and notice that the size $|E| = O(\varepsilon^{-4}n\log n)$ of the sample set is all we need. In Section 4 we show a counterpart of Lemma 6 which provides similar guarantees for practitioners who choose to relax it using SVM, for which fast solvers exist.

Proof For any permutation $\sigma \in \Pi(V_1,\ldots,V_k)$, it is clear that

$$\tilde{C}(\sigma) - C(\sigma,V,W) = C_E(\sigma,\{V_1,\ldots,V_k\},W) - \sum_{i\in\mathcal{B}} C(\sigma_{|V_i},V_i,W_{|V_i})\ .$$

By Proposition 5, with probability at least $1-e^{-n}$ the absolute value of the RHS is bounded by $\varepsilon^3\sum_{i\in\mathcal{B}}\binom{n_i}{2}$, which is at most $\varepsilon\min_{\pi\in\Pi(V)}C(\pi,V,W)$ by (2). This establishes (6). Inequality (7) is obtained from (6) together with (3) and the triangle inequality. ■

3. A Query Efficient Algorithm for ε-Good Decomposing

The section is dedicated to proving the following:

Theorem 7 *Given a set* V *of size* n*, a preference oracle* W *and an error tolerance parameter* $0 < \varepsilon < 1$*, there exists a polynomial time algorithm which returns, with constant probability, an* ε*-good partition of* V*, querying at most* $O(\varepsilon^{-6}n\log^5 n)$ *locations in* W *on expectation. The running time of the algorithm (counting computations) is* $O(n\,\mathrm{poly}(\log n,\varepsilon^{-1}))$*.*

Before describing our algorithm, we need some definitions.

Definition 8 *Let* π *denote a permutation over* V*. Let* $v \in V$ *and* $i \in [n]$*. We define* $\pi_{v\to i}$ *to be the permutation obtained by moving the rank of* v *to* i *in* π*, and leaving the rest of the elements in the same order. For example, if* $V = \{x,y,z\}$ *and* $(\pi(1),\pi(2),\pi(3)) = (x,y,z)$*, then* $(\pi_{x\to 3}(1),\pi_{x\to 3}(2),\pi_{x\to 3}(3)) = (y,z,x)$*.*

Definition 9 *Fix a permutation π over V, an element $v \in V$ and an integer $i \in [n]$. We define the number* $\mathrm{TestMove}(\pi, V, W, v, i)$ *as the decrease in the cost $C(\cdot, V, W)$ achieved by moving from π to $\pi_{v \to i}$. More precisely,* $\mathrm{TestMove}(\pi, V, W, v, i) = C(\pi, V, W) - C(\pi_{v \to i}, V, W)$ *. Equivalently, if $i \geq \rho_\pi(v)$ then*

$$\mathrm{TestMove}(\pi, V, W, v, i) = \sum_{u : \rho_\pi(u) \in [\rho_\pi(v)+1, i]} (W_{uv} - W_{vu}) \ .$$

A similar expression can be written for $i < \rho_\pi(v)$.

Now assume that we have a multi-set $E \subseteq \binom{V}{2}$. We define $\mathrm{TestMove}_E(\pi, V, W, v, i)$*, for $i \geq \rho_\pi(v)$, as*

$$\mathrm{TestMove}_E(\pi, V, W, v, i) \quad = \quad \frac{|i - \rho_\pi(v)|}{|\tilde{E}|} \sum_{u : (u,v) \in \tilde{E}} (W(u,v) - W(v,u)) \ ,$$

where the multiset $\tilde{E}$ is defined as $\{(u,v) \in E : \rho_\pi(u) \in [\rho_\pi(v)+1, i]\}$. Similarly, for $i < \rho_\pi(v)$ we define

$$\mathrm{TestMove}_E(\pi, V, W, v, i) \quad = \quad \frac{|i - \rho_\pi(v)|}{|\tilde{E}|} \sum_{u : (u,v) \in \tilde{E}} (W(v,u) - W(u,v)) \ , \tag{8}$$

where the multiset $\tilde{E}$ is now defined as $\{(u,v) \in E : \rho_\pi(u) \in [i, \rho_\pi(v)-1]\}$.

Lemma 10 *Fix a permutation π over V, an element $v \in V$, an integer $i \in [n]$ and another integer N. Let $E \subseteq \binom{V}{2}$ be a random (multi)-set of size N with elements $(v, u_1), \ldots, (v, u_N)$, drawn so that for each $j \in [N]$ the element u_j is chosen uniformly at random from among the elements lying between v (exclusive) and position i (inclusive) in π.*
Then $\mathbf{E}[\mathrm{TestMove}_E(\pi, V, W, v, i)] = \mathrm{TestMove}(\pi, V, W, v, i)$*. Additionally, for any $\delta > 0$, except with probability of failure δ,*

$$|\mathrm{TestMove}_E(\pi, V, W, v, i) - \mathrm{TestMove}(\pi, V, W, v, i)| = O\left(|i - \rho_\pi(v)| \sqrt{\frac{\log \delta^{-1}}{N}} \right) \ .$$

The lemma is easily proven using, for example, Hoeffding tail bounds, using the fact that $|W(u,v)| \leq 1$ for all u, v.

3.1 The Decomposition Algorithm

Our decomposition algorithm SampleAndRank is detailed in Algorithm 1, with subroutines in Algorithms 2 and 3. It can be viewed as a query efficient improvement of the main algorithm of Kenyon-Mathieu and Schudy (2007). Another difference is that we are not interested in an approximation algorithm for MFAST: Whenever we reach a small block (line 3) or a big block with a probably approximately sufficiently high cost (line 8) in our recursion of Algorithm 2), we simply output it as a block in our partition. Denote the resulting outputted partition by $V_1, \ldots, V_k$. Denote by $\hat{\pi}$ the minimizer of $C(\cdot, V, W)$ over $\Pi(V_1, \ldots, V_k)$. Most of the analysis is dedicated to showing that $C(\hat{\pi}, V, W) \leq (1+\varepsilon) \min_{\pi \in \Pi(V)} C(\pi, V, W)$, thus establishing (3).

In order to achieve an efficient query complexity compared to that of Kenyon-Mathieu and Schudy (2007), we use procedure ApproxLocalImprove (Algorithm 3) to replace a greedy local

improvement step there which is *not* query efficient. Aside from the aforementioned differences, we also raise here the reader's awareness to the query efficiency of QuickSort, which was established by Ailon and Mohri (2010).

SampleAndRank (Algorithm 1) takes the following arguments: The set V we want to rank, the preference matrix W and an accuracy argument ε. It is implicitly understood that the argument W passed to SampleAndRank is given as a query oracle, incurring a unit cost upon each access to a matrix element by the procedure and any nested calls.

The first step in SampleAndRank is to obtain an expected constant factor approximation π to MFAST on V,W, incurring an expected low query cost. More precisely, this step returns a random permutation π with an expected cost of $O(1)$ times that of the optimal solution to MFAST on V,W. The query complexity of this step is $O(n\log n)$ *on expectation* (Ailon and Mohri, 2010). Before continuing, we make the following assumption, which holds with constant probability using Markov probability bounds.

Assumption 11 *The cost $C(\pi,V,W)$ of the initial permutation π computed line 2 of* SampleAndRank *is at most $O(1)$ times that of the optimal solution π^* to MFAST on (V,W), and the query cost incurred in the computation is $O(n\log n)$.*

Following QuickSort, a recursive procedure SampleAndDecompose is called. It implements a divide-and-conquer algorithm. Before branching, it executes the following steps. Lines 5 to 9 are responsible for identifying local chaos, with sufficiently high probability. The following line 10 calls a procedure ApproxLocalImprove (Algorithm 3) which is responsible for performing query-efficient approximate greedy steps. We devote the next Sections 3.2-3.4 to describing this procedure. The establishment of the ε-goodness of SampleAndRank's output (establishing (3)) is deferred to Section 3.5.

3.2 Approximate Local Improvement Steps

The procedure ApproxLocalImprove takes as input a set V of size N, the preference oracle W, a permutation π on V, two numbers C_0, ε and an integer n. The number n is the size of the input in the root call to SampleAndDecompose, passed down in the recursion, and used for the purpose of controlling the success probability of each call to the procedure (there are a total of $O(n\log n)$ calls, and a union bound will be used to bound a failure probability, hence each call may fail with probability inversely polynomial in n). The goal of the procedure is to repeatedly identify, with high probability, single vertex moves that considerably decrease the cost. Note that in the PTAS of Kenyon-Mathieu and Schudy (2007), a crucial step in their algorithms entails identifying single vertex moves that decrease the cost by a magnitude which, given our sought query complexity, would not be detectable. Hence, our algorithm requires altering this crucial part in their algorithm.

The procedure starts by creating a *sample ensemble* $\mathcal{S}=\{E_{v,i} : v\in V, i\in[B,L]\}$, where $B=\log\lfloor\Theta(\varepsilon N/\log n)\rfloor$ and $L=\lceil\log N\rceil$. The size of each $E_{v,i}\in\mathcal{S}$ is $\Theta(\varepsilon^{-2}\log^2 n)$, and each element $(v,x)\in E_{v,i}$ was added (with possible multiplicity) by uniformly at random selecting, with repetitions, an element $x\in V$ positioned at distance at most 2^i from the position of v in π. Let $\mathcal{D}_\pi$ denote the distribution space from which $\mathcal{S}$ was drawn, and let $\Pr_{X\sim\mathcal{D}_\pi}[X=\mathcal{S}]$ denote the probability of obtaining a given sample ensemble $\mathcal{S}$.

We want $\mathcal{S}$ to enable us to approximate the improvement in cost obtained by moving a single element u to position j.

Definition 12 *Fix $u \in V$ and $j \in [n]$, and assume $\log|j-\rho_\pi(u)| \geq B$. Let $\ell = \lceil \log|j-\rho_\pi(u)| \rceil$. We say that $\mathcal{S}$ is successful at u, j if $|\{x : (u,x) \in E_{u,\ell}\} \cap \{x : \rho_\pi(x) \in [\rho_\pi(u), j]\}| = \Omega(\varepsilon^{-2}\log^2 n)$.*

In words, success of $\mathcal{S}$ at u, j means that sufficiently many samples $x \in V$ such that $\rho_\pi(x)$ is between $\rho_\pi(u)$ and j are represented in $E_{u,\ell}$. Conditioned on $\mathcal{S}$ being successful at u, j, note that the denominator of TestMove_E (defined in (8)) does not vanish, and we can thereby define:

Definition 13 *$\mathcal{S}$ is a* good approximation *at u, j if*

$$\left|\text{TestMove}_{E_{u,\ell}}(\pi, V, W, u, j) - \text{TestMove}(\pi, V, W, u, j)\right| \leq \frac{1}{2}\varepsilon|j-\rho_\pi(u)|/\log n \ ,$$

where ℓ is as in Definition 12.

In words, $\mathcal{S}$ being a good approximation at u, j allows us to approximate a quantity of interest $\text{TestMove}(\pi, V, W, u, j)$, and to detect whether it is sufficiently large, and more precisely, at least $\Omega(\varepsilon|j-\rho_\pi(u)|/\log n)$.

Definition 14 *We say that $\mathcal{S}$ is a good approximation if it is successful and a good approximation at all $u \in V$, $j \in [n]$ satisfying $\lceil \log|j-\rho_\pi(u)| \rceil \in [B, L]$.*

Using Chernoff bounds to ensure that $\mathcal{S}$ is successful $\forall u, j$ as in Definition 14, then using Hoeffding to ensure that $\mathcal{S}$ is a good approximation at all such u, j and finally union bounding we get

Lemma 15 *Except with probability $1 - O(n^{-4})$, $\mathcal{S}$ is a good approximation.*

Algorithm 1 SampleAndRank(V, W, ε)

1: $n \leftarrow |V|$
2: $\pi \leftarrow$ Expected $O(1)$-approx solution to MFAST using $O(n \log n)$ W-queries on expectation using QuickSort (Ailon et al., 2008a)
3: **return** SampleAndDecompose$(V, W, \varepsilon, n, \pi)$

3.3 Mutating the Pair Sample To Reflect a Single Element Move

Line 17 in ApproxLocalImprove requires elaboration. In lines 15-20, we check whether there exists an element u and position j, such that moving u to j (giving rise to $\pi_{u\to j}$) would considerably improve the MFAST cost of the procedure input, based on a high probability approximate calculation. The approximation is done using the sample ensemble $\mathcal{S}$. If such an element u exists, we execute the exchange $\pi \leftarrow \pi_{u\to j}$. With respect to the new value of the permutation π, the sample ensemble $\mathcal{S}$ becomes *stale*. By this we mean, that if $\mathcal{S}$ was a good approximation with respect to π, then it is no longer necessarily a good approximation with respect to $\pi_{u\to j}$. We must refresh it. Before the next iteration of the while loop, we perform in line 17 a transformation $\varphi_{u\to j}$ to $\mathcal{S}$, so that the resulting sample ensemble $\varphi_{u\to j}(\mathcal{S})$ is distributed according to $\mathcal{D}_{\pi_{u\to j}}$. More precisely, we will define a transformation φ such that

$$\varphi_{u\to j}(\mathcal{D}_\pi) = D_{\pi_{u\to j}} \ , \tag{9}$$

where the left hand side denotes the distribution obtained by drawing from $\mathcal{D}_\pi$ and applying $\varphi_{u\to j}$ to the result. The transformation $\varphi_{u\to j}$ is performed as follows. Denoting $\varphi_{u\to j}(\mathcal{S}) = \mathcal{S}' = \{E'_{v,i} : v \in V, i \in [B, L]\}$, we need to define each $E'_{v,i}$.

Algorithm 2 SampleAndDecompose(V,W,ε,n,π)

$N \leftarrow |V|$
if $N \leq \log n/\log\log n$ **then**
  return trivial partition $\{V\}$
end if
$E \leftarrow$ random subset of $O(\varepsilon^{-4}\log n)$ elements from $\binom{V}{2}$ (with repetitions)
$C \leftarrow C_E(\pi,V,W)$ (C is an additive $O(\varepsilon^2N^2)$ approximation of C w.p. $\geq 1-n^{-4}$)
if $C = \Omega(\varepsilon^2N^2)$ **then**
  return trivial partition $\{V\}$
end if
$\pi_1 \leftarrow$ ApproxLocalImprove(V,W,π,ε,n)
$k \leftarrow$ random integer in the range $[N/3,2N/3]$
$V_L \leftarrow \{v \in V : \rho_\pi(v) \leq k\}$, $\pi_L \leftarrow$ restriction of π_1 to V_L
$V_R \leftarrow V \setminus V_L$, $\pi_R \leftarrow$ restriction of π_1 to V_R
return concatenation of decomposition SampleAndDecompose$(V_L,W,\varepsilon,n,\pi_L)$ and decomposition SampleAndDecompose$(V_R,W,\varepsilon,n,\pi_R)$

Algorithm 3 ApproxLocalImprove(V,W,π,ε,n) (Note: π used as both input and output)

$N \leftarrow |V|$, $B \leftarrow \lceil\log(\Theta(\varepsilon N/\log n)\rceil$, $L \leftarrow \lceil\log N\rceil$
if $N = O(\varepsilon^{-3}\log^3 n)$ **then**
  return
end if
for $v \in V$ **do**
  $r \leftarrow \rho_\pi(v)$
  for $i = B\ldots L$ **do**
    $E_{v,i} \leftarrow \emptyset$
    for $m = 1..\Theta(\varepsilon^{-2}\log^2 n)$ **do**
      $j \leftarrow$ integer uniformly at random chosen from $[\max\{1,r-2^i\},\min\{n,r+2^i\}]$
      $E_{v,i} \leftarrow E_{v,i} \cup \{(v,\pi(j))\}$
    end for
  end for
end for
while $\exists u \in V$ and $j \in [n]$ s.t. *(setting* $\ell := \lceil\log|j-\rho_\pi(u)|\rceil$*):*

$$\ell \in [B,L] \text{ and } \text{TestMove}_{E_{u,\ell}}(\pi,V,W,u,j) > \varepsilon|j-\rho_\pi(u)|/\log n$$

do
  for $v \in V$ and $i \in [B,L]$ **do**
    refresh sample $E_{v,i}$ with respect to the move $u \to j$ (see Section 3.3)
  end for
  $\pi \leftarrow \pi_{u\to j}$
end while

Definition 16 *We say that $E_{v,i}$ is* interesting *in the context of π and $\pi_{u\to j}$ if the two sets T_1, T_2 defined as*

$$\begin{aligned} T_1 &= \{x \in V : |\rho_\pi(x) - \rho_\pi(v)| \le 2^i\} \\ T_2 &= \{x \in V : |\rho_{\pi_{u\to j}}(x) - \rho_{\pi_{u\to j}}(v)| \le 2^i\} \end{aligned}$$

differ.

We set $E'_{v,i} = E_{v,i}$ for all v, i for which $E_{v,i}$ is *not* interesting.

Observation 17 *There are at most $O(|\rho_\pi(u) - j| \log n)$ interesting choices of v, i. Additionally, if $v \neq u$, then for T_1, T_2 as in Definition 16, $|T_1 \Delta T_2| = O(1)$, where Δ denotes symmetric difference.*

Fix one interesting choice v, i. Let T_1, T_2 be as in Definition 16. By the last observation, each of T_1 and T_2 contains $O(1)$ elements that are not contained in the other. Assume $|T_1| = |T_2|$, let $X_1 = T_1 \setminus T_2$, and $X_2 = T_2 \setminus T_1$. Fix any injection $\alpha : X_1 \to X_2$, and extend $\alpha : T_1 \to T_2$ so that $\alpha(x) = x$ for all $x \in T_1 \cap T_2$. Finally, define

$$E'_{v,i} = \{(v, \alpha(x)) : (v, x) \in E_{v,i}\} \ . \tag{10}$$

(The case $|T_1| \neq |T_2|$ may occur due to the clipping of the ranges $[\rho_\pi(v) - 2^i, \rho_\pi(v) + 2^i]$ and $[\rho_{\pi_{u\to j}}(v) - 2^i, \rho_{\pi_{u\to j}}(v) + 2^i]$ to a smaller range. This is a simple technicality which may be taken care of by formally extending the set V by N additional elements $\tilde{v}^L_1, \ldots, \tilde{v}^L_N$, extending the definition of ρ_π for all permutation π on V so that $\rho_\pi(\tilde{v}^L_a) = -a + 1$ for all a and similarly $N = |V|$ additional elements $\tilde{v}^R_1, \ldots, \tilde{v}^R_N$ such that $\rho_\pi(\tilde{v}^R_a) = N + a$. Formally extend W so that $W(v, \tilde{v}^L_a) = W(\tilde{v}^L_a, v) = W(v, \tilde{v}^R_a) = W(\tilde{v}^R_a, v) = 0$ for all $v \in V$ and a. This eliminates the need for clipping ranges in line 10 in ApproxLocalImprove.)

Finally, for $v = u$ we create $E'_{v,i}$ from scratch by repeating the loop in line 7 for that v.

It is easy to see that (9) holds. We need, however, something stronger that (9). Since our analysis assumes that $\mathcal{S} \sim \mathcal{D}_\pi$ is successful, we must be able to measure the distance (in total variation) between the random variable ($\mathcal{D}_\pi|$ success) defined by the process of drawing from $\mathcal{D}_\pi$ and conditioning on the result's success, and $\mathcal{D}_{\pi_{u\to j}}$. By Lemma 15, the total variation distance between ($\mathcal{D}_\pi|$ success) and $\mathcal{D}_{\pi_{u\to j}}$ is $O(n^{-4})$. Using a simple chain rule argument, we conclude the following:

Lemma 18 *Fix π^0 on V of size N, and fix $u_1, \ldots, u_k \in V$ and $j_1, \ldots, j_k \in [n]$. Consider the following process. We draw $\mathcal{S}^0$ from $\mathcal{D}_{\pi^0}$, and define*

$$\begin{aligned} \mathcal{S}^1 = \varphi_{u_1\to j_1}(\mathcal{S}^0), S^2 = \varphi_{u_2\to j_2}(\mathcal{S}^1), \quad \cdots \quad , S^k = \varphi_{u_k\to j_k}(\mathcal{S}^{k-1}) \\ \pi^1 = \pi^0_{u_1\to j_1}, \pi^2 = \pi^1_{u_2\to j_2}, \quad \cdots \quad , \pi^k = \pi^{k-1}_{u_k\to j_k} \ . \end{aligned}$$

Consider the random variable S^k conditioned on $S^0, S^1, \ldots, S^{k-1}$ being successful for $\pi_0, \ldots, \pi^{k-1}$, respectively. Then the total variation distance between the distribution of S^k and the distribution $\mathcal{D}_{\pi^k}$ is at most $O(kn^{-4})$.

3.4 Bounding the Query Complexity of Computing $\varphi_{u\to j}(\mathcal{S})$

We now need a notion of distance between $\mathcal{S}$ and $\mathcal{S}'$, measuring how many extra pairs were introduced ino the new sample family. These pairs may incur the cost of querying W. We denote this measure as $\mathrm{dist}(\mathcal{S},\mathcal{S}')$, and define it as $\mathrm{dist}(\mathcal{S},\mathcal{S}') := \left|\bigcup_{v,i} E_{v,i}\Delta E'_{v,i}\right|$.

Lemma 19 *Assume $\mathcal{S}\sim\mathcal{D}_\pi$ for some permutation π, and $\mathcal{S}'=\varphi_{u\to j}$. Then* $\mathbf{E}[\mathrm{dist}(\mathcal{S},\mathcal{S}')] = O(\varepsilon^{-3}\log^3 n)$.

Proof Denote $\mathcal{S}=\{E_{v,i}\}$ and $\mathcal{S}'=\{E'_{v,i}\}$. Fix some $v\neq u$. By construction, the sets $E_{v,i}$ for which $E_{v,i}\neq E'_{v,i}$ must be interesting, and there are at most $O(|\rho_\pi(u)-j|\log n)$ such, using Observation 17. Fix such a choice of v,i. By (10), $E_{v,i}$ will indeed differ from $E'_{v,i}$ only if it contains an element (v,x) for some $x\in T_1\setminus T_2$. But the probability of that is at most

$$1-(1-O(2^{-i}))^{\Theta(\varepsilon^{-2}\log^2 n)} \le 1-e^{-\Theta(\varepsilon^{-2}2^{-i}\log^2 n)} = O(\varepsilon^{-2}2^{-i}\log^2 n)$$

(We used the fact that $i\geq B$, where B is as defined in line 1 of ApproxLocalImprove, and $N=\Omega(\varepsilon^{-3}\log^3 n)$ as guaranteed in line 3 of ApproxLocalImprove.) Therefore, the expected size of $E'_{v,i}\Delta E_{v,i}$ (counted with multiplicities) is $O(\varepsilon^{-2}2^{-i}\log^2 n)$.

Now consider all the interesting sets $E_{v_1.i_1},\ldots,E_{v_P,i_P}$. For each possible value i it is easy to see that there are at most $2|\rho_\pi(u)-j|$ p's for which $i_p=i$. Therefore,

$$\mathbf{E}\left[\sum_{p=1}^{P}|E'_{v_p,i_p}\Delta E_{v_p,i_p}|\right] = O\left(\varepsilon^{-2}|\rho_\pi(u)-j|\log^2 n\sum_{i=B}^{L}2^{-i}\right) ,$$

where B,L are defined in line 1 in ApproxLocalImprove. Summing over $i\in[B,L]$, we get at most $O(\varepsilon^{-3}|\rho_\pi(u)-j|\log^3 n/N)$. For $v=u$, the set $\{E_{v,i}\}$ is drawn from scratch, clearly contributing $O(\varepsilon^{-2}\log^3 n)$ to $\mathrm{dist}(\mathcal{S},\mathcal{S}')$. The claim follows. ■

3.5 Analysis of SampleAndDecompose

Throughout the execution of the algorithm, various *high probability* events must occur in order for the algorithm guarantees to hold. Let $\mathcal{S}_1,\mathcal{S}_2,\ldots$ denote the sample families that are given rise to through the executions of ApproxLocalImprove, either between lines 5 and 14, or as a mutation done between lines 15 and 20. We will need the first $\Theta(n^4)$ to be good approximations, based on Definition 14. Denote this favorable event $\mathcal{E}_1$. By Lemma 18, and using a union bound, with constant probability (say, 0.99) this happens. We also need the cost approximation C obtained in line 5 to be successful. Denote this favorable event $\mathcal{E}_2$. By Hoeffding tail bounds, this happens with probability $1-O(n^{-4})$ for each execution of the line. This line is obviously executed at most $O(n\log n)$ times, and hence we can lower bound the probability of success of all executions by 0.99.

From now throughout, we make the following assumption, which is true by the above with probability at least 0.97.

Assumption 20 *Events $\mathcal{E}_1$ and $\mathcal{E}_2$ hold true.*

Note that by conditioning the remainder of our analysis on this assumption may bias some expectation upper bounds derived earlier and in what follows. This bias can multiply the estimates by at most $1/0.97$, which can be absorbed in the O-notation of these bounds.

Let π^* denote the optimal permutation for the root call to SampleAndDecompose with V, W, ε. The permutation π is, by Assumption 11, a constant factor approximation for MFAST on V, W. Using the triangle inequality, we conclude that $d_\tau(\pi,\pi^*) \leq C(\pi,V,W) + C(\pi^*,V,W)$. Hence, $E[d_\tau(\pi,\pi^*)] = O(C(\pi^*,V,W))$. From this we conclude, using (4), that

$$E[d_{\text{foot}}(\pi,\pi^*)] = O(C(\pi^*,V,W)) .$$

Now consider the recursion tree $\mathcal{T}$ of SampleAndDecompose. Denote I the set of internal nodes, and by $\mathcal{L}$ the set of leaves (i.e., executions exiting from line 8). For a call SampleAndDecompose corresponding to a node X in the recursion tree, denote the input arguments by $(V_X, W, \varepsilon, n, \pi_X)$. Let $L[X], R[X]$ denote the left and right children of X respectively. Let k_X denote the integer k in 11 in the context of $X \in I$. Hence, by our definitions, $V_{L[X]}, V_{R[X]}, \pi_{L[X]}$ and $\pi_{R[X]}$ are precisely V_L, V_R, π_L, π_R from lines 12-13 in the context of node X.

Take, as in line 1, $N_X = |V_X|$. Let π^*_X denote the optimal MFAST solution for instance $(V_X, W_{|V_X})$. By $\mathcal{E}_1$ we conclude that the first $\Theta(n^4)$ times in which we iterate through the while loop in ApproxLocalImprove (counted over all calls to ApproxLocalImprove), the cost of $\pi_{X\,u\to j}$ is an actual improvement compared to π_X (for the current value of π_X, u and j in iteration), and the improvement in cost is of magnitude at least $\Omega(\varepsilon|\rho_{\pi_X}(u) - j|/\log n)$, which is $\Omega(\varepsilon^2 N_X/\log^2 n)$ due to the use of B defined in line 1. But this means that the number of iterations of the while loop in line 15 of ApproxLocalImprove is

$$O(\varepsilon^{-2} C(\pi_X, V_X, W_{|V_X}) \log^2 n / N_X) .$$

Indeed, otherwise the true cost of the running solution would go below 0. Since $C(\pi_X, V_X, W_{|V_X})$ is at most $\binom{N_X}{2}$, the number of iterations is hence at most $O(\varepsilon^{-2} N_X \log^2 n)$. By Lemma 19 the expected query complexity incurred by the call to ApproxLocalImprove is therefore $O(\varepsilon^{-5} N_X \log^5 n)$. Summing over the recursion tree, the total query complexity incurred by calls to ApproxLocalImprove is, on expectation, at most $O(\varepsilon^{-5} n \log^6 n)$.

Now consider the moment at which the while loop of ApproxLocalImprove terminates. Let π_{1X} denote the permutation obtained at that point, returned to SampleAndDecompose in line 10. We classify the elements $v \in V_X$ to two families: V_X^{short} denotes all $u \in V_X$ s.t. $|\rho_{\pi_{1X}}(u) - \rho_{\pi^*_X}(u)| = O(\varepsilon N_X/\log n)$, and V_X^{long} denotes $V_X \setminus V_X^{\text{short}}$. We know, by assumption, that the last sample ensemble $\mathcal{S}$ used in ApproxLocalImprove was a good approximation, hence for all $u \in V_X^{\text{long}}$,

$$\text{TestMove}(\pi_{1X}, V_X, W_{|V_X}, u, \rho_{\pi^*_X}(u)) = O(\varepsilon|\rho_{\pi_{1X}}(u) - \rho_{\pi^*_X}(u)|/\log n). \tag{11}$$

Definition 21 *(Kenyon-Mathieu and Schudy, 2007) For $u \in V_X$, we say that u crosses k_X if the interval $[\rho_{\pi_{1X}}(u), \rho_{\pi^*_X}(u)]$ contains the integer k_X.*

Let V_X^{cross} denote the (random) set of elements $u \in V_X$ that cross k_X as chosen in line 11. We define a key quantity T_X as in Kenyon-Mathieu and Schudy (2007) as follows:

$$T_X = \sum_{u \in V_X^{\text{cross}}} \text{TestMove}(\pi_{1X}, V_X, W_{|V_X}, u, \rho_{\pi^*_X}(u)) .$$

Following (11), the elements $u \in V_X^{\text{long}}$ can contribute at most

$$O\left(\varepsilon \sum_{u \in V_X^{\text{long}}} |\rho_{\pi_{1X}}(u) - \rho_{\pi^*_X}(u)|/\log n\right)$$

to T_X. Hence the total contribution from such elements is, by definition,

$$O(\varepsilon d_{\text{foot}}(\pi_{1X}, \pi_X^*)/\log n)$$

which is, using (4) at most $O(\varepsilon d_\tau(\pi_{1X}, \pi_X^*)/\log n)$. Using the triangle inequality and the definition of π_X^*, the last expression, in turn, is at most $O(\varepsilon C(\pi_{1X}, V_X, W_{|V_X})/\log n)$.

We now bound the contribution of the elements $u \in V_X^{\text{short}}$ to T_X. The probability of each such element to cross k is $O(|\rho_{\pi_{1X}}(u) - \rho_{\pi_X^*}(u)|/N_X)$. Hence, the total expected contribution of these elements to T_X is

$$O\left(\sum_{u \in V_X^{\text{short}}} |\rho_{\pi_{1X}}(u) - \rho_{\pi_X^*}(u)|^2/N_X\right) . \tag{12}$$

Under the constraints $\sum_{u \in V_X^{\text{short}}} |\rho_{\pi_{1X}}(u) - \rho_{\pi_X^*}(u)| \leq d_{\text{foot}}(\pi_{1X}, \pi_X^*)$ and $|\rho_{\pi_{1X}}(u) - \rho_{\pi_X^*}(u)| = O(\varepsilon N_X/\log n)$, the maximal value of (12) is

$$O(d_{\text{foot}}(\pi_{1X}, \pi_X^*)\varepsilon N_X/(N_X \log n)) = O(d_{\text{foot}}(\pi_{1X}, \pi_X^*)\varepsilon/\log n) .$$

Again using (4) and the triangle inequality, the last expression is

$$O(\varepsilon C(\pi_{1X}, V_X, W_{|V_X})/\log n) .$$

Combining the accounting for V^{long} and V^{short}, we conclude

$$E_{k_X}[T_X] = O(\varepsilon C(\pi_X^*, V_X, W_{|V_X})/\log n) , \tag{13}$$

where the expectation is over the choice of k_X in line 11 of SampleAndDecompose.

We are now in a position to use a key Lemma by Kenyon-Mathieu and Schudy (2007). First we need a definition: Consider the optimal solution π'_X respecting $V_{L[X]}, V_R[X]$ in lines 12 and 13. By this we mean that π'_X must rank all of the elements in V_{XL} before (to the left of) V_{RX}. For the sake of brevity, let C_X^* be shorthand for $C(\pi_X^*, V_X, W_{|V_X})$ and C'_X for $C(\pi_X^{\cdot}, V_X, W_{|V_X})$.

Lemma 22 *(Kenyon-Mathieu and Schudy, 2007) With respect to the distribution of the number k_X in line 11 of* SampleAndDecompose*,*

$$E[C'_X] \leq O\left(\frac{d_{\text{foot}}(\pi_{1X}, \pi_X^*)^{3/2}}{N_X}\right) + E[T_X] + C_X^* . \tag{14}$$

Using (4), we can replace $d_{\text{foot}}(\pi_{1X}, \pi_X^*)$ with $d_\tau(\pi_{1X}, \pi_X^*)$ in (14). Using the triangle inequality, we can then, in turn, replace $d_\tau(\pi_{1X}, \pi_X^*)$ with $C(\pi_{1X}, V_X, W_{|V_X})$.

3.6 Summing Over the Recursion Tree

Let us study the implication of (14) for our purpose. Recall that $\{V_1, \ldots, V_k\}$ is the decomposition returned by SampleAndRank, where each V_i corresponds to a leaf in the recursion tree. Also recall that $\hat{\pi}$ denotes the minimizer of $C(\cdot, V, W)$ over all permutations in $\Pi(V_1, \ldots, V_k)$ respecting the decomposition. Given Assumption 20 it suffices, for our purposes, to show that $\hat{\pi}$ is a (relative)

small approximation for MFAST on V, W. Our analysis of this account is basically that of Kenyon-Mathieu and Schudy (2007), with slight changes stemming from bounds we derive on $E[T_X]$. We present the proof in full detail for the sake of completeness. Let RT denote the root node.

For $X \in I$, let β_X denote the contribution of the split $L[X], R[X]$ to the LHS of (3). More precisely,

$$\beta_X = \sum_{u \in L[X], v \in R[X]} \mathbf{1}_{W(v,u)=1} ,$$

so we get $\sum_{1 \le i < j \le k} \sum_{(u,v) \in V_i \times V_j} \mathbf{1}_{W(v,u)=1} = \sum_{X \in I} \beta_X$.

For any $X \in I$, note also that by our definitions $\beta_X = C'_X - C^*_{L[X]} - C^*_{R[X]}$. Hence, using Lemma 22 and the ensuing comment,

$$E[\beta_X] \le O\left(E\left[\frac{C(\pi_{1X}, V_X, W_{|V_X})^{3/2}}{N_X} \right] \right) + E[T_X] + E[C^*_X] - E[C^*_{L[X]}] - E[C^*_{R[X]}] ,$$

where the expectations are over the entire space of random decisions made by the algorithm execution. Summing the last inequality over $X \in I$, we get (minding the cancellations):

$$E\left[\sum_{X \in I} \beta_X \right] \le O\left(\sum_{X \in I} E\left[\frac{C(\pi_{1X}, V_X, W_{|V_X})^{3/2}}{N_X} \right] \right) + E\left[\sum_{X \in I} T_X \right] + C^*_{\mathrm{RT}} - \sum_{X \in \mathcal{L}} E[C^*_X] . \tag{15}$$

The expression $E[\sum_{X \in I} T_X]$ is bounded by $O(E[\sum_{X \in I} \varepsilon \sum C^*_X / \log n])$ using (13) (which depends on Assumption 20). Clearly the sum of C^*_X for X ranging over nodes $X \in I$ in a particular level is at most $C(\pi_{\mathrm{RT}}, V, W)$ (again using Assumption 20 to assert that the cost of π_{1X} is less than the cost of π_X at each node X). By taking Assumption 11 into account, $C(\pi_{\mathrm{RT}}, V, W)$ is $O(C^*_{\mathrm{RT}})$. Hence, summing over all $O(\log n)$ levels,

$$E\left[\sum_{X \in I} T_X \right] = O(\varepsilon C^*_{\mathrm{RT}}) .$$

Let $C_{1X} = C(\pi_{1X}, V_X, W_{|V_X})$ for all $x \in I$. Denote by F the expression in the O-notation of the first summand in the RHS of (15), more precisely:

$$F = \sum_{X \in I} E\left[\frac{C_{1X}^{3/2}}{N_X} \right] , \tag{16}$$

where we remind the reader that $N_X = |V_X|$. It will suffice to show that under Assumption 20, the following inequality holds with probability 1:

$$G((C_{1X})_{X \in I}, (N_X)_{X \in I}) := \sum_{X \in I} C_{1X}^{3/2} / N_X \le c_3 \varepsilon C_{1\mathrm{RT}} , \tag{17}$$

where $c_3 > 0$ is some global constant. This turns out to require a bit of elementary calculus. A complete proof of this assertion is not included in the extended abstract of Kenyon-Mathieu and Schudy (2007). We present a version of the proof here for the sake of completeness.

Under assumption 20, the following two constraints hold uniformly for all $X \in I$ with probability 1: Letting $C_X = C(\pi_X, V_X, W_{|V_X})$,

$(A1)$ If X is other than RT, let Y be its sibling and P their parent. In case $Y \in I$:

$$C_{1X} + C_{1Y} \leq C_{1P} \ . \tag{18}$$

(In case $Y \in \mathcal{L}$, we simply have that $C_{1X} \leq C_{1P}$.[6]) To see this, notice that $C_{1X} \leq C_X$, and similarly, in case $Y \in I$, $C_{1Y} \leq C_Y$. Clearly $C_X + C_Y \leq C_{1P}$, because π_X, π_Y are simply restrictions of π_{1P} to disjoint blocks of V_P. The required inequality (18) is proven.

$(A2)$ $C_{1X} \leq c_2 \varepsilon^2 N_X^2$ for some global $c_2 > 0$.

In order to show (17), we may increase the values C_{1X} for $X \neq$ RT in the following manner: Start with the root node. If it has no children, there is nothing to do because then $G = 0$. If it has only one child $X \in I$, continuously increase C_{1X} until either $C_{1X} = C_{1\mathrm{RT}}$ (making $(A1)$ tight) or $C_{1X} = c_2 \varepsilon^2 N_X^2$ (making $(A2)$ above tight). Then recurse on the subtree rooted by X. In case RT has two children $X, Y \in I$ (say, X on left), continuously increase C_{1X} until either $C_{1X} + C_{1Y} = C_{1\mathrm{RT}}$ ($(A1)$ tight) or until $C_{1X} = c_2 \varepsilon^2 N_X^2$ ($(A2)$ tight) . Then do the same for C_{1Y}, namely, increase it until $(A1)$ is tight or until $C_{1Y} = c_2 \varepsilon^2 N_Y^2$ ($(A2)$ tight). Recursively perform the same procedure for the subtrees rooted by X, Y.

After performing the above procedure, let I_1 denote the set of internal nodes X for which $(A1)$ is tight, namely, either the sibling Y of X is a leaf and $C_{1X} = C_{1P}$ (where P is X's parent) or the sibling $Y \in I$ and $C_{1X} + C_{1Y} = C_{1P}$ (in which case also $Y \in I_1$). Let $I_2 = I \setminus I_1$. By our construction, for all $X \in I_2$, $C_{1X} = c_2 \varepsilon^2 N_X^2$.

Note that if $X \in I_2$ then its children (more precisely, those in I) cannot be in I_1. Indeed, this would violate $(A2)$ for at least one child, in virtue of the fact that N_Y lies in the range $[N_X/3, 2N_X/3]$ for any child Y of X. Hence, the set $I_1 \cup \{\mathrm{RT}\}$ forms a connected subtree which we denote by $\mathcal{T}_1$. Let $P \in \mathcal{T}_1$ be an internal node in $\mathcal{T}_1$. Assume it has one child in $\mathcal{T}_1$, call it X. Then $C_{1X} = C_{1P}$ and in virtue of $N_X \leq 2N_P/3$ we have $C_{1P}^{3/2}/N_P \leq (2/3)^{3/2} C_{1X}^{3/2}/N_X$. Now assume P has two children $X, Y \in \mathcal{T}_1$. Then $C_{1X} + C_{1Y} = C_{1P}$. Using elementary calculus, we also have that $C_{1P}^{3/2}/N_P \leq (C_{1X}^{3/2}/N_X + C_{1Y}^{3/2}/N_Y)/\sqrt{2}$ (indeed, the extreme case occurs for $N_X = N_Y = N_P/2$ and $C_{1X} = C_{1Y} = C_{1P}/2$). We conclude that for any P internal in $\mathcal{T}_1$, the corresponding contribution $C_{1P}^{3/2}/N_P$ to G is geometrically dominated by that of its children in I_1. Hence the entire sum $G_1 = \sum_{X \in I_1 \cup \{\mathrm{RT}\}} C_{1X}^{3/2}/N_X$ is bounded by $c_4 \sum_{X \in \mathcal{L}_1} C_{1X}^{3/2}/N_X$ for some constant c_4, where $\mathcal{L}_1$ is the set of leaves of $\mathcal{T}_1$. For each such leaf $X \in \mathcal{L}_1$, we have that $C_{1X}^{3/2}/N_X \leq c_2^{3/2} \varepsilon C_{1X}$ (using $(A2)$), hence $\sum_{X \in \mathcal{L}_1} C_{1X}^{3/2}/N_X \leq \sum_{X \in \mathcal{L}_1} c_2^{3/2} \varepsilon C_{1X} \leq c_2^{3/2} \varepsilon C_{1R}$ (the rightmost inequality in the chain follows from $\{V_X\}_{X \in \mathcal{L}_1}$ forming a disjoint cover of $V = V_{\mathrm{RT}}$, together with (A_1)). We conclude that $G_1 \leq c_4 c_2^{3/2} \varepsilon C_{1R}$.

To conclude (17), it remains to show that $G_2 = G - G_1 = \sum_{X \in I_2}$. For $P \in G_2$, clearly $C_{1P}^{3/2}/N_P = c_2^{3/2} \varepsilon^3 N_P^3$. Hence, if $X, Y \in G_2$ are children of P in I_2 then $C_{1P}^{3/2}/N_P \geq c_5 C_{1X}^{3/2}/N_X + C_{1Y}^{3/2}/N_Y$ and if X is the unique child of P in I_2, then $C_{1P}^{3/2}/N_P \geq c_5 C_{1X}^{3/2}/N_X$, for some global $c_5 > 1$. In other words, the contribution to G_2 corresponding to P geometrically dominates the sum of the corresponding contributions of its children. We conclude that G_2 is at most some constant c_6 times $\sum_{X \in \mathrm{root}(I_2)} C_{1X}^{3/2}/N_X$, where $\mathrm{root}(I_2)$ is the set of roots of the forest induced by I_2. As before, it is clear that $\{V_X\}_{X \in \mathrm{root}(I_2)}$ is a disjoint collection, hence as before we conclude that $G_2 \leq c_7 \varepsilon C_{1R}$ for some global $c_7 > 0$. The assertion (17) follows, and hence (16).

6. We can say something stronger in this case, but we won't need it here.

Plugging our bounds in (15), we conclude that

$$E\left[\sum_{X\in I}\beta_X\right] \leq C^*_{\mathrm{RT}}(1+O(\varepsilon)) - \sum_{X\in\mathcal{L}} E[C^*_X] \ .$$

Clearly $C(\hat{\pi},V,W) = \sum_{X\in I}\beta_X + \sum_{X\in\mathcal{L}} C^*_X$. Hence

$$E[C(\hat{\pi},V,W)] = (1+O(\varepsilon))C^*_{\mathrm{RT}} = (1+O(\varepsilon))C^* \ .$$

We conclude the desired assertion on expectation. Assumption 20, together with a simple counting of accesses to W gives our main result, Theorem 7, as a simple corollary. A simple counting of accesses to W proves Theorem 7.

4. Using Our Decomposition as a Preconditioner for SVM

We consider the following practical scenario, which is can be viewed as an improvement over a version of the well known SVMrank (Joachims, 2002; Herbrich et al., 2000) for the preference label scenario.

Consider the setting developed in Section 2.1, where each element u in V is endowed with a feature vector $\varphi(u) \in \mathbb{R}^d$ for some d (we can also use infinite dimensional spaces via kernels, but the effective dimension is never more than $n = |V|$). Assume, additionally, that $\|\phi(u)\|_2 \leq 1$ for all $u \in V$ (otherwise, normalize). Our hypothesis class $\mathcal{H}$ is parametrized by a weight vector $w \in \mathbb{R}^d$, and each associated permutation π_w is obtained by sorting the elements of V in decreasing order of a score given by $\mathrm{score}_w(u) = \langle\varphi(u), w\rangle$. In other words, $u \prec_{\pi_w} v$ if $\mathrm{score}_w(u) > \mathrm{score}_w(v)$ (in case of ties, assume any arbitrary tie breaking scheme).

The following SVM formulation is a convex relaxation for the problem of optimizing $C(h,V,W)$ over our chosen concept class $\mathcal{H}$:

$$\begin{array}{lrl}
\text{(SVM1)} & \text{minimize} & F_1(w,\xi) = \sum_{u,v}\xi_{u,v} \\
& \text{s.t. } \forall u,v : W(u,v) = 1 & \mathrm{score}_w(u) - \mathrm{score}_w(v) \geq 1 - \xi_{u,v} \\
& \forall u,v & \xi_{u,v} \geq 0 \\
& & \|w\| \leq c
\end{array}$$

Instead of optimizing (SVM1) directly, we make the following observation. An ε-good decomposition $V_1,\ldots,V_k$ gives rise to a surrogate learning problem over $\Pi(V_1,\ldots,V_k) \subseteq \Pi(V)$, such that optimizing over the restricted set does not compromise optimality over $\Pi(V)$ by more than a relative regret of ε (property (3)). In turn, optimizing over $\Pi(V_1,\ldots,V_k)$ can be done separately for each block V_i. A natural underlying SVM corresponding to this idea is captured as follows:

$$\begin{array}{lrl}
\text{(SVM2)} & \text{minimize} & F_2(w,\xi) = \sum_{u,v\in\Delta_1\cup\Delta_2}\xi_{u,v} \\
& \text{s.t. } \forall (u,v) \in \Delta_1\cup\Delta_2 & \mathrm{score}_w(u) - \mathrm{score}_w(v) \geq 1 - \xi_{u,v} \\
& \forall u,v & \xi_{u,v} \geq 0 \\
& & \|w\| \leq c \ ,
\end{array}$$

where $\Delta_1 = \bigcup_{1\le i<j\le k} V_i \times V_j$ and $\Delta_2 = \bigcup_{i=1}^{k} \{(u,v) : u,v \in V_i \wedge W(u,v) = 1\}$.

Abusing notation, for $w \in \mathbb{R}^d$ s.t. $\|w\| \le c$, let $F_1(w)$ denote $\min F_1(w,\xi)$, where the minimum is taken over all ξ that satisfy the constraints of SVM1. Observe that $F_1(w)$ is simply $F_1(w,\xi)$, where ξ is taken as:

$$\xi_{u,v} = \begin{cases} \max\{0, 1 - \text{score}_w(u) + \text{score}_w(v)\} & W(u,v) = 1 \\ 0 & \text{otherwise} \end{cases}.$$

Similarly define $F_2(w)$ as the minimizer of $F_2(w,\xi)$, which is obtained by setting:

$$\xi_{u,v} = \begin{cases} \max\{0, 1 - \text{score}_w(u) + \text{score}_w(v)\} & (u,v) \in \Delta_1 \cup \Delta_2 \\ 0 & \text{otherwise} \end{cases}. \tag{19}$$

Let π^* denote the optimal solution to MFAST on V, W.

We do not know how to directly relate the optimal solution to SVM1 and that of SVM2. However, we can can replace SVM2 with a careful sampling of constraints thereof, such that (i) the solution to the subsampled SVM is optimal to within a relative error of ε as a solution to SVM2, and (ii) the sampling is such that only $O(n\,\text{poly}(\log n, \varepsilon^{-1}))$ queries to W are necessary in order to construct it. This result, which we quantify in what follows, strongly relies on the local chaos property of the ε-good decomposition (2) and some combinatorics on permutations.

Our subsampled SVM which we denote by SVM3, is obtained as follows. For ease of notation we assume that all blocks $V_1, \ldots, V_k$ are big in V, otherwise a simple accounting of small blocks needs to be taken care of, adding notational clutter. Let Δ_3 be a subsample of size M (chosen shortly) of Δ_2, each element chosen uniformly at random from Δ_2 (with repetitions - hence Δ_3 is a multi-set). Define:

$$\begin{array}{lrl} \text{(SVM3)} & \text{minimize} & F_3(w,\xi) = \sum_{u,v\in\Delta_1} \xi_{u,v} + \frac{\sum_{i=1}^{k} \binom{n_i}{2}}{M} \sum_{u,v\in\Delta_3} \xi_{u,v} \\ & \text{s.t. } \forall (u,v) \in \Delta_1 \cup \Delta_3 & \text{score}_w(u) - \text{score}_w(v) \ge 1 - \xi_{u,v} \\ & \forall u,v & \xi_{u,v} \ge 0 \\ & & \|w\| \le c \end{array}$$

As before, define $F_3(w)$ to be $F_3(w,\xi)$, where $\xi = \xi(w)$ is the minimizer of $F_3(w,\cdot)$ and is taken as

$$\xi_{u,v} = \begin{cases} \max\{0, 1 - \text{score}_w(u) + \text{score}_w(v)\} & (u,v) \in \Delta_1 \cup \Delta_3 \\ 0 & \text{otherwise} \end{cases}.$$

Our ultimate goal is to show that for quite small M, SVM3 is a good approximation of SVM2. To that end we first need another lemma.

Lemma 23 *Any feasible solution (w,ξ) for SVM1 satisfies $\sum_{u,v} \xi_{u,v} \ge C(\pi^*, V, W)$.*

Proof The following has been proven by Ailon et al. (2008a): Consider *non-transitive* triangles induced by W: These are triplets (u,v,y) of elements in V such that $W(u,v) = W(v,y) = W(y,u) = 1$. Note that any permutation must disagree with at least one pair of elements contained in a non-transitive triangle. Let T denote the set of non-transitive triangles. Now consider an assignment of

non-negative weights β_t for each $t \in T$. We say that the weight system $\{\beta_t\}_{t\in T}$ *packs* T if for all $u,v \in V$ such that $W(u,v) = 1$, the sum $\sum_{(u,v) \text{ in } t} \beta_t$ is at most 1. (By u,v *in* t we mean that u,v are two of the three elements inducing t.) Let $\{\beta_t^*\}_{t\in T}$ be a weight system packing T with the maximum possible value of the sum of weights. Then

$$\sum_{t\in T} \beta_t^* \geq C(\pi^*, V, W)/3 \; . \tag{20}$$

Now consider one non-transitive triangle $t = (u,v,y) \in T$. We lower bound $\xi_{u,v} + \xi_{v,y} + \xi_{y,u}$ for any ξ such that w, ξ is a feasible solution to SVM1. Letting $a = \text{score}_w(u) - \text{score}_w(v), b = \text{score}_w(v) - \text{score}_w(y), c = \text{score}_w(y) - \text{score}_w(u)$, we get from the constraints in SVM1 that $\xi_{u,v} \geq 1-a, \xi_{v,y} \geq 1-b, \xi_{y,u} \geq 1-c$. But clearly $a+b+c = 0$, hence

$$\xi_{u,v} + \xi_{v,y} + \xi_{y,u} \geq 3 \; . \tag{21}$$

Now notice that the objective function of SVM1 can be bounded from below as follows:

$$\begin{aligned} \sum_{u,v} \xi_{u,v} &\geq \sum_{t=(u,v,y)\in T} \beta_t^* (\xi_{u,v} + \xi_{v,y} + \xi_{y,u}) \\ &\geq \sum_{t=(u,v,y)\in T} \beta_t^* \cdot 3 \\ &\geq C(\pi^*, V, W) \; . \end{aligned}$$

(The first inequality was due to the fact that $\{\beta_t^*\}_{t\in T}$ is a packing of the non-transitive triangles, hence the total weight corresponding to each pair u,v is at most 1. The second inequality is from (21) and the third is from (20).) This concludes the proof. ■

Theorem 24 *Let* $\varepsilon \in (0,1)$ *and* $M = O(\varepsilon^{-6}(1+2c)^2 d \log(1/\varepsilon))$. *Then with high constant probability, for all w such that* $\|w\| \leq c$,

$$|F_3(w) - F_2(w)| = O(\varepsilon F_2(w)) \; .$$

Proof Let $B_d(c) = \{z \in \mathbb{R}^d : \|z\| \leq c\}$. Fix a vector $w \in B_d(c)$. Over the random choice of Δ_3, it is clear that $E[F_3(w)] = F_2(w)$. We need a strong concentration bound. From the observation that $|\xi_{u,v}| \leq 1+2c$ for all u,v, we conclude (using Hoeffding bound) that for all $\mu > 0$,

$$\Pr[|F_3(w) - F_2(w)| \geq \mu] \leq \exp\left\{ \frac{-\mu^2 M}{\left(\sum_{i=1}^k \binom{n_i}{2}(1+2c)\right)^2} \right\} \; . \tag{22}$$

Let $\eta = \varepsilon^3$ and consider an η-net of vectors w in the ball $B_d(c)$. By this we mean a subset $\Gamma \subseteq B_d(c)$ such that for all $z \in B_d(c)$ there exists $w \in \Gamma$ s.t. $\|z-w\| \leq \eta$. Standard volumetric arguments imply that there exists such a set Γ of cardinality at most $(c/\eta)^d$.

Let $z \in \Gamma$ and $w \in B_d(c)$ such that $\|w-z\| \leq \eta$. From the definition of F_2, F_3, it is clear that

$$|F_2(w) - F_2(z)| \leq \sum_{i=1}^k \binom{n_i}{2} \varepsilon^3, \quad |F_3(w) - F_3(z)| \leq \sum_{i=1}^k \binom{n_i}{2} \varepsilon^3 \; . \tag{23}$$

Using (22), we conclude that for any $\mu > 0$, by taking $M = O(\mu^{-2}(\sum \binom{n_i}{2})^2(1+2c)^2 d\log(c\eta^{-1}))$, with constant probability over the choice of Δ_3, uniformly for all $z \in \Gamma$:

$$|F_3(z) - F_2(z)| \leq \mu \, .$$

Take $\mu = \varepsilon^3 \sum_{i=1}^{k} \binom{n_i}{2}$. We conclude (plugging in our choice of μ and the definition of η) that by choosing

$$M = O(\varepsilon^{-6}(1+2c)^2 d\log(c/\varepsilon)) \, ,$$

with constant probability, uniformly for all $z \in \Gamma$:

$$|F_3(z) - F_2(z)| \leq \varepsilon^3 \sum_{i=1}^{k} \binom{n_i}{2} \, .$$

Using (23) and the triangle inequality, we conclude that for all $w \in B_d(c)$,

$$|F_3(w) - F_2(w)| \leq 3\varepsilon^3 \sum_{i=1}^{k} \binom{n_i}{2} \, . \tag{24}$$

By property (2) of the ε-goodness definition, (24) implies

$$|F_3(w) - F_2(w)| \leq 3\varepsilon \min_{\pi \in \Pi(V)} \sum_{i=1}^{k} C(\pi_{|V_i}, V_i, W_{|V_i}) = 3\varepsilon \sum_{i=1}^{k} \min_{\sigma \in \Pi(V_i)} C(\sigma, V_i, W_{|V_i}) \, .$$

By Lemma 23 applied separately in each block V_i, this implies

$$|F_3(w) - F_2(w)| \leq 3\varepsilon \sum_{i=1}^{k} \sum_{u,v \in V_i} \xi_{u,v} = 3\varepsilon F_2(w),$$

(where $\xi = \xi(w)$ is as defined in (19).) This concludes the proof. ■

5. Limitations and Future Work

Optimality. The exponent of ε^{-6} in Theorem 7 seems rather high, and it would be interesting to improve it. A better dependence of ε^{-4} has been recently claimed by Ailon et al. (2011). It would be interesting to find the correct bound.

Practicality. Our bounds are asymptotic, and our work calls for experimentation in order to determine in which cases our sampling technique beats uniform sampling.

Searching in natural permutation subspaces. Algorithm 1, which leads to our main result Theorem 7, is heavily based on dividing and conquering. This is also the main limitation of this work. To understand this limitation, consider the scenario of Section 4. There, the practitioner searches in the limited space of *linearly induced permutations*, namely, permutations induced by a linear functional applied to the features endowing the elements in V. It is not hard to conceive a scenario in which our divide and conquer step constrains the algorithm to search in a region of permutations that does not intersect this restricted search space. This, in fact, was the reason for our inability to relate between SVM1 and SVM2 (and its subsampled counterpart, SVM3). There is nothing special about linearly

induced permutations, for this matter. In a recently studied scenario (Jamieson and Nowak, 2011), for example, one searches in the space of permutations induced by score functions computed as the distance from a fixed point from some metric space in which V is embedded. The same problem exists there as well: our sampling algorithm cannot be used to find almost optimal solutions within any restricted permutation subspace. Interestingly, the main result of Ailon et al. (2011), achieved while this work has been under review, has alleviated this problem using new techniques.

Acknowledgments

The author gratefully acknowledges the help of Warren Schudy with derivation of some of the bounds in this work. Special thanks to Ron Begleiter as well as to anonymous reviewers for helpful comments.

Appendix A. Linear VC Bound of Permutation Set

To see why the VC dimension of the set of permutations viewed as binary function over the set of all possible $\binom{n}{2}$ preferences, it is enough to show that any collection of n pairs of elements cannot be *shattered* by the set of permutation. (Refer to the definition of VC dimension by Vapnik and Chervonenkis (1971) for a definition of shattering). Indeed, any such collection must contain a cycle, and the set of permutations cannot direct a cycle cyclically.

Appendix B. Why The Disagreement Coefficient Does Not Help Here

We now show why a straightforward application of the disagreement coefficient (Hanneke, 2007) is not useful in our setting. The key idea of Hanneke (2007) is a definition of a measure of distance between concepts, equalling the volume of data points on which they disagree on. Using this measure, one then defines a ball $B_r(\pi)$ of radius r around a concept (a permutation) π, in an obvious way. The disagreement coefficient Θ is then defined as the smallest possible number bounding as a linear function Θr the volume of points on which the hypotheses in B_r are not unanimous on. Adopting this idea here, the underlying distance between hypotheses (permutations) is simply the Kendall-tau distance $d_\tau(\pi,\sigma)$ divided by $\binom{n}{2}$. We need to normalize this distance because Hanneke's work, as does most statistical machine learning work, assumes a probability measure on the space of instances (pairs of elements), while we used the counting measure for various reasons of simplicity. We define the normalized distance function as $\hat{d}_\tau(\pi,\sigma) = \binom{n}{2}^{-1} d_\tau(\pi,\sigma)$.

If we consider a ball $B_r(\pi)$ of radius $r > 2/n$ around some permutation π on V, then it is easy to see that there does not exist a pair of elements $u,v \in V$ on which $B_r(\pi)$ is unanimous on. Indeed, a simple swap of any two elements results in a permutation π' satisfying $\hat{d}_\tau(\pi,\pi') \leq 2/n$. This means that the disagreement coefficient, by definition, is it least $\Omega(n)$. Recall that the VC dimension of the space of permutations, viewed as $\binom{n}{2}$-dimensional binary preference vectors, is at most n. Plugging these bounds into the analysis of Hanneke (2007) of the famous A2 algorithm using the disagreement coefficient results in a sample complexity which is $\Omega(n^3)$ for any desired error rate. Clearly this is suboptimal because the number of pairs is only $O(n^2)$.

References

Nir Ailon and Mehryar Mohri. Preference based learning to rank. *Machine Learning*, 80:189–212, 2010.

Nir Ailon and Kira Radinsky. Ranking from pairs and triplets: Information quality, evaluation methods and query complexity. In *Proceedings of the 4th ACM International Conference on Web Search and Data Mining (WSDM)*, 2011.

Nir Ailon, Bernard Chazelle, Seshadhri Comandur, and Ding Liu. Estimating the distance to a monotone function. *Random Struct. Algorithms*, 31(3):371–383, 2007.

Nir Ailon, Moses Charikar, and Alantha Newman. Aggregating inconsistent information: Ranking and clustering. *J. ACM*, 55(5), 2008a.

Nir Ailon, Bernard Chazelle, Seshadhri Comandur, and Ding Liu. Property-preserving data reconstruction. *Algorithmica*, 51(2):160–182, 2008b.

Nir Ailon, Ron Begleiter, and Esther Ezra. A new active learning scheme with applications to learning to rank from pairwise preferences. *arxiv:1110.2136*, 2011.

Miklos Ajtai, Vitaly Feldman, Avinatan Hassidim, and Jelani Nelson. Sorting and selection with imprecise comparisons. In *Automata, Languages and Programming*, volume 5555 of *Lecture Notes in Computer Science*, pages 37–48. Springer Berlin / Heidelberg, 2009.

Noga Alon. Ranking tournaments. *SIAM J. Discret. Math.*, 20(1):137–142, 2006.

Dana Angluin. Queries revisited. *Theor. Comput. Sci.*, 313(2):175–194, 2004.

Les Atlas, David Cohn, Richard Ladner, Mohamed A. El-Sharkawi, and Robert J. Marks. Training connectionist networks with queries and selective sampling. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, pages 566–573, 1990.

Les Atlas, David Cohn, and Richard Ladner. Improving generalization with active learning. *Machine Learning*, 15(2):201–221, 1994.

Maria-Florina Balcan, Nikhil Bansal, Alina Beygelzimer, Don Coppersmith, John Langford, and Gregory B. Sorkin. Robust reductions from ranking to classification. *Machine Learning*, 72 (1-2):139–153, 2008.

Maria-Florina Balcan, Alina Beygelzimer, and John Langford. Agnostic active learning. *J. Comput. Syst. Sci.*, 75(1):78–89, 2009.

Maria-Florina Balcan, Steve Hanneke, and Jennifer Vaughan. The true sample complexity of active learning. *Machine Learning*, 80:111–139, 2010.

Yoram Baram, Ran El-Yaniv, and Kobi Luz. Online choice of active learning algorithms. *Journal of Machine Learning Research*, 5:255–291, 2004.

Ron Begleiter, Ran El-Yaniv, and Dmitry Pechyony. Repairing self-confident active-transductive learners using systematic exploration. *Pattern Recognition Letters*, 29(9):1245–1251, 2008.

Mark Braverman and Elchanan Mossel. Noisy sorting without resampling. In *Proceedings of the nineteenth Annual ACM-SIAM Symposium on Discrete algorithms (SODA)*, pages 268–276, Philadelphia, PA, USA, 2008.

Ben Carterette, Paul N. Bennett, David Maxwell Chickering, and Susan T. Here or there: Preference judgments for relevance. In *Proceedings of the 30th European Conference on Information Retrieval (ECIR)*, pages 16–27, 2008.

William W. Cohen, Robert E. Schapire, and Yoram Singer. Learning to order things. In *Proceedings of the 10th Conference on Advances in Neural Information Processing Systems (NIPS)*, pages 451–457, 1998.

Koby Crammer and Yoram Singer. Pranking with ranking. In *Proceedings of the 14th Conference on Advances in Neural Information Processing Systems (NIPS)*, pages 641–647, 2001.

Aron Culotta and Andrew McCallum. Reducing labeling effort for structured prediction tasks. In *Proceedings of the 20th Conference on Artificial Intelligence (AAAI)*, pages 746–751, 2005.

Sanjoy Dasgupta. Coarse sample complexity bounds for active learning. In *Proceedings of the 18th Conference on Advances in Neural Information Processing Systems (NIPS)*, pages 235–242, 2005.

Sanjoy Dasgupta, Daniel Hsu, and Claire Monteleoni. A general agnostic active learning algorithm. In *Proceedings of the 21st Conference on Advances in Neural Information Processing Systems (NIPS)*, 2007.

Sanjoy Dasgupta, Adam Tauman Kalai, and Claire Monteleoni. Analysis of perceptron-based active learning. *Journal of Machine Learning Research*, 10:281–299, 2009.

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

Irit Dinur and Shmuel Safra. On the importance of being biased. In *Proceedings of the 34th Annual Symposium on the Theory of Computing (STOC)*, pages 33–42, 2002.

Ran El-Yaniv and Yair Wiener. On the foundations of noise-free selective classification. *J. Machine Learning Research*, 11:1605–1641, 2010.

Uri Feige, David Peleg, Prabhakar Raghavan, and Eli Upfal. Computing with unreliable information. In *Proceedings of the 22nd Annual Symposium on the Theory of Computing (STOC)*, pages 128–137, 2002.

Shai Fine, Ran Gilad-Bachrach, and Eli Shamir. Query by committee, linear separation and random walks. *Theoretical Computer Science*, 284(1):25–51, 2002.

Yoav Freund, H. Sebastian Seung, Eli Shamir, and Naftali Tishby. Selective sampling using the query by committee algorithm. *Mach. Learn.*, 28(2-3):133–168, 1997.

Eric J. Friedman. Active learning for smooth problems. *In Proceedings of the* 22^{nd} *Annual Conference on Learning Theory (COLT)*, 2009.

Shirley Halevy and Eyal Kushilevitz. Distribution-free property-testing. *SIAM J. Comput.*, 37(4): 1107–1138, 2007.

Steve Hanneke. A bound on the label complexity of agnostic active learning. In *Proceedings of the 24th International Conference on Machine Learning (ICML)*, pages 353–360, 2007.

Ralf Herbrich, Thore Graepel, and Klaus Obermayer. *Advances in Large Margin Classifiers*, chapter 7 (Large Margin Rank Boundaries for Ordinal Regression), pages 115–132. MIT Press, 2000.

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

Kevin G. Jamieson and Robert D. Nowak. Active ranking using pairwise comparisons. In *Proceedings of the 25th Conference on Advances in Neural Information Processing Systems (NIPS)*, 2011.

Thorsten Joachims. Optimizing search engines using clickthrough data. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 133–142, 2002.

Richard M. Karp. Reducibility among combinatorial problems. In *Complexity of Computer Computations*, pages 85–104. Plenum Press, New York, 1972.

Claire Kenyon-Mathieu and Warren Schudy. How to rank with few errors. In *Proceedings of the 39th Annual Symposium on the Theory of Computing (STOC)*, pages 95–103, 2007.

Michael Lindenbaum, Shaul Markovitch, and Dmitry Rusakov. Selective sampling for nearest neighbor classifiers. *Machine Learning*, 54:125–152, 2004.

Dan Roth and Kevin Small. Margin-based active learning for structured output spaces. In *Proceedings of the European Conference on Machine Learning (ECML)*, pages 413–424, 2006.

Vladimir N. Vapnik and Alexey Ya. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, 16(2):264–280, 1971.

Kai Yu, Jinbo Bi, and Volker Tresp. Active learning via transductive experimental design. In *Proceedings of the 23rd International Conference on Machine Learning (ICML)*, pages 1081–1088, 2006.