Approximate Sorting of Preference Data

Ludwig M. Busse **Morteza Haghir Chehreghani**

Joachim M. Buhmann
Department of Computer Science
ETH Zurich
8092 Zurich, Switzerland
{ludwig.busse, morteza.chehreghani, jbuhmann}@inf.ethz.ch

Abstract

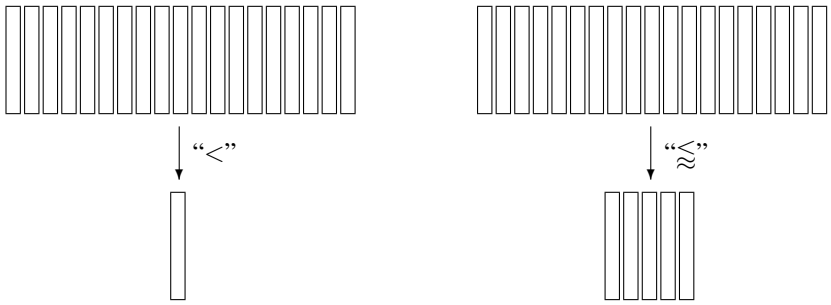
We consider sorting data in noisy conditions. Whereas sorting itself is a well studied topic, ordering items when the comparisons between objects can suffer from noise is a rarely addressed question. However, the capability of handling noisy sorting can be of a prominent importance, in particular in applications such as preference analysis. Here, orderings represent consumer preferences ("rankings") that should be reliably computed despite the fact that individual, simple pairwise comparisons may fail.
This paper derives an information theoretic method for approximate sorting. It is optimal in the sense that it extracts as much information as possible from the given observed comparison data conditioned on the noise present in the data. The method is founded on the maximum approximation capacity principle [2, 3]. All formulas are provided together with experimental evidence demonstrating the validity of the new method and its superior rank prediction capability.

1 Introduction

Sorting items into ascending or descending order defines a fundamental topic studied in the context of algorithm design. This topic arises frequently in many real-world problems. General-purpose sorting routines are well studied over the past 100 years. A very good and in-depth overview of sorting algorithms is provided by [1], starting with an elementary discussion on permutations. In fact, sorting technologies can be understood as mechanisms that operate on the symmetric group $\mathbb{S}_n$ (the space of all permutations over n items = rankings/orderings). At the beginning of a sorting procedure, every element of $\mathbb{S}_n$ (each ordering) is equally likely. With increasing run-time, the sorting algorithm gathers more and more information, i.e. paired comparisons are sequentially queried and provided by an oracle. Thereby, the set of possible orderings is reduced. Finally the algorithm outputs one ordering that is (maximally) compatible with the observed pairwise comparisons. In other words, sorting algorithms aim at constructing a set of partial orders, reducing its cardinality so as to ultimately end up with a single total order indicating the sorting result (see figure 1(a)).

There are a lot of different sorting algorithms proposed in the literature. Sorting approaches can be schematically classified into sorting by insertion, exchanging, selection, and merging [1]. Technically, they differ in the order that the pairwise comparisons are queried from the oracle. The various sorting techniques offer different computational complexities (typically either $O(n^2)$ or $O(n \log n)$), but also different level of robustness to errors that can occur during the sort.

We propose a new method to *optimally sort data in noisy conditions*. In this setting, the pairwise comparison provided by the oracle might be contaminated by noise, i.e. the algorithm might receive



(a) paired comparison-based sorting (b) approximate sorting

Figure 1: A paired comparison-based sorting algorithm reduces the number possible orderings until the final order is obtained (Figure 1(a)). As it is illustrated in Figure 1(b), when noise is involved in the pairwise comparisons, sorting from a data source accounts for computing an approximation set of orders (where applicable, the orders are weighted). The more noise in the data, the larger the approximation set will be.

an inverted result for its query "$<$". Several generative scenarios are imaginable: (i.) The items are very similar to each other. Therefore, their relative order is confused. (ii.) Features are underlying the items, and comparing items involves sampling from the feature distribution. (iii.) When eliciting preferences of users or customers, people state wrong comparisons because they are unsure, indifferent, tired, or unaware of consistent decision dimensions. (iv.) In sports games (tournaments), it is natural to observe no absolute order of the teams' performances.

Our approach takes such noise into account by not commiting to a single final order, but rather determining a *set* of orderings that exhibit statistically equivalent (or weighted) approximate sortings of the items (see figure 1(b)). The output of our algorithm hence is a set of orderings, called the *approximation set*. Its cardinality is a function of the noise level in the data. In the limit of vanishing noise, we can shrink the approximation set down to one element (which constitutes the special case of standard sorting). We wish to automatically adjust the cardinality of the approximation set based on the noise type in the data in an effort as to extract as much reliable partial order information as possible without being adversely affected by noise. Thereby, there is a tradeoff between informativeness (favors a small approximation set) and stability (advocates a large approximation set).

In order to optimally balance this tradeoff, we rely on an information theoretic model validation principle called *Approximation Set Coding* (ASC) [2, 3]. The principle benefits from a generic set-based coding and communication scenario. However, there exists a conceptual analogy between communication and learning: For communication, one demands a high rate (a large amount of information transferred per channel use) together with a decoding rule that is stable under the perturbations of the messages by the noise in the channel [4]. In this paper, we formulate sorting as a communication problem by setting up a protocol that uses sorting solutions as codewords. Optimal communication then prefers total orders which imply maximal partial order information but still are stable under the noise. We demonstrate how to transform a sorting task to a communication setting and how to compute the *capacity* of sorting solutions.

1.1 Relevant work

Sorting in noisy environments, despite its fundamental importance, is rarely discussed in the literature. [5] derives a lower bound on the number of comparisons required to approximate an arbitrary ranking within a certain expected (Spearman's footrule) distance. In [6], the robustness of sorting and tournament algorithms against faulty comparisons is analyzed - asserting that in general, there exists a tradeoff between the number of comparisons and the accuracy of the result. The work of [7] examines ranking a collection of objects using pairwise comparisons, where the goal is to minimize

the number of comparisons (by exploiting an assumed embeddability of the items into an Euclidean space).

1.2 Application to preference learning

A set of items can be characterized in the most elementary form by a preference relation. Such pairwise comparisons, that yield so-called paired comparison data, encode the preferences of items in many different contexts. Comparing two items with an operator $<$, i.e. measuring whether an item is bigger, higher, more preferred, ... than the other item endows an otherwise unstructured pair of items with a very elementary piece of information (1-bit). The data comes from paired comparison experiments: Many situations naturally produce pairwise comparisons such as sporting events which involve two teams (e.g. football, basketball). Then, the record of wins and losses for the teams constitute the data. In other situations, such as food tasting, pairwise comparisons are helpful due to the difficulty of distinguishing preferences when more than two items are considered simultaneously. An approach to preference modeling is the use of rankings, where e.g. items, values, products or websites are ordered according to their degree of preference or importance. Although direct rankings are popular e.g. in music, movies and food, giving a ranking of more than, say, 5 items poses quite a difficult and time-consuming challenge for an interviewee to complete. Deciding between just 2 items at a time is easier than immediately eliciting complete rankings and thus generates data of superior quality. Thereby, pairwise preference data arises often naturally or implicitly, e.g. click data. With many items up for consideration (e.g. products or websites), one should expect missing or noisy entries in the reported pairwise preference data. Sorting then refers to learning the complete preference structure, i.e. the reconstruction of the total order from pairwise observed and potentially error-prone preferences.

2 Approximate Sorting

2.1 Problem formulation

Let n be the number of items to be sorted. From a mathematical point of view, ordering is a permutation denoted by π. In an optimization context, the problem of sorting then technically rephrases as: Among all permutations π, find the one that orders the items best. In learning terminology we say that, consequently, the hypothesis class is the symmetric group of order n, i.e. $\mathcal{H} = \{\pi \in \mathbb{S}_n\}$.

The pairwise comparisons (the actual data) that a sorting algorithm can query/has access to are provided in a pairwise comparison matrix $\mathbf{X} = (x_{ij}) \in \{0,1\}^{n\times n}$, $i,j = 1,\dots,n$. In order to stay within the standard conditions of sorting, it is assumed that multiple queries to the same pairwise comparison are not possible. A value of $x_{ij} = 1$ means that item i is larger than/preferred over item j. W.r.t. the ranking-based notation, $\mathbb{I}[\pi_i > \pi_j]$ indicates that the rank of item i (as induced by the ranking π) is higher than the rank of item j, meaning that item i is smaller/less preferred (i.e. please note that a ranking contains the ranks of items, not their order!). We can now write down a cost function, which specifies good sorting solution

$$\mathcal{R}^{\text{Sorting}}(\pi|\mathbf{X}) = \sum_{i,j} x_{ij}\mathbb{I}[\pi_i > \pi_j] \tag{1}$$

This cost function counts the number of disagreements between the *provided* pairwise comparisons $\{x_{ij}\}$ and the pairwise comparisons $\mathbb{I}[\pi_i > \pi_j]$ induced by a proposal ranking π from the hypothesis class of orderings $\mathbb{S}_n$. It optimizes for a ranking that exhibits the least number of contradictions between the ordering and the pairwise comparisons.
For inference, we will have to evaluate sums over the hypothesis space (partition sums) such as

$$Z \;=\; \sum_{\pi\in\mathbb{S}_n} \exp\Big(-\beta(\sum_{i,j} x_{ij}\mathbb{I}[\pi_i > \pi_j])\Big) \;=\; \sum_{\pi\in\mathbb{S}_n}\prod_{i,j} \exp\Big(-\beta(x_{ij}\mathbb{I}[\pi_i > \pi_j])\Big), \tag{2}$$

which is intractable for large n due to its non-factored nature. Also, item-rank assignments (the assignments of items to ranks) are statistically dependent in this non-factored model.

2.2 Mean-field equations for Sorting

However, we can set up a mean-field approximation which approximates *average* item-rank assignments $q_{ik} = \mathbb{E}\{\mathbb{I}[\pi_i = k]\}$ with other assignment variables by a *mean-field* $\mathcal{E}_{ik}$. Within an $\mathcal{E}$-parametrized family of factorial distributions $\mathbf{Q}(\mathcal{E})$, the most similar one is determined by minimizing the Kullback-Leibler divergence as

$$\begin{aligned} D_{\mathrm{KL}}(\mathbf{Q}||\mathbf{P}^{\text{Sorting}}) &= \sum_{\pi\in\mathbb{S}_n} \mathbf{Q}\log\frac{\mathbf{Q}}{\exp(-\beta(\mathcal{R}^{\text{Sorting}} - \mathcal{F}^{\text{Sorting}}))} \\ &= \sum_{i=1}^{n}\sum_{k=1}^{n} q_{ik}\log q_{ik} + \beta\mathbb{E}_{\mathbf{Q}}\{\mathcal{R}^{\text{Sorting}}\} - \beta\mathcal{F}^{\text{Sorting}} \end{aligned} \tag{3}$$

An upper bound of the free energy $\mathcal{F}^{\text{Sorting}}$ is given by the non-negativity of the KL-divergence. Upper bound minimization is performed under the constraint $\sum_{k=1}^{n} q_{ik} = 1, \forall i$, yielding

$$\begin{aligned} 0 &= \frac{\partial}{\partial q_{ik}}\sum_{j=1}^{n}\sum_{k=1}^{n} q_{jk}\log q_{jk} + \beta\mathbb{E}_{\mathbf{Q}}\{\mathcal{R}^{\text{Sorting}}\} + \sum_{j=1}^{n}\lambda_j(\sum_{k=1}^{n} q_{jk} - 1) \\ &= \sum_{\pi\in\mathbb{S}_n}\prod_{j\le n: j\neq i} q_{j,\pi_j}\mathbb{I}[\pi_i = k]\mathcal{R}^{\text{Sorting}} + \frac{1}{\beta}(\log q_{ik} + 1) + \lambda_i \end{aligned} \tag{4}$$

Subject to the constraint of assigning item i to rank k, the expectation over all assignments is $\mathbb{E}_{\mathbf{Q}_{i\to k}}\{\mathcal{R}^{\text{Sorting}}\}$. For the extremum of the bound, the necessary condition determines the mean-field assignments

$$q_{ik} = \frac{\exp(-\beta\mathcal{E}_{ik})}{\sum_{k'}\exp(-\beta\mathcal{E}_{ik'})}, \quad \text{with} \quad \mathcal{E}_{ik} = \mathbb{E}_{\mathbf{Q}_{i\to k}}\{\mathcal{R}^{\text{Sorting}}\}. \tag{5}$$

To calculate the mean-fields, $\mathcal{R}^{\text{Sorting}}$ is decomposed into contributions which depend on item i and on the costs of all other items. Each q_{ik} is influenced uniquely by the part which depends on the item i. For sorting, meanfield approximation yields the parameters

$$\mathcal{E}_{ik} = \sum_{j\neq i}\left((x_{ij}(\sum_{k'=1}^{k+1} q_{jk'})) + (x_{ji}(\sum_{k'=k+1}^{n} q_{jk'}))\right) + \texttt{const}\,. \tag{6}$$

An iterative EM-like procedure computes the mean-fields and the probabilities by mutual conditioning. The t^{th} iteration of the algorithm consists of two main steps. First, $q_{ik}^{(t)}$ is estimated as a function of $\mathcal{E}_{ik}^{(t-1)}$ (the q_{ik}'s can be used to determine the ranking by placing item i on rank $\arg\max_k\ q_{ik}$). Second, $\mathcal{E}_{ik}^{(t)}$ is calculated for given $q_{ik}^{(t)}$.

Thereby, given the mean-fields and the assignment probabilities, an equivalent cost function for the sorting problem can now be written as

$$\mathcal{R}^{\text{Sorting}}(M|\mathcal{E}) = \sum_{i=1}^{n}\sum_{k=1}^{n} M_{ik}\mathcal{E}_{ik}, \tag{7}$$

where rankings are now represented via the item-rank assignment matrix M ($M_{ik} = 1$ iff item i is on rank k, $M_{ik} = 0$ otherwise, i.e. $M_{ik} = \mathbb{I}[\pi_i = k]$), and the mean-fields $\mathcal{E}_{ik}$ have been computed from the data (cf. with equation 1). In this factorial model, partition functions can be calculated in a computationally efficient way:

$$\begin{aligned} Z &= \sum_{M}\exp\left(-\beta(\sum_{i=1}^{n}\sum_{k=1}^{n} M_{ik}\mathcal{E}_{ik})\right) \\ &= \sum_{M}\prod_{i=1}^{n}\exp\left(-\beta(M_{ik}\mathcal{E}_{ik})\right) = \prod_{i=1}^{n}\sum_{k=1}^{n}\exp\left(-\beta\mathcal{E}_{ik}\right) \end{aligned} \tag{8}$$

With a cost function that captures the problem at hand (here: sorting), and being able to compute the associated partition sum over the hypothesis space computationally efficient, we now have all ingredients to turn to the problem of approximate sorting.

2.3 The principle of Approximation Set Coding for Sorting Problems

The sorting model is characterized by its cost function $\mathcal{R}^{\text{Sorting}}(\pi|\mathbf{X})$, where the solution (order of the items) is captured in π.
Let $\mathbf{X}^{(m)}, m \in \{1, 2\}$ be two datasets from the same total order, but with different noise instantiations. In most cases, their empirically minimal total orders are different due to different noise realizations. Hence, sorting the data by determining the empirical optimal total order lacks robustness. The Approximation Set Coding (ASC) framework [2, 3] suggests to rank all sorting solutions by *approximation weights* $w(\pi, \mathbf{X})$. ASC uses the family of Boltzmann weights $w(\pi, \mathbf{X}) := \exp(-\beta\mathcal{R}(\pi, \mathbf{X}))$, parameterized by the inverse computational temperature β. Thereby two weight sums are defined

$$Z^{(m)} := Z(\mathbf{X}^{(m)}) = \sum_{\pi\in\mathbb{S}_n} \exp\Big(-\beta\mathcal{R}(\pi, \mathbf{X}^{(m)})\Big), \qquad m = 1, 2 \tag{9}$$

and a joint weight sum is defined as

$$Z^{(12)} = \sum_{\pi\in\mathbb{S}_n} \exp\Big(-\beta\Big(\mathcal{R}(\pi, \mathbf{X}^{(1)}) + \mathcal{R}(\pi, \mathbf{X}^{(2)})\Big)\Big) \tag{10}$$

Essentially, $Z^{(m)}$ defines the resolution of the solution space. Very large β renders selecting the empirical minimizer, which lacks robustness. Smaller β indicates involving more solutions which yields to a coarser resolution of the solution space, i.e. a more stable solution set. Therefore, the key problem of learning is to find the optimal resolution, i.e. the optimal tradeoff between informativeness and stability. To answer this question, the ASC principle requires to optimize the mutual information $\mathcal{I}_\beta$ defined as

$$\mathcal{I}_\beta(\mathbf{X}^{(1)}, \mathbf{X}^{(2)}) = \frac{1}{n}\log(\frac{|\{\sigma\}|Z^{12}}{Z^1 \cdot Z^2}). \tag{11}$$

The cardinality $|\{\sigma\}|$ is the number of all distinct sorting solutions on $\mathbf{X}$, i.e. the entropy of the type of the empirical minimizer $\pi^\perp(\mathbf{X}^{(1)})$. The maximum of $\mathcal{I}_\beta$ with respect to β is called approximation capacity. It determines the best tradeoff between stability and informativeness. Using these entities, sorting solutions π are ranked according to their approximation capacity. The ordering π with the maximal capacity is chosen as superior one.

2.4 Approximation Set Coding for Sorting

Consider a sorting task of n given items. The potential $\mathcal{E}_{ik}$ (as derived in section 2.2) indicates the cost of placing item i on rank k. In our sorting problem, the potentials $\mathcal{E}_{ik}, 1 \le i \le n, 1 \le k \le n$ are provided from the mean-field (see section 2.2). Given the potentials, approximation capacity is computed by (i.) calculating the weight sums $Z^{(m)}, m = 1, 2$, and the joint weight sum $Z^{(12)}$ and (ii.) maximizing $\mathcal{I}_\beta$ (Eq. 11) with respect to β.
The weight sums are

$$\begin{aligned} Z^{(m)} &= \sum_{\pi\in\mathbb{S}_n} \exp\Big(-\beta\sum_{i=1}^{n}\sum_{k=1}^{n} M_{ik}\mathcal{E}_{ik}^{(m)}\Big) \\ &= \prod_{i=1}^{n}\sum_{k=1}^{n} \exp\Big(-\beta\mathcal{E}_{ik}^{(m)}\Big) \end{aligned} \tag{12}$$

M with $M_{ik} \in \{0, 1\}$ and $\sum_k M_{ik} = 1, \forall i$ denotes the item-rank assignment matrix which encodes the sorting solution π.
Similarly, the joint weight is calculated by

$$\begin{aligned} Z^{(12)} &= \sum_{\pi\in\mathbb{S}_n} \exp\Big(-\beta\sum_{i=1}^{n}\sum_{k=1}^{n} M_{ik}(\mathcal{E}_{ik}^{(1)} + \mathcal{E}_{ik}^{(2)})\Big) \\ &= \prod_{i=1}^{n}\sum_{k=1}^{n} \exp\Big(-\beta(\mathcal{E}_{ik}^{(1)} + \mathcal{E}_{ik}^{(2)})\Big) \end{aligned} \tag{13}$$

Thus, the mutual information $\mathcal{I}_\beta$ yields

$$\mathcal{I}_\beta = \frac{1}{n}\log|\{\sigma\}| + \frac{1}{n}\sum_{i=1}^{n}\log\sum_{k=1}^{n}\exp\Big(-\beta(\mathcal{E}_{ik}^{(1)} + \mathcal{E}_{ik}^{(2)})\Big) \\ -\frac{1}{n}\sum_{i=1}^{n}\log\Big(\sum_{k=1}^{n}\exp(-\beta\mathcal{E}_{ik}^{(1)})\sum_{k'=1}^{n}\exp(-\beta\mathcal{E}_{ik'}^{(2)})\Big). \tag{14}$$

The approximation capacity is defined as the maximum of the mutual information $\mathcal{I}_\beta$ over β, providing us with the corresponding β_{optimal}. With this technology at hand, data drawn from a noisy pairwise comparison source can be sorted. The sorting results are sampled from the Gibbs distribution associated with the cost function given at the temperature $T_{\text{optimal}} = 1/\beta_{\text{optimal}}$). They optimally balance the tradeoff between extracting as much reliable order information as possible from the data, while at the same time being maximally robust against the noise type actually present in the data.

2.5 Experimental validation

First experimental results demonstrate that the new approximate sorting technique is capable of sensibly sorting noisy data. Experiments were performed for $n = 50$ items with varying noise levels. For generating the synthetic data, a simple noise model was assumed: the order between items which are close to each other is confused with the noise fraction specified in the respective experiments, whereas comparisons between distant items (e.g. between the first and the last item in the correct order) are corrupted with vanishing probability. Flip probabilities are interpolated between these extremes. This is a basic noise mechanism.

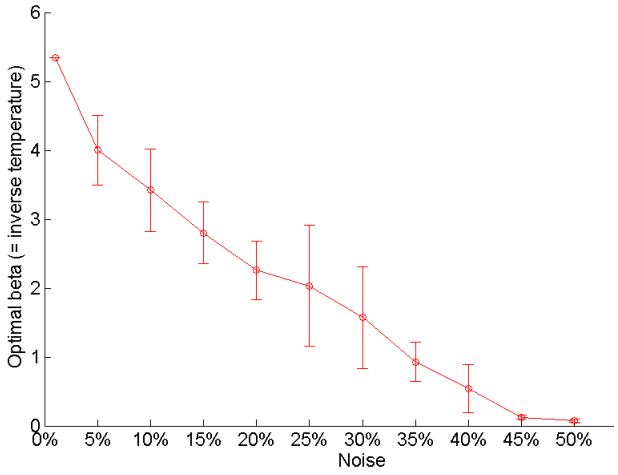
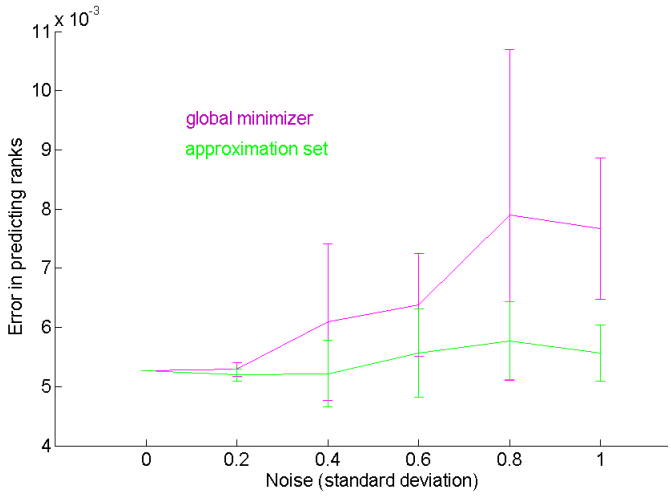


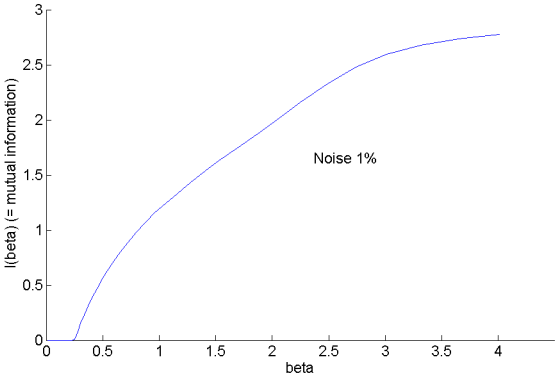
Figure 2: The resolution, at which we can look at orderings (as reflected in β), decreases with noise.

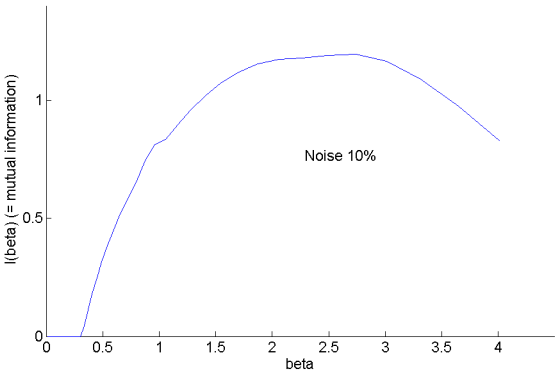
Figure 2 demonstrates that β_{optimal} (as inverse computational temperature) decreases with an increase in the noise level. The more noise in the data, the higher the temperature must be at which we analyse the data (the less order information we can extract). With little noise, on contrary, one is able to extract much structure (high beta/low temperature).
The following diagnostic plots visualize the mutual information $\mathcal{I}_\beta$ versus β for noise contingents of 1%, 10%, and 25%, respectively. We can see that the mutual information strictly increases with β (in a setting with as little noise as 1%), then, for increasing noise (10% and 25%) its maximum shifts to the left and less mutual information is found, declining again after β_{optimal} is passed. For data highly contaminated with noise, the mutual information decreases for too high choices of β (too low temperature, resolution at which we look at the sortings is inadequately high given the noise).

We then investigated the prediction performance of our approximation set in comparison with the global minimizer solution. To check the ability of recovering ranks, pairwise comparisons were impurified with noise by comparing two random draws from Normal distributions centered around the true ranks of the items under comparison (i.e. $\tilde{\pi}_i \sim \mathcal{N}(\mu = \pi_i, \sigma^2)$ and $\tilde{\pi}_j \sim \mathcal{N}(\mu = \pi_j, \sigma^2)$, $x_{ij} = \mathbb{I}[\tilde{\pi}_i < \tilde{\pi}_j]$. In figure 3, the standard deviation σ adjusting the noise was increased from 0 to 1. For a fair comparison, the global minimizer was computed over both datasets $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(2)}$, i.e. the ranking π which minimizes $\mathcal{R}^{\text{Sorting}}(\pi|\mathbf{X}^{(1)}, \mathbf{X}^{(2)}) = \mathcal{R}^{\text{Sorting}}(\pi|\mathbf{X}^{(1)}) + \mathcal{R}^{\text{Sorting}}(\pi|\mathbf{X}^{(2)})$ was chosen for predicting ranks. The approximation set was fully automatically determined as proposed in this article. As we can see in figure 3, rank predictions based on approximate sorting are substantially better than considering only the global minimizer (which corresponds to cooling down to zero temperature). The proposed method is particularly helpful when there is a portion of noise in the data (as we can expect in many real-world situations, e.g. soccer matches). In the noise free case, both approaches naturally perform the same.

In an analysis of real-world data, we exploited the superior rank prediction capability of the new method to determine both robust and informative ranks of German soccer teams based on records of wins and losses.







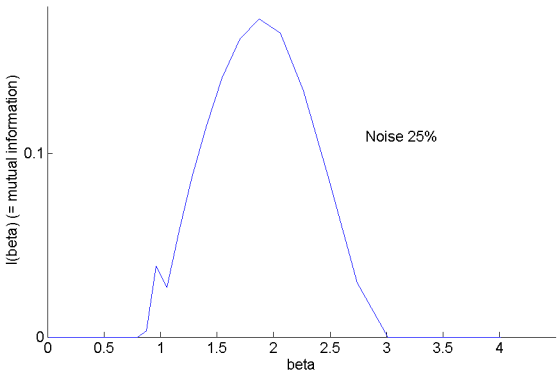


Figure 3: The error in predicting ranks versus noise present in the data. This result promises that predictions based on the approximation set outperform the global minimizer substantially! The superiority in particular is more pronounced when more noise is involved. In the noise free case, both cases fall together.

3 Outlook

D. E. Knuth [1] believes that *"virtually every important aspect of programming arises somewhere in the context of sorting or searching"*. This contribution considers sorting items under noisy conditions. An optimal sorting procedure based on information-theoretic model validation was derived so as to extract as much reliable order information as possible from the data. Our approach also exemplifies how Approximation Set Coding is employed to extract both informative and robust structure in a context sensitive way. The search for optimal structures is guided by a problem dependent cost function.

Future work includes the design of new sorting algorithms that actively choose paired comparisons adaptively to the data (and their noise types) with the goal of minimizing query costs while maximizing information content and robustness. Such studies will also shed new light on the relationship between computational complexity and statistical complexity. The hypothesis that robust solutions can be efficiently found [8] can also be tested in the context of approximate sorting.

Acknowledgments

We thank Mario Frank for many helpful discussions on ASC theory.

References

[1] D. E. Knuth, *The Art of Computer Programming, VOLUME 3, Sorting and Searching*, 2nd ed., Addison-Wesley, 1998.

[2] J. M. Buhmann, *Information theoretic model validation for clustering*, ISIT'10. IEEE, 2010.

[3] J. M. Buhmann, *Context Sensitive Information: Model Validation by Information Theory*, Pattern Recognition, Lecture Notes in Computer Science, Volume 6718, Springer, 2011.

[4] J. M. Buhmann, M. Haghir Chehreghani and A. P. Streich and M. Frank, *Information Theoretic Model Selection for Pattern Analysis*, ICML Workshop on Unsupervised and Transfer Learning, 2011.

[5] J. Giesen and E. Schuberth and M. Stojakovic, *Approximate Sorting*, Lecture Notes in Computer Science 3887, 2006.

[6] W. Elmenreich and T. Ibounig and I. Fehervari, *Robustness versus Performance in Sorting and Tournament Algorithms*, Acta Polytechnica Hungarica, 2009.

[7] K. G. Jamieson and R. D. Nowak, *Active Ranking using Pairwise Comparisons*, NIPS, 2011.

[8] Y. Bilu and N. Linial, *Are stable instances easy?* 1st Symp. Innovations in Computer Science (ICS), 2010.