Scaling Up Ordinal Embedding: A Landmark Approach

Jesse Anderton [1] **Javed Aslam** [1]

Abstract

Ordinal Embedding is the problem of placing n objects into $\mathbb{R}^d$ to satisfy constraints like "object a is closer to b than to c." It can accommodate data that embeddings from features or distances cannot, but is a more difficult problem. We propose a novel landmark-based method as a partial solution. At small to medium scales, we present a novel combination of existing methods with some new theoretical justification. For very large values of n optimizing over an entire embedding breaks down, so we propose a novel method which first embeds a subset of $m \ll n$ objects and then embeds the remaining objects independently and in parallel. We prove a distance error bound for our method in terms of m and that it has $O(dn \log m)$ time complexity, and show empirically that it is able to produce high quality embeddings in a fraction of the time needed for any published method.

1. Introduction

We consider the problem of converting pairwise distance comparisons of a set of n objects into a d-dimensional Euclidean representation $X \in \mathbb{R}^{n \times d}$ which preserves the order, particularly in the large n setting. This task is variously known as ordinal embedding or non-metric multidimensional scaling. Algorithms take as input a set of *triplets* (a, b, c) meaning that the embedding should satisfy $\|x_a - x_b\| < \|x_a - x_c\|$ (denoting the ith row of X by x_i). The triplets may be fixed in advance or actively selected by the embedding algorithm.

In many other embedding tasks, one either has access to a vector representation of the objects or one can somehow estimate pairwise distances between the objects. In these cases, efficient and effective embedding algorithms already exist. Ordinal embedding addresses embedding when such positions and distances are not readily available. It was originally used with triplets sourced from human comparisons of phenomena for which answering distance or position questions is effectively impossible. For example, early experiments in color theory (Borg & Groenen, 2005) asked human participants to identify the most perceptually-similar pair of colors from among three examples. Ordinal embedding is also of use when precise distances are very expensive to compute but comparing distances is more tractable; for example, one could perform a breadth-first search of a large graph down to some fixed depth and infer that any nodes not visited must be more distant from the source than any node already seen. A third domain of interest involves data which is highly sparse or noisy, so one would prefer to rely only on the order of distance estimates rather than their magnitudes. A final possibility is in exploring novel distance or ranking functions for which deriving distance-based embedding algorithms may be laborious; ordinal embedding is agnostic about the distances used to derive triplets, and exploratory embeddings can be produced with out-of-the-box ordinal methods to compare various distance functions. High-quality ordinal embeddings of large scale datasets can also yield interesting new algorithms for important order-dependent tasks such as similarity search, product recommendation, metric/kernel learning, or information retrieval.

Unfortunately, state-of-the-art ordinal embedding algorithms are unable to handle large datasets. Methods which optimize over a Gram matrix are inherently quadratic in n, while current methods which optimize over point positions are non-convex, so with large enough n and d a random initialization is highly unlikely to be near the global optimum. Improving the scalability of these methods is an active research topic, but to date the published results show at most $n = 10,000$ points embedded into $d = 3$ dimensions. Even the recent work of Cucuringu & Woodworth (2015), designed explicitly for large-n datasets, only shows results for up to $5,000$ points. See Section 1.1 for a further discussion.

This work allows ordinal embeddings of much larger datasets, raising the number of points supported by the state-of-the-art from tens of thousands to millions or more. We recommend a novel combination of existing methods for up to tens of thousands of points, and when n is very large we provide a fast new landmark-based algorithm with

[1] College of Computer and Information Science, Northeastern University, Boston, Massachusetts. Correspondence to: Jesse Anderton <jesse@ccs.neu.edu>.

Proceedings of the 36th International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019. Copyright 2019 by the author(s).

guarantees on embedding quality.

Our contributions are two-fold: (1) At small-to-medium scales, we demonstrate a novel combination of existing methods which achieves much better performance than found in current practice. (2) At large scales, we introduce a new embedding algorithm which permits high-quality embeddings at scales of at least two orders of magnitude larger than any previously published research.

Our Algorithm. Our proposed algorithm [1], Large-scale Landmark Ordinal Embedding (LLOE), works by: (1) uniformly sampling a subset of $m \ll n$ points, (2) using the medium-scale algorithm from Section 2 to embed the subset, and finally (3) using the subset as landmarks to embed the remaining points.

We thus contribute (1) L-SOE, a medium-scale algorithm which consists of a novel combination of existing methods and which reliably handles values of n up to the tens of thousands (Section 2), and (2) LLOE, a fast ordinal method for large n, which can embed real-world datasets under dimensionally-constrained circumstances, preserving order comparably to distance-based methods. LLOE uses only $O(dn \log m)$ triplets for some $m \ll n$, below the lower bound of $\Omega(dn \log n)$ for exact embeddings (Section 3). We analyze LLOE in Section 4, proving an error bound in terms of m for appropriately "smooth" datasets when m is sufficiently large and when the subset is accurately embedded, and show empirical results in Section 5.

LLOE depends critically on first embedding a large enough subset to act as landmarks; this is difficult for datasets with large "holes" between dense, disconnected clusters of points or with high intrinsic dimensionality. We do not solve this problem, instead relying on the state-of-the-art tools which can potentially fail and thus lead to poor performance overall. However, LLOE can readily take advantage of improvements in the state-of-the-art by replacing the subset embedding method as technology advances.

1.1. Related Work

Many methods are designed to embed ordinal datasets, e.g., Agarwal et al. (2007); Tamuz et al. (2011); Van der Maaten & Weinberger (2012); Terada & von Luxburg (2014); Hashimoto et al. (2015). They typically embed thousands of data points into five or fewer dimensions, often for the purpose of visualizing similarity inferred from human assessments. Many have $\Omega(n^2)$ or worse objectives, particularly those which optimize a Gram matrix of point positions, making them ill-suited to a large n setting.

Cucuringu & Woodworth (2015) present a method designed for larger n. They use a spectral clustering of the kNN adjacency graph to divide the objects into subsets, embed each subset using Local Ordinal Embedding (Terada & von Luxburg, 2014), and align points shared by different subsets to merge them. Compared to our approach, this method has two main drawbacks. First, it assumes access to the kNN adjacency graph for the data. This graph can be costly to obtain, taking up to $O(n^2)$ triplets. Second, it involves various matrix operations which become expensive for larger sets. Our approach is simpler and more efficient.

The landmark-based method of Davenport (2013) is similar to ours; they embed points individually using triplets based on a set of previously-embedded landmarks. They choose triplets to embed a new point x by sampling random pairs of landmarks and comparing their distances to x; in the limit, this will place x at an arbitrary position within the correct cell of the Voronoi diagram of landmarks. In contrast, our method: (1) uses more precise triplets which constrain new points to lie in an intersection of spherical shells, (2) comes with theoretical convergence guarantees, (3) empirically exhibits successful embeddings for much larger datasets in higher dimensionalities, and (4) even compares tolerably well to simple distance-based methods, which are trained on more informative data unavailable to us.

Recent theoretical advances have established that when the underlying point positions meet certain smoothness guarantees, preserving the distance ordering between sufficiently many points must also preserve distances. In fact, various subsets of triplets also suffice to preserve distances (Kleindessner & von Luxburg, 2014; Arias-Castro, 2017). Convergence rates and error bounds have been proven for embeddings of randomly-selected noisy triplets (Jain et al., 2016), and it is established that a lower bound of at least $\Omega(dn \log n)$ adaptively-selected triplets is needed to recover the distances between n objects in $\mathbb{R}^d$ up to global scaling (Jamieson & Nowak, 2011). Our method is not exact, but uses only $O(dn \log m)$ triplets for subset size $m \ll n$.

Jain et al. (2016) show a convergence result for $O(dn \log n)$ randomly-selected triplets, but for a slightly different problem. They assume triplets are less predictable when the compared distances are almost equal, and we do not. In a noise-free setting, the triplet selection method we use is more accurate than random triplets (see Figure 2). In fact, Jamieson & Nowak (2011) proved that $\Omega(n^3)$ randomly-selected triplets are needed for the precision we obtain.

Our methods amount to sorting a subset of the objects by distance to some fixed object, so it is also worth mentioning the large literature on sorting algorithms for crowdsourcing (e.g., Marcus et al. (2011); Niu et al. (2015)) and on noise-tolerant sorting and selection algorithms (e.g., Feige et al. (1990); Ergün et al. (1998); Alonso et al. (2004); Braverman & Mossel (2008); Ajtai et al. (2009); Hadjicostas & Lakshmanan (2011)), especially as the latter seem somewhat

[1] Available at https://github.com/jesand/lloe.

overlooked by the learning community.

2. Landmark Soft Ordinal Embedding

In this section, we improve the state-of-the-art for small-to-medium-scale ordinal embeddings with a novel combination of existing methods that effectively solves the ordinal embedding problem at these scales, backed up by empirical results substantially stronger than are currently found in the literature and novel theoretical support for our method.

We call this method Landmark Soft Ordinal Embedding (L-SOE). L-SOE optimizes the Soft Ordinal Embedding (Terada & von Luxburg, 2014) objective using L-BFGS (Liu & Nocedal, 1989), over a set of triplets chosen according to the LNM-MDS algorithm of Jamieson & Nowak (2011). The SOE objective,

$$\mathcal{L}(X;\mathcal{T},\lambda) \tag{1}$$
$$= \sum_{(a,b,c)\in\mathcal{T}} \max\left(0, \|x_a - x_b\| - \|x_a - x_c\| + \lambda\right)^2,$$

incurs zero loss when points b and c are correctly ordered with respect to a (where λ sets the scale).

2.1. Triplet Selection.

For ordinal objectives to succeed, it is critical to select the right set of triplets to adequately constrain the solution. While there are $O(n^3)$ total correct triplets, one can *obtain* all triplets with $O(n^2 \log n)$ adaptive triplets (by sorting the objects from each member) and *represent* all triplets using just $O(n^2)$ triplets (by transitivity). However, these numbers lead to a quadratic cost to compute the objective or its gradient, which is not practical for large n.

Four selection methods are proposed in the literature: (1) random triplets (e.g. sample (a,b,c) s.t. $\|x_a - x_b\| < \|x_a - x_c\|$) are easily collected and have error bounds provided by Jain et al. (2016); (2) Tamuz et al. (2011) offer an active learning method which iteratively selects triplets based on embeddings of the triplets gathered so far, but these embeddings are time-consuming to produce, only approximately-optimal triplets can be chosen, and it is not clear how many triplets are needed; (3) Local Ordinal Embedding (Terada & von Luxburg, 2014) derives triplets from the k-nearest neighbors adjacency matrix and provides theoretical convergence guarantees for large n, but implicitly uses $nk(n-k)$ triplets which is $\Omega(n^2)$ and $O(n^3)$; and (4) Landmark-based methods use a subset of points as "landmarks" and orders the set relative to them.

LNM-MDS Triplets. Without loss of generality, let our set of object identifiers be $[n] := \{1, \ldots, n\}$. The LNM-MDS algorithm samples L landmarks $\mathcal{L} \subset [n]$ and collects two subsets of triplets: (1) the entire collection is sorted by distance to each landmark, and (2) the landmarks are sorted by distance to each point in the collection. This costs $O(Ln \log n)$ triplets to find the ordering, and can be filtered by transitivity to $O(Ln)$ triplets for the objective. Once obtained, the triplets are embedded using Nonmetric Multidimensional Scaling (NM-MDS).

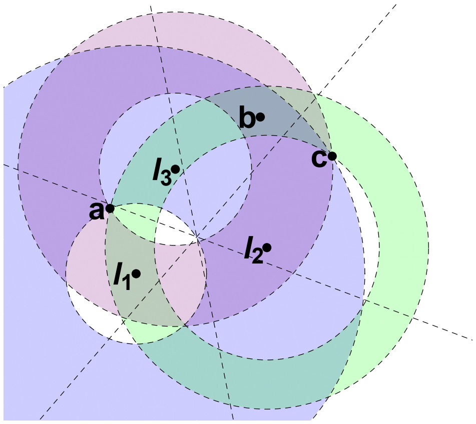


Figure 1: With landmarks l_1, l_2, l_3 and b placed in shells centered on the landmarks and with boundary points a and c, L-SOE constrains b to lie within the intersection of shaded regions and the correct Voronoi cell of landmarks.

LNM-MDS received little empirical support in Jamieson & Nowak (2011), perhaps due to the limitations of NM-MDS. The only experiment shown exhibits up to $n = 30, d = 2$. We are not aware of any further published use of the algorithm. However, Arias-Castro (2017) proved that when the collection are n i.i.d. draws from a measure taking support in a connected union of balls in $\mathbb{R}^d$, any embedding preserving the triplets chosen for LNM-MDS will preserve all pairwise distances up to a global scaling constant, with error that drives to zero as $n \to \infty$, and as the set of landmarks grows dense in the background space. This lends L-SOE some theoretical support, as it uses the same triplets; indeed, L-SOE empirically outperforms embeddings of an equal number of randomly-selected triplets.

2.2. Geometric Insights

We will now provide some intuition on why L-SOE can be expected to perform well. An ordinal triplet (a,b,c) constrains a to lie in the same halfspace as b, on the same side of the hyperplane supported by the vector halfway from b to c. The d-dimensional intersection of such halfspaces arising from a set of triplets is commonly called a *d-cell*. Each point $a \in [n]$ is thus constrained to lie in the *d-cell* corresponding to some set of triplets $\{(a, b_i, c_i)\}_i$ for each $b_i, c_i \in \mathcal{L}$. This *d-cell* is a subset of the cell of the Voronoi diagram of landmarks. Meanwhile, point b is constrained to lie within $\mathrm{B}(a,c)$, while c is constrained to lie outside

$\mathrm{B}\,(a, b)$. See Figure 1 for an illustration.

More subtly, preserving distance order from each landmark preserves a partial ordering of the orthogonal projections of the points along the vectors between pairs of landmarks. This is proven in the appendix as the following lemma.

Lemma 1 (Vector Projection Order)**.** *For distinct $p, q, x, y \in \mathbb{R}^d$, let x' and y' be the magnitudes of the projections of x and y, respectively, onto $\vec{pq}$. If $\|x_p - x_x\| < \|x_p - x_y\|$ and $\|x_q - x_x\| > \|x_q - x_y\|$ then $x' < y'$.*

In other words, the triplets (p, x, y) and (q, y, x) constrain x and y so the order of their orthogonal projections onto $\vec{pq}$ is preserved. Thus, for any of the $\binom{L}{2}$ vectors between pairs of landmarks, large subsets of points must be well-ordered by projection onto those vectors. If the landmarks are in general position, then any $d + 1$ landmarks create a set of (more than) d linearly-independent vectors establishing a basis of $\mathbb{R}^d$ onto which the points must be well-ordered in each dimension. Finally, by enforcing the total distance ordering of points w.r.t. each landmark $a \in \mathcal{L}$ we impose the same scale onto corresponding distances for each vector $\vec{ab}, b \in \mathcal{L} \setminus \{a\}$, as long as there are points near each vector at the required distances. We thus make the following conjecture.

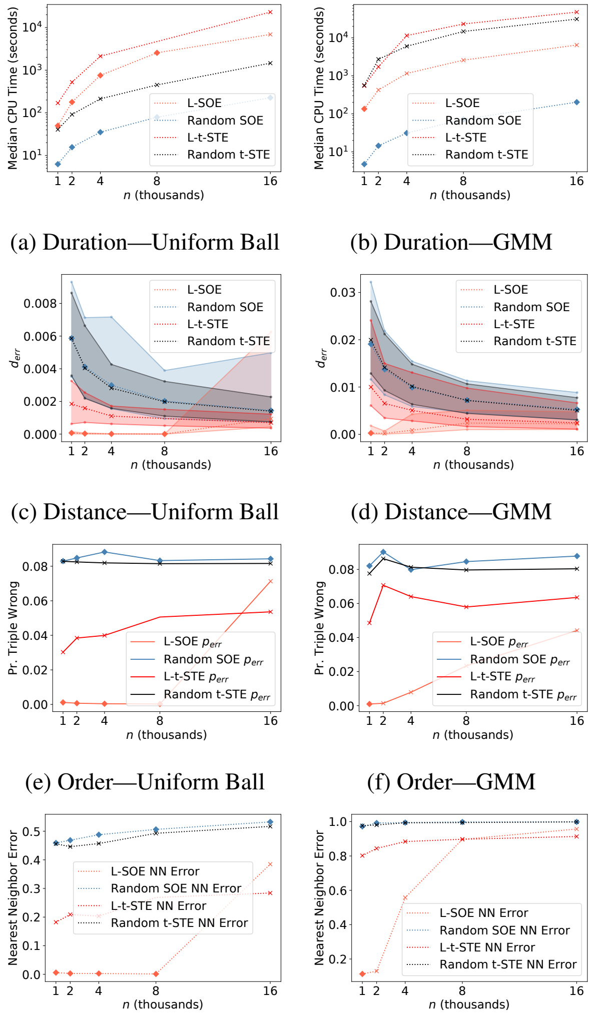
Conjecture 1 (L-SOE Convergence)**.** *Let $Y \in \mathbb{R}^{n \times d}$ be n i.i.d. draws from a Lipschitz measure over a bounded, connected subset of $\mathbb{R}^d$. For any $\epsilon, \delta > 0$ there is n_0 such that with probability at least $1 - \delta$, if $n > n_0$ any embedding X satisfying L-SOE for $L = O(d)$ landmarks satisfies $\|x_i - y_i\| < \epsilon, \forall i \in [n]$ after an appropriate similarity transformation.*

If true, this would confirm that the triplet cost of ordinal embedding is indeed $\Theta(dn \log n)$ as proposed by Jamieson & Nowak (2011).

2.3. Empirical Results

We will now present some empirical results for L-SOE, with further results in the Appendix. Recall that our primary aim here is to show that L-SOE can produce embeddings of adequate quality to serve as a reference for LLOE.

Figure 2 exhibits L-SOE for a variety of values of n and d, on two simulated datasets—a set of points sampled from a uniform ball, and a set of points sampled from a Gaussian Mixture Model (GMM) with 10 components [2]. The uniform ball represents the simplest type of data for ordinal embedding, with uniform density throughout the convex hull of the set and no large gaps in the data. The GMM is more realistic, and presents more of a challenge for ordinal methods. On each dataset, we compare L-SOE using $L = 100$ landmarks and $O(Ln)$ triplets as described above to an equal number of correct triplets selected uniformly at random. We also compare against t-STE (Van der Maaten & Weinberger, 2012) run with the same two sets of triplets. Triplets are embedded into a space of the correct dimensionality. Each embedding proceeds for up to 1,000 rounds of L-BFGS, with early termination if no loss decrease is observed. We report the best embedding from 20 random initializations, and the minimum CPU time for any of these embeddings [3].



(a) Duration—Uniform Ball

(b) Duration—GMM

(c) Distance—Uniform Ball

(d) Distance—GMM

(e) Order—Uniform Ball

(f) Order—GMM

(g) NN Error—Uniform Ball

(h) NN Error—GMM

Figure 2: L-SOE Performance in $\mathbb{R}^{30}$. A diamond marks an embedding with SOE loss < 1, and an X marks embeddings further from a global minimum.

We present four standard evaluation measures used for ordinal embedding methods. Distance error (denoted d_{err}) presents the distribution over distances (min, max, and median) between points in the true configuration and the em-

[2] Generated with `sklearn.datasets.make_blobs()` (Pedregosa et al., 2011)

[3] Embeddings run on a late 2013 15" quad core MacBook Pro with 2 GHz Intel Core i7 CPU and 16GB of RAM.

bedding, after a Procrustes transform (which minimizes d_{err}) and with the matrices scaled so the sum of squared vector norms equals one. We also estimate the probability p_{err} that a random triplet is incorrect in the embedding; note that the average Kendall's τ across all points satisfies $\bar{\tau} = 2\mathbb{E}[1 - p_{err}] - 1$. Finally, we estimate a top-heavy version of this, related to Average Precision and to the average kNN score across all values of k, in which a and c are sampled and then b is sampled s.t. $\|x_a - x_b\| < \|x_a - x_c\|$ in the true configuration; we present the probability p_{AP} that this order is violated in the embedding. The τ_{AP} measure (Yilmaz et al. (2008); commonly used in Information Retrieval) corresponds to $\tau_{AP} = 2\mathbb{E}[1 - p_{AP}] - 1$. Each of these probabilities is estimated from 100,000 triplets. Finally, we present nearest-neighbor error, the fraction of points for which the nearest neighbor in the embedding matches the nearest neighbor in the true configuration.

Random triplets are known to perform well overall, and in our experiments they consistently converge in under 1,000 rounds. This leads to faster embeddings than with L-SOE triplets. However, embeddings satisfying the L-SOE triplets exhibit nearly perfect performance. Even L-SOE embeddings which violate many triplets, leading to relatively high SOE loss, often perform better than optimal embeddings of random triplets. This empirically supports the theoretical convergence guarantees for L-SOE and suggests a generous rate of convergence; already by $n = 1,000$ excellent embeddings are observed. We also tried in preliminary experiments adding random triplets to L-SOE triplets, but did not observe a decrease in embedding time. This seems to merit further study.

3. Large-Scale Ordinal Embedding

At very large scales, it is not practical to find an embedding satisfying the SOE objective. Embeddings become increasingly time-consuming to produce as n increases, and the local optima obtained violate so many triplets that the output is not useful. To address this, we present our algorithm, Large-scale Landmark Ordinal Embedding (LLOE). LLOE produces ordinal embeddings at arbitrarily large scales using time linear in n and d. We are thus able to present the first ordinal embeddings of truly large-scale datasets (i.e. where n^2 runtime is too expensive). We also provide theoretical embedding quality guarantees, and justify our method with empirical performance of comparable scale to distance-based embedding methods (which are not available in general for ordinal datasets).

LLOE (Alg 1) proceeds in two phases, first identifying the largest number m of points which can be accurately embedded with L-SOE by a successive doubling strategy, and then embedding a subset of m items with L-SOE and then using a fast landmark approach for embedding the remaining $n - m$ items. It receives a collection of n items, w.l.o.g. only as a list of identifiers $[n] := \{1, \ldots, n\}$, a target dimensionality d, the number L of landmarks for the first (L-SOE) phase, and a triplet oracle $o : [n]^2 \to \{-1, +1, \lambda\}$. The triplet oracle answers distance comparisons (a, b, c), answering with -1 if $\|x_a - x_b\| < \|x_a - x_c\|$, with $+1$ if $\|x_a - x_b\| > \|x_a - x_c\|$, and with λ if the answer can not be determined (e.g. due to data sparsity).

Algorithm 1 $LLOE(n, d, L, o)$

```
input: object number $n$, landmark number $L$, dimension $d$, triplet oracle $o$
output: ordinal embedding $X \in \mathbb{R}^{n \times d}$
{L-SOE Phase}
randomly permute object identifiers
$m \leftarrow 50L$
repeat
  $m \leftarrow \min(2m, n)$
  $[x_1, \ldots, x_m], loss \leftarrow L\text{-}SOE(m, L, d, \prec)$
until $loss > 1$ or $m = n$
$m \leftarrow m/2$
$[x_1, \ldots, x_m] \leftarrow$ last successful embedding of $m$ points
{LLOE Phase}
for $q \leftarrow m + 1, \ldots, n$ (in parallel) do
  {Pick anchors for $x_i$}
  $shells \leftarrow$ empty list
  for $a \leftarrow 1, \ldots, m$ in FFT order do
    $p \leftarrow \arg\max_{\{i \in [m]: o(a,i,q) = -1\}} \|x_a - x_i\|$
    $r \leftarrow \arg\min_{\{i \in [m]: o(a,q,i) = -1\}} \|x_a - x_i\|$
    if $p$ and $r$ were both found then
      $shells.append(x_a, \|x_a - x_p\|, \|x_a - x_r\|)$
      if $|shells| = 2(d + 1)$ then
        $x_q \leftarrow$ minimum of Eq. 2 for $shells$
        break
return $[x_1, \ldots, x_n]$
```

Fast Landmark Embedding (LLOE Phase). In the second phase of LLOE, our goal is to choose positions $X = \{x_{m+1}, \ldots, x_n\}$ of all remaining objects so that $o(a, b, c) = -1 \Rightarrow \|x_a - x_b\| < \|x_a - x_c\|$ while using time subquadratic in n. We embed each point $j \in [n] \setminus [m]$ independently and in parallel. A standard landmark embedding approach would sort the members of $[m]$ by distance to j and then use a modified form of a standard ordinal embedding method to position x_j within $\mathbb{R}^d$. However, this would consume $O(nm \log m)$ total triplets, above the lower bound on the problem, and most of these triplets are redundant (Jamieson & Nowak, 2011). We take a more efficient approach, using $O(nd \log m)$ total triplets and thus beating the lower bound for "exact" embeddings (i.e. embeddings which preserve all $O(n^3)$ true triplets). For example, we will show results on $n = 1,000,000$ points embedded into

$\mathbb{R}^{30}$ with high accuracy and taking time growing linearly with n. In any case, obtaining an exact embedding likely involves fine-tuning the positions of the points in the subset, which makes embedding the remaining points dependent on each other through the subset. This dependency of each coordinate on all the others is part of what makes Ordinal Embedding difficult, and we wish to avoid it.

Our approach is to choose widely-separated points in $[m]$ to serve as "anchors," and to insert j into the ranking of $[m]$ for each anchor using binary search. This constrains x_j to lie inside a spherical shell centered on each anchor. That is, suppose that for some anchor $a \in [m]$ we find that j lies just beyond $b \in [m]$ but just before $c \in [m]$. Then x_j should be embedded inside $\mathrm{B}\left(x_a, x_c\right)$ but outside $\mathrm{B}\left(x_a, x_b\right)$, where $\mathrm{B}\left(i, j\right)$ denotes the set of points with distance at most $\|x_i - x_j\|$ from x_i. We will henceforth denote this difference of balls as $\mathrm{Shell}\left(a, b, c\right)$, which means that $\|x_a - x_b\| < \|x_a - x_i\| < \|x_a - x_c\|$. We will need at least $d + 1$ anchors for the intersection of their shells to hope to constrain x_j to a small, simply-connected region, and we also need m large enough that the thickness of each of these shells $(= d_{y_a y_{b+1}} - d_{y_a y_b})$ is small. We will later show in Section 4 that m need not be very large for x to be well-constrained. This is further validated by our empirical results in Section 5. In practice, we will choose $2(d+1)$ anchors to provide redundancy to deal with possible measurement error and unfortunate subset sampling.

To select anchors to embed a new point, we need to account for the possibility that comparisons can not be determined for any particular member of $[m]$. We choose a farthest-first traversal order of $[m]$; this tends to pick points as far apart as possible, but gradually closes in on the position of the new point. We will choose the first $2(d + 1)$ points in this order which can be compared to the new point. This ordering is re-used for all future points, so it is a one-time cost.

1. Choose as the first item the farthest-ranked point from an arbitrary member of $[m]$. (This point will lie on the convex hull of $[m]$.)
2. Given points $a_1, \ldots, a_{i-1}$ already chosen and set $\mathcal{T} \subset [m]$ not yet chosen, choose point a_i having $\max_{k \in \mathcal{T}} \min_{j<i} \|x_k - x_j\|$.

We relabel the set so that $[m]$ is in this order, and thus any prefix $x_1, \ldots, x_k, k < m$ forms an ϵ-net of $\{x_1, \ldots, x_m\}$. That is, there is some constant ϵ_k such that all points $x_1, \ldots, x_k$ have distance at least ϵ_k from each other, and all points $x_{k+1}, \ldots, x_m$ have distance at most ϵ_k to the nearest point in $x_1, \ldots, x_k$.

We embed x_j to lie as close as possible to the intersection of the $2(d + 1)$ spherical shells we have obtained. We can think of $\mathrm{Shell}\left(a, b, c\right)$ as the points within distance $(\|x_a - x_c\| - \|x_a - x_b\|)/2$ of the sphere with radius $(\|x_a - x_c\| + \|x_a - x_b\|)/2$ centered on a. Thus, we use a margin-relaxed version of the sphere intersection objective suggested by Coope (2000). Given a set of shells $\mathrm{Shell}\left(a_i, b_i, c_i\right)$, we define radius $r_i := (\|x_{a_i} - x_{b_i}\| + \|x_{a_i} - x_{c_i}\|)/2$ and margin $m_i := |\|x_{a_i} - x_{b_i}\| - \|x_{a_i} - x_{c_i}\||/2$ and seek the position x_j within margin m_i of each $\mathrm{Sphere}\left(a_i, r_i\right)$, using L-BFGS to optimize our objective,

$$\mathcal{L}(y) = \sum_{i=1}^{2(d+1)} \max(0, (\|y - a_i\| - r_i)^2 - m_i^2). \qquad (2)$$

This objective is quite similar to Figure 2, except there is no Voronoi cell constraint and the landmarks are in fixed positions.

4. Convergence of LLOE

We now provide results showing an error bound for our method. Section 5 demonstrates the method on a variety of generated and real datasets, to show how it behaves in practice.

Bounding shell thickness. As a first step, we want to show that the shells we use for phase two of our algorithm are not too thick. We will begin by showing that the number of points from the full set which lie within each shell is close to uniform. Intuitively speaking, this happens because choosing our subset uniformly at random produces i.i.d. draws from the underlying density. Since the underlying density is Lipschitz smooth (by assumption), the distances between pairs of points is similarly smooth. We formalize this as follows.

Let $a \in \mathbb{R}^d$ and $\{x_1, \ldots, x_m\} \subset \mathbb{R}^d$ be i.i.d. draws from a Lipschitz-smooth density over a bounded, connected subspace of $\mathbb{R}^d$. Now let $d_1, \ldots, d_m$ be the Euclidean distance from a to each of the x_i, sorted in increasing order. These distances represent the radii to the shell boundaries one can use to position new points with respect to anchor a. Since the underlying density over points is Lipschitz smooth with constant L_X (by assumption), the induced density over distances is also Lipschitz smooth with some related constant $L_{\mathcal{D}}$. We prove the following theorem in the appendix.

Theorem 1 (Anchor distance representation)**.** *Let distances* $d_1 < \cdots < d_m$ *be i.i.d. draws from a Lipschitz smooth measure over the positive reals having Lipschitz constant* $L_{\mathcal{D}}$*. Then with probability of at least* $1 - \delta$ *for any* $\delta \in (0, 1)$*, the largest gap* $d_i - d_{i-1}$ *written as* $\epsilon d_m L_{\mathcal{D}}$*, satisfies* $\epsilon < \frac{2}{m} \ln \frac{m}{\delta}$.

Theorem 1 tells us that the size of the thickest shell for any given anchor scales like $(\ln m)/m$ as the subset size m grows. Doubling the subset size causes the largest shell thickness to be reduced by roughly $1/3$. However, as the probability of drawing a point from a shell increases with the shell volume the shells will tend not to be evenly-spaced — most shells will be much thinner than this, especially in high-dimensional spaces. Critically, the shells will tend to be thinner in areas of higher point concentration, where more precision is needed for accurate distance recovery.

Bounding embedding quality. In the appendix we prove the following upper bound on the distance of any point embedded in phase two from its correct position, provided that phase one recovered exact point positions.

Theorem 2 (Embedding accuracy)**.** *Let $X \subset \mathbb{R}^d$ be n i.i.d. draws from a Lipschitz-smooth measure over a bounded, connected subspace of $\mathbb{R}^d$. Let $\mathcal{S} \subset X$ be a uniformly-sampled subset of size $m \gg d$ with known positions, and let $\mathcal{A} \subset \mathcal{S}$ be a set of at least $d+1$ anchors chosen by farthest-first traversal. For any $x \in X$, let $\hat{x} \in \mathbb{R}^d$ be any point satisfying the distance constraints to the members of $\mathcal{A}$ imposed by the order of $\mathcal{S} \cup \{x\}$. Then there is a constant $c \in \mathbb{R}$ such that with probability at least $1-\delta$ for $\delta \in (0,1)$,*

$$\|x - \hat{x}\| < \frac{cd}{m} \ln \frac{m}{\delta}.$$

The proof employs our Vector Projection Order lemma (Lemma 1), using the known explicit distances to derive a range of possible projections for a given point onto each vector between pairs of anchors. This range depends on the inter-anchor distances; using the fact that they form an ϵ-net lets us tighten the projection interval. Finally, we use that we have small projection intervals onto d linearly independent vectors to derive a distance bound within $\mathbb{R}^d$.

On Noisy Triplets. When only noisy triplets are available, e.g. when triplets are obtained from human assessments or behavior, a slight variation of our distance bound can be proven to hold. LLOE should be modified by using a noise-tolerant sorting algorithm (such as those listed in Section 1.1) to insert each point x into the anchors' rankings with a high probability bound on the number of inversions. The lower and upper distance bounds from x to each anchor can be expanded using this bounded number of inversions, producing a confidence interval over the distance to each anchor and expanding the bound in Theorem 2 by a corresponding constant factor.

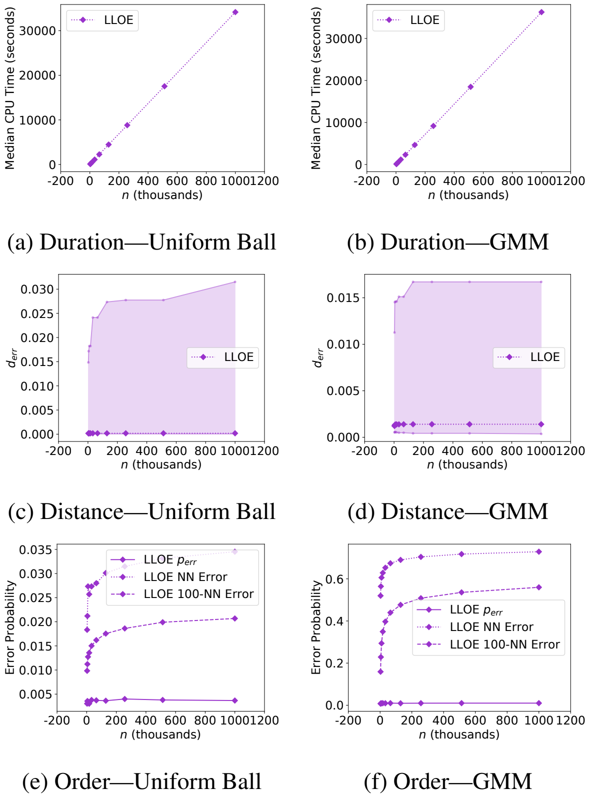
5. Empirical Results for LLOE

A natural baseline for LLOE would use the same ordinal input, but no known ordinal method can handle such large collections. There are no published results at such scales, and no appropriate public datasets of triplets. Instead, we compare to LSA (Deerwester et al., 1990), which uses strictly more information about the data (as it has access to the original distance matrix). One would expect LSA to strongly outperform our method, as it has access to this additional information which is typically not available for ordinal problems, and as our ordinal embeddings are approximate (relying on reference embeddings of very small sizes m). We nonetheless produce embeddings of comparable quality to LSA.

We exhibit LLOE on a variety of datasets, showing the first published high-quality ordinal embeddings of large datasets in high dimensionality, competitive with a LSA baseline. Additional results are in the appendix. Please refer to Section 2.3 for descriptions of our synthetic datasets and evaluation measures. These results (Figure 3) are largely what one would expect from an embedding method which preserves distances to at least $d+1$ points to within some small error. Regardless of the size of n embedded we observe a roughly constant distance error and a duration linear in n. The maximum distance error is well above the median error; over 990,000 points for the Uniform Ball and over 920,000 points for the GMM had distance error below 0.003. On the test hardware used, and with $d = 5$, LLOE embedded about 1,100 points per second, while SOE embeds about 16 points per second with L-SOE triplets and about 670 points per second with random triplets[4].

We also tested LLOE on two real datasets. We embed `20 Newsgroups` ($n = 18,846$, $d = 101,631$ TF-IDF scores) and `MNIST Digits` ($n = 70,000$, $d = 784$ pixels), and compare the results to LSA in Figure 4. These datasets pose challenges that the synthetic data does not; namely, (1) the distributions are naturally occurring and may violate our modeling assumptions (e.g. containing large gaps between dense clusters of points), and (2) we embed into fewer dimensions than are required to fully preserve distances, so a perfect embedding is not possible. As a consequence, our reference embeddings tend to be relatively imprecise. Recall that LSA is a strong baseline, as it uses exact distances between points which are typically not available for ordinal datasets. We do not quite match its performance, but we argue that our results are of comparable quality for many practical uses. It is worth pointing out that for some lower dimensionalities LLOE preserves order better than LSA; see the appendix for those results. Our takeaway is that LLOE is particularly useful when exact distances are not available, and methods like LSA can not be used, though it may outperform LSA on a given dataset in a given dimensionality. It would be interesting to compare to other algorithms such as word2vec, but we do not do so here. Further work would need to be done to build an ordinal oracle which corresponds to the distance function which word2vec seeks to preserve, and perhaps to deal with the special sparsity structure of word embeddings. We leave the design of ordinal word

[4] SOE typically used 1,000 rounds for L-SOE, while for random triplets it converged in far fewer rounds; its performance per round depends on d and on the number of triplets.



(a) Duration—Uniform Ball

(b) Duration—GMM

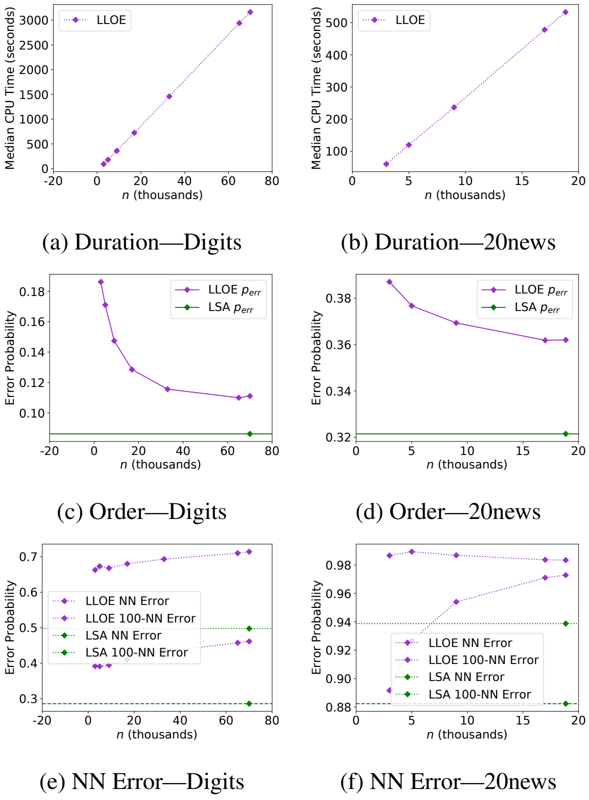
(c) Distance—Uniform Ball

(d) Distance—GMM

(e) Order—Uniform Ball

(f) Order—GMM

Figure 3: LLOE on synthetic data in $\mathbb{R}^{30}$; $m = 1,000$. PCA not shown, as original positions are already in $\mathbb{R}^{30}$.



(a) Duration—Digits

(b) Duration—20news

(c) Order—Digits

(d) Order—20news

(e) NN Error—Digits

(f) NN Error—20news

Figure 4: LLOE on real data in $\mathbb{R}^{30}$; $m = 1,000$.

embedding algorithms to future work.

Interestingly, performance on held-out triplets tends to improve on the real datasets as the number of points embedded increases. We believe this is due to the relatively poor quality of reference embeddings used. The reference embedding quality dominates performance as the first few points are embedded. As the embedding progresses, however, since the reference embedding causes the space to be warped *consistently* for all new points, performance on held-out triplets (which relies on order rather than distance) improves until the points embedded by LLOE dominate the order rather than the points in the reference embedding.

These experiments show that our method can handle large n and d. We believe that our method can produce embeddings of much larger dimensionality (e.g. hundreds of dimensions) once research advances to the point that such a high-dimensional reference embedding can be obtained. Such an advance will be valuable, as many natural ordinal datasets (e.g. document and word embeddings) seem to require such high dimensionality for acceptable representations. In the meantime, this method competes with distance-based methods at moderately-high dimension.

6. Final Thoughts

Based both on our results and the existing theoretical support, we would recommend the use of L-SOE for any ordinal embedding of up to tens of thousands of points. While the embeddings are somewhat slow compared to random triplets, the process might be sped by injecting a small number of random triplets to a L-SOE set. When L-SOE succeeds in finding a good embedding, the result is a nearly-perfect recovery of the original configuration.

For larger values of n, we have shown that LLOE exhibits excellent performance and produces strong embeddings, even recovering somewhat from poor reference embedding performance. The main missing pieces, given this work, are in extending our ability to embed higher-dimensional datasets ($d = 300$ is common for PCA/word2vec) and solving the distributional challenges of ordinal embedding (e.g. when large gaps exist between dense clusters of points).

References

Agarwal, S., Wills, J., Cayton, L., Lanckriet, G., Kriegman, D., and Belongie, S. Generalized non-metric multidimensional scaling. In *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics*, San Juan, Puerto Rico, 2007.

Ajtai, M., Feldman, V., Hassidim, A., and Nelson, J. Sorting and Selection with Imprecise Comparisons. *ICALP*, 2009.

Alonso, L., Chassaing, P., Gillet, F., Janson, S., Reingold, E. M., and Schott, R. Quicksort with Unreliable Comparisons: A Probabilistic Analysis. *Combinatorics, Probability and Computing*, 13(4-5):419–449, July 2004.

Arias-Castro, E. Some theory for ordinal embedding. *Bernoulli*, 23(3):1663–1693, 08 2017. doi: 10.3150/15-BEJ792.

Borg, I. and Groenen, P. *Modern Multidimensional Scaling: Theory and Applications*. Springer New York, New York, NY, 2005. ISBN 978-0-387-28981-6. doi: 10.1007/0-387-28981-X.

Braverman, M. and Mossel, E. Noisy sorting without resampling. *SODA*, 2008.

Coope, I. D. Reliable computation of the points of intersection of n spheres in R^n. *ANZIAM Journal*, 42(0): 461–477, December 2000.

Cucuringu, M. and Woodworth, J. Ordinal embedding of unweighted kNN graphs via synchronization. In *IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP)*, pp. 1–6. IEEE, 2015.

Davenport, M. A. Lost without a compass: Nonmetric triangulation and landmark multidimensional scaling. In *5th IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*. IEEE, 2013.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., and Harshman, R. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41:391–407, 1990.

Ergün, F., Kannan, S., Kumar, S. R., Rubinfeld, R., and Viswanathan, M. Spot-checkers. In *Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing*, STOC '98, pp. 259–268, New York, NY, USA, 1998. ACM. ISBN 0-89791-962-9. doi: 10.1145/276698.276757.

Feige, U., Peleg, D., Raghavan, P., and Upfal, E. Computing with unreliable information. In *STOC '90: Proceedings of the twenty-second annual ACM symposium on Theory of computing*, pp. 128–137, New York, New York, USA, April 1990. ACM Request Permissions.

Hadjicostas, P. and Lakshmanan, K. B. Recursive merge sort with erroneous comparisons. *Discrete Applied Mathematics*, 159(14):1398–1417, August 2011.

Hashimoto, T. B., Sun, Y., and Jaakkola, T. S. Metric recovery from directed unweighted graphs. *AISTATS*, 2015.

Jain, L., Jamieson, K. G., and Nowak, R. Finite sample prediction and recovery bounds for ordinal embedding. In *Advances In Neural Information Processing Systems*, pp. 2711–2719, 2016.

Jamieson, K. G. and Nowak, R. D. *Low-dimensional embedding using adaptively selected ordinal data*. IEEE, 2011.

Kleindessner, M. and von Luxburg, U. Uniqueness of Ordinal Embedding. *COLT*, 2014.

Liu, D. C. and Nocedal, J. On the limited memory bfgs method for large scale optimization. *Mathematical programming*, 45(1-3):503–528, 1989.

Marcus, A., Wu, E., Karger, D., Madden, S., and Miller, R. Human-powered sorts and joins. *Proc. VLDB Endow.*, 2011.

Niu, S., Lan, Y., Guo, J., Cheng, X., Yu, L., and Long, G. Listwise Approach for Rank Aggregation in Crowdsourcing. *WSDM*, 2015.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Tamuz, O., Liu, C., Belongie, S., Shamir, O., and Kalai, A. Adaptively learning the crowd kernel. In *Proceedings of the International Conference on Machine Learning*, 2011.

Terada, Y. and von Luxburg, U. Local ordinal embedding. In *Proceedings of the 31st International Conference on Machine Learning*, 2014.

Van der Maaten, L. and Weinberger, K. Stochastic triplet embedding. *2012 IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, pp. 1–6, 2012.

Yilmaz, E., Aslam, J. A., and Robertson, S. E. A new rank correlation coefficient for information retrieval. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 587, New York, New York, USA, July 2008. ACM Request Permissions.