Euclidean Co-Embedding of Ordinal Data for Multi-Type Visualization

Dung D. Le[*]   Hady W. Lauw[†]

Abstract

Embedding deals with reducing the high-dimensional representation of data into a low-dimensional representation. Previous work mostly focuses on preserving similarities among objects. Here, not only do we explicitly recognize multiple types of objects, but we also focus on the ordinal relationships across types. Collaborative Ordinal Embedding or COE is based on generative modelling of ordinal triples. Experiments show that COE outperforms the baselines on objective metrics, revealing its capacity for information preservation for ordinal data.

1 Introduction

We are interested in embedding, a visualization that maps a high-dimensional representation of data to a lower-dimensional one. The emphasis is on its capacity to preserve as much information as possible. Each data point is represented by a coordinate in a low-dimensional Euclidean space, and the relationship among data points are visualizable through Euclidean distances in that visualization space. Most of the previous works on embedding focus on metric embedding, whose objective is to preserve the pairwise distances among data points [19, 20, 18, 4]. This is applicable when the main relationship among objects is similarity, e.g., images of handwritten digits or human faces [4].

Ordinal data refers to data where the ranking established by numerical values are more significant than the exact values. Such a representation is applicable to various domains, e.g., preferences [16], document retrieval [8]. As a focusing point, and without loss of generality, subsequently, we primarily use the example of the domain of preferences, where users express how much they like various items. For instance, after purchasing a product on Amazon, a user may leave an explicit rating. While listening to music at Spotify, a user leaves implicit traces of her liking for a track or an artist by the frequencies at which she consumes them. In both explicit and implicit cases, it is important to model the relative sense of whether an item is preferred to another.

---

[*]School of Information Systems, Singapore Management University. Email: ddle.2015@phdis.smu.edu.sg.

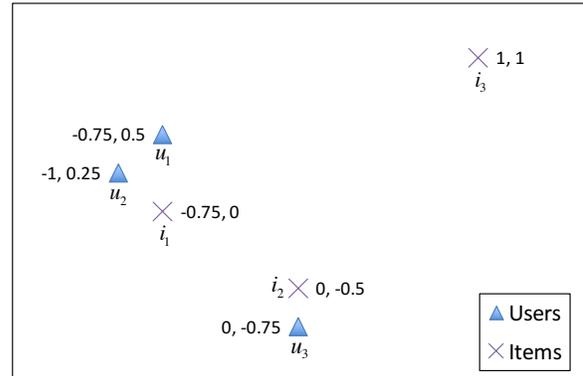[†]School of Information Systems, Singapore Management University. Email: hadywlauw@smu.edu.sg.

Figure 1: Euclidean Embedding of Users & Items

Problem. Embedding for ordinal data seeks to preserve the ordinal relationships among data points. Our goal is ordinal *co-embedding*, where multiple object types are involved (e.g., users and items), and cross-type ordinal relationships are key (e.g., users express preferences over items). We discuss the scenario of a preference dataset. Suppose for each user, we are given pairwise rankings over items. A triple $\langle u, i, j \rangle$ indicates that a user u prefers an item i to a different item j. As output, every user and every item would be respectively assigned a latent coordinate (to be learned) in a D-dimensional Euclidean space. We assume $D = 2$ or 3 for their appropriateness for visualization. User u's preference for item i to item j is visualizable through a shorter distance between u and i than between u and j.

Figure 1 illustrates an example 2D embedding for three users (blue triangles) and three items (purple crosses), specifying their respective coordinates. Through our spatial perception of the relative distances, we can immediately tell that the user u_1 prefers item i_1 the most (closest), followed by item i_2, and item i_3 the least (furthest). Such information leaps out at us without our having to consciously compute the distances.

In addition to visualization, embedding could also enable other applications arising from its Euclidean metric properties. One potential application is *retrieval* for recommendation queries, such as which items are the closest (most preferred) to a user. Euclidean geometry fits the mould of spatial data management, allowing it

to benefit from such developments as spatial indexing [3] and efficient nearest-neighbor query processing [17]. For another potential application, as embedding relies on building a compact model for user preferences, it may eventually enable an interactive interface for training recommender systems. In text domain [12], we may seek an embedding that preserves the relative importance of words to a document (for summarization).

Approach. While there has been prior work on ordinal embedding [11, 1, 21], our work is novel in a couple of fundamental respects. First, the "classical" ordinal embedding is formulated mainly for one object type, e.g., cities [21], images [1]. It enforces that for same-type quadruple of objects $\langle i, j, k, l \rangle$, if i is closer to j in the original data than k is to l, the same ordinal relationship should hold in the embedding space. This presumes that the primary information is similarity among objects. In contrast, our primary objective is based on *ranking*. More specifically, the ranking of objects of one type (e.g., items) by an object of a different type (e.g., user). For instance, it is possible for two users to be "similar", say in terms of their demographics or their habits of watching horror movies, and yet to have different rankings over specific items.

Moreover, because classical ordinal embedding deals with within-type ordinal relationships, it implicitly assumes that there is one underlying reality to approximate, e.g., distances of cities in the map [21]. However, for many ordinal datasets, there may not be a singular ground-truth reality. For preference data, each user imposes his or her own ranking on the items, and these rankings may be different and at times conflicting. This fundamental difference motivates two distinguishing aspects of our approach. Because a common embedding space needs to accommodate the diverse preferences of users, we harness the collaborative effect among users and among items. In order to capture the variance in the rankings induced by preferences of different users or items in a principled way, we also formulate our model in terms of probabilistic generative modelling.

Contributions and Organization. We provide the formal problem statement in Section 2. In this paper, we make the following contributions towards the problem. *First*, in Section 3, we propose a new embedding model, called *Collaborative Ordinal Embedding* or COE. This model is notable in its generative modeling of ordinal embedding allowing various types of triples, as well as in its objective function with both a penalty component for violated observations and a reward component for preserved observations on a smooth continuous spectrum modeled by probabilistic Sigmoid or Gompertz distributions. *Second*, in Section 3.3, we describe COE's learning algorithm to derive the embedding co-

ordinates that maximize the posterior probability of the generative model based on stochastic gradient ascent for both Sigmoid and Gompertz. *Third*, in Section 5, comprehensive experiments on publicly available datasets show that COE outperforms the baselines, both in preserving the observed pairwise comparisons and in predicting unseen pairwise comparisons expressed as relative distances in the Euclidean space. We review the related work in Section 4, and conclude in Section 6.

2 Problem Formulation

We formally define the problem addressed in this paper, which is *co-embedding* of objects based on *cross-type* ordinal relationships. Moreover, for ease of reference, we adopt the language of preference dataset, and refer to one of the types as "users", and the other type as "items". Note that this is merely nomenclature, and does not limit the object types in the ordinal data.

Input. The set of users is $\mathcal{U}$, and u or v refers to a user. The set of items is $\mathcal{I}$, and i or j refers to an item. The input is a multiset of triples $\mathcal{T} = \mathcal{T}_A \cup \mathcal{T}_B$, consisting of "type-A" triples $\mathcal{T}_A \subset \mathcal{U} \times \mathcal{I} \times \mathcal{I}$ and "type-B" triples $\mathcal{T}_B \subset \mathcal{U} \times \mathcal{U} \times \mathcal{I}$. A type-A triple $t_{uij} \in \mathcal{T}_A$ relates a user $u \in \mathcal{U}$ and two different items $i, j \in \mathcal{I}$, indicating u's preferring i to j. A type-B $t_{uvi} \in \mathcal{T}_B$ indicates a user u has greater preference over i than user v does.

Such triples form a general representation of preferences over one object type as expressed by the other object type. There are examples abound in both explicit and implicit feedback scenarios. Triples can be derived from ratings, e.g., when u assigns a higher rating to i than to j. Other than ratings, it could also model implicit feedback [16]. For cable TV, u may watch the channel i but not j, or spend a longer time watching i than j [7]. For Web search, u may click on the result i after skipping j [15]. Outside of preference domain, in text, a word i may be more frequent than another word j in document u. Alternatively, document u may be more relevant to word i than document v does.

While we focus on cross-type triples, it is feasible to accommodate triples involving three objects of the same type, e.g., u is more "similar" to v than to v'. Here, we will not concentrate on such similarity-based triples.

More generally, we can use triple form $(o^1_{\tau_1}, o^2_{\tau_2}, o^3_{\tau_3})$, where $o^i_{\tau_i}$ are objects of types $\tau_i, (i = 1, 2, 3)$ respectively, to represent ordinal relations among multiple objects. The framework can be extended naturally by adding latent variables for objects of each type. For simplicity, we only present our model with two types.

Output. Given $\mathcal{T}$, the goal is to assign a coordinate $x_u \in \mathbb{R}^D$ to each user $u \in \mathcal{U}$, as well as a coordinate $y_i \in \mathbb{R}^D$ to each item $i \in \mathcal{I}$, such that their distances in $\mathbb{R}^D$ preserve the relative ordering indicated by the
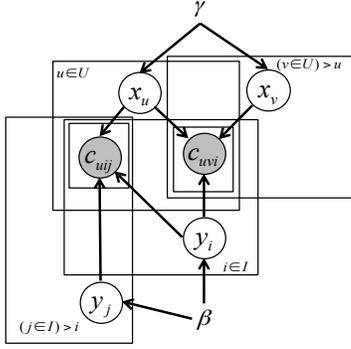
Figure 2: Collaborative Ordinal Embedding (COE)

triples. We denote the collection of all user coordinates as X and the collection of all item coordinates as Y. The coordinates of users and items lie in the same D-dimensional Euclidean space, where D is 2 or 3.

PROBLEM 1. (ORDINAL CO-EMBEDDING) *Given a set of triples $\mathcal{T}$, find the set of user coordinates X and item coordinates Y, so as to meet the following respective condition for as many triples in $\mathcal{T}$ as possible, i.e.,*

$$t_{uij} \in \mathcal{T}_A \Rightarrow ||x_u - y_i|| < ||x_u - y_j||,$$
$$t_{uvi} \in \mathcal{T}_B \Rightarrow ||x_u - y_i|| < ||x_v - y_i||$$

3 Methodology

We now describe our proposed model, called *Collaborative Ordinal Embedding* or COE. The challenge is integrating the diverse triples into the same low-dimensional Euclidean space. The input triples $\mathcal{T}$ may also suffer from sparsity, variance, and uncertainties, in the form of incompleteness (not all possible triples are specified), inconsistency (some triples are conflicting), and repetitions (some triples may occur more than once). Yet the final objective is a unified view for all items and users.

3.1 Generative Model To achieve this, we harness the "collaborative" effect. Since item coordinates are shared across users, users with similar coordinates would have similar ordinal relationships with items. To develop this probabilistically, we design a graphical model, whose plate notation is illustrated in Figure 2.

We model each user coordinate and each item coordinate as real-valued latent random variables x_u and y_i respectively. For each triple $\langle u, i, j \rangle$ where $i < j$, we associate it with a binary random variable c_{uij}. When c_{uij} takes on the value of 1, it corresponds to an instance of $t_{uij} \in \mathcal{T}$. When $c_{uij} = 0$, it corresponds to an instance of $t_{uji} \in \mathcal{T}$. In Figure 2, c_{uij} is shaded and lies within its own plate, i.e., it is observed and there could be multiple instances. Correspondingly, for

each triple $\langle u, v, i \rangle$ where $u < v$, we associate it with a variable c_{uvi}. The state of c_{uij} (or c_{uvi}) and the generation of t_{uij} (or t_{uvi}) are related to user and item coordinates through the following generative process.

The generative process of COE is as follows:

1. For each user $u \in \mathcal{U}$:
   Draw u's coordinate: $x_u \sim \text{Normal}(0, \gamma^2 \mathbf{I})$,

2. For each item $i \in \mathcal{I}$:
   Draw i's coordinate: $y_i \sim \text{Normal}(0, \beta^2 \mathbf{I})$,

3. For each triple $\langle u, i, j \rangle \in \mathcal{T}_A$:
   - Draw $c_{uij} \sim \text{Bernoulli}(\text{P}(c_{uij} = 1 \mid x_u, y_i, y_j))$,
   - If $c_{uij} = 1$, generate a triple instance t_{uij},
   - Else ($c_{uij} = 0$), generate a triple instance t_{uji}.

4. For each triple $\langle u, v, i \rangle \in \mathcal{T}_B$:
   - Draw $c_{uvi} \sim \text{Bernoulli}(\text{P}(c_{uvi} = 1 \mid x_u, x_v, y_i))$.
   - If $c_{uvi} = 1$, generate a triple instance t_{uvi},
   - Else ($c_{uvi} = 0$), generate a triple instance t_{vui}.

In Step 1 and Step 2, we generate the users' and items' coordinates, placing zero-mean multi-variate spherical Gaussian priors on these coordinates, with γ^2 and β^2 controlling the respective variances of the Normal distributions. $\mathbf{I}$ denotes the identity matrix.

In Step 3, we generate type-A triples involving one user and two items, by drawing the outcome for c_{uij} from a Bernoulli process, where the parameter is specified by the probability $\text{P}(c_{uij} = 1 \mid x_u, y_i, y_j)$ of generating a triple instance t_{uij}. In Step 4, we generate type-B triples involving two users and one item.

3.2 Triple Probability Function A crucial component is how the latent coordinates of users and items would generate the pairwise comparisons in $\mathcal{T}$. This bridge between the hidden variables and the observations is the triple probability function. To keep the discussion streamlined, in the following we discourse on type-A triples of the form $\langle u, i, j \rangle$, but a similar principle applies in a symmetric manner to type-B triples.

The principle in relating latent coordinates to a triple $\langle u, i, j \rangle$ is: if u prefers i to j, the distance from x_u to y_i is shorter than that from x_u to y_j. The more evidence there is that u prefers i to j, the closer x_u should be to y_i than to y_j. To realize this intuition, we express the probability $\text{P}(c_{uij} = 1 \mid x_u, y_i, y_j)$ in terms of the Euclidean distances $||x_u - y_i||$ and $||x_u - y_j||$. Let Δ_{uij} be a quantity expressed in terms of these distances, such that Δ_{uij} is higher the more u prefers i to j. One realization of Δ_{uij} is Equation 3.1.

$$(3.1) \qquad \Delta_{uij} = ||x_u - y_j|| - ||x_u - y_i||$$
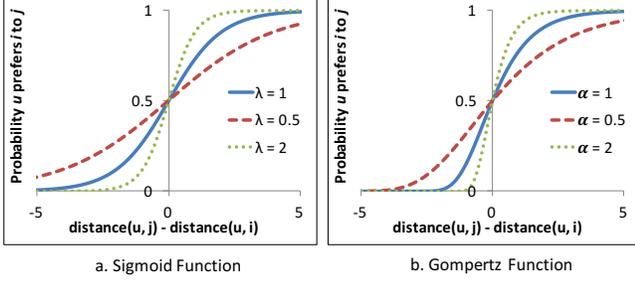
a. Sigmoid Function   b. Gompertz Function

Figure 3: Triple Probability Function

Because t_{uij} and t_{uji} are opposites, we have $P(c_{uij} = 1 \mid x_u, y_i, y_j) = 1 - P(c_{uij} = 0 \mid x_u, y_i, y_j)$. Δ_{uij} has a bearing on these probabilities. For $\Delta_{uij} > 0$, the triple t_{uij} is more likely. For $\Delta_{uij} < 0$, t_{uji} is more likely. For $\Delta_{uij} = 0$, the two triples are equally likely.

To model the probabilities of triples as a function of Δ_{uij} (or Δ_{uvi}), we identify two possible functions.

Sigmoid Function. The first is Sigmoid in Equation 3.2, where λ is a scaling parameter. Figure 3(a) shows that the probability that u prefers i to j tends towards 1 as $\Delta_{uij} \to \infty$, and 0 as $\Delta_{uij} \to -\infty$.

$$(3.2) \qquad P(c_{uij} = 1 \mid x_u, y_i, y_j) = \frac{1}{1 + e^{-\lambda \cdot \Delta_{uij}}}$$

This function allows us to model both a penalty for violating observed triples (probability mass < 0.5), and a reward for preserving observed triples (probability mass > 0.5). This is different from classical ordinal embedding. For instance, the state-of-the-art SOE [21] (see Section 4) only has a penalty component, but no reward. This holds two advantages for COE. First, there is a smoother spectrum of penalty and reward over a continuous function vs. the cliff effect for SOE. Second, there is discrimination among triples with more vs. less evidence earning different probability masses. The scaling parameter λ controls the slope of the function. The greater is λ, the steeper is the penalty/reward. The λ setting may empirically tuned.

Gompertz Function. Sigmoid is symmetrical, which implies that the penalty component is commensurate with the reward component. There may be instances when we seek to model penalty and reward asymmetrically. In particular, we may place greater importance on penalty, i.e., steeper slope for negative Δ_{uij} and gentler slope for positive Δ_{uij}. This can be modeled by the Gompertz function, as shown in Equation 3.3.

$$(3.3) \qquad P(c_{uij} = 1 \mid x_u, y_i, y_j) = a \cdot e^{-b \cdot e^{-\alpha \cdot \Delta_{uij}}}$$

To fit the triple probability function, we set $a = 1$ so as to put the range of values between 0 and 1 (reflecting

probability). Moreover, since $\Delta_{uij} = 0$ correlates with uncertainty of 0.5 probability, we set $b = \ln 2$. In turn, α is a scaling parameter to be tuned. Figure 3(b) shows that the left side $\Delta_{uij} < 0$ has steeper drop, while the right side has gentler gain. In turn the greater α is, the steeper is the slope overall.

3.3 Learning Algorithms Given $\mathcal{T}$ as input observations, our goal is to learn the latent coordinates X and Y with the highest posterior probability $P(X, Y \mid \mathcal{T})$. Through Bayes' Theorem, we have $P(X, Y \mid \mathcal{T}) = P(\mathcal{T}, X, Y) / P(\mathcal{T})$. Since $P(\mathcal{T})$ does not affect the model parameters, the goal is to maximize the joint probability, as shown in Equation 3.4.

$$(3.4) \qquad \arg\max_{X, Y} P(\mathcal{T}, X, Y \mid \gamma, \beta)$$

The joint probability is decomposed into four terms corresponding to the steps in the generative process.

$$P(\mathcal{T}, X, Y \mid \gamma, \beta) = P(X \mid \gamma) \times P(Y \mid \beta) \times P(\mathcal{T} \mid X, Y),$$

$$P(X \mid \gamma) = \prod_{u \in \mathcal{U}} (2\pi\gamma^2)^{-\frac{D}{2}} e^{-\frac{1}{2\gamma^2} \|x_u\|^2},$$

$$P(Y \mid \beta) = \prod_{i \in \mathcal{I}} (2\pi\beta^2)^{-\frac{D}{2}} e^{-\frac{1}{2\beta^2} \|y_i\|^2},$$

$$P(\mathcal{T}_A \mid X, Y) = \prod_{t_{uij} \in \mathcal{T}_A} P(c_{uij} = 1 \mid x_u, y_i, y_j),$$

$$P(\mathcal{T}_B \mid X, Y) = \prod_{t_{uvi} \in \mathcal{T}_B} P(c_{uvi} = 1 \mid x_u, x_v, y_i).$$

Maximizing the joint probability is equivalent to maximizing its logarithm, shown below. To simplify the parameters, we set $\gamma = \beta$, and equate both $\frac{1}{\gamma^2}$ and $\frac{1}{\beta^2}$ to a common regularization parameter η.

$$\mathcal{L} = \ln P(X \mid \gamma) + \ln P(Y \mid \beta) + \ln P(\mathcal{T} \mid X, Y)$$
$$= \ln P(\mathcal{T} \mid X, Y) - \eta \sum_{u \in \mathcal{U}} \|x_u\|^2 - \eta \sum_{i \in \mathcal{I}} \|y_i\|^2$$

To find the coordinates that maximize the joint probability, we employ stochastic gradient ascent for computationally efficiency, an important factor given the potentially huge size of pairwise comparisons.

Sigmoid Function. For the Sigmoid function, the gradient of $\mathcal{L}$ w.r.t. each user coordinate x_u is:

$$\frac{\partial \mathcal{L}}{\partial x_u} = \sum_{\{i,j: \, t_{uij} \in \mathcal{T}_A\}} \frac{\lambda e^{-\lambda \Delta_{uij}}}{1 + e^{-\lambda \Delta_{uij}}} \left(\frac{x_u - y_j}{\|x_u - y_j\|} - \frac{x_u - y_i}{\|x_u - y_i\|} \right)$$

$$+ \sum_{\{i,v: \, t_{uvi} \in \mathcal{T}_B\}} \frac{\lambda e^{-\lambda \Delta_{uvi}}}{1 + e^{-\lambda \Delta_{uvi}}} \left(\frac{y_i - x_u}{\|y_i - x_u\|} \right)$$

$$+ \sum_{\{i,v: \, t_{vui} \in \mathcal{T}_B\}} \frac{\lambda e^{-\lambda \Delta_{vui}}}{1 + e^{-\lambda \Delta_{vui}}} \left(\frac{-y_i + x_u}{\|y_i - x_u\|} \right) - \eta \cdot x_u$$

The gradient w.r.t. each item coordinate y_i is:

$$\frac{\partial \mathcal{L}}{\partial y_i} = \sum_{\{u,v:\ t_{uvi} \in \mathcal{T}_B\}} \frac{\lambda e^{-\lambda \Delta_{uvi}}}{1 + e^{-\lambda \Delta_{uvi}}} \left(\frac{y_i - x_v}{||y_i - x_v||} - \frac{y_i - x_u}{||y_i - x_u||} \right)$$

$$+ \sum_{\{u,j:\ t_{uij} \in \mathcal{T}_A\}} \frac{\lambda e^{-\lambda \Delta_{uij}}}{1 + e^{-\lambda \Delta_{uij}}} \left(\frac{x_u - y_i}{||x_u - y_i||} \right)$$

$$+ \sum_{\{u,j:\ t_{uji} \in \mathcal{T}_A\}} \frac{\lambda e^{-\lambda \Delta_{uji}}}{1 + e^{-\lambda \Delta_{uji}}} \left(\frac{-x_u + y_i}{||x_u - y_i||} \right) - \eta \cdot y_i$$

Algorithm 1 describes the stochastic gradient ascent algorithm for the version COE-S with Sigmoid function. It first initializes the coordinates of users and items. In each iteration, a triple is randomly selected from $\mathcal{T}$, and the model parameters are updated based on the gradients above, with a decaying learning rate ϵ over time. The complexity is $\mathcal{O}(|\mathcal{U}| \times |\mathcal{I}|^2 + |\mathcal{U}|^2 \times |\mathcal{I}|)$. In case of having triples of multi-type ordinal relations among multiple objects, the complexity is still a polynomial of variables with highest degree is 3.

Gompertz Function. For the Gompertz function, the gradient of $\mathcal{L}$ w.r.t. each user coordinate x_u is:

$$\frac{\partial \mathcal{L}}{\partial x_u} = \sum_{\{i,j:\ t_{uij} \in \mathcal{T}_A\}} \alpha \ln(2) e^{-\alpha \Delta_{uij}} \left(\frac{x_u - y_j}{||x_u - y_j||} - \frac{x_u - y_i}{||x_u - y_i||} \right)$$

$$+ \sum_{\{i,v:\ t_{uvi} \in \mathcal{T}_B\}} \alpha \ln(2) e^{-\alpha \Delta_{uvi}} \left(\frac{y_i - x_u}{||y_i - x_u||} \right)$$

$$+ \sum_{\{i,v:\ t_{vui} \in \mathcal{T}_B\}} \alpha \ln(2) e^{-\alpha \Delta_{vui}} \left(\frac{-y_i + x_u}{||y_i - x_u||} \right) - \eta \cdot x_u$$

The gradient w.r.t. each item coordinate y_i is:

$$\frac{\partial \mathcal{L}}{\partial y_i} = \sum_{\{u,v:\ t_{uvi} \in \mathcal{T}_B\}} \alpha \ln(2) e^{-\alpha \Delta_{uvi}} \left(\frac{y_i - x_v}{||y_i - x_v||} - \frac{y_i - x_u}{||y_i - x_u||} \right)$$

$$+ \sum_{\{u,j:\ t_{uij} \in \mathcal{T}_A\}} \alpha \ln(2) e^{-\alpha \Delta_{uij}} \left(\frac{x_u - y_i}{||x_u - y_i||} \right)$$

$$+ \sum_{\{u,j:\ t_{uji} \in \mathcal{T}_A\}} \alpha \ln(2) e^{-\alpha \Delta_{uji}} \left(\frac{-x_u + y_i}{||x_u - y_i||} \right) - \eta \cdot y_i$$

The algorithm and the complexity for the version COE-G with Gompertz function are similar to those for COE-S, but with the corresponding gradients above.

4 Related Work

We now relate to several categories of previous work.

Ordinal Embedding. Given a set of data points, ordinal embedding seeks to preserve the relative comparisons of pairwise distances among data points [11]. In Section 5, we compare to a representative: the state-of-the-art SOE [21], which was shown to be more efficient and accurate than GNMDS [1]. Our key differences from SOE include the explicit modeling of cross-type ordinal relationships, and our probabilistic modeling that has both penalty and reward components. [22] investigated embedding for similarity-based triplets.

Algorithm 1 Stochastic Gradient Ascent for COE-S (with Sigmoid triple probability function)

1: Initialize x_u for $u \in \mathcal{U}$
2: Initialize y_i for $i \in \mathcal{I}$
3: **while** not converged **do**
4:     Draw a triple at random from $\mathcal{T}$.
5:     **if** it is a type-A triple $t_{uij} \in \mathcal{T}_A$ **then**
6:         $x_u \leftarrow x_u + \epsilon \cdot \left[\frac{\lambda e^{-\lambda \Delta_{uij}}}{1 + e^{-\lambda \Delta_{uij}}} \left(\frac{x_u - y_j}{||x_u - y_j||} - \frac{x_u - y_i}{||x_u - y_i||} \right) - \eta \cdot x_u \right]$
7:         $y_i \leftarrow y_i + \epsilon \cdot \left[\frac{\lambda e^{-\lambda \Delta_{uij}}}{1 + e^{-\lambda \Delta_{uij}}} \left(\frac{x_u - y_i}{||x_u - y_i||} \right) - \eta \cdot y_i \right]$
8:         $y_j \leftarrow y_j + \epsilon \cdot \left[\frac{\lambda e^{-\lambda \Delta_{uij}}}{1 + e^{-\lambda \Delta_{uij}}} \left(\frac{-x_u + y_j}{||x_u - y_j||} \right) - \eta \cdot y_j \right]$
9:     **if** it is a type-B triple $t_{uvi} \in \mathcal{T}_B$ **then**
10:    $x_u \leftarrow x_u + \epsilon \cdot \left[\frac{\lambda e^{-\lambda \Delta_{uvi}}}{1 + e^{-\lambda \Delta_{uvi}}} \left(\frac{y_i - x_u}{||y_i - x_u||} \right) - \eta \cdot x_u \right]$
11:    $x_v \leftarrow x_v + \epsilon \cdot \left[\frac{\lambda e^{-\lambda \Delta_{uvi}}}{1 + e^{-\lambda \Delta_{uvi}}} \left(\frac{-y_i + x_v}{||y_i - x_v||} \right) - \eta \cdot x_v \right]$
12:    $y_i \leftarrow y_i + \epsilon \cdot \left[\frac{\lambda e^{-\lambda \Delta_{uvi}}}{1 + e^{-\lambda \Delta_{uvi}}} \left(\frac{y_i - x_v}{||y_i - x_v||} - \frac{y_i - x_u}{||y_i - x_u||} \right) - \eta \cdot y_i \right]$
13: Return $\{x_u\}_{u \in \mathcal{U}}$ and $\{y_i\}_{i \in \mathcal{I}}$

Metric Embedding. Metric embedding seeks to preserve similarity or distance values. In working with preference data, our work is related to CFEE [10], which fits rating values. CFEE expressed a rating $\hat{r}_{ui}$ by user u on item i in terms of the squared Euclidean distance between x_u and y_i. Fitting ratings directly may not necessarily preserve the pairwise comparisons, as we will see in Section 5. In embedding two object types, our work is related to embedding co-occurrences, e.g., documents and words [6] or words and images [24]. The idea is to express co-occurrence frequencies in terms of Euclidean distances. In Section 5 we include a comparison to CODE [6] to show fitting co-occurrences may not preserve comparisons. [13] analyzes generalized convex formulation for co-embedding.

Matrix Factorization. Embedding and matrix factorization are recognized as different problems. The latter's objective is to find a latent vector U for each user and V for each item, such that the inner product $U^T V$ approximates ratings [14] or pairwise comparisons [16, 23]. A tenuous link between *squared* Euclidean distance and inner product, i.e., $||U - V||^2 = ||U||^2 + ||V||^2 - 2U^T V$, does not imply monotonicity because of the vector magnitudes. [2] proposed post facto transformation, by extending output latent vectors by one dimension and using that extra dimension to equalize the magnitude of item vectors. This could only preserve either of user-centric or item-centric triples, but not both. In Section 5, we compare to the composite of BPR [16], followed by [2]'s transformation.

Table 1: Datasets

	users/ docs	items/ words	ratings/ observ- ations	type-A $\langle u, i, j \rangle$ triples	type-B $\langle u, v, i \rangle$ triples
MovieLens	943	1,413	99,543	7.80×10^6	8.22×10^6
Netflix	429,102	17,769	99,841,834	2.68×10^9	2.51×10^{11}
Last.fm	1,772	3,521	72,955	1.50×10^6	3.87×10^6
20News	15,744	14,414	1,076,900	5.61×10^7	2.19×10^8

5 Experiments

Our objective is to investigate the effectiveness of COE, for visualization in low-dimensional Euclidean space.

Datasets. While COE assumes ordinal triples as inputs, we experiment with publicly available datasets with numerical values and derive the triples accordingly. This allows us to compare to baselines that work directly with the numerical values. We work with four datasets of two categories, and their sizes are listed in Table 1.

The first category includes rating-based preference datasets: *MovieLens*[1] and *Netflix*[2]. The object types are users and movies (items). The raw observations are ratings. As in [5], we apply Z-score normalization, which compensates for different rating means and rating spreads to make ratings more comparable across users. We then generate a type-A triple t_{uij} for each instance where a user u has higher normalized rating on an item i than on item j, and a type-B triple t_{uvi} for each instance where a user u has higher normalized rating on i than v does. We do not generate any triple involving non-rated items. For *MovieLens, Netflix*, each user has been pre-conditioned by the original dataset to have at least 20 ratings. We further ensure that each item has at least 4 ratings. We find similar practice in other works [16].

The second category are based on cooccurrences: *Last.fm*[3] and *20News*[4]. *Last.fm* contains users' listening frequencies to music artists (items). As in above, we retain users with at least 20 items, and items with at least 4 users. To show applicability beyond preferences, we include the text-based *20News*, which has documents ("users") and words ("items"). We downloaded the dataset with stop words removed and the remaining words stemmed. Following the standard practice by the baseline [6], we filter out extremely infrequent words (less than 5 documents), and extremely frequent words (top 100 most frequent). For both datasets, the raw observation is the term frequency of a word (or an item) in a document (or a user). To normalize the effect of

document length, we divide each word's frequency by the document length, and generate triples from these normalized term frequencies.

Because of the different natures of the two categories of datasets, which involve some different comparative baselines, in the following we organize the experiments into two sections, one for each dataset category.

5.1 Rating-based Datasets Because the main purpose is visualization, all comparisons are based on embedding in two-dimensional space. We experiment with two versions of our model. The first uses the Sigmoid function, referred to as COE-S. The second uses the Gompertz function, referred to as COE-G.

The first baseline is a representative of the traditional ordinal embedding SOE [21]. We use the authors' implementation[5]. The second baseline is the embedding designed to fit the numerical rating values, i.e., CFEE [10]. As its authors have not made their implementation available, we implement it in Java. The third baseline is matrix factorization based on pairwise comparisons BPR [16] with one dimension, followed by [2]'s Euclidean transformation into two dimensions, denoted as BPR+. For BPR, we use the Java implementation in LibRec[6]. The justifications for the baselines were discussed in Section 4. We tune the respective parameters for the best performance on each dataset.

Metrics. We apply several metrics that allow an evaluation of the various methods in terms of information preservation in two-dimensional Euclidean space.

As is common for dimensionality reduction [9], the primary aim is how well the reduced dimensionality preserves the observed data. The first and main metric is *preservation accuracy*, the extent to which the information within the observed triples is preserved by the coordinates. For a user u, let $\mathcal{T}^u_{observed}$ denote the triples involving u. For u, the preservation accuracy is defined as the fraction of her triples for which the coordinates reflect the preference direction in the triples. Overall, the preservation accuracy is the average of users' preservation accuracies, as shown in Equation 5.5. By doing so, it is not biased towards few users with many ratings at the expense of many users with few ratings.

(5.5)
$$\frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \frac{|\{t_{uij} \in \mathcal{T}^u_{observed} \ : \ ||x_u - y_i|| < ||x_u - y_j||\}|}{|\mathcal{T}^u_{observed}|}$$

As mentioned in Section 2, we do not presume that the input set of triples are complete. It is therefore interesting to study how well the learnt coordinates

---

[1]http://grouplens.org/datasets/movielens/
[2]http://www.cs.uic.edu/~liub/Netflix-KDD-Cup-2007.html
[3]http://files.grouplens.org/datasets/hetrec2011/hetrec2011-lastfm-2k.zip
[4]http://web.ist.utl.pt/acardoso/datasets/

[5]http://rpackages.ianhowson.com/cran/loe/man/SOE.html
[6]http://www.librec.net/

Table 2: Rating-based Dataset (MovieLens - 100K Sample): COE vs. Ordinal Embedding

	Preservation Accuracy			Prediction Accuracy			1-NN Avg Rating			5-NN Avg Rating		
	Type-A	Type-B	**H-Mean**	Type-A	Type-B	**H-Mean**	Users	Items	**H-Mean**	Users	Items	**H-Mean**
COE-S	70.1%	57.3%	**63.0%**	62.7%	57.4%	**59.9%**	4.38	3.66	**3.99**	4.24	3.48	**3.82**
COE-G	70.0%	57.5%	**63.2%**	62.8%	57.9%	**60.2%**	4.41	3.67	**4.01**	4.24	3.48	**3.82**
SOE	69.4%	55.9%	**61.9%**	62.5%	56.0%	**59.1%**	4.29	3.44	**3.82**	4.22	3.38	**3.75**

could generalize to unseen triples. We introduce a secondary metric, *prediction accuracy*, the extent to which the coordinates can infer the preference directions of hidden triples $\mathcal{T}_{hidden}$. For an embedding solution as a whole, the prediction accuracy is derived from user-level accuracies, as shown in Equation 5.6.

$$(5.6) \quad \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \frac{|\{t_{uij} \in \mathcal{T}^u_{hidden} \; : \; ||x_u - y_i|| < ||x_u - y_j||\}|}{|\mathcal{T}^u_{hidden}|}$$

The above definitions are for type-A triples. A corresponding version is defined for type-B triples. We will present the results both types separately, as well as together by taking their harmonic mean (H-Mean).

We split the ratings randomly into 80% $\mathcal{R}_{observed}$ and 20% $\mathcal{R}_{hidden}$, in a stratified manner to maintain the same ratio for every user. The observed set of triples $\mathcal{T}_{observed}$ are formed within $\mathcal{R}_{observed}$. The hidden set of triples $\mathcal{T}_{hidden}$ include triples formed within $\mathcal{R}_{hidden}$, as well as triples involving one rating each from $\mathcal{R}_{observed}$ and $\mathcal{R}_{hidden}$. Ordinal-based methods learn from $\mathcal{T}_{observed}$, while the rest learn from with $\mathcal{R}_{observed}$. Both preservation and prediction accuracies range from 0% (worst) to 100% (best). For statistical significance, we average the results across 10 random (80:20) splits.

These metrics are general for ordinal triples. Since the ordinal triples are derived from ratings, we include a rating-based third measure: *average rating among k-nearest neighbors* (*k*-NN). Intuitively, a good embedding with high preservation should place higher-rated items closer to the user. Given a user, we identify the *k*-nearest rated items based on their Euclidean distances in the embedding space, and average the user's ratings on those items. Symmetrically, this can be measured from each item's point of view. We average this across users and items respectively for $k = 1$ and $k = 5$.

Versus Ordinal Embedding. Existing ordinal embedding packages do not scale to large datasets. The author implementation of SOE limits the number of input size to 100K. We sample 100K triples from $\mathcal{T}_{observed}$, and use them to compare SOE and COE. Yet, this is only applicable to *MovieLens*, as SOE cannot cope with the number of users and items in *Netflix*.

Table 2 shows the performance of the methods on the 100K sample of MovieLens for both type-A and type-B triples. Focusing on the overall figures (harmonic mean in bold), we see that the preservation accuracies of COE-S and COE-G are similar at 63.0% and 63.2%. Both are higher than SOE's 61.9%, whose lower performance is statistically significant. For prediction accuracies, the figures are slightly lower overall, but the relative trend is the same. For visualization based on dimensionality reduction, preservation is the greater objective, as the intent is to represent the observed data.

Table 2 also shows the comparison of the average rating among 1-nearest neighbors (1-NN), as well as 5-NN. Again, we take the harmonic mean (H-Mean) between users' and items' rating averages. Evidently, the nearest neighbors around every user or item tend to have high ratings (in the scale of 1 to 5). COE-G and COE-S are similar, while SOE is significantly lower.

Versus Other Baselines. In Table 3, we employ the full data to compare to the other baselines. COE-S and COE-G have significantly higher results in Table 3, because they run with the full set of observed triples.

CFEE, which fits rating values directly, generally achieves lower accuracies. Since rating and visualization spaces are distinct, forcing their unification may not obtain the best embedding to preserve the triples. BPR+, which learns matrix factorization by pairwise ranking, followed by Euclidean transformation, also achieves lower results. As mentioned in Section 4, the Euclidean transformation applied to BPR's output could only preserve the pairwise comparisons of either type-A triples or type-B triples (not both at once). However, we present the best results for both transformations, which evidently are still lower than COE's. This signifies that for visualization, directly modelling Euclidean distance, such as in COE, leads to better visualization.

Table 4 shows the results for the much-larger *Netflix* dataset, which also support the major observations made above. The differences between COE's variants and the baselines are statistically significant.

Visualization. Figure 4 shows an example of three users U887 (blue), U222 (red), U903 (green) in *MovieLens*, and the 17 items (crosses) that all three have rated. For instance, U222 and U903 are closer to Fargo (which they rated 5) than U887 is (who rated it 2). Interestingly, U222 is closer to U903 than U222 is to U887, supported by the Pearson correlation of their

Table 3: Rating-based Dataset (MovieLens): COE vs. Other Baselines

	Preservation Accuracy			Prediction Accuracy			1-NN Avg Rating			5-NN Avg Rating		
	Type-A	Type-B	**H-Mean**	Type-A	Type-B	**H-Mean**	Users	Items	**H-Mean**	Users	Items	**H-Mean**
COE-S	75.0%	65.0%	**69.6%**	64.0%	59.0%	**61.4%**	4.48	3.93	**4.19**	4.33	3.58	**3.92**
COE-G	75.0%	65.0%	**69.6%**	64.0%	59.0%	**61.4%**	4.48	3.87	**4.15**	4.33	3.55	**3.90**
CFEE	67.2%	62.4%	**64.7%**	59.7%	60.3%	**60.0%**	4.07	3.63	**3.84**	4.03	3.50	**3.75**
BPR+	68.4%	60.9%	**64.5%**	62.1%	59.1%	**60.5%**	4.14	3.63	**3.87**	4.13	3.40	**3.73**

Table 4: Rating-based Dataset (Netflix): COE vs. Other Baselines

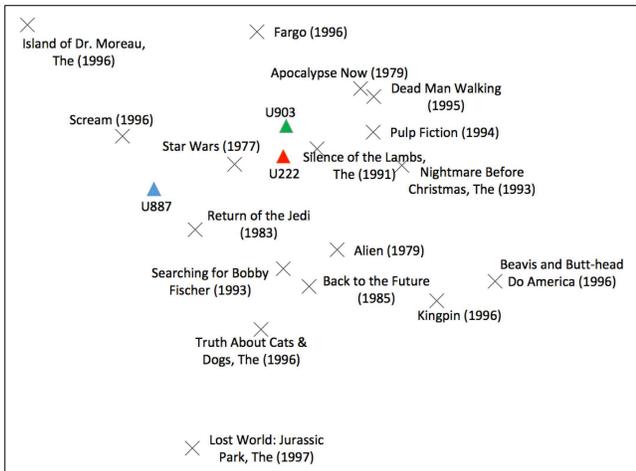|	Preservation Accuracy			Prediction Accuracy			1-NN Avg Rating			5-NN Avg Rating		
	Type-A	Type-B	**H-Mean**	Type-A	Type-B	**H-Mean**	Users	Items	**H-Mean**	Users	Items	**H-Mean**
COE-S	75.2%	66.3%	**70.4%**	63.3%	61.2%	**62.2%**	4.63	4.06	**4.32**	4.51	3.74	**4.09**
COE-G	74.9%	65.5%	**69.9%**	63.1%	60.7%	**61.9%**	4.66	4.05	**4.34**	4.52	3.72	**4.08**
CFEE	66.0%	62.4%	**64.2%**	58.9%	61.4%	**60.2%**	4.15	3.93	**4.04**	4.10	3.74	**3.91**
BPR+	68.2%	60.2%	**64.0%**	60.3%	58.8%	**59.6%**	4.07	3.16	**3.56**	4.00	3.15	**3.52**

Figure 4: Example Visualization of Users (triangles) and Items (crosses) in MovieLens

ratings on items: 0.31 between (U222, U903), and -0.21 between (U222, U887). The layout of movies are also intuitive. Horror films Scream and Island of Dr. Moreau are on the top left. Science fictions Star Wars, Return of the Jedi, and Back to the Future are at the centre. Darker dramas Fargo, Apocalypse Now are on the top right. Comedies such as Kingpin and Beavis and Butt-head are on the far right. Family-oriented Searching for Bobby Fischer and Lost World are towards the bottom.

Efficiency is not our major focus here. The learning algorithms can be run offline. On MovieLens and LastFM, COE takes approximately a minute on a PC with Intel Core i5 3.2GHz CPU and 12GB RAM. For 20News, the running time of COE is around 15 minutes. Our efficiency is comparable to other models running on pairwise comparisons, e.g., BPR, and is much faster than ordinal embedding, i.e., SOE.

5.2 Cooccurrence-based Datasets We now discuss the comparisons for the other two datasets based on cooccurrences: *Last.fm* and *20News*. Here, we focus on the comparison to CODE [6], which fits co-occurrence frequencies. We use the implementation[7] by its author.

For the metrics, we again rely on preservation and prediction accuracies. In addition, we adapt the "average rating" concept to the cooccurrence scenario. Since the raw observation is normalized term frequency, we evaluate the average term frequencies among the k-nearest neighbors of a document or a word respectively. The higher it is, the more successful is the embedding in placing the closest words to a document (vice versa).

Table 5 for *Last.fm* and Table 6 for *20News* show that both COE versions have significantly higher preservation and prediction accuracies than the baseline CODE. This experiment showcases that the information within ordinal triples is not easily approximated by fitting probabilities of co-occurrences (which is semantically closer to similarity/distance-based embedding). This is also evident from the comparison of average normalized term frequencies among the k-NN. The values seem deceptively low, these frequencies are actually high, considering that each document consists of many words. For instance, in Table 6, COE achieves 0.050 for $k = 1$, which implies that the nearest word to a document is expected to cover 5% of the document.

We have also compared to ordinal embedding SOE, and COE is also better than SOE on these datasets.

6 Conclusion

We address the problem of ordinal co-embedding based on cross-type ordinal relationships, whereby every user and every item is respectively associated with a la-

---

[7] http://ai.stanford.edu/~gal/

Table 5: Cooccurrence-based Dataset (Last.fm): COE vs. Cooccurrence Embedding

	Preservation Accuracy			Prediction Accuracy			1-NN Avg Frequency			5-NN Avg Frequency		
	Type-A	Type-B	**H-Mean**	Type-A	Type-B	**H-Mean**	Users	Items	**H-Mean**	Users	Items	**H-Mean**
COE-S	64.5%	85.6%	**73.5%**	51.7%	63.2%	**56.9%**	0.048	0.047	**0.047**	0.041	0.032	**0.036**
COE-G	64.0%	85.7%	**73.3%**	51.4%	63.1%	**56.6%**	0.048	0.047	**0.047**	0.040	0.032	**0.036**
CODE	53.3%	52.8%	**53.1%**	49.8%	54.7%	**52.2%**	0.032	0.031	**0.032**	0.032	0.032	**0.032**

Table 6: Cooccurrence-based Dataset (20News): COE vs. Cooccurrence Embedding

	Preservation Accuracy			Prediction Accuracy			1-NN Avg Frequency			5-NN Avg Frequency		
	Type-A	Type-B	**H-Mean**	Type-A	Type-B	**H-Mean**	Docs	Words	**H-Mean**	Docs	Words	**H-Mean**
COE-S	78.9%	90.3%	**84.3%**	51.0%	69.2%	**58.7%**	0.050	0.049	**0.050**	0.039	0.029	**0.037**
COE-G	77.0%	88.0%	**82.1%**	50.8%	68.7%	**58.4%**	0.049	0.047	**0.048**	0.038	0.028	**0.036**
CODE	59.7%	56.2%	**57.9%**	48.7%	52.8%	**50.7%**	0.035	0.022	**0.027**	0.033	0.020	**0.025**

tent coordinate in a low-dimensional Euclidean space. The objective is to place a user closer to a more preferred item. This accommodates datasets including ratings and co-occurrences. Experiments on public datasets show that Collaborative Ordinal Embedding or COE outperforms comparable baselines in information preservation in the low-dimensional visualization space.

Acknowledgments

References

[1] S. Agarwal, J. Wills, L. Cayton, G. Lanckriet, D. J Kriegman, and S. Belongie. Generalized non-metric multidimensional scaling. In *AISTATS*, 2007.

[2] Y. Bachrach, Y. Finkelstein, R. Gilad-Bachrach, L. Katzir, N. Koenigstein, N. Nice, and U. Paquet. Speeding up the xbox recommender system using a euclidean transformation for inner-product spaces. In *RecSys*, 2014.

[3] N. Beckmann, H.-P. Kriegel, R. Schneider, and B. Seeger. The R*-tree: An efficient and robust access method for points and rectangles. In *SIGMOD*, 1990.

[4] L. Van der Maaten and G. Hinton. Visualizing data using t-SNE. *JMLR*, 9, 2008.

[5] M. D. Ekstrand, J. T. Riedl, and J. A. Konstan. Collaborative filtering recommender systems. *Foundations and Trends in Human-Computer Interaction*, 4(2):81–173, 2011.

[6] A. Globerson, G. Chechik, F. Pereira, and N. Tishby. Euclidean embedding of co-occurrence data. *JMLR*, 8:2047–2076, 2007.

[7] Y. Hu, Y. Koren, and C. Volinsky. Collaborative filtering for implicit feedback datasets. In *ICDM*, 2008.

[8] T. Joachims. Training linear svms in linear time. In *KDD*, 2006.

[9] I. Jolliffe. *Principal Component Analysis*. Wiley Online Library, 2005.

[10] M. Khoshneshin and W. N. Street. Collaborative filtering via euclidean embedding. In *RecSys*, 2010.

[11] J. B. Kruskal. Nonmetric multidimensional scaling: a numerical method. *Psychometrika*, 29(2), 1964.

[12] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to information retrieval*. 2008.

[13] F. Mirzazadeh, Y. Guo, and D. Schuurmans. Convex co-embedding. In *AAAI*, 2014.

[14] A. Mnih and R. Salakhutdinov. Probabilistic matrix factorization. In *NIPS*, 2007.

[15] F. Radlinski and T. Joachims. Query chains: learning to rank from implicit feedback. In *KDD*, 2005.

[16] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme. BPR: Bayesian personalized ranking from implicit feedback. In *UAI*, 2009.

[17] N. Roussopoulos, S. Kelley, and F. Vincent. Nearest neighbor queries. In *SIGMOD*, 1995.

[18] S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290, 2000.

[19] R. N. Shepard. The analysis of proximities: Multidimensional scaling with an unknown distance function. i. *Psychometrika*, 27(2), 1962.

[20] J. B. Tenenbaum, V. De Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290, 2000.

[21] Y. Terada and U. V Luxburg. Local ordinal embedding. In *ICML*, 2014.

[22] Laurens Van der Maaten and Kilian Weinberger. Stochastic triplet embedding. In *MLSP*, pages 1–6, 2012.

[23] M. Weimer, A. Karatzoglou, Q. V. Le, and A. Smola. Cofirank - maximum margin matrix factorization for collaborative ranking. In *NIPS*, 2007.

[24] J. Weston, S. Bengio, and N. Usunier. Wsabie: Scaling up to large vocabulary image annotation. In *IJCAI*, 2011.