Active Ordinal Querying for Tuplewise Similarity Learning

Gregory Canal,[*1] **Stefano Fenu,**[*2] **Christopher Rozell**[1]
[1]School of Electrical and Computer Engineering
[2]School of Interactive Computing
Georgia Institute of Technology, Atlanta, GA
{gregory.canal, sfenu3, crozell}@gatech.edu

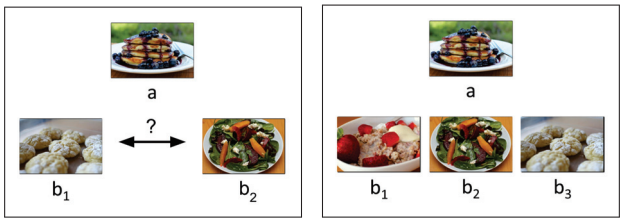
Abstract

Many machine learning tasks such as clustering, classification, and dataset search benefit from embedding data points in a space where distances reflect notions of relative similarity as perceived by humans. A common way to construct such an embedding is to request triplet similarity queries to an oracle, comparing two objects with respect to a reference. This work generalizes triplet queries to tuple queries of arbitrary size that ask an oracle to rank multiple objects against a reference, and introduces an efficient and robust adaptive selection method called InfoTuple that uses a novel approach to mutual information maximization. We show that the performance of InfoTuple at various tuple sizes exceeds that of the state-of-the-art adaptive triplet selection method on synthetic tests and new human response datasets, and empirically demonstrate the significant gains in efficiency and query consistency achieved by querying larger tuples instead of triplets.

Introduction

Similarity learning is the process of assigning point coordinates to objects in a dataset such that distances between objects in the learned space are consistent with notions of similarity as perceived by humans. While these objects usually exist in some high-dimensional space (e.g., images, audio), very often the semantic information humans attribute to these objects lies in a low-dimensional space (e.g., items, words). Once this low-dimensional embedding is learned, existing intelligent algorithms (Jamieson and Nowak 2011a; Canal et al. 2019) can be used to search the dataset with query complexity scaling in the embedding dimension, allowing large datasets to be searched quickly in applications such as task selection for robot learning from demonstration (Argall et al. 2009), object recognition (Ferrari, Tuytelaars, and Van Gool 2004), or image retrieval (Yang et al. 2010).

To construct such an embedding for a given set of objects, queries that capture the similarity statistics between the objects in question must be made to human experts. While there exist several types of similarity queries that can be made (e.g., relative attributes between objects (Parikh and Grauman 2011)), we focus on relative similarity queries posed to an oracle comparing objects with respect to a "head" (i.e., reference) object. Relative similarity queries are useful because they gather object similarity information using only object identifiers rather than predetermined features or attributes, allowing similarity learning methods to be applied to any collection of uniquely identifiable objects. In contrast, if a head object were not specified, an oracle would need to use a feature-based criterion for ranking the object set, which is not viable in many applications of interest (e.g., learning human preferences).



(a) (b)

Figure 1: In 1(a) it is ambiguous which item should be chosen as more similar to the head object, since both comparison items are similar in distinct ways. In 1(b), adding one more comparison item can add context to disambiguate this choice.

Such relative similarity queries typically come in the form of triplet comparisons (i.e., "is object a more similar to object b or c?") (Tamuz et al. 2011; Van Der Maaten and Weinberger 2012; Hoffer and Ailon 2015). *In our first main contribution, we extend these queries to larger rank orderings of tuples of k objects to gather more information at once for similarity learning*. This query type takes the form "rank objects b_1 through b_{k-1} according to their similarity to object a." To the best of our knowledge, this study is the first attempt to leverage this generalized query type in similarity learning. The use of this query type is motivated by the fact that comparing multiple objects simultaneously provides increased context for a human expert (Fernando et al. 2015), which can increase labeling consistency without a significant increase in human effort per query (Liang and Grauman 2014) and has demonstrated benefits in settings such as rank learning (Cao et al. 2007). In technical terms, tuplewise queries capture joint dependence between objects that isn't captured in

[*]Equal contribution.

triplet comparisons (which are often incorrectly modeled as independent queries). To illustrate this point, consider the difference between the triplet query and tuple query presented in Figure 1. In the triplet query, multiple attributes could be used to rank a given query, increasing the ambiguity about which item should be chosen as more similar to the reference. Adding an item to the tuple can provide additional context about the entire dataset to the oracle, clarify which criterion should be used to rank the tuple and thereby making the query less ambiguous.

While tuple queries are appealing, their use presents two major challenges. First, in a dataset of N objects queried with tuples of size k there are $N\binom{N-1}{k-1}$ possible tuples. Labeling these individual tuples is prohibitively time consuming for large datasets. Even if uniformly random query selection is used to downsample this set, there is evidence that such a strategy is still punitively expensive (Jamieson and Nowak 2011b). Requesting an exhaustive number of queries is also inefficient from an information standpoint, since there is redundancy in the set of all tuple rankings. Second, in many settings of interest, the oracle answering such queries may be stochastic. For example, crowd oracles may aggregate responses from experts with differing similarity judgements (Tamuz et al. 2011), and individual oracles can be unreliable over time (especially for queries regarding similar objects).

These issues can be ameliorated in part by leveraging tools from *active learning*, the goal of which is to minimize the total labeling cost including the number of expert interactions (usually corresponding to monetary cost), aggregate response time, and computational cost needed to dynamically select queries. This is achieved through adaptive approaches that increase learning efficiency by using previous query responses to determine which information about a model is still "missing" as well as model the oracle's stochasticity. In this framework, unlabeled data points that optimize a measure of informativeness are selected for expert labeling. One such metric, *mutual information*, is a popular way to assess the reduction in uncertainty a query provides about unknown learning parameters (Settles 2012; Lindley 1956; MacKay 1992). In active similarity learning, the state-of-the-art is a strategy called "Crowd Kernel Learning" (CKL) that selects triplets that maximize the mutual information between a query response and the embedding coordinates of the head object (Tamuz et al. 2011). However, CKL does not apply to ordinal queries of general tuples sizes ($k > 3$), and its formulation of mutual information only measures the information a query provides about the embedding coordinates of the head object, disregarding information about the locations of the other objects in the query.

In our second main contribution, we address these deficiencies and the lack of an active similarity learning strategy for our new query type by introducing a novel method for efficient and robust adaptive selection of tuplewise queries of arbitrary size. Our method, called *InfoTuple*, maximizes the mutual information a query response provides about the *entire* embedding, which is a direct measure of query informativeness that leverages the high degree of coupling between all of the objects in a query. InfoTuple relies on a novel set of simplifying yet reasonable assumptions for tractable mutual information estimation from a single batch of Monte Carlo samples. Our approach accounts for all objects in a query, while avoiding the need to decompose mutual information into a prohibitive number of terms. We demonstrate the performance of this method across datasets, oracle models, and tuple sizes, using both synthetic tests and newly collected large-scale human response datasets. In particular, we empirically show that InfoTuple's performance exceeds that of CKL and random queries, and furthermore that it benefits significantly from using larger tuples even after normalizing for tuple size. We also demonstrate the utility of our novel query type by showing an increase in query consistency for larger tuples over triplets, and show that these advantages can be gained without excessive labeling-time increases.

Related Work

Similarity learning from triplets is increasingly commonplace in modern AI, and popular deep learning architectures have been developed to leverage triplet labels (Hoffer and Ailon 2015). Frameworks such as that of (Liu et al. 2012) or t-STE (Van Der Maaten and Weinberger 2012) are relatively ubiquitous in the visualization community, and attempt to directly capture a notion of visual similarity close to that observed in psychometrics literature (e.g. (Chater and Brown 1999)). However, for large datasets it is often punitively expensive to collect such exhaustive relationship data from labelers, so the development of approximate methods of learning such embeddings is a matter of interest to the AI community.

The bulk of the existing literature on active selection of ordinal queries for constructing these embeddings focuses on the case where distance relationships between objects can be determined with absolute certainty. This deterministic case is well studied, and lower bounds exist on the sample complexity needed to learn high-quality embeddings (Jamieson and Nowak 2011b). In reality, responses are often not deterministic for a number of practical reasons and probabilistic MDS methods have been proposed to model such cases (Tamuz et al. 2011). Analytic results do exist characterizing bounds on prediction error in this setting (Jain, Jamieson, and Nowak 2016), but determining optimal strategies for query selection in the stochastic setting remains largely an open problem.

Specifically, to the best of our knowledge there have been no previous attempts to adaptively select relative comparisons with respect to a head object for general tuple sizes ($k \geq 3$) in the context of similarity learning. Prior work (Liang and Grauman 2014; Yu 2005) develops an active strategy for sampling tuples, but the query task is relative attribute ranking within the tuple according to some pre-specified attribute as opposed to comparison against a head object. Other work (Qian et al. 2013) actively samples the same query type as our study, but in the context of classification via label propagation. Research exists that is similar to our learning scenario since they actively sample tuples for relative similarity comparisons to a head for the sake of learning and searching an embedding of objects (Cao and Ai 2015), but these comparisons are ternary 'similar', 'dissimilar', or 'neither' labels and their methodology differs from the mutual information approach presented here. Similarly, other work (Patterson et al. 2015) actively

samples tuplewise queries with binary 'similar' or 'dissimilar' label responses with respect to a head, but in the context of classification. Finally, the prior work (Wilber, Kwak, and Belongie 2014) also employs such tuplewise binary queries for similarity learning, but with randomly selected queries. While no previous study addresses the similarity learning problem that we explore here, the existing literature demonstrates the effectiveness, efficiency, and feasibility of queries involving multiple objects and provides support for the practical use of our proposed query type.

Methods

The problem of adaptively selecting a tuplewise query can be formulated as follows: for a dataset $\mathcal{X}$ of N objects, assume that there exists a d-dimensional vector of embedding coordinates for each object which are concatenated as columns in matrix $M \in \mathbb{R}^{d \times N}$. The *similarity matrix* corresponding to M is given by $K = M^T M$, which implies an $N \times N$ matrix D of distances between the objects in $\mathcal{X}$. Specifically, the squared distance between the ith and jth objects in the dataset is given by $D^2_{i,j} = K_{i,i} - 2K_{i,j} + K_{j,j}$. These distances are assumed to be consistent in expectation with similarity comparisons from an *oracle* (e.g., human expert or crowd) such that similar objects are closer and dissimilar objects are farther apart. Since relative similarity comparisons between tuples of objects inform their relative embedding distances rather than their absolute coordinates, our objective is to learn similarity matrix K rather than M, which can be recovered from K up to a change in basis (Tamuz et al. 2011).

A tuplewise oracle query at time step n is composed of a "body" of objects $B^n = \{b^n_1, b^n_2, \dots b^n_{k-1}\}$ which the oracle ranks by similarity with respect to some "head" object a_n. Letting $Q_n = \{a_n\} \cup B^n$ denote the nth posed tuple, we denote the oracle's ranking response as $R(Q_n) = \{R_1(Q_n), R_2(Q_n), \dots R_{k-1}(Q_n)\}$ which is a permutation of B^n such that $R_1(Q_n) \prec R_2(Q_n) \cdots \prec R_{k-1}(Q_n)$ where $b_i \prec b_j$ indicates that the oracle ranks object b_i as more similar to a_n than object b_j. Since the oracle is assumed to be stochastic, $R(Q_n)$ is a random permutation of B^n governed by a distribution that is assumed to depend on K. This assumed dependence is natural because oracle consistency is likely coupled with notions of object similarity, and therefore with distances between the objects in M. The actual recorded oracle ranking is a random variate of $R(Q_n)$ denoted as $r(Q_n)$. Letting $r^n = \{r(Q_1), r(Q_2), \dots r(Q_n)\}$, define $\widehat{K}^n$ as an estimate of K learned from previous rankings r^n, with corresponding distance matrix $\widehat{D}^n$.

Suppose that tuples $Q_1, Q_2, \dots Q_{n-1}$ have been posed as queries to the oracle with corresponding ranking responses r^{n-1}, and consider a Bayes optimal approach where after the nth query we estimate the similarity matrix as the maximum a-posteriori (MAP) estimator over a similarity matrix posterior distribution given by $f(K|r^n)$, i.e. $\widehat{K}_n = \arg\max_K f(K|r^n)$. To choose the query Q_n, a reasonable objective is to select a query that maximizes the achieved posterior value of the resulting MAP estimator (or equivalently one that maximizes the achieved logarithm of the posterior), corresponding to a higher level of confidence in the estimate. However, because the oracle response $r(Q_n)$ is unknown before a query is issued, the resulting maximized posterior value is unknown. Instead, a more reasonable objective is to select a query that maximizes the *expected* value over the posterior of $R(Q_n)$. This can be stated as

$$\arg\max_{Q_n} \mathop{\mathbb{E}}_{R(Q_n)} \left[\max_K \log f(K|R(Q_n), r^{n-1}) \mid r^{n-1} \right].$$

In practice, this optimization is infeasible since each expectation involves the calculation of several MAP estimates. Noting that maximization is lower bounded by expectation, this optimization can be relaxed by replacing the maximization over K with an expectation over its posterior distribution given $R(Q_n)$ and r^{n-1}, resulting in a feasible maximization of a lower bound given by

$$\arg\max_{Q_n} -h(K \mid R(Q_n), r^{n-1}), \tag{1}$$

where $h(K \mid R(Q_n), r^{n-1})$ denotes *conditional differential entropy* (Cover and Thomas 2012). Let the *mutual information* between K and $R(Q_n)$ given r^{n-1} be defined by

$$I(K; R(Q_n) \,|\, r^{n-1}) = h(K \,|\, r^{n-1}) - h(K \,|\, R(Q_n), r^{n-1}),$$

and note that the second term is equal to (1) while the first term does not depend on the choice of Q_n. Thus, maximizing (1) over Q_n is equivalent to maximizing $I(K, R(Q_n) \mid r^{n-1})$. Hence, we can adaptively select tuples that maximize mutual information as a means of greedily maximizing a lower bound on the log-posterior achieved by a MAP estimator, corresponding to a high estimator confidence.

However, calculating (1) for a candidate tuple is an expensive procedure that involves estimating the differential entropy of a combinatorially large number of posterior distributions, since the expectation with respect to $R(Q_n)$ is taken over $(k-1)!$ possible rankings. Instead, in the spirit of (Houlsby et al. 2012) we leverage the symmetry of mutual information to write the equivalent objective

$$\arg\max_{Q_n} H(R(Q_n) \mid r^{n-1}) - H(R(Q_n) \mid K, r^{n-1}) \tag{2}$$

where $H(\cdot \mid \cdot)$ denotes conditional entropy of a discrete random variable. Estimating (2) for a candidate tuple only involves averaging ranking entropy over a *single* posterior $f(K \mid r^{n-1})$, regardless of the value of k. This insight, along with suitable probability models discussed in the next sections, allows us to efficiently estimate mutual information for a candidate tuple over a single batch of Monte Carlo samples, rather than having to sample from $(k-1)!$ posteriors.

Furthermore, by interpreting entropy of discrete random variables as a measure of uncertainty, this form of mutual information maximization has a satisfying qualitative interpretation. The first entropy term in (2) prefers tuples whose rankings are uncertain, preventing queries from being wasted on predictable or redundant responses. Meanwhile, the second term discourages tuples that have high expected uncertainty when conditioned on K; this prevents the selection of tuples that, even if K were somehow revealed, would *still* have uncertain rankings. Such queries are inherently

ambiguous, and therefore uninformative to the embedding. Thus, maximizing mutual information optimizes the balance between these two measures of uncertainty and therefore prefers queries that are unknown to the learner but that can still be answered consistently by the oracle.

Estimating Mutual Information

To tractably estimate the entropy terms in (2) for a candidate tuple, we employ several simplifying assumptions concerning the joint statistics of the query sequence and the embedding that allow for efficient Monte Carlo sampling:

(A1) As is common in active learning settings, we assume that each query response $R(Q_n)$ is statistically independent of previous responses r^{n-1}, when conditioned on K.

(A2) The distribution of $R(Q_n)$ conditioned on K is only dependent on the distances between a_n and the objects in B^n, notated as set $D_{Q_n} := \{D_{a_n,b} : b \in B\}$. This direct dependence of tuple ranking probabilities on inter-object distances is rooted in the fact that the distance relationships in the embedding are assumed to capture oracle response behavior, and is a common assumption in ordinal embedding literature (Van Der Maaten and Weinberger 2012; Tamuz et al. 2011). Furthermore, this conditional independence of $R(Q_n)$ from objects $x \notin Q_n$ is prevalent in probabilistic ranking literature (Stern 1990). In the next section, we describe a reasonable ranking probability model that satisfies this assumption.

(A3) D is conditionally independent of r^{n-1}, given $\widehat{D}^{n-1}$. This assumption is reasonable because embedding methods used to estimate $\widehat{K}^{n-1}$ (and subsequently $\widehat{D}^{n-1}$) are designed such that distances in the estimated embedding preserve the response history contained in r^{n-1}. In practice, it is more convenient to model an embedding posterior distribution by conditioning on $\widehat{D}^{n-1}$, learned from the previous responses r^{n-1}, rather than by conditioning on r^{n-1} itself. This is in the same spirit of CKL, where the current embedding estimate is used to approximate a posterior distribution over points.

(A4) Conditioned on $\widehat{D}^{n-1}$, the posterior distribution of D_{Q_n} is normally distributed about the corresponding values in $\widehat{D}^{n-1}_{Q_n}$, i.e. $D^{n-1}_{a_n,b} \sim \mathcal{N}(\widehat{D}^{n-1}_{a_n,b}, \sigma^2_{n-1})\ \forall b \in B$, where σ^2_{n-1} is a variance parameter. Imposing Gaussian distributions on inter-object distances is a recent approach to modeling uncertainty in ordinal embeddings (Lohaus, Hennig, and von Luxburg 2019) that allows us to approximate the distance posterior with a fixed batch of samples from a simple distribution. Furthermore, the combination of this model with (A2) means that we only need to sample from the normal distributions corresponding to the objects in Q_n. We choose σ^2_{n-1} to be the sample variance of all entries in $\widehat{D}^{n-1}$, which is a heuristic that introduces a source of variation that preserves the scale of the embedding.

Combining these assumptions, with a slight abuse of notation by writing $H(X) = H(p(X))$ for a random variable X with probability mass function $p(X)$, and $\mathcal{N}^{n-1}_{Q_n}$ to represent normal distribution $\mathcal{N}(\widehat{D}^{n-1}_{Q_n}, \sigma^2_{n-1})$, we have

$$
\begin{aligned}
H(R(Q_n) \mid r^{n-1}) &= H\left(\mathop{\mathbb{E}}_{K}\left[p(R(Q_n) \mid K, r^{n-1}) \mid r^{n-1}\right]\right) \\
&= H\left(\mathop{\mathbb{E}}_{K}\left[p(R(Q_n) \mid K) \mid r^{n-1}\right]\right) \quad \text{(A1)} \\
&= H\left(\mathop{\mathbb{E}}_{D_{Q_n}}\left[p(R(Q_n) \mid D_{Q_n}) \mid r^{n-1}\right]\right) \quad \text{(A2)} \\
&= H\left(\mathop{\mathbb{E}}_{D_{Q_n}}\left[p(R(Q_n) \mid D_{Q_n}) \mid \widehat{D}^{n-1}\right]\right) \quad \text{(A3)} \\
&= H\left(\mathop{\mathbb{E}}_{D_{Q_n} \sim \mathcal{N}^{n-1}_{Q_n}}\left[p(R(Q_n) \mid D_{Q_n})\right]\right) \quad \text{(A4)}
\end{aligned}
$$

Similarly, we have

$$
H(R(Q_n) \mid K, r^{n-1}) = \mathop{\mathbb{E}}_{D_{Q_n} \sim \mathcal{N}^{n-1}_{Q_n}}\left[H\left(p(R(Q_n) \mid D_{Q_n})\right)\right]
$$

This formulation allows a fixed-sized batch of samples to be drawn and evaluated over, the size of which can be tuned based on real-time performance specifications. This enables us to separate our computational budget and mutual information estimation accuracy from the size of the tuple query.

Embedding Technique

In order to maximize the flexibility of our approach and draw a closer one-to-one comparison to existing methods for similarity learning, we train our embedding on our actively selected tuples by first decomposing a tuple ranking into $k-2$ *constituent triplets* defined by the set $\{R_i(Q_m) \prec R_{i+1}(Q_m) : 1 \le i \le k-2,\ m \le n\}$, and then learning an embedding from these triplets with any triplet ordinal embedding algorithm of choice. Since we compare performance against CKL in our experiments, our proposed embedding technique follows directly from the probabilistic MDS formulation in (Tamuz et al. 2011) so as to evaluate the effectiveness of our novel query selection strategy in a controlled setting. We wish to constrain our learned similarity matrix to the set of symmetric unit-length PSD matrices, so we consider the set S of such matrices: $S = \{K \succeq 0 | K_{11} = K_{22} = \cdots = K_{NN} = 1\}$. We denote the closest matrix in S to K as $P_S(K) = \arg\min_{A \in S} \sum_{ij}(K_{ij} - A_{ij})^2$. Projecting to the element in S closest to K is a quadratic program, which we solve by gradient projection descent on K. We do this by selecting an initial K^0 arbitrarily, and for each iteration computing $K^{t+1} = P_S(K^t - \eta \nabla l_t(K^t))$ with l_t being the empirical log-loss at iteration t i.e. $l_t = \log \frac{1}{p}$, and p being the probability that the oracle correctly ordered the constituent triplets of the selected tuples. For the response probability of an individual triplet, we adopt the model in (Tamuz et al. 2011) that is reminiscent of Bradley-Terry pairwise score models (Bradley and Terry 1952): for parameter $\mu > 0$, $p(b_1 \prec b_2) = (D^2_{a,b_2} + \mu)/(D^2_{a,b_1} + D^2_{a,b_2} + 2\mu)$.

Tuple Response Model

Our proposed technique is compatible with any tuple ranking model that satisfies (A2). However, since we use the triplet

response model listed above in the probabilistic MDS formulation, combined with the need for a controlled test against CKL, we extend their model to the tuplewise case as follows: we first decompose an oracle's ranking into its constituent triplets, and then apply

$$p(R(Q_n) \mid D_{Q_n}) \propto \prod_{i=1}^{k-2} \frac{D^2_{a,R_{i+1}(Q_n)} + \mu}{D^2_{a,R_i(Q_n)} + D^2_{a,R_{i+1}(Q_n)} + 2\mu},$$

for parameter $\mu > 0$. This model corresponds to oracle behavior that ranks objects proportionally to the ratio of their distances with respect to a, such that closer (resp. farther) objects are more (resp. less) likely to be deemed similar. Models of this type are generally held to be similar to the scale-invariant models present in some human perceptual systems (Chater and Brown 1999).

Adaptive Algorithm

Combining these concepts, we have the following algorithm titled InfoTuple, summarized in Algorithm 1: the algorithm requires that some initial set of randomly selected tuples be labeled to provide a reasonable initialization of the learned similarity matrix. Since the focus of this work is on the effectiveness of various adaptive selection methods, this initialization is standardized across methods considered in our results. Specifically, following established practice (Tamuz et al. 2011), a "burn-in" period is used where T_0 random triplets are posed for each object a in object set $\mathcal{X}$, with a being the head of each query. Then, for each time step n we learn a similarity matrix $\widehat{K}^{n-1}$ on the set of previous responses r^{n-1} by using probabilistic MDS. To make a comparison to CKL, we follow their procedure and subsequently pose a single tuple for each head $a \in \mathcal{X}$. However, it is possible to adaptively choose a with our method by searching over both head and body objects for a maximally informative tuple. The body of each tuple, given some head a, is chosen by uniformly downsampling the set of possible bodies and selecting the one that maximizes the mutual information, calculated using the aforementioned probability model in our estimation procedure. This highlights the importance of computational tractability in estimating mutual information, since for a fixed computing budget per selected query, less expensive mutual information estimation allows for more candidate bodies to be considered. For a tuple size of k we denote the run of an algorithm using that tuple size as InfoTuple-k.

Algorithm 1 InfoTuple-k

Require: object set $\mathcal{X}$, rate ω, sample size N_f, horizon T
  $r^0 \leftarrow \emptyset$ initialize set of oracle responses
  $\widehat{K}^0 \leftarrow$ initialize embedding
  for $n = 1$ **to** T **do**
    $\widehat{D}^{n-1} \leftarrow$ calculate pairwise distances from $\widehat{K}^{n-1}$
    $\sigma^2_{n-1} \leftarrow \frac{1}{N^2} \sum_{d \in \widehat{D}^{n-1}} \left(d - \frac{1}{N^2} \sum_{d \in \widehat{D}^{n-1}} d \right)^2$
    for all $a \in \mathcal{X}$ **do**
      $\beta \leftarrow$ downsampled $k-1$ sized bodies at rate ω
      for all $B \in \beta$ **do**
        $Q \leftarrow \{a\} \cup B$
        $D_s \sim \mathcal{N}(\widehat{D}^{n-1}_Q, \sigma^2_{n-1})$, drawn N_f times
        $I_B \leftarrow H\left(\sum_{D \in D_s} \frac{p(R(Q)|D)}{N_f} \right) - \sum_{D \in D_s} \frac{H(p(R(Q)|D))}{N_f}$
      end for
      $B \leftarrow \arg\max_{B \in \beta} I_B$
      $r \leftarrow$ oracle ranks objects in B relative to a
      $r^n \leftarrow r^{n-1} \cup r$
    end for
    $\widehat{K}^n \leftarrow \text{probabilisticMDS}(r^n)$
  end for
Ensure: $\widehat{K}^T$

Experiments

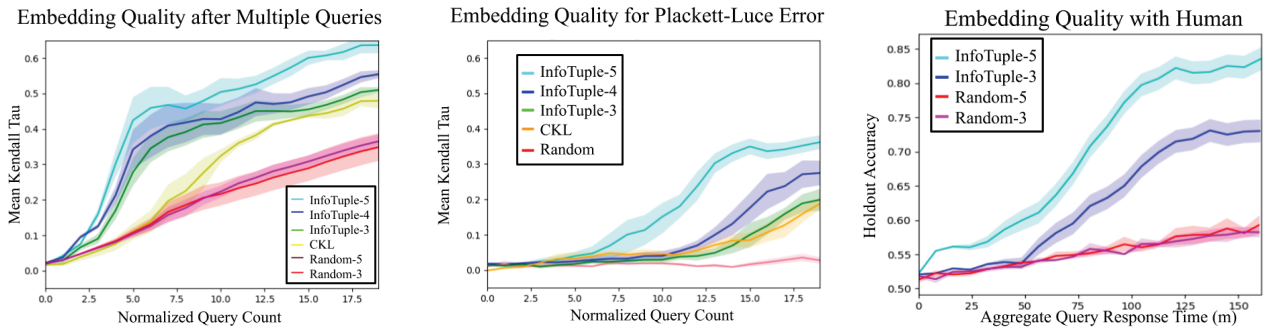
Our results on synthetic and human response datasets show that InfoTuple's adaptive selection outperforms both random query selection and that of CKL[1]. This is true even when normalizing for changes in tuple size and when normalizing for labeling effort, showing that the incurred benefit is not only due to the increased information inherently present in larger tuples but also due to our improved adaptive selection. We also show that there are inherent consistency benefits to the use of larger queries, and that human labelers can respond to these query types in practice without undue cost.

[1]Code available at https://github.com/siplab-gt/infotuple

Datasets

To evaluate algorithm performance in a controlled setting, we constructed a synthetic evaluation dataset by generating a point cloud drawn from a d-dimensional multivariate normal distribution. To simulate oracle responses for this dataset, we use the popular Plackett-Luce permutation model to sample a ranking for a given head and body (Cao et al. 2007; Guiver and Snelson 2009). In this response model, each object in a tuple body is assigned a score according to a scoring function, which in our case is based on the distance in the underlying space between each object and the head. For a given subset of body objects, the probability of an object being ranked as most similar to the head is its score divided by the scores of all objects in that subset, and we generate each simulated oracle response by sequentially sampling objects without replacement from a tuple according to this model. We chose this tested response model to differ from the one we use to estimate mutual information in order to demonstrate the robustness of our method to mismatched noise models, and evaluate an additional Gaussian noise model in the appendix. This dataset was used to compare InfoTuple-3, InfoTuple-4, InfoTuple-5, CKL, Random-3, and Random-5 across noiseless, Gaussian, and Plackett-Luce oracles.

To demonstrate the broader applicability of our work in real-world settings and evaluate our proposed technique on perceptual similarity data, we also collected a large dataset of human responses to tuplewise queries through Amazon Mechanical Turk. Drawing 3000 food images from the Food-10k dataset (Wilber et al. 2015), we presented over 7000 users with a total of 192,000 varying-size tuplewise queries chosen using Infotuple-3, InfoTuple-5, Random-3, and Random-5 as selection strategies across three repeated runs of each al-



(a) (b) (c)

Figure 2: 2(a) and 2(b) show a comparison of the fidelity of the learned embedding to the ground truth embedding with a simulated deterministic (left) and a stochastic (right) oracle, plotted with ± 1 standard error. Results shown are for a synthetic dataset of $N = 500$ points from a two-dimensional dataset. 2(c) shows holdout accuracy on human-subject tests with $N = 5000$.

gorithm. Users were evaluated with one repeat query out of 25, and users who responded inconsistently to the repeat query were discarded. Query bodies were always shuffled when presented to minimize the impact of any possible order effect, and it was not found to be the case that there was any significant order effect in the human responses. Initial embeddings for each of these methods were trained on 5,000 triplet queries drawn from (Wilber et al. 2015). Although experimental costs prevented us from extending the experiments in Figure 2(c) to larger tuple sizes, in order to verify the feasibility of having humans respond to larger tuples in practice we performed a separate data collection in which we asked users to rank randomly selected tuples up to a size of $k = 10$ and recorded the labeling time for each response.

Evaluation Metrics

In order to directly measure the preservation of object rankings between the ground truth object coordinates and the embedding learned from oracle responses, we use Kendall's Tau rank correlation coefficient (Kendall 1938). To get an aggregate measure of quality when comparing an estimated embedding to a ground-truth embedding, we take the mean of Kendall's Tau across the total rankings obtained by setting each object as the head and sorting all objects by embedding distance to the head. In our experiments with human respondents it is not possible to use this measure, as the "ground truth" embedding that corresponds to human preferences is not known. In these cases we instead measure the accuracy with respect to a held-out set of queries drawn from the Food-10k dataset (Wilber et al. 2015), which is a common embedding quality metric (Van Der Maaten and Weinberger 2012; Wilber et al. 2015). The holdout accuracy is the fraction of a held out set of triplet comparisons that agrees with distances in the final learned embedding. To capture a notion of the internal coherence between a set of oracle responses and an embedding that is learned from them, we measure the mean rank correlation between each response in this set and the ranking over the same objects imputed from the learned embedding–we refer to this as the *coherence* of a set of tuples.

One issue that naturally arises when comparing results from strategies that select tuples of different size is normalization, as larger tuples will naturally be more informative. In human-response studies normalization is relatively straightforward, as we can simply normalize with respect to the total time spent labeling queries in order to reflect the total labeling cost. While other more comprehensive measures of labeler effort exist, labeling time is a first-order approximation for the cognitive load of a labeling task and is the most salient metric for determining the cost of a large-scale data collection. In the case of synthetic data, we instead compute a *normalized query count* corresponding to the number of constituent triplet comparisons defining the relation of each body point to the head in the tuple. This is justified since in practice we decompose tuples in this way when feeding them into the embedding algorithm, and corresponds to the size of a tuple's transitive reduction (a common representation in learning-to-rank literature (Kingston and Egerstedt 2009)). Additional experimental details such as hyperparameter selection are available in the appendix.

Experimental Results

Using simulated data, we show a direct comparison of embedding quality from using InfoTuple, CKL, and Random queries under a simulated deterministic oracle (Figure 2(a)) and two simulated stochastic oracles (Figure 2(b)), and note that InfoTuple consistently outperformed the other methods. We note two important observations from these results: first, regardless of the oracle used, larger tuple sizes for InfoTuple tended to perform better and converge faster than did smaller tuple sizes even after normalizing for the tuple size, showing the benefit of larger tuples beyond just providing more constituent triplets. Recalling that the Plackett-Luce oracle was not directly modeled in our estimate of mutual information, this lends support to the robustness of our technique to various oracle distributions. Second, results on Random-3, Random-4 and Random-5 are comparable, implying both that the improvements seen in InfoTuple are not solely due to the difference in tuple sizes and that our choice of normalization is appropriate. Note that since random query performance did not change with tuple size, Figure 2(b) only shows Random-3

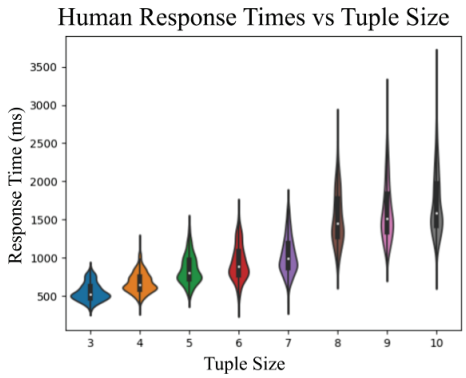


Figure 3: This violin plot shows the distribution of timing responses for random queries from size $k = 3$ to $k = 10$, for the purpose of measuring labeler effort. The response time for $k \leq 7$ shows only modest increases in cost, although responses above these sizes require significantly more effort.

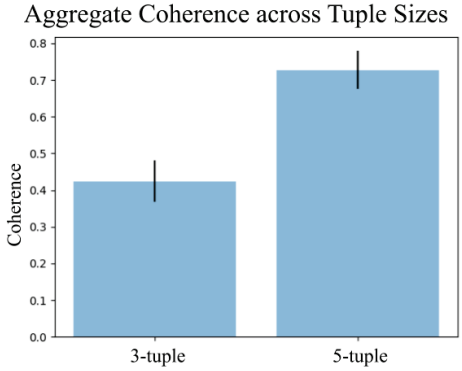


Figure 4: Measuring the aggregate coherence for all tuples of size 3 and size 5 (i.e. over 80,000 tuples at each size) with respect to an aggregate embedding learned for each tuple size, we find that there is a significant difference in their internal coherence as measured by a t-test (p=0.007181). We hypothesize that the difference is due to an increase in context available to the oracle. Error bars depict ± 1 standard error.

for the sake of visual clarity.

Using the Mechanical Turk dataset described previously, we also show that these basic results extend to real data situations when the stochastic response model is not exactly known, and allows us to examine the complexity of acquiring data with increasing tuple sizes. While larger tuples sizes produce more informative queries, it is possible that the information gained incurs a hidden cost in the complexity or labeler effort involved in acquiring the larger query. Specifically, it can be the case that maximizing query informativeness can produce queries that are more difficult to answer (Baldridge and Palmer 2009). Fortunately, the results on tuplewise comparisons collected for our Mechanical Turk dataset indicate that this is not an issue for our proposed use case. In particular, Figure 2(c) shows the accuracy results when predicting the labels from a held out set of 1200 triplet queries. These results show an increase in the effectiveness of InfoTuple adaptive selection as well as increasing tuples sizes when plotted against the aggregate query response time. In other words, any increase in query complexity (measured by response time) is more than compensated for by the increased information acquired by the query and the increase in the resulting quality of the learned embedding.

Figure 3 explores this issue further by examining the response times for our additional timing dataset as a function of query size. There are only modest increases in the ranking time cost with increasing tuple size, leading to the significant gains observed in normalized information efficiency in this range of tuple sizes. While it is true that complexity cost will continue to increase for larger tuple sizes and the gains in information efficiency are not guaranteed to increase indefinitely and there may also be additional factors in the choice of optimal tuple size for a given problem, we show that up to a modest tuple size it is strictly more useful to ask tuplewise queries than triplet queries.

One possible reason for why tuples outperform triplets is that asking a query that contains more objects provides additional context for the oracle about the contents of the dataset, allowing it to more reliably respond to ambiguous comparisons than if these were asked as triplet queries. As a result of this increase in context, oracles tend to respond to larger queries significantly more coherently than they do to smaller ones, as shown in Figure 4. We note that this is not guaranteed to increase indefinitely as larger tuples are considered, but the effect is noticeable for modest increases in tuple sizes and is clear when comparing 5-tuples to triplets.

Discussion

In this paper we proposed InfoTuple, an adaptive tuple selection strategy based on maximizing mutual information for relative tuple queries for similarity learning. We introduce the tuple query for similarity learning, present a novel set of assumptions for efficient estimation of mutual information, and through the collection of new user-response datasets, provide new insights into the gains acquired by using larger tuples in learning efficiency and query consistency. After testing on synthetic and real datasets, InfoTuple was found to more effectively learn similarity-based object embeddings than random queries and state-of-the-art triplet queries for both synthetic data (with a typical oracle model) and in a real world experiment. The performance gains were especially evident for larger tuples and even after normalizing for tuple size, indicating that the proposed selection objective that maximizes the mutual information between the query response and the entire embedding yields information gains that are not simply due to an increase in tuple size. Taken together, these results suggest that large tuples selected with InfoTuple supply richer and more robust embedding information than their triplet and random counterparts.

In practice, larger tuple sizes can provide more context for the oracle, increasing the reliability of the responses without significant increases in labeling effort. In the pathological extreme, the level of effort almost certainly outweighs the benefits of larger tuples, as an oracle would have to provide

a ranking over the entire dataset. Despite this downside in extreme tuple sizes, our human study results indicate that performance increases hold up in the real-world for moderate tuple sizes. This interesting tradeoff between informativeness per query and real-world oracle behavior merits a more comprehensive study on the psychometric aspects of the problem, in the spirit of (Miller 1956).

Acknowledgements

This work is partially supported by NSF CAREER award CCF-1350954, ONR grant number N00014-15-1-2619 and AFRL contract number FA8750-19-C-0200.

Appendix

Experimental Details

For each of the human-subject experiments, μ was set to 0.1 and d was set to 4 per the hyperparameter search shown in Figure 5. The validation set for this search was an additional 500 heldout triplets from the Food10k dataset. In the synthetic experiments provided, μ was set to 0.5 and d was set to 2 to match the dimensionality of the generating distribution. The stochastic oracle had a high noise level, inverting 33% of tuple responses. Higher tuple sizes were strongly correlated with both higher performance and higher robustness to error (even when normalized by the effective number of pairwise queries), indicating performance gains for InfoTuple that are not simply due to increasing tuple sizes. A heuristic was used to pick a number of samples for the Monte Carlo estimation of the mutual information, with $\frac{N}{10}$ samples being used in practice.

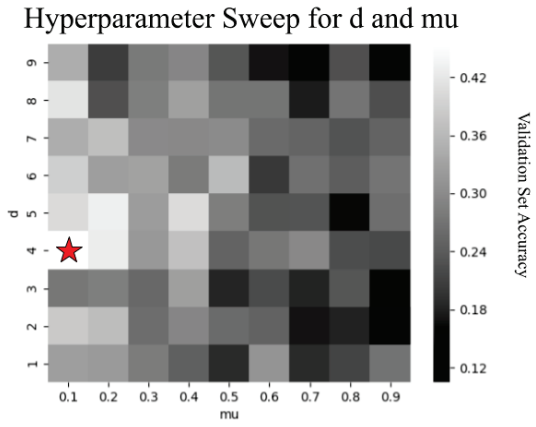


Figure 5: Hyperparameter sweep for Food10k dataset. Experimental values of $d = 4$ and $\mu = 0.1$ were found to be the most effective on a held-out validation set of triplets.

Figure 2 in the paper body shows empirical performance for query selection algorithms on predicting labels from held out triplet queries in the Mechanical Turk dataset described. Experimental horizons for human subject experiments were chosen based on estimates of the initial steps of convergence and had to be limited due to high experimental costs. Turk subjects were presented with queries in batches of 25, with one repeated tuple across the batch as a test for validity. If the repeat query was not answered the same way by the user both times it was asked the batch was discarded. Order effects were controlled for by shuffling queries prior to presenting them to users for labeling, ensuring that any queries presented to multiple users would appear in different orders and that the test queries would also appear differently each time.

Oracle Details

Two different models of oracle noise were used in our synthetic experiments, Plackett-Luce noise and Gaussian noise (Figure 6). These models were chosen to be different from the one we use to estimate mutual information in order to demonstrate the robustness of our method. In the body of the paper we describe the selection process used by the Plackett-Luce oracle noise, which works by assigning latent scores to objects on the basis of their distances in some synthetic "ground truth" embedding space. The Gaussian noise model, instead of applying noise directly at the level of the ranking responses, applies noise at the level of the oracle's representation of the "ground truth" embedding by adding Gaussian noise to the coordinates of each point drawn from the "ground truth" embedding before imputing a ranking from distances in the oracle's noisy interpretation of the space. For the Plackett-Luce error model results shown in the paper body, 33% of individual rankings were inverted.

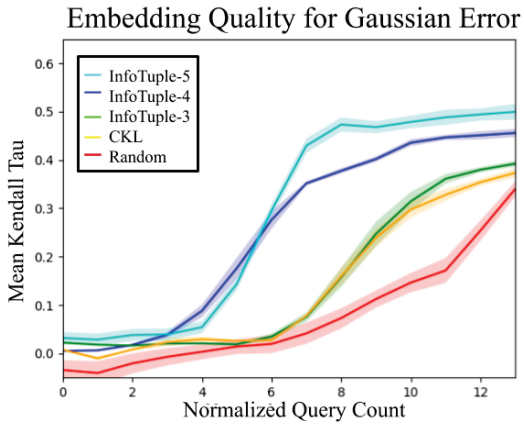


Figure 6: Synthetic experiment results using an oracle with Gaussian noise. Results were broadly consistent with those of the Plackett-Luce oracle in spite of the mismatch between the oracle noise and the embedding.

Computational Complexity

The computational complexity of the embedding calculation is that of a typical MDS algorithm- for any $M \in \mathbb{R}^{d \times N}$ an approximate solution can be found in $O(N)$ for $d < N$ (Aflalo and Kimmel 2013). Our case has an N far greater than d while still being of manageable size, allowing for a fast linear-time approximation.

With respect to the entropy calculation itself, the inner loop computing the mutual information from a given tuple is

computable in $O(N_f k^2)$. However, the computational complexity for a given algorithm iteration is dominated by the $O(\omega\binom{N}{k-1})$ cost of generating and iterating over large pools of candidate tuples, meaning that the run-time is heavily dependent on the choice of the sampling rate ω and distance sample size N_f, and the question of how to efficiently estimate similar mutual information quantities without the use of Monte Carlo methods remains open.

References

Aflalo, Y., and Kimmel, R. 2013. Spectral multidimensional scaling. *Proceedings of the National Academy of Sciences* 110(45):18052–18057.

Argall, B. D.; Chernova, S.; Veloso, M.; and Browning, B. 2009. A survey of robot learning from demonstration. *Robotics and autonomous systems* 57(5):469–483.

Baldridge, J., and Palmer, A. 2009. How well does active learning actually work?: Time-based evaluation of cost-reduction strategies for language documentation. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*, 296–305. Association for Computational Linguistics.

Bradley, R. A., and Terry, M. E. 1952. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika* 39(3/4):324–345.

Canal, G.; Massimino, A.; Davenport, M.; and Rozell, C. 2019. Active embedding search via noisy paired comparisons. In *International Conference on Machine Learning*, 902–911.

Cao, C., and Ai, H.-Z. 2015. Facial similarity learning with humans in the loop. *Journal of Computer Science and Technology* 30(3):499–510.

Cao, Z.; Qin, T.; Liu, T.-Y.; Tsai, M.-F.; and Li, H. 2007. Learning to rank: from pairwise approach to listwise approach. In *Proceedings of the 24th international conference on Machine learning*, 129–136. ACM.

Chater, N., and Brown, G. D. 1999. Scale-invariance as a unifying psychological principle. *Cognition* 69(3):B17–B24.

Cover, T. M., and Thomas, J. A. 2012. *Elements of information theory*. John Wiley & Sons.

Fernando, B.; Gavves, S.; Muselet, D.; and Tuytelaars, T. 2015. Learning to rank based on subsequences. In *Proceedings ICCV 2015*, 2785–2793.

Ferrari, V.; Tuytelaars, T.; and Van Gool, L. 2004. *Simultaneous object recognition and segmentation by image exploration*. Berlin, Heidelberg: Springer Berlin Heidelberg. 40–54.

Guiver, J., and Snelson, E. 2009. Bayesian inference for plackett-luce ranking models. In *proceedings of the 26th annual international conference on machine learning*, 377–384. ACM.

Hoffer, E., and Ailon, N. 2015. Deep metric learning using triplet network. In *International Workshop on Similarity-Based Pattern Recognition*, 84–92. Springer.

Houlsby, N.; Huszar, F.; Ghahramani, Z.; and Hernández-Lobato, J. M. 2012. Collaborative gaussian processes for preference learning. In *Advances in neural information processing systems*, 2096–2104.

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

Jamieson, K. G., and Nowak, R. 2011a. Active ranking using pairwise comparisons. In *Advances in Neural Information Processing Systems*, 2240–2248.

Jamieson, K. G., and Nowak, R. D. 2011b. Low-dimensional embedding using adaptively selected ordinal data. In *Communication, Control, and Computing (Allerton), 2011 49th Annual Allerton Conference on*, 1077–1084. IEEE.

Kendall, M. G. 1938. A new measure of rank correlation. *Biometrika* 30(1/2):81–93.

Kingston, P., and Egerstedt, M. 2009. Comparing apples and oranges through partial orders: An empirical approach. In *American Control Conference, 2009. ACC'09.*, 5434–5439. IEEE.

Liang, L., and Grauman, K. 2014. Beyond comparing image pairs: Setwise active learning for relative attributes. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 208–215.

Lindley, D. V. 1956. On a measure of the information provided by an experiment. *The Annals of Mathematical Statistics* 986–1005.

Liu, E. Y.; Guo, Z.; Zhang, X.; Jojic, V.; and Wang, W. 2012. Metric learning from relative comparisons by minimizing squared residual. In *Data Mining (ICDM), 2012 IEEE 12th International Conference on*, 978–983. IEEE.

Lohaus, M.; Hennig, P.; and von Luxburg, U. 2019. Uncertainty estimates for ordinal embeddings. *arXiv preprint arXiv:1906.11655*.

MacKay, D. J. 1992. Information-based objective functions for active data selection. *Neural computation* 4(4):590–604.

Miller, G. A. 1956. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological review* 63(2):81.

Parikh, D., and Grauman, K. 2011. Relative attributes. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, 503–510. IEEE.

Patterson, G.; Van Horn, G.; Belongie, S. J.; Perona, P.; and Hays, J. 2015. Tropel: Crowdsourcing detectors with minimal training. In *HCOMP*, 150–159.

Qian, B.; Wang, X.; Wang, F.; Li, H.; Ye, J.; and Davidson, I. 2013. Active learning from relative queries. In *IJCAI*, 1614–1620.

Settles, B. 2012. Active learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning* 6(1):1–114.

Stern, H. 1990. Models for distributions on permutations. *Journal of the American Statistical Association* 85(410):558–564.

Tamuz, O.; Liu, C.; Belongie, S.; Shamir, O.; and Kalai, A. T. 2011. Adaptively learning the crowd kernel. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, 673–680. Omnipress.

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

Wilber, M.; Kwak, I. S.; Kriegman, D.; and Belongie, S. 2015. Learning concept embeddings with combined human-machine expertise. In *International Conference on Computer Vision (ICCV)*.

Wilber, M. J.; Kwak, I. S.; and Belongie, S. J. 2014. Cost-effective hits for relative similarity comparisons. In *Second AAAI Conference on Human Computation and Crowdsourcing*.

Yang, L.; Jin, R.; Mummert, L.; Sukthankar, R.; Goode, A.; Zheng, B.; Hoi, S. C.; and Satyanarayanan, M. 2010. A boosting framework for visuality-preserving distance metric learning and its application to medical image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32(1):30–44.

Yu, H. 2005. Svm selective sampling for ranking with application to data retrieval. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, 354–363. ACM.