

## Cost-Effective HITs for Relative Similarity Comparisons

**Michael J. Wilber**  
Cornell University

**Iljung S. Kwak**  
University of California, San Diego

**Serge J. Belongie**  
Cornell University

### Abstract

Similarity comparisons of the form “Is object  $a$  more similar to  $b$  than to  $c$ ?” form a useful foundation in several computer vision and machine learning applications. Unfortunately, an embedding of  $n$  points is only uniquely specified by  $n^3$  triplets, making collecting every triplet an expensive task. In noticing this difficulty, other researchers investigated more intelligent triplet sampling techniques, but they do not study their effectiveness or their potential drawbacks. Although it is important to reduce the number of collected triplets to generate a good embedding, it is also important to understand how best to display a triplet collection task to the user to better respect the worker’s human constraints. In this work, we explore an alternative method for collecting triplets and analyze its financial cost, collection speed, and worker happiness as a function of the final embedding quality. We propose best practices for creating cost effective human intelligence tasks for collecting triplets. We show that rather than changing the sampling algorithm, simple changes to the crowdsourcing UI can drastically decrease the cost of collecting similarity comparisons. Finally, we provide a food similarity dataset as well as the labels collected from crowd workers.

### Introduction

Recently in machine learning (Tamuz et al. 2011; Jamieson and Nowak 2011; van der Maaten and Weinberger 2012; McFee 2012), there has been a growing interest in collecting human similarity comparisons of the form “Is  $a$  more similar to  $b$  than to  $c$ ?” These comparisons are asking humans to provide constraints of the form  $d(a, b) < d(a, c)$ , where  $d(x, y)$  represents some perceptual distance between  $x$  and  $y$ . We refer to these constraints as *triplets*. Each triplet provides a small unit of information about one point’s location with respect to the other points in human perceptual space.

These triplets are useful for a variety of computer vision problems. Collection is usually the first stage of a much larger pipeline: first, researchers collect several thousand triplets until they feel they “have enough triplets” or reach their crowdsourcing budget. Then, an embedding algorithm such as t-Stochastic Triplet Embedding (t-STE) (van der Maaten and Weinberger 2012) embeds each entry of the dataset into a space where Euclidean distances between

Copyright © 2014, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

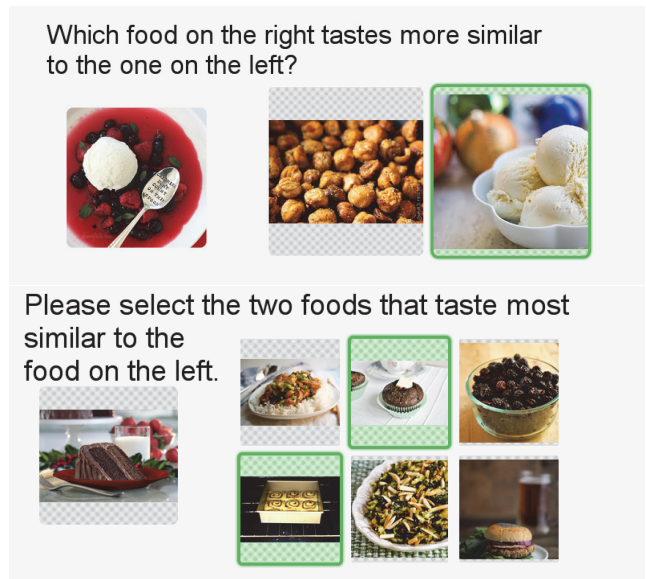


Figure 1: Questions of the form “Is object  $a$  more similar to  $b$  than to  $c$ ?” are one useful way to collect similarity comparisons from crowd workers. Traditionally, these comparisons are collected individually using the UI shown at the top. In this work, we collect triplets using a grid of  $n$  images and ask the user to select the  $k$  most similar tasting foods to the food on the left. This grid UI, bottom, allows us to collect 8 triplets whereas the individual triplet UI, top, only yields a single triplet.

points approximates their perceptual similarity. Specific applications then use this embedding in different ways. Some systems provide tools to search for images similar to an imagined image in a user’s head (Ferecatu and Geman 2009). Other systems use triplet embeddings in conjunction with computer vision to aid classification of bird images (Wah et al. 2014). Still others use triplet embeddings for metric learning (Frome et al. 2007), learning semantic clusters (Gomes et al. 2011), and finding similar musical genres and artists (van der Maaten and Weinberger 2012; McFee 2012). The common idea behind all of this work is that these authors must collect thousands of groundtruth

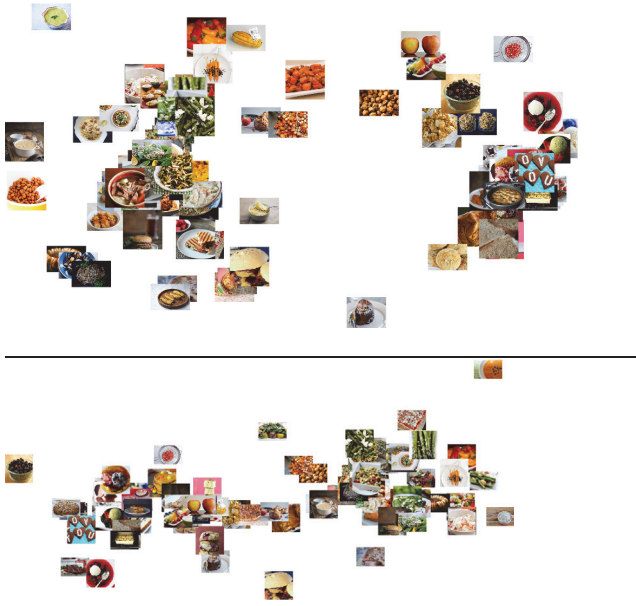


Figure 2: **Top:** An example cuisine embedding, collected with our 16-choose-4 grid UI strategy. The data for this embedding cost \$5.10 to collect and used 408 screens, but yielded 19,199 triplets. The result shows good clustering behavior with desserts gathered into the top left. The meats are close to each other, as are the salads. **Bottom:** An embedding with 408 random triplets. The data for this embedding also cost \$5.10 to collect, but the result is much dirtier, with worse separation and less structure. Salads are strewn about the right half of the embedding and a steak lies within the dessert area. From our experiments, we know that an embedding of such low quality would have cost us less than \$0.10 to collect using our grid strategy. To see these results in closer detail, visit our Food Embedding Explorer on our companion web site at <http://cost-effective-hits.mjwilber.org/>.

triplets before they can solve their specific problem. That’s where we come in: our main goal is to provide a more efficient way to collect huge numbers of these constraints.

For any set of  $n$  points, there are at most  $n^3$  unique triplets (Kleindessner and von Luxburg 2014). Unfortunately, collecting such a large amount of triplets from crowd workers quickly becomes intractable for larger datasets. For this reason, some research groups propose more intelligent sampling techniques to better capture the same space with less information (Tamuz et al. 2011; Jamieson and Nowak 2011). However, the difficulty of collecting a large number of triplets is also related to the time and monetary cost of collecting data from humans. To investigate this relationship more closely, we chose to study how the design of the triplet collection Human Intelligence Task (HIT) affects not only the time and cost of collecting triplets, but also the quality of the embedding, which is usually the researcher’s primary concern.

Traditionally, an MTurk task designed to collect triplets

would show crowd workers three images, labeled  $a$ ,  $b$ ,  $c$ . The worker is then asked to select either image  $b$  or image  $c$ , whichever looks more similar to image  $a$ . See the top of Fig. 1 for an example. We call this the *individual format* because it samples triplets individually. Although this is the most direct design to collect triplets, it is potentially inefficient. Instead, we chose to investigate triplets collected from a grid of images. In the *grid format*, a probe image— analogous to image “ $a$ ” in the triplet representation—is shown next to a grid of  $n$  images. The crowd worker is then asked to choose the  $k$  most similar images from the grid. We are not the first to realize that a grid is more efficient for collecting triplets—such techniques were also used by (Wah et al. 2014; Tamuz et al. 2011) and studied by (Rao and Katz 1971) in the field of marketing research—but we believe we are the first to investigate more thoroughly the effectiveness of triplets collected with a grid. This is important because previous authors do not quantify the efficiency gain and do not acknowledge the potential drawbacks of the grid triplets they rely on.

This paper outlines several UI modifications that allow researchers to multiply the number of triplets collected per screen for perceptual similarity learning. We show that simple changes to the crowdsourcing UI—*instead* of fundamental changes to the algorithm — can lead to much higher quality embeddings. In our case, using our grid format allows us to collect several triplets per screen, leading to much faster convergence than asking one triplet question at a time. Researchers with tight deadlines can create reasonable embeddings with off-the-shelf algorithms and a low crowdsourcing budget by following our guidelines.

Our contributions are:

- A set of guidelines to use when collecting similarity embeddings, with insights on how to manage the trade-off between user burden, embedding quality, and cost;
- A series of synthetic and human-powered experiments that establish our methods’ effectiveness;
- Evidence that each individual triplet sampled with a grid may capture less information than a uniformly random triplet, but that their quantity outweighs the potential quality decrease;
- A dataset of 100 food images, ingredient annotations, and roughly 39% of the triplets that describe it. <sup>1</sup>

## Related Work

In our work, we collect human similarity measurements of images in the form of triplets. The authors of (Heikinheimo and Ukkonen 2013) proposed an algorithm for collecting triplets from humans as well. However, in that work, the triplets that were collected did not have a probe image because they formulated the question differently. Similarly, (Yi et al. 2013) focuses on estimating user preferences from crowd sourced similarity comparisons. However (Yi et al. 2013) uses pairwise comparisons where triplets are generally more stable (Kendall and Gibbons 1990).

<sup>1</sup>See <http://cost-effective-hits.mjwilber.org/>

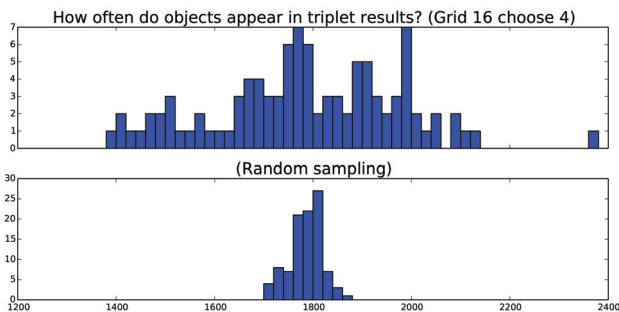


Figure 3: **Random triplets have a different distribution than grid triplets.** The top histogram shows the occurrences of each object within human answers for “Grid 16 choose 4” triplets from our human experiments. The bottom histogram shows a histogram of sampling random triplets individually. 59520 triplets were collected for both histograms. Each object occurs in our answers about  $\hat{\mu} = 1785$  times, but the variation when using grid triplets (top) is much wider ( $\hat{\sigma} \approx 187.0$ ) than the variation when sampling triplets uniformly (bottom,  $\hat{\sigma} = 35.5$ ). This effect is not recognized in the literature by authors who use grids to collect triplets. We study its impact in our experiments.

Our work bears much similarity to Crowd Kernel Learning (Tamuz et al. 2011) and Active MDS (Jamieson and Nowak 2011). These algorithms focus on collecting triplets one at a time by sampling the *best* triplets first. The idea behind these systems is that the bulk of the information in the embedding can be captured within a very small number of triplets since most triplets convey redundant information. For instance, Crowd Kernel Learning (Tamuz et al. 2011) considers each triplet individually, modeling the information gain learned from that triplet as a probability distribution over embedding space. Active MDS (Jamieson and Nowak 2011) considers a set of triplets as a partial ranking with respect to each object in the embedding, placing geometric constraints on the locations where each point may lie. In our work we focus on altering UI design to improve speed and quality of triplet collection.

## Method

Instead of asking “Is  $a$  more similar to  $b$  or  $c$ ?”, we present a probe image and  $n$  other images to humans. We then ask “Select  $k$  images that are most similar to the probe,” as in Fig. 1. This layout allows us to collect  $k$  images that are more similar to the probe image than the remaining  $n - k$  images, yielding  $k(n - k)$  triplets per answer. We can change the number of triplets per grid answer by varying  $n$  and  $k$ , but this also affects the amount of effort a crowd worker must exert to answer the question. However, it allows crowd workers to avoid having to wait for multiple screens to load, especially in cases where one or more of the images in the queried triplets does not change, and allows crowd workers to benefit from the parallelism in the low-level human visual system (Wolfe 1994). Since many of these observations involve human issues, we conclude that the right way

of measuring embedding quality is with respect to *human cost* rather than the number of triplets. This human cost is related to the time it takes crowd workers to complete a task and the pay rate of a completed task. Our goal is to formalize their intuitive notions into hard guidelines.

It is important to note that the *distribution of grid triplets is not uniformly random*, even when the grid entries are selected randomly and even with perfect answers. To our knowledge, no authors that use the grid format acknowledge this potential bias even though it deteriorates each triplet’s quality, as we will show in our experiments. Figure 3 shows a histogram of how many times each object occurs in our triplet answers of our synthetic experiments mentioned below. When using the grid format, some objects can occur far more often than others, suggesting that the quality of certain objects’ placement within the recovered embedding may be better than others. The effect is less pronounced in random triplets, where objects appear with roughly equal frequency. This observation is important to keep in mind because the unequal distribution influences the result.

## Synthetic Experiments

Our proposed use of grid triplets impacts the collection process in two ways: first, the distribution of grid triplets is fundamentally different than the distribution of uniformly random triplets, which may impact the embedding quality. Second, the new question format imposes a different kind of burden on the worker. Since it is best to investigate these issues separately, we chose to run synthetic experiments to investigate the quantitative differences in triplet distributions in addition to our human experiments to investigate the human impacts of our proposed framework.

We aimed to answer two questions: *Are grid-format triplets of lower quality than triplets acquired individually?* Second, *even if grid-format triplets are lower quality, does their quantity outweigh that effect?* To find out, we ran synthetic “Mechanical Turk-like” experiments on synthetic workers. For each question, we show a probe and a grid of  $n$  objects. The synthetic workers use Euclidean distance within a groundtruth embedding to choose  $k$  grid entries that are most similar to the probe. As a baseline, we randomly sample triplets from the groundtruth embedding using the same Euclidean distance metric. After collecting the test triplets, we build a query embedding with t-STE (van der Maaten and Weinberger 2012) and compare this embedding to the groundtruth. This way, we can measure the quality of our embedding with respect to the total number of worker tasks. This is not a perfect proxy for human behavior, but it does let us validate our approach and should be considered in conjunction with the actual human experiments that we describe later.

**Dataset.** We evaluate our system on the music similarity dataset from (van der Maaten and Weinberger 2012) The music dataset’s groundtruth contains 9,107 human-supplied triplets for 412 artists.

**Metrics.** Since our goal is to evaluate the impact that different UIs have on the embedding quality, we evaluate each embedding’s quality using two metrics from (van der Maaten and Weinberger 2012): Triplet Generalization Error,

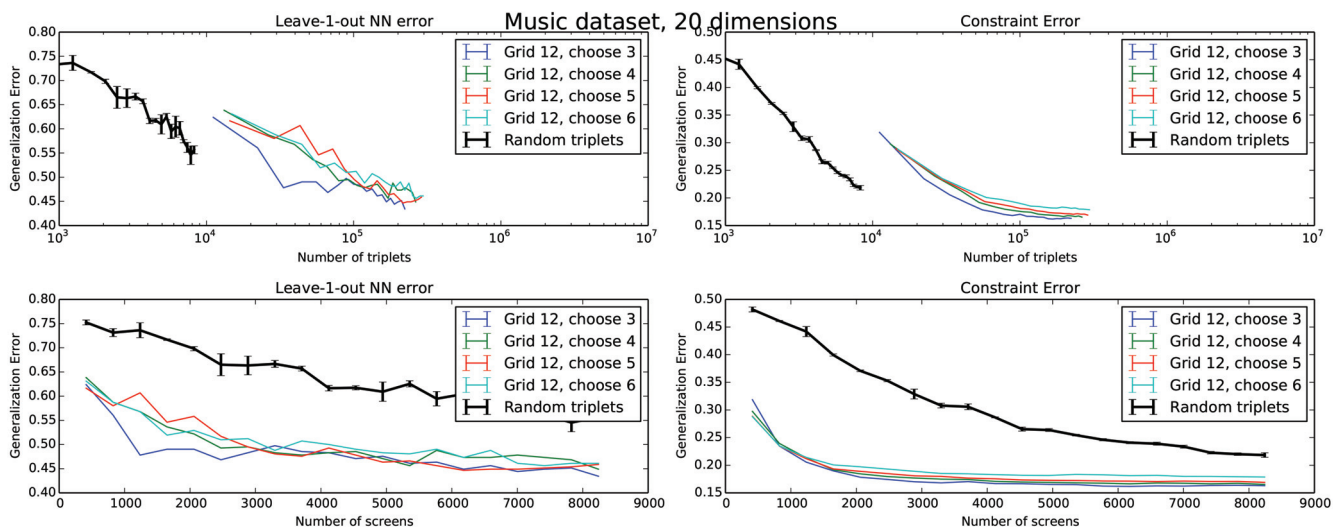


Figure 4: Over the course of a synthetic experiment, we collect triplets, either randomly one at a time (thick black line) or in batches using our grid UI (colored lines). When the embedding quality is viewed as the number of triplets gathered (top two graphs), it appears that sampling random triplets one at a time yields a better embedding. However, when viewed as a function of human effort, grid triplets create embeddings that converge much faster than individually sampled triplets. Here, quantity outweighs quality as measured by Leave-One-Out NN Error (left graphs) and Triplet Generalization Error (right graphs).

which counts the fraction of the groundtruth embedding’s triplet constraints that are violated by the recovered embedding; and Leave-One-Out Nearest Neighbor error, which measures the percentage of points that share a category label with their closest neighbor within the recovered embedding. As pointed out by (van der Maaten and Weinberger 2012), these metrics measure different things: Triplet Generalization Error measures the triplet generator UI’s ability to generalize to unseen constraints, while Leave-One-Out Nearest Neighbor error reveals how well the embedding models the (hidden) human perceptual similarity distance function.

**Results.** Our experiments show that even though triplets acquired via the grid format converge faster than triplets acquired via the individual format, each grid-format triplet *is of lower quality* than an individual random triplet. Figure 4 shows how the music dataset embedding quality converges with respect to the number of triplets. If triplets are sampled one at a time (top two graphs), individual triplets converge much faster on both quality metrics than triplets acquired via grid questions. However, this metric does not reveal the full story because the grid UI yields multiple triplets at once. When viewed with respect to the number of *screens* (human task units), as in the bottom two graphs in Figure 4, we now see that the grid triplets can converge far faster than individual triplets with respect to the total amount of human work. This leads us to conclude that “quality of the embedding *with respect to* number of triplets” is the wrong metric to optimize because framing the question in terms of triplets gives researchers the wrong idea about how fast their embeddings converge. A researcher who only considers the inferior performance of grid triplets on the “per-triplet” metric will prefer sampling triplets individually, but they could achieve

much better accuracy using the grid format even in spite of the reduced quality of each triplet. In other words, efficient collection UIs are better than individual, random sampling, even though each triplet gathered using such UIs does not contain as much information.

Why does this happen? In all cases, the 12 images within the grid were chosen randomly; intuitively, we expect a uniform distribution of triplets. However, because certain objects are more likely than others chosen as “Near” by humans, certain objects will appear in triplets more often than others. This leads to a nonuniform distribution of correct triplets, as shown in Fig. 3. Here, we can see that the non-uniformity may create a difference in performance.

## Human Experiments

The synthetic experiments validate our approach but cannot capture any human considerations which are also paramount in any crowdsourcing task (Kittur et al. 2013). In particular, we must consider the burden the two UIs impose on our workers, both in terms of the time it takes to complete each task and how much money they can make per hour—metrics that are impossible to gather via synthetic means. To verify that these approaches build better embeddings even when humans provide inconsistent triplets, we ran Mechanical Turk experiments on a set of 100 food images sourced from Yummy recipes. We only use images that contain roughly one entree. For example we avoid images of sandwiches alongside soups. Some sample images are shown in Fig. 5. For each experiment, we allocated the same amount of money for each HIT, allowing us to quantify embedding quality with respect to cost. This dataset and the human annotations are available for download at our companion web-



Figure 5: Example images from our dataset. The images in our dataset span a wide range of foods and imaging conditions. The dataset as well as the collected triplets are available on our companion website.

site, <http://cost-effective-hits.mjwilber.org/>.

**Design.** For each task, we show a random probe and a grid of  $n$  random foods. We ask the user to select the  $k$  objects that “taste most similar” to the probe. To study the effects of different grid size, we varied  $n$  across (4, 8, 12, 16) and varied  $k$  across (1, 2, 4). We ran three independent repetitions of each experiment. We paid \$0.10 per HIT, which includes 8 usable grid screens and 2 catch trials. To evaluate the quality of the embedding returned by each grid size, we use the same “Triplet Generalization Error” as in our synthetic experiments: we gather all triplets from all grid sizes and construct a reference embedding via t-STE. Then, to evaluate a set of triplets, we construct a target embedding and count how many of the reference embedding’s constraints are violated by the target embedding. Varying the number of HITs shows how fast the embedding’s quality converges.

**Baseline.** Since we wish to show that grid triplets produce better-quality embeddings at the same cost as random triplets, we should collect random  $(a, b, c)$  comparisons from our crowd workers for comparison. Unfortunately, collecting all comparisons one at a time is infeasible (see our “Cost” results below), so instead, we construct a groundtruth embedding from all grid triplets and uniformly sample random constraints from this reference embedding. This is unlikely to lead to much bias because we were able to collect 39% of the possible unique triplets, meaning that t-STE only has to generalize to constraints that are likely to be redundant. All evaluations are performed relative to this reference embedding.

## Results

Two example embeddings are shown in Fig. 2.

**Cost.** Across all experiments, we collected 14,088 grids, yielding 189,519 unique triplets. Collecting this data cost us \$158.30, but sampling this many random triplets one at a time would have cost us \$2,627.63, which is far outside

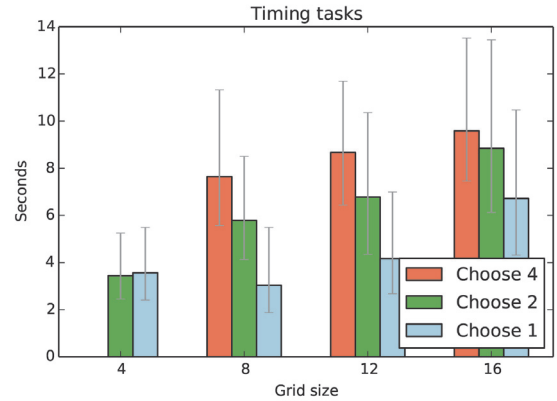


Figure 6: We show the median time that it takes a human to complete one grid task. The time per each task increases with a higher grid size (more time spent looking at the results) and with a higher required number of near answers (which means more clicks per task). Error bars are 25 and 75-percentile.

our budget<sup>2</sup>. If we had used the 16-choose-4 grid strategy (which yields 48 triplets per grid), we would be able to sample all unique triplets for about \$140—a feat that would cost us \$6,737.50 by sampling one at a time.

Grid $n$	choose $k$	Error at \$1	Time/screen (s)	Wages (\$/hr)
$n: 4,$	$k: 1$	0.468	3.57	<b>\$10.09</b>
	$k: 2$	0.369	3.45	<b>\$10.45</b>
$n: 8,$	$k: 1$	0.400	3.04	<b>\$11.85</b>
	$k: 2$	0.311	5.79	<b>\$6.22</b>
	$k: 4$	0.273	7.65	\$4.71
$n: 12,$	$k: 1$	0.406	4.17	<b>\$8.64</b>
	$k: 2$	0.294	6.78	\$5.31
	$k: 4$	0.235	8.67	\$4.15
$n: 16,$	$k: 1$	0.413	6.72	\$5.36
	$k: 2$	0.278	8.84	\$4.07
	$k: 4$	0.231	9.59	\$3.76
Random		0.477	–	–
CKL		0.403	–	–

Table 1: Results of our actual Mechanical Turk experiments. We ask workers to choose the  $k$  most similar objects from a grid of  $n$  images. We invest \$1 worth of questions, giving us 100 grid selections. When  $n$  and  $k$  are large, each answer yields more triplets. Large grids require more time to complete, but many of our tasks (bold) still pay a respectable wage of more than \$6 per hour.

**Quality.** As we spend more money, we collect more triplets, allowing t-STE to do a better job generalizing to unseen redundant constraints. All embeddings converge to lower error when given more triplets, but this convergence is not monotonic because humans are fallible and there is

<sup>2</sup>There are  $100 \cdot 99 \cdot 98/2 = 485,100$  possible unique triplets and each triplet answer would cost one cent. We additionally need to allocate 10% to Amazon’s cut and 20% of our tasks are devoted to catch trials.

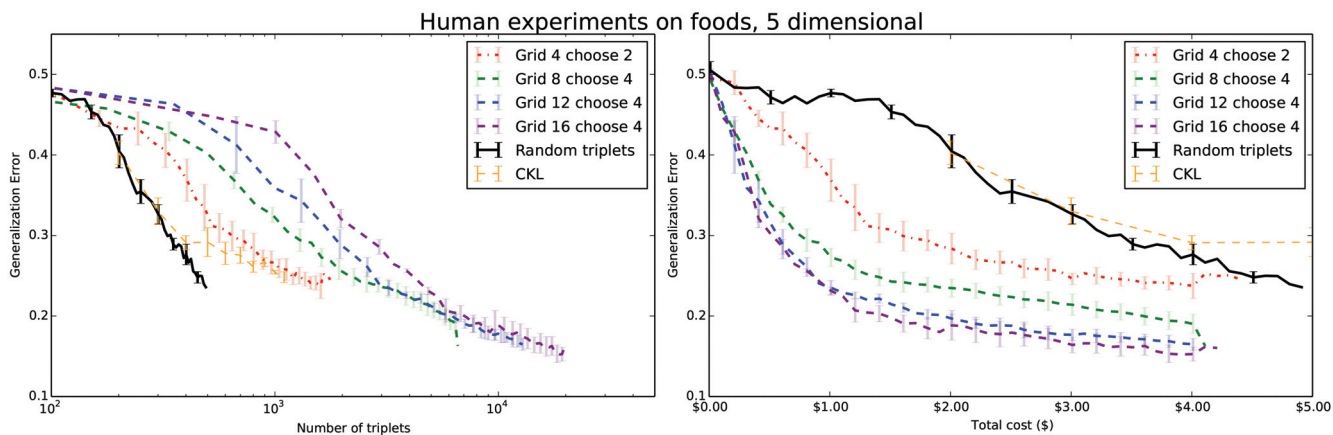


Figure 7: Results of our human experiments on the food dataset. Left graph: Triplet generalization error when viewed with respect to the total number of triplets. Right: The same metric when viewed with respect to the total cost (to us) of constructing each embedding. The left graph implies that a randomly-sampled embedding appears to converge faster. However, when quality is viewed with respect to cost, we find that an embedding generated using a 16-choose-4 grid cost \$0.75, while an embedding with random triplets of similar quality costs \$5.00. It is clear that the grid UI saves money; in this case, by over a factor of 6.

randomness in the embedding construction. See Fig. 7 for a graphical comparison of grids with size 4,8,12, and 16. When viewed with respect to the number of triplets, random triplets again come out ahead; but when viewed with respect to cost, the largest grid converges more quickly than others, and even the smallest grid handily outperforms random triplet sampling.

Unlike our synthetic experiments, we now observe a large separation between the performance of various grid sizes. Grid 16-choose-4, which yields  $4 \cdot 12 = 48$  triplets per answer, uniformly outperforms the rest, with Grid 12-choose-4 (at  $4 \cdot 8 = 32$  triplets per answer) close behind. Both of these outperform 8-choose-4 (16 triplets/answer) and 4-choose-2 (4 triplets/answer).

We also compare our performance with the adaptive triplet sampling strategy of (Tamuz et al. 2011). CKL picks triplets one-at-a-time but attempts to select the best triplet possible to ask by maximizing the information gain from each answer. In our experiments, it did not outperform random sampling; further analysis will be future work.

Though catch trials comprised 20% of the grid answers we collected, we found that the results were generally of such high quality that no filtering or qualification was required.

**Time.** Fig. 6 shows the time each human takes to answer one grid question. Our smallest task was completed in 3.5 seconds on average, but even our largest grid (16 choose 4) can be completed within 10 seconds. Times vary widely between workers: our fastest worker answered 800 questions in an average of 2.1 seconds per grid task for 8-choose-1 grids.

**Worker Satisfaction.** In this work we discussed ways to optimize the cost of collecting triplets from crowd workers. Clearly, we can save money by paying our crowdworkers less money while having them do more work. However,

there is no reason for us to believe that crowd workers would continue to complete our HITs under these conditions. In order for us to avoid underpaying our workers and collect a large amount of triplets at a reasonable cost, we closely monitored Turker responses to our hits.

Across all of our experiments, we received no complaints from “Turkopticon” or from Amazon itself. Our tasks were featured on multiple HIT aggregators including Reddit’s `HitsWorthTurkingFor` community and the “TurkerNation” forums as examples of bountiful HITs. This tells us that our workers did not feel exploited in spite of the increased burden of grid triplets. According to the `HitsWorthTurkingFor` FAQ<sup>3</sup>, the “good HITs” featured on that page generally pay at least \$6/hour. Although it is not clear if Turkers truly enjoyed working on our HITs, it did seem that they felt they were being compensated reasonably for their efforts. See Tab. 1 for a tabular comparison.

Because we were featured in forums such as `HitsWorthTurkingFor`, our HITs were visible to a very large audience of potential skilled Turkers. Though high payouts mean higher cost, in our case, the benefit outweighed the drawback. We feel that in order to truly optimize the interaction between computer algorithms and humans, it is important to understand the perspective of the Turker.

## Guidelines and conclusion

Throughout this paper, we show that taking advantage of simple batch UI tricks can save researchers significant amounts of money when gathering crowdsourced perceptual similarity data. Our recommendations can be summarized as follows:

- Rather than collecting comparisons individually, re-

<sup>3</sup><http://reddit.com/r/HitsWorthTurkingFor/wiki/index>

searchers **should use the grid format** to sample comparisons in batch, or should use some other UI paradigm appropriate to their task. However, researchers should not assume that such “batch” comparisons are of identical quality to uniformly random sampling—this is a trade-off that should be considered.

- If cost is an issue, researchers should consider **quantifying their results with respect to dollars spent** rather than the number of units of information gathered, because simply choosing the UI format that leads to the highest quality-per-triplet will lead researchers to prefer inferior collection methods.
- Researchers should **continuously monitor the human effort of their tasks**, so that they can calculate an appropriate target wage and stand a better chance of being featured on “Good HIT” lists and be seen by more skilled Turkers.
- When using grids to collect triplets, researchers should **consider the trade-off between size and effort**. Consider that an  $n$ -choose- $k$  grid can yield

$$k(n - k) \quad (1)$$

triplets per answer. Since this has a global maximum at  $n = 2k$ , one appropriate strategy is to select the largest  $n$  that yields a wage of \$6/hour and set  $k$  equal to  $n/2$ .

There are several opportunities for future work. First, we should better quantify the relationship between  $n$ ,  $k$ , and task completion time to build a more accurate model of human performance. Second, we should continue investigating triplet sampling algorithms such as “CKL” as there may be opportunities to populate grids adaptively so as to converge faster than random, giving us advantages of both strategies.

## Acknowledgments

We especially thank Jan Jakeš, Tomáš Matera, and Edward Cheng for their software tools that helped us collect grid triplets so quickly. We also thank Vicente Malave for helpful discussions. This work was partially supported by an NSF Graduate Research Fellowship award (NSF DGE-1144153, Author 1) and a Google Focused Research award (Author 3).

## References

Ferecatu, M., and Geman, D. 2009. A statistical framework for image category search from a mental picture. *IEEE TPAMI*.

Frome, A.; Singer, Y.; Sha, F.; and Malik, J. 2007. Learning globally-consistent local distance functions for shape-based image retrieval and classification. In *IEEE ICCV*.

Gomes, R.; Welinder, P.; Krause, A.; and Perona, P. 2011. Crowdclustering. *NIPS*.

Heikinheimo, H., and Ukkonen, A. 2013. The crowd-median algorithm. In *First AAAI Conference on Human Computation and Crowdsourcing*.

Jamieson, K., and Nowak, R. 2011. Low-dimensional embedding using adaptively selected ordinal data. In *Allerton Conference on Communication, Control, and Computing*.

Kendall, M., and Gibbons, J. D. 1990. *Rank Correlation Methods*. Charles Griffin Book Series. Oxford University Press, 5th edition.

Kittur, A.; Nickerson, J. V.; Bernstein, M.; Gerber, E.; Shaw, A.; Zimmerman, J.; Lease, M.; and Horton, J. 2013. The future of crowd work. In *Conference on Computer Supported Cooperative Work*, 1301–1318. ACM.

Kleindessner, M., and von Luxburg, U. 2014. Uniqueness of ordinal embedding. *JMLR*.

McFee, B. 2012. *More like this: machine learning approaches to music similarity*. Ph.D. Dissertation, University of California, San Diego.

Rao, V. R., and Katz, R. 1971. Alternative multidimensional scaling methods for large stimulus sets. *Journal of Marketing Research* 8(4):488–494.

Tamuz, O.; Liu, C.; Belongie, S.; Shamir, O.; and Kalai, A. T. 2011. Adaptively learning the crowd kernel. In *ICML*.

van der Maaten, L., and Weinberger, K. 2012. Stochastic triplet embedding. In *IEEE MLSP*.

Wah, C.; Horn, G. V.; Branson, S.; Maji, S.; Perona, P.; and Belongie, S. 2014. Similarity comparisons for interactive fine-grained categorization. In *IEEE CVPR*.

Wolfe, J. M. 1994. Guided search 2.0 a revised model of visual search. *Psychonomic Bulletin & Review* 1.

Yi, J.; Jin, R.; Jain, S.; and Jain, A. 2013. Inferring users preferences from crowdsourced pairwise comparisons: A matrix completion approach. In *HCOMP*.