Contents lists available at ScienceDirect

Visual Informatics

journal homepage: www.elsevier.com/locate/visinf

Group optimization for multi-attribute visual embedding

Qiong Zeng [a,*], Wenzheng Chen [b,1], Zhuo Han [c], Mingyi Shi [a], Yanir Kleiman [d], Daniel Cohen-Or [e], Baoquan Chen [a], Yangyan Li [a]

[a] School of Computer Science & Technology, Shandong University, Qingdao, China
[b] Department of Computer Science, University of Toronto, Toronto, Canada
[c] Viterbi School of Engineering, University of Southern California, CA, United States
[d] DNEG Visual Effects, London, United Kingdom
[e] School of Computer Science, Tel Aviv University, Tel Aviv, Israel

ARTICLE INFO

Article history:
Received 12 August 2018
Received in revised form 25 September 2018
Accepted 27 September 2018
Available online xxxx

Keywords:
Embedding
Semantic similarity
Visual retrieval

ABSTRACT

Understanding semantic similarity among images is the core of a wide range of computer graphics and computer vision applications. However, the visual context of images is often ambiguous as images can be perceived with emphasis on different attributes. In this paper, we present a method for learning the semantic visual similarity among images, inferring their latent attributes and embedding them into multi-spaces corresponding to each latent attribute. We consider the multi-embedding problem as an optimization function that evaluates the embedded distances with respect to qualitative crowdsourced clusterings. The key idea of our approach is to collect and embed qualitative pairwise tuples that share the same attributes in clusters. To ensure similarity attribute sharing among multiple measures, image classification clusters are presented to, and solved by users. The collected image clusters are then converted into groups of tuples, which are fed into our group optimization algorithm that jointly infers the attribute similarity and multi-attribute embedding. Our multi-attribute embedding allows retrieving similar objects in different attribute spaces. Experimental results show that our approach outperforms state-of-the-art multi-embedding approaches on various datasets, and demonstrate the usage of the multi-attribute embedding in image retrieval application.

1. Introduction

Understanding semantic similarity among images is the core of a wide range of computer graphics and computer vision applications, especially in image retrieval (Douze et al., 2011). However, it is a particularly challenging task as it reflects how humans perceive images, a task that cannot be inferred by low-level analysis. Supervised learning is a common means of studying such semantic problem, for which, the ground truth of how humans perceive similarity among images is critical.

However, the semantic context of images is often ambiguous as images can be perceived with emphasis on different attributes (see Fig. 1). One example out of many is the separation of categorization and style (*e.g.*, color, light, scene type, *etc.*). One can claim that two images are similar due to their categorization and another may find two images of similar categorization different due to their style. Similarities between the images may be measured in multiple attributes, which can be contradictory to each other.

Humans cannot state a consistent meaningful measure of semantic similarity for a large batch of images. Therefore, annotations about semantic similarity collected by crowd queries are qualitative in nature. In addition, they only contain a partial view of a whole dataset. To consolidate the partial and possibly conflicting views, the collected qualitative data is often embedded into a common Euclidean space that hosts all the images, such that the quantitative metric distances among images reflect the aggregate of the qualitative measures as much as possible (Kleiman et al., 2016; Tamuz et al., 2011). However, since semantic similarity may reflect various attributes, one embedding space cannot represent well multiple distances among images. A few existing works address such contradictions by disentangling similarities in multiple attribute spaces, based on assuming latent embedding distributions (Amid and Ukkonen, 2015), explicitly specifying attributes (Veit et al., 2017), or learning similarities along with worker and context information (ho Kim et al., 2018).

[*] Corresponding author.
E-mail address: qiong.zn@gmail.com (Q. Zeng).
[1] First co-author.

Peer review under responsibility of Zhejiang University and Zhejiang University Press.

https://doi.org/10.1016/j.visinf.2018.09.004
2468-502X/

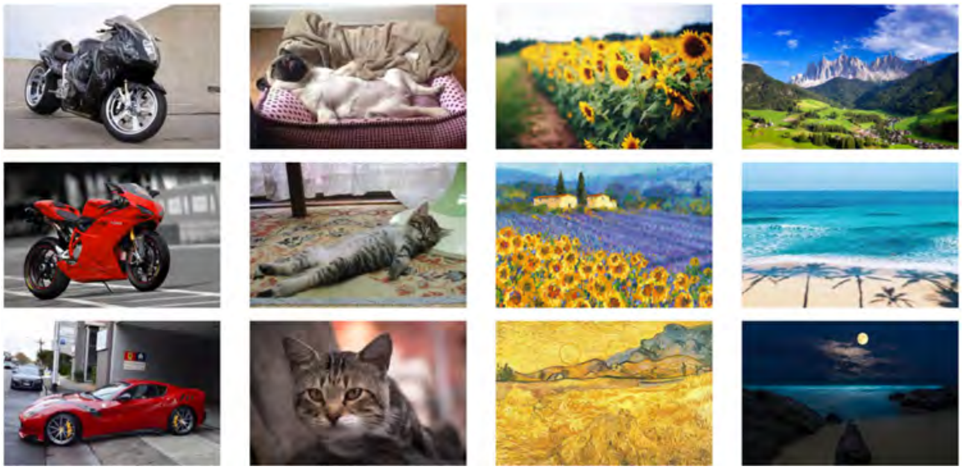

Fig. 1. Often the semantics of images are ambiguous. It is unclear whether images in the second row are more similar to the top or bottom row. Images in the top row have similar category to the second row, while images in the bottom row are similar in style (*e.g.*, color, pose, rendering and light).

In this paper, we present an unsupervised method for multi-attribute embedding, which disentangles similarity annotations based on their latent attributes and embeds them into multiple corresponding embedding spaces. The distances in each embedding space well represent object similarity under the corresponding attribute. The task is challenging since it encapsulates a two-fold problem. First, the semantic similarity among images has no clear quantitative measure and thus, it must be deduced from qualitative measures. Second, the attribute that each crowd member relies on is unknown.

To addressed the challenges, we collect qualitative semantic similarities from crowdsourced clustering queries, and embed images into multiple spaces by optimizing an objective function that evaluates the embedded distances with respect to the qualitative similarities. A critical issue in the optimization is to infer which attribute is used in answering a particular query. Thus, each query is associated with an additional variable on top of the unknown coordinates of the embedded elements. The key idea of our approach is to collect and embed qualitative measures in *groups*. The grouped measures necessarily share the same attributes, which significantly reduces the number of unknown variables.

More specifically, the task we use is designed as classifying a collection of images into clusters (see Fig. 2). This necessarily leads the user to use a single attribute in providing a series of qualitative measures on the collection of images. Each clustering annotation is then converted into a group of $T(i, j, \theta)$-like tuples, where θ indicates whether image O_i is similar to image O_j ($\theta = 1$) or not ($\theta = 0$), and fed into our embedding optimization. As we shall show, the optimization with tuple groups requires less variables, leading to higher quality embeddings.

Besides, we further explore the usage of multi-attribute embedding in image retrieval by leveraging recent progress in the field of Deep Neural Networks (Krizhevsky et al., 2012). A CNN model is presented to map an image into the multi-attribute embeddings, so that it lies near those of other images containing similar objects in different attribute spaces.

We evaluate our multi-attribute embedding approach on various synthetic and crowdsourcing datasets, and show that it outperforms the state-of-the-art multi-embedding approaches. We also show that our method can support intuitive image retrieval by turning different attributes on and off .

2. Related work

Together with the availability of massive data, crowdsourcing allows supervised learning to be performed in large scale (Russakovsky et al., 2015), providing an efficient way to measure human perception in various contexts, such as product design (Bell and Bala, 2015), illustration style (Garces et al., 2014), font similarity (O'Donovan et al., 2014) and entity matching (Wang et al., 2012). For a recent comprehensive study of crowdsourcing, please refer to Chittilappilly et al. (2016).

Single embedding metric learning. The problem of consolidating numerous instances of information, which is often quantitative, into a single consistent space is referred to as *metric learning*, and is widely studied (Globerson and Roweis, 2005; Wang et al., 2011; Xie and Xing, 2013; Xing et al., 2002). Quantifying human similarity perception is challenging since it is often qualitative and relative. A number of metric learning methods focus on recovering a single embedding space from such relative similarity measures, in the form of paired comparisons (Agarwal et al., 2007) or relative triplets (Tamuz et al., 2011). Some recent methods emphasize the importance of qualitative measures from crowd clustering (Gomes et al., 2011; Wilber et al., 2014), which provide more information compared to pairs or triplets of images. Kleiman et al. (2016) leveraged crowd clustering to learn semantic similarity. In addition, some methods learn single embedding similarity metric by combining both quantitative representation and qualitative measures. For example, Li et al. (2018) proposed a continuous dissimilarity metric by leveraging quantitative deep neural features and qualitative inter-class measures.

However, such methods often assume that similarity between two objects can be depicted by a single scalar value, and thus a single embedding space can capture similarity among a set of objects. Similarity measures, which might be from different attributes, are "fused" into one embedding space. Instead, we model similarity between two objects as a multi-dimension vector, *i.e.*, two objects may have different degree of similarity under different attributes. We propose to "disentangle" similarities and embed them in multiple embeddings by their attributes, which can be separately explored.

Multi-attribute embedding. Learning multiple embeddings in general cases has not been explored much, even though it is often essential for various human-computational applications. Recent research in natural language processing has proposed a number of models in which words are associated with several corresponding embeddings based on human word similarity judgments (Li and Jurafsky, 2015; Liu et al., 2015; Wu and Giles, 2015). However, these models use additional information such as local co-occurrence and sentence context which are not available in the general case.

Please cite this article in press as: Zeng Q., et al., Group optimization for multi-attribute visual embedding. Visual Informatics (2018), https://doi.org/10.1016/j.visinf.2018.09.004.

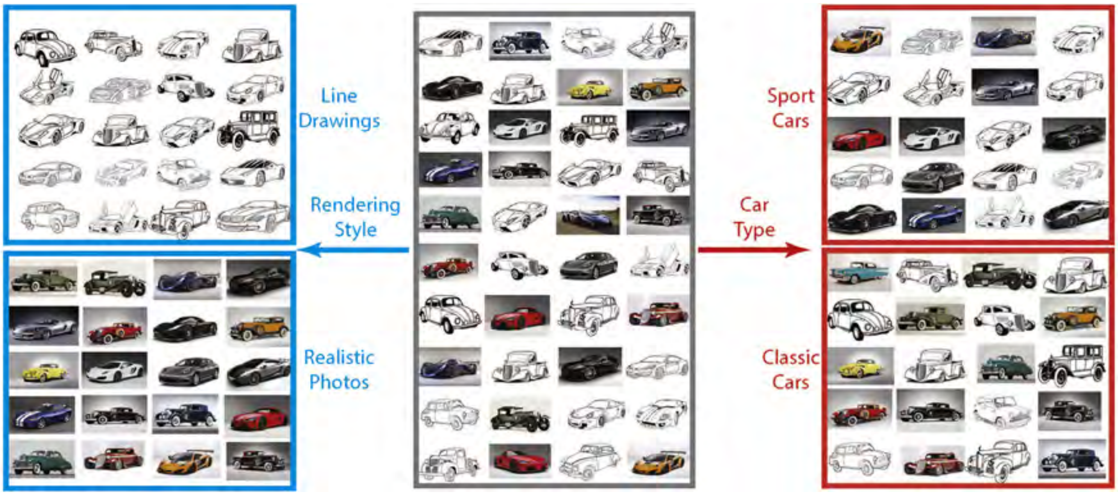

Fig. 2. A collection of images are classified into clusters by crowd users. Different clusterings reflect similarity in different attributes, which may be contradictory to each other. For example, cars may be clustered by either considering the rendering style shown in the leftmost column, or car type as shown in the rightmost column.

Amid and Ukkonen (2015) proposed a multi-view triplet embedding algorithm (MVTE) to reveal multiple embeddings by maximizing the sum of log-probabilities of triplet-embedding mapping over all triplets. The triplet-embedding mapping is defined as an heuristic indicator function of the embedding based on distribution assumptions of the underlying embedding spaces. The method alternates between optimizing the embeddings with fixed indicators and deriving the indicators from embeddings. Unlike MVTE, we do not make any distribution assumption of the underlying embedding spaces. Instead, we introduce a set of *attribute inference variables* that represent the mapping probabilities.

A deep model to learn multi-attribute similarity is proposed by Veit et al. (2017). In this work, multi-attribute embeddings are learned directly from image features through a supervised way: the network is provided with a given set of triplets and their corresponding attribute. Different to this work, none of our data is labeled with accurate attribute. The gist of our work is to estimate which similarity triplet is associated with each attribute, and at the same time generate multi-attribute embeddings that fit each of these unknown attributes.

ho Kim et al. (2018) introduced an end-to-end deep context embedding networks (CENs) to learn multi-attribute image embeddings by modeling crowd worker bias and image context. They collect image similarity annotations from crowd clusterings, as well as detailed worker and context information. Images in each cluster are considered as similar ones, while others are dissimilar ones. Semantic attributes for embeddings are modeled and encoded from worker annotation behavior and clustering context. Rather than learning attributes from workers and context, we directly optimize our attribute inference variables simultaneously with the embedding variables, to model complex multi-attribute embedding relationships.

3. Group-based crowd queries

The design of annotation task is the key for gathering information from a crowd. Inspired by Kleiman et al. (2016), we ask workers to perform crowd clusterings. A set of images is presented to each worker, who is requested to classify the images into multiple groups. Note that this classification task is cost-effective since it can yield a large amount of similarity annotations from these clusters. More importantly, such a classification naturally leads the worker to use a single similarity attribute while performing the task. Thus, all the derived pairwise similarity annotations can be assigned to the same embedding space, i.e., they are grouped. The group queries greatly reduce the amount of affiliations to be inferred, as instead of inferring the affiliation of each triplet, only a single affiliation for each group is required.

Formally, in a query, we ask a worker to classify $\mathbb{N}$ images into at most $\mathbb{B}$ bins/clusters $\{\mathcal{S}_c\}$. The aforementioned $T(i,j,k)$-like triplets can be derived from the clusters as $\{T(i,j,k)\}$, where, $O_i, O_j \in \mathcal{S}_x, O_k \in \mathcal{S}_y$ and $x \neq y$, i.e., two images from the same cluster are considered to be more similar than the third one from another cluster. In practice, we chose to use a simpler representation of qualitative similarities—$T(i,j,\theta)$-like pairwise tuples. These tuples can be derived from the clusters $\{\mathcal{S}_c\}$ by producing a tuple $\{T(i,j,1)\}$ where $O_i, O_j \in \mathcal{S}_x$ and a tuple $\{T(i,j,0)\}$ where $O_i \in \mathcal{S}_x, O_j \in \mathcal{S}_y$. In other words, two images are considered to be similar/dissimilar if they are from same/different clusters. We denote pairwise tuples derived from query $q \in \mathcal{Q}$ as $\mathcal{T}^q = \{T(i,j,\theta)\}$. In the next section, we present a group optimization algorithm that takes grouped tuples $T(i,j,\theta)$ as input.

4. Multi-attribute embedding

As discussed above, a multitude of attributes cannot be captured in a consistent way within a single embedding space. Thus, we compute multiple embedding spaces $\mathcal{E} = \{E_s\}$ dedicated to different similarity attributes.

To associate grouped $T(i,j,\theta)$-tuples with appropriate embedding spaces, there are two sets of variables to solve. One set contains the attribute inference variables α_s^q, which indicates the likelihood that query q is based on the sth similarity attribute. The other set contains the coordinates of the images in each embedding.

Let us denote the coordinates of image O_* in embedding E_s as $O_{*,s}$. We use contrastive loss (Chopra et al., 2005) to model how well tuple $T(i,j,\theta)$ fits in the sth embedding E_s:

$$L(T(i,j,\theta),E_s) = \theta \times d(O_{i,s},O_{j,s})^2 \\ + (1-\theta) \times \max(0, m - d(O_{i,s},O_{j,s}))^2, \tag{1}$$

where $d(O_{i,s},O_{j,s}) = \|O_{i,s} - O_{j,s}\|_2$ and m is a margin for embedding dissimilar images apart from each other.

The loss of associating grouped tuples $\mathcal{T}^q$ with embedding E_s is then:

$$L(\mathcal{T}^q, E_s) = \sum_{T(i,j,\theta)\in\mathcal{T}^q} L(T(i,j,\theta),E_s). \tag{2}$$

Intuitively, $L(\mathcal{T}^q, E_s)$ is small when the tuples $\mathcal{T}^q$ from query q are associated with the embedding space that corresponds to the similarity attribute used by query q. However, it is unknown which embedding space is the best fit. We introduce attribute inference variables α_s^q to address this problem. Formally, the aggregate loss of grouped tuples $\mathcal{T}^q$ with respect to multiple embeddings $\{E_s\}$ is:

$$L(\mathcal{T}^q) = \sum_{E_s \in \mathcal{E}} \alpha_s^q \times L(\mathcal{T}^q, E_s), \qquad \sum_{s=1}^{|\mathcal{E}|} \alpha_s^q = 1, \alpha_s^q > 0. \tag{3}$$

An inference variable α_s^q can be interpreted as the probability that query q is based on the sth similarity attribute. As the optimization progresses, α_s^q gradually converge to associate query q with a specific embedding.

Finally, we sum the loss for all queries, and the optimization can be written as:

$$\operatorname*{argmin}_{\alpha_s^q,\ O_{*,s}} \sum_{q \in \mathcal{Q}} \sum_{E_s \in \mathcal{E}} \alpha_s^q \times \sum_{T(i,j,\theta) \in \mathcal{T}^q} L(T(i,j,\theta), E_s), \tag{4}$$

where $\sum_s \alpha_s^q = 1$, and $\alpha_s^q > 0$. Note that there is one attribute inference variable α_s^q per query per embedding, i.e., their total number is $|\mathcal{Q}| \times |\mathcal{E}|$. The loss function is differentiable with respect to variables α_s^q and $O_{*,s}$, thus it can be optimized with gradient descent based optimizers. We solve these two sets of variables simultaneously.

Initialization. If the embedding spaces are initialized with the same coordinates, or symmetrically with respect to the queries, the gradients are exactly the same. Thus, the gradient descent optimization updates them in the same way, which leads to identical embeddings. To avoid this, we use random initialization for the embedding coordinates. It can be assumed that the initial random embeddings are not equivalent or symmetric to one another with respect to the queries. However, in the beginning, such asymmetry is probably weak, i.e., there is no strong tendency for a query to belong to a specific embedding. The asymmetry is gradually reinforced by our algorithm, and the embeddings evolve into quite different spaces corresponding to multi-attributes. We initialize α_s^q to $\frac{1}{|\mathcal{E}|}$, indicating that the queries have the same probability to be based on any of the unknown attributes. For specific applications, where relevant prior information can be leveraged, they can also be initialized with bias for different attributes.

5. Algorithm analysis

Two distinctive features of our algorithm are the group optimization and the use of attribute inference variables. We study their effectiveness by comparison to alternatives approaches. Instead of using subjective crowdsourcing data, we leverage objective synthetic data as ground truth to analyze our algorithm.

Analysis settings. We introduce a synthetic "AOB" dataset, which contains 214 points ($|O_*| = 214$), distributed to form "A", "O", and "B"utterfly shapes in the ground truth embeddings $\mathcal{E}_{gt} = \{E_{gtA}, E_{gtO}, E_{gtB}\}$ (see Fig. 3). The points are indexed sequentially according to point coordinates in E_{gtA} and E_{gtB}, so the embeddings are different but not completely independent. In E_{gtO}, the points are indexed randomly, so that the embedding is completely independent of the other two. This way, the dataset simulates both dependent and independent attributes. We color the points by smoothly mapping their indices into continuous colors, *i.e.*, points with neighboring indices have similar colors.

We generate $|\mathcal{Q}|(= 600)$ random queries from each ground truth embedding and attempt to recover them by simulated query answers. Each query contains $\mathbb{N}(= 20)$ randomly sampled objects. Note that the random sampling strategy does not use any prior knowledge of the embeddings, to simulate actual crowdsourcing scenario where the ground truth is unknown. The answers are generated using K-means clustering of the samples, with $\mathbb{B}(= 5)$ seeds. The clustering is based on the position of objects in one of the embeddings (selected in random), to simulate users query answers which are based on a single unknown attribute. 114,000 tuples are inferred from the clustering query answers of each embedding.

We evaluate the quality of recovered embeddings $\mathcal{E}$ based on the Normalized Discounted Cumulative Gain (NDCG) metric (Järvelin and Kekäläinen, 2000). NDCG is widely used in evaluating information retrieval relevance (Burges et al., 2005; Clarke et al., 2008), and suitable for evaluating the recovery quality of the similarity based embedding spaces. More specifically, we first compute K-nearest ($K = 0.1 \times |O_*|$) neighbors for each point in each recovered embedding in $\mathcal{E}$ and its corresponding embedding[2] in $\mathcal{E}_{gt}$, then with the corresponding ranked lists, NDCG are computed and averaged over all points.

5.1. Group optimization

A common approach for computing multi-embeddings is to collect and optimize over individual tuples. In this approach, each tuple may be based on a different attribute, thus the optimization has to infer attributes for each tuple individually. Formally, the non-group optimization problem can be written as:

$$\operatorname*{argmin}_{\alpha_s^{(i,j,\theta)},\ O_{*,s}} \sum_{E_s \in \mathcal{E}} \sum_{T(i,j,\theta) \in \mathcal{T}} \alpha_s^{(i,j,\theta)} \times L(T(i,j,\theta), E_s), \tag{5}$$

where $\alpha_s^{(i,j,\theta)}$ is the attribute inference variable indicating the probability of associating tuple $T(i,j,\theta) \in \mathcal{T}$ with embedding E_s, and $\sum_s \alpha_s^{(i,j,\theta)} = 1$, and $\alpha_s^{(i,j,\theta)} > 0$. The tuples can either be collected using single-tuple queries or inferred from a clustering query. Clearly, this formulation leads to many more variables to optimize than the group optimization, since the number of attribute inference variables is proportional to the number of tuples.

We apply group and non-group optimizations on AOB dataset, and show a visual comparison in Fig. 3. Note that in both optimizations the embeddings are computed from the same random initialization. As can be seen in the figure, group optimization leads to a significantly better recovery of the ground truth embeddings than the non-group version. The group optimization (top row) produces distinct embeddings that resemble the ground truth, while the non-group optimization (bottom row) produces noisy embeddings that are quite similar to each other. The recovery quality is also evident in the color coding of the results. A high quality recovery should present color coding which is similar to the ground truth. While this is true for the group optimization results, in the non-group optimization results the color coding of all embeddings is similar only to the "A" and butterfly shapes and not to the "O" shape. This suggests that the attributes are not separated correctly, as all embeddings are influenced by tuples that represent similar color coding.

We also quantitatively measure the quality of the multi-attri bute embedding recovery. At each iteration, we compute the average NDCG of the three embedding results, with group and non-group optimization. As can be seen in Fig. 4, the group optimization converges much faster to more accurate embeddings.

[2] In case of poor recovery, the correspondence between $\mathcal{E}$ and $\mathcal{E}_{gt}$ is not clear (see Fig. 3 middle lower). In this case, we compute NDCG for all possible mappings between them, and pick the one with the highest NDCG as the most likely mapping.

Please cite this article in press as: Zeng Q., et al., Group optimization for multi-attribute visual embedding. Visual Informatics (2018), https://doi.org/10.1016/j.visinf.2018.09.004.

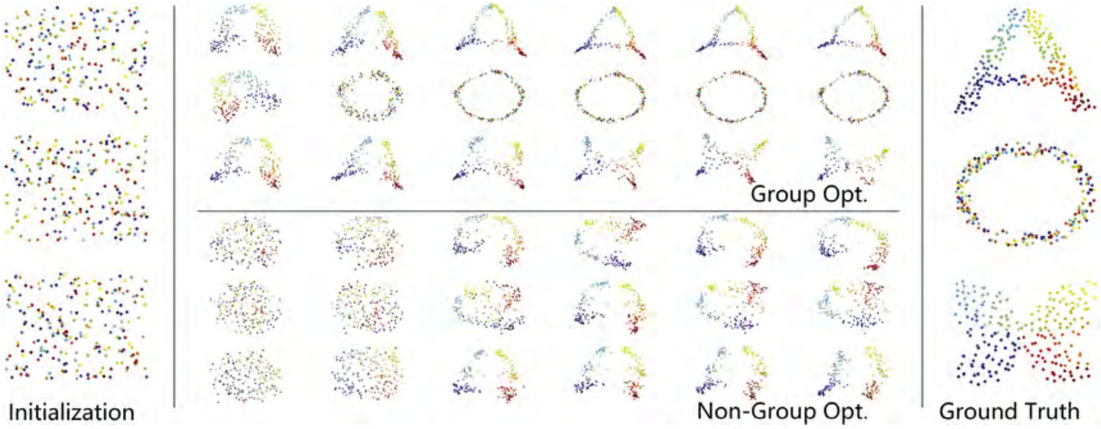

Fig. 3. A visual comparison of embedding results w/wo group optimization at iteration 5, 10, 20, 30, 60 and 100. Both methods are optimized from the same initialization, but group optimization converges faster.

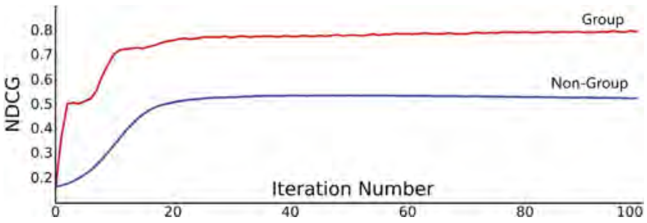

Fig. 4. A quantitative comparison of Fig. 3 results using NDCG metric. It shows that the group optimization converges faster to more accurate embeddings.

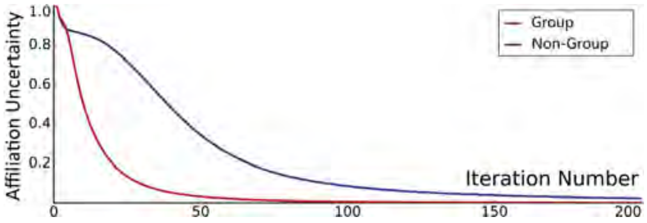

Fig. 5. Affiliation uncertainty curve during recovery of "AO" embeddings, with and without group optimization.

5.2. Attribute inference variables

As discussed above, the optimization starts with random initialization of multiple embedding spaces, and progressively evolves to differentiate between distinct attributes. We examine this phenomenon in the task of recovering E_{gtA} and E_{gtO} from simulated queries and answers, with an affiliation uncertainty metric:

$$\mathscr{U} = \frac{1}{|\mathcal{Q}|} * \sum_{q\in\mathcal{Q}} \frac{\min(\alpha_1^q, \alpha_2^q)}{\max(\alpha_1^q, \alpha_2^q)}, \tag{6}$$

where α_1^q and α_2^q are the attribute inference variables associated with E_A and E_O in query q. As shown in Fig. 5, in the beginning, $\mathscr{U}$ is high, as the two initial embeddings are still in chaos state and it is not significant whether a query is associated with E_A or E_O. However, since E_A and E_O are not likely to be symmetric with respect to the queries, the asymmetry is gradually reinforced while one of the embeddings is evolving towards E_A and the other one towards E_O. This can be observed from the reduction of affiliation uncertainty as the optimization progresses.

Similarly, we can define the affiliation uncertainty metric for non-group optimization as:

$$\mathscr{U}_n = \frac{1}{|\mathcal{T}|} * \sum_{T(i,j,\theta)\in\mathcal{T}} \frac{\min(\alpha_1^{(i,j,\theta)}, \alpha_2^{(i,j,\theta)})}{\max(\alpha_1^{(i,j,\theta)}, \alpha_2^{(i,j,\theta)})}, \tag{7}$$

where $\alpha_1^{(i,j,\theta)}$ and $\alpha_2^{(i,j,\theta)}$ are the attribute inference variables associated with E_A and E_O for tuple $T(i,j,\theta)$. $\mathscr{U}_n$ is also plotted in Fig. 5. $\mathscr{U}_n$ also reduces as the optimization progresses, which shows the attribute inference variables are somewhat effective without the group optimization. Still, the group optimization reduces affiliation uncertainty more effectively than the non-group version.

6. Experimental results

We implement our algorithm with Tensorflow (Abadi et al., 2015). In Eqs. (4) and (5), instead of directly optimizing α_s to satisfy $\sum_s \alpha_s = 1$ and $\alpha_s^q > 0$, we let $\alpha_s = softmax(\beta_s)$, and optimize β_s without constraints. We use Adam method (Kingma and Ba, 2014) with learning rate 0.01 for the optimizations. The optimization process will stop when the number of iterations is larger than a threshold, or the loss function has no changes in a certain number of iterations. All code and data will be opensourced.

In this section, we mainly focus on experimental evaluations based on crowdsourcing data and synthetic data. We collect crowd data from Amazon Mechanical Turk (AMT), and perform our experiments on the following two datasets. Our crowd data collection details can be found in the supplementary material.

Chair Dataset contains 6,777 images rendered from the chair category of ShapeNet (Chang et al., 2015). For this dataset, crowd workers were required to cluster queries by considering one of the following predefined attributes: arms, legs and back of the chairs. This constraint allows us to verify qualitative accuracy. In total, 2,709 workers clustered 41,287 valid clustering queries, yielding in 7,953,827 pairwise similarity tuples.

Poster Dataset is a film poster dataset with rich semantic information, which contains 2,00 images collected by Kleiman et al. (2016). For this dataset, we collect data without any instructions on attributes. In total, 74 workers clustered 840 valid clustering queries, yielding in 36,000 pairwise similarity tuples.

6.1. Multi-attribute embeddings from Crowd Data

For Chair Dataset, we recover appropriate embeddings of corresponding attributes *without* the prior knowledge of the predefined attributes that users were asked to consider. Fig. 6 shows the final

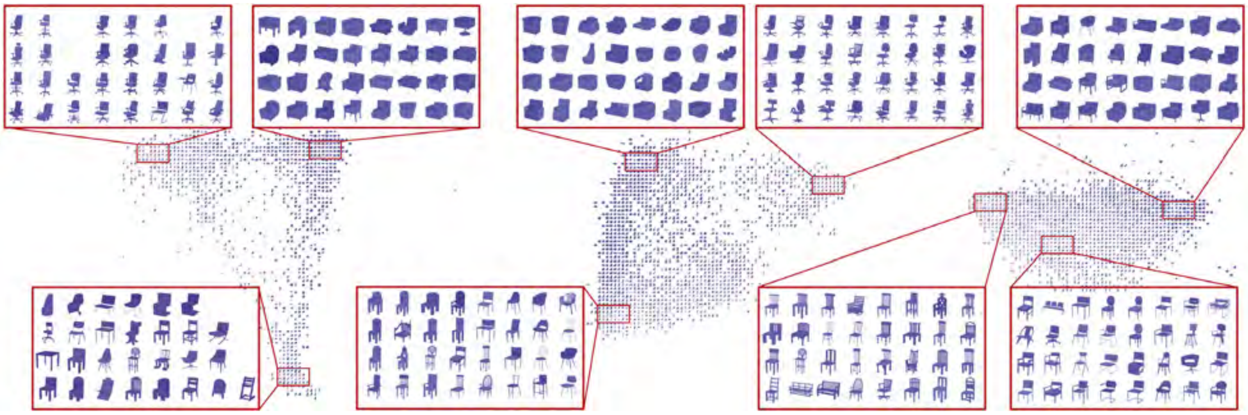

Fig. 6. Recovering multiple attributes using our approach on the chair dataset with three attributes. The results show that the embeddings reflect specific attributes, from left to right: arms, legs and back of and chairs.

recovered embeddings for three attributes. It can be seen that from left to right each embedding reflects a separate attribute. For example, in the left figure chairs are clustered according to their arms, while those in a cluster show obvious differences in their legs and backs. This demonstrates the ability of our method to automatically distinguish and classify the query answers according to their unknown attributes.

Fig. 7 shows the corresponding multi-attribute embeddings from Poster Dataset. For the purpose of display clarity, we only present 20 sampled images out of the total in the dataset. The nature of images in the dataset and the unconstrained settings of the experiment suggest the images may be categorized by the crowd workers using a large number of attributes. Thus, each embedding may reflect a mix of several attributes. Still, we can identify a meaningful distinction between the three embeddings. Embedding (a) reflects the appearance of the posters, in terms of color, composition and the content of the poster. For example, posters with white background (see marked images) appear close to each other in this embedding, even though the movies belong to various genres. Embedding (b) reflects external context such as the genre of the movie or the actors that play in it. Note that horror movies appear on the top right, sci-fi movies appear on the top left, and family movies appear on the bottom. In embedding (c), the distinction between animated movies and live action movies takes precedence over other attributes, creating two tight groups of movies, animated and non-animated.

6.2. Comparisons with (t-)MVTE

Our general approach to the problem is similar to the one presented by Amid and Ukkonen (2015), aiming to deduce multi-attribute embeddings from pairwise comparisons without constructing deep neural models. They define a heuristic indicator function for the triplet embedding based on distribution assumptions of the underlying embedding spaces. Therefore, we conduct an experiment to evaluate the effectiveness of our algorithm and compare it with (t-)MVTE multi-view embedding algorithm.

We use six synthetic datasets with multiple attributes, as well as different dimensions, as can be seen in Fig. 8. For example, 3D-2 is a set of three-dimensional points composed of two geometric models that correspond to two-attribute distributions. Each 2D data has 214 instances, while 1,600 instances for each 3D data. 600 and 4,800 clustering queries are randomly sampled from each attribute in the synthetic 2D and 3D data, which are used to produce triplets and grouped tuples. Triplets or grouped tuples in each attribute are mixed together as input to (t-)MVTE or our method for optimizing multi-attribute embedding.

Table 1
NDCG performance compared with (t-)MVTE algorithms on various synthetic datasets.

	AO	AOB	AOB8	3D-2	3D-3	3D-4
MVTE (%)	92.9	69.9	64.6	82.9	61.8	59.2
t-MVTE (%)	95.5	75.1	56.7	82.4	62.3	40.6
Ours (%)	**95.8**	**94.8**	**96.3**	**83.2**	**83.7**	**85.3**

Fig. 8 visualizes the qualitative multi-attribute embeddings recovered by (t-)MVTE and our algorithm. Compared with (t-)MVTE, our algorithm performs better in mapping embedding positions and preserving global distribution. Table 1 summarizes the NDCG of the corresponding multi-embeddings based on ground truth. Our algorithm outperforms (t-)MVTE and is more stable when dealing with complex data of higher dimensions.

6.3. Application: Cross-attribute image retrieval

The multi-embedding spaces associate each image with a spacial position in different attribute spaces. A query image may be similar to different images in various attribute spaces. For a query image, we can firstly embed it to the multi-embedding spaces and retrieve similar images cross multiple attributes. Therefore, we propose a cross-attribute image retrieval framework based on our multi-attribute embeddings.

We leverage convolutional neural structure to learn a mapping between images and their positions in the multi-embedding spaces. Our embedding positions are used as supervised annotations for the training process. More specifically, for one specific attribute embedding, the training data consists of a collection of pairs, (I_i^s, P_i^s), where I_i^s indexes an image i under an attribute space s, and P_i^s is its embedding position. Our convolutional model is a function f that receive as input an image and is expected to output its spatial position Pi. f is depended with network parameters θ. We measure the mapping error with a Euclidean loss function L:

$$L^s(\subseteq) = \sum_i \left\| f(I_i^s; \theta) - P_i^s \right\|^2 . \tag{8}$$

We adopt the network with AlexNet (Krizhevsky et al., 2012), which might be replaced with more sophisticated networks.

We experiment our image retrieval application on the Chair Dataset. The dataset is randomly separated into training and validation set, with 5,000 and 1,777 images respectively. We firstly rerender training images with a solid color to leave out noises caused by texture and shading. This schema enables us to achieve a higher prediction accuracy in validation set.

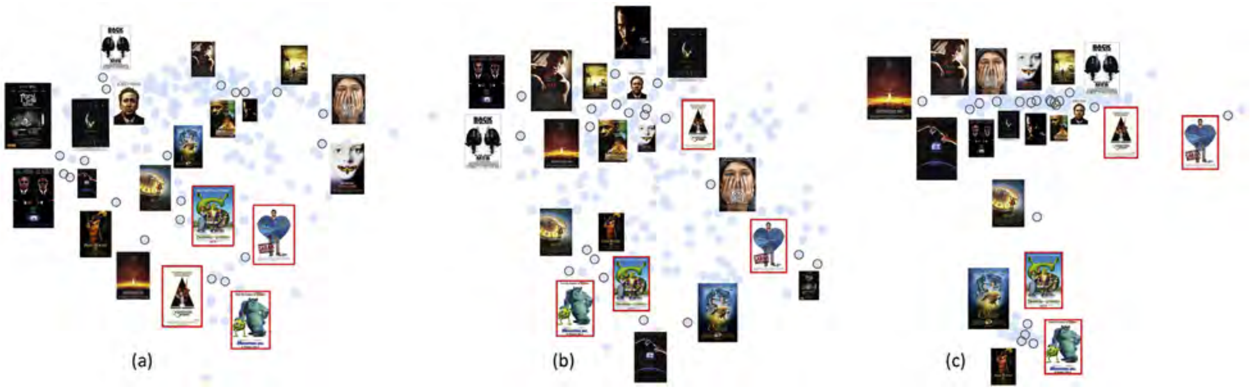

Fig. 7. Recovering multiple attributes using our approach on the poster dataset with unknown attributes. The results show that the embeddings are ordered under different attributes: color and appearance (a), genre (b), animation/live action (c).

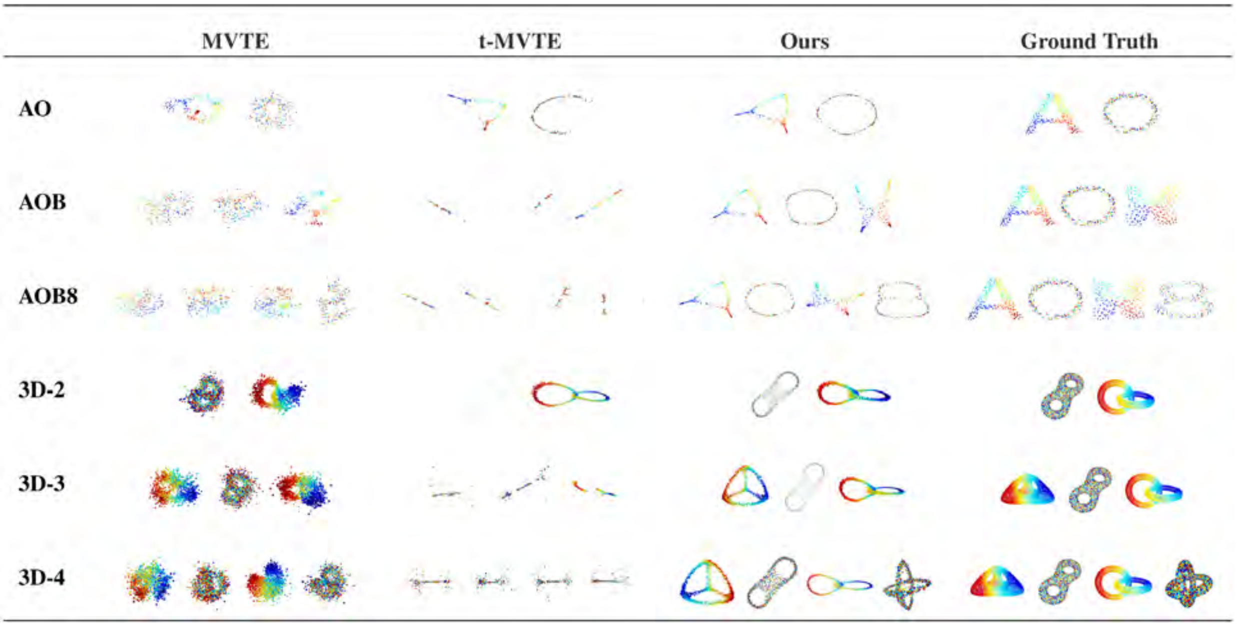

Fig. 8. Multi-attribute embedding visualization compared with (t-)MVTE algorithms on various synthetic datasets. Data points are smoothly mapped into continuous colors by their indices. Our algorithm outperform (t-)MVTE in preserving complex data distribution.

Given a new query image, our deep model can retrieve images containing similar chairs in different attributes (*e.g.*, chair arms, legs and back). The retrieved images under different attributes can be quite diverse in terms of semantics. Fig. 9 presents our cross-attribute retrieval results. The query images are randomly chosen from the validation dataset, while the retrieved images are from the training dataset. It shows that our system can retrieve different images in different semantic attributes. Take (a) for example, the first row shows retrieved images with similar arms—that all of them are no arms; the second row presents neighboring chairs with similar legs, in which their arm and back attributes are quite different; while the third row shows retrieval feedbacks with similar backs.

7. Conclusions

We have presented a method for multi-attribute embedding. The method takes qualitative measures and solves an optimization problem that embeds them into multiple spaces such that the quantitative measures in the embedded spaces agree with the qualitative measures. The optimization solves two sets of unknown parameters simultaneously: one is the embedded coordinates of the points and the other is the classification variables of the measure to the unknown attributes. We presented a group optimization and showed its power to infer the attribute classification and produce embedding coordinates. Our experimental results on crowdsourced data demonstrate the competence of our method to produce multi-embedding from inconsistent and redundant data. Our method can also be applied to intuitive image retrieval by turning on and off different attributes.

Limitations. Different similarity attributes may have different popularity, but our method does not take this into account. In addition, the number of different embeddings in our method needs to be manually set. While larger number of embeddings can better reflect more attributes, there is a risk that they actually represent noise or outlier measures. An interesting future work is to try and differentiate inliers and outliers measures, and automatically pick a suitable number of embedding spaces.

Future work. A possible future work is to try and differentiate inliers and outliers measures, and automatically pick a suitable number of embedding spaces. Another interesting future work is to extend our method to sketch based image retrieval (Yu et al., 2016). Since people often emphasize different aspects in sketches,

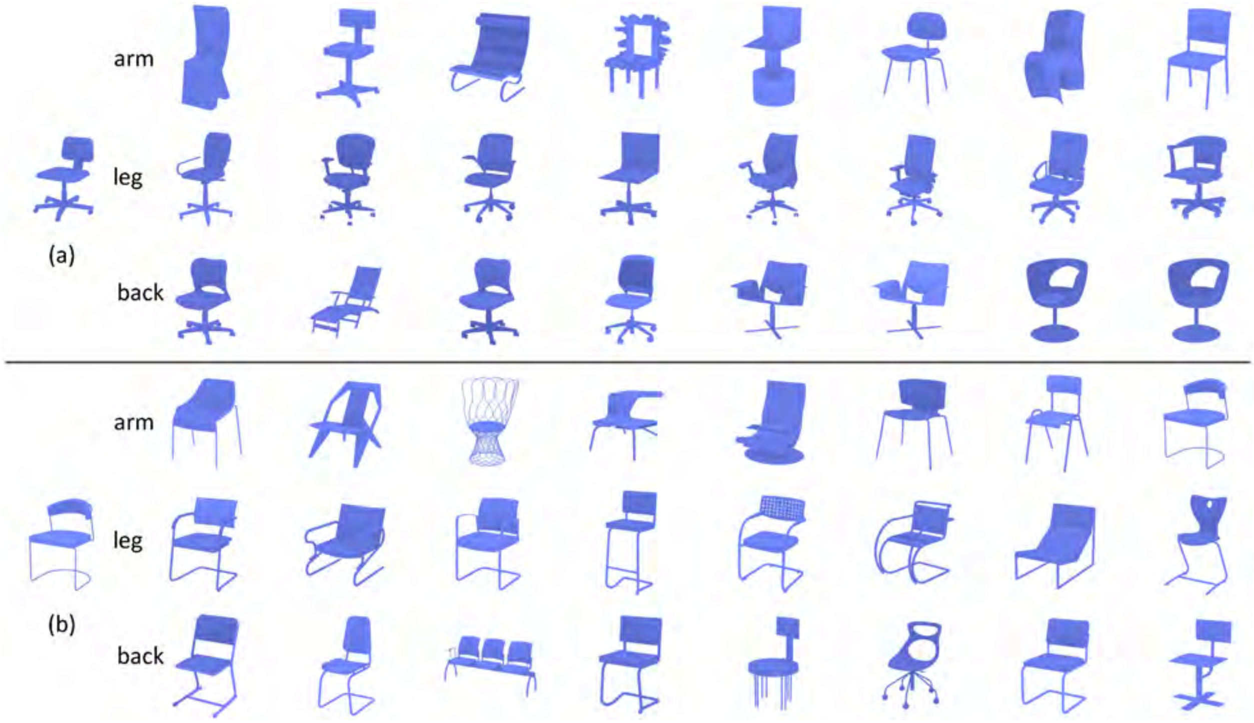

Fig. 9. Top 8 nearest neighbors of query images (a) and (b) under three attributes. From top to bottom, we turn on arm, leg and back attribute respectively for retrieval task.

learning multiple similarity embeddings among sketches and real world images may improve the effectiveness of SBIR.

Acknowledgments

This study was funded by National Key Research & Development Plan of China (No. 2016YFB1001404) and National Natural Science Foundation of China (No. 61602273).

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.visinf.2018.09.004.

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., Zheng, X., 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Agarwal, S., Wills, J., Cayton, L., Lanckriet, G., Kriegman, D., Belongie, S., 2007. Generalized non-metric multidimensional scaling. In: Proc. of Int. Conf. on AI and Statistics, San Juan, Puerto Rico.

Amid, E., Ukkonen, A., 2015. Multiview triplet embedding: learning attributes in multiple maps. In: Proc.of ICML. pp. 1472–1480.

Bell, S., Bala, K., 2015. Learning visual similarity for product design with convolutional neural networks. ACM Trans. Graph. 34 (4), 98:1–98:10. http://dx.doi.org/10.1145/2766959.

Burges, C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N., Hullender, G., 2005. Learning to rank using gradient descent. In: Proceedings of the 22nd International Conference on Machine Learning. ICML '05, ACM, New York, NY, USA, pp. 89–96. http://dx.doi.org/10.1145/1102351.1102363.

Chang, A.X., Funkhouser, T.A., Guibas, L.J., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., Xiao, J., Yi, L., Yu, F., 2015. ShapeNet: An information-rich 3D model repository. CoRR. abs/1512.03012.

Chittilappilly, A.I., Chen, L., Amer-Yahia, S., 2016. A survey of general-purpose crowdsourcing techniques. IEEE Trans. Knowl. Data Eng. 28 (9), 2246–2266. http://dx.doi.org/10.1109/TKDE.2016.2555805.

Chopra, S., Hadsell, R., LeCun, Y., 2005. Learning a similarity metric discriminatively, with application to face verification. In: Proc. of CVPR, Vol. 1. IEEE, pp. 539–546.

Clarke, C.L., Kolla, M., Cormack, G.V., Vechtomova, O., Ashkan, A., Büttcher, S., MacKinnon, I., 2008. Novelty and diversity in information retrieval evaluation. In: Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. SIGIR '08, ACM, New York, NY, USA, pp. 659–666. http://dx.doi.org/10.1145/1390334.1390446.

Douze, M., Ramisa, A., Schmid, C., 2011. Combining attributes and fisher vectors for efficient image retrieval. In: Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition. CVPR '11, IEEE Computer Society, Washington, DC, USA, pp. 745–752. http://dx.doi.org/10.1109/CVPR.2011.5995595.

Garces, E., Agarwala, A., Gutierrez, D., Hertzmann, A., 2014. A similarity measure for illustration style. ACM Trans. Graph. 33 (4), 93:1–93:9. http://dx.doi.org/10.1145/2601097.2601131.

Globerson, A., Roweis, S., 2005. Metric learning by collapsing classes. In: Proc. of NIPS. MIT Press, Cambridge, MA, USA, pp. 451–458.

Gomes, R.G., Welinder, P., Krause, A., Perona, P., 2011. Crowdclustering. In: Proc. of NIPS. Curran Associates, Inc., pp. 558–566.

Järvelin, K., Kekäläinen, J., 2000. IR evaluation methods for retrieving highly relevant documents. In: Proc. of SIGIR. ACM, New York, NY, USA, pp. 41–48. http://dx.doi.org/10.1145/345508.345545.

ho Kim, K., Mac Aodha, O., Perona, P., 2018. Context embedding networks. In: Computer Vision and Pattern Recognition, CVPR. CVPR '18.

Kingma, D.P., Ba, J., 2014. Adam: A method for stochastic optimization. In: Proceedings of the 3rd International Conference on Learning Representations, ICLR.

Kleiman, Y., Goldberg, G., Amsterdamer, Y., Cohen-Or, D., 2016. Toward semantic image similarity from crowdsourced clustering. Vis. Comput. 32 (6–8), 1045–1055.

Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks. In: Pereira, F., Burges, C.J.C., Bottou, L., Weinberger, K.Q. (Eds.), Advances in Neural Information Processing Systems 25. Curran Associates, Inc., pp. 1097–1105.

Li, M., Fish, N., Cheng, L., Tu, C., Cohen-Or, D., Zhang, H., Chen, B., 2018. Class-sensitive shape dissimilarity metric. Graph. Models 98, 33–42.

Li, J., Jurafsky, D., 2015. Do multi-sense embeddings improve natural language understanding? In: Conf. on Empirical Methods for Natural Language Processing.

Liu, Y., Liu, Z., Chua, T.S., Sun, M., 2015. Topical word embeddings. In: Proc. of AAAI. AAAI Press, pp. 2418–2424.

O'Donovan, P., Lībeks, J., Agarwala, A., Hertzmann, A., 2014. Exploratory font selection using crowdsourced attributes. ACM Trans. Graph. 33 (4), 92:1–92:9. http://dx.doi.org/10.1145/2601097.2601110.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L., 2015. ImageNet large scale visual recognition challenge. Int. J. Comput. Vision 115 (3), 211–252. http://dx.doi.org/10.1007/s11263-015-0816-y.

Tamuz, O., Liu, C., Belongie, S., Shamir, O., Kalai, A., 2011. Adaptively learning the crowd kernel. In: Getoor, L., Scheffer, T. (Eds.), Proc. of ICML. ACM, New York, NY, USA, pp. 673–680.

Veit, A., Belongie, S., Karaletsos, T., 2017. Conditional similarity networks. In: Computer Vision and Pattern Recognition, CVPR, Honolulu, HI.

Wang, J., Do, H., Woznica, A., Kalousis, A., 2011. Metric learning with multiple kernels. In: Proc. of NIPS. Curran Associates Inc., USA, pp. 1170–1178.

Wang, J., Kraska, T., Franklin, M.J., Feng, J., 2012. CrowdER: crowdsourcing entity resolution. Proc. VLDB Endow. 5 (11), 1483–1494. http://dx.doi.org/10.14778/2350229.2350263.

Wilber, M.J., Kwak, I.S., Belongie, S.J., 2014. Cost-effective hits for relative similarity comparisons. In: 2nd AAAI Conference on Human Computation and Crowdsourcing.

Wu, Z., Giles, C.L., 2015. Sense-aware semantic analysis: A multi-prototype word representation model using Wikipedia. In: Proc. of AAAI. AAAI Press, pp. 2188–2194.

Xie, P., Xing, E.P., 2013. Multi-modal distance metric learning. In: Proc. of IJCAI. AAAI Press, pp. 1806–1812.

Xing, E.P., Ng, A.Y., Jordan, M.I., Russell, S., 2002. Distance metric learning, with application to clustering with side-information. In: Proc. of NIPS. MIT Press, Cambridge, MA, USA, pp. 521–528.

Yu, Q., Liu, F., SonG, Y.Z., Xiang, T., Hospedales, T., Loy, C.C., 2016. Sketch me that shoe. In: Computer Vision and Pattern Recognition.