Image-based recommendations on styles and substitutes

Julian McAuley
UC San Diego
jmcauley@ucsd.edu

Christopher Targett
University of Adelaide
christopher.targett@student.adelaide.edu.au

Qinfeng ('Javen') Shi
University of Adelaide
javen.shi@adelaide.edu.au

Anton van den Hengel
University of Adelaide
anton.vandenhengel@adelaide.edu.au

ABSTRACT

Humans inevitably develop a sense of the relationships between objects, some of which are based on their appearance. Some pairs of objects might be seen as being alternatives to each other (such as two pairs of jeans), while others may be seen as being complementary (such as a pair of jeans and a matching shirt). This information guides many of the choices that people make, from buying clothes to their interactions with each other. We seek here to model this human sense of the relationships between objects based on their appearance. Our approach is not based on fine-grained modeling of user annotations but rather on capturing the largest dataset possible and developing a scalable method for uncovering human notions of the visual relationships within. We cast this as a network inference problem defined on graphs of related images, and provide a large-scale dataset for the training and evaluation of the same. The system we develop is capable of recommending which clothes and accessories will go well together (and which will not), amongst a host of other applications.

1. INTRODUCTION

We are interested here in uncovering relationships between the appearances of pairs of objects, and particularly in modeling the human notion of which objects complement each other and which might be seen as acceptable alternatives. We thus seek to model what is a fundamentally human notion of the visual relationship between a pair of objects, rather than merely modeling the visual similarity between them. There has been some interest of late in modeling the visual style of places [6, 27], and objects [39]. We, in contrast, are not seeking to model the individual appearances of objects, but rather how the appearance of one object might influence the desirable visual attributes of another.

There are a range of situations in which the appearance of an object might have an impact on the desired appearance of another. Questions such as 'Which frame goes with this picture', 'Where is the lid to this', and 'Which shirt matches these shoes' (see Figure 1) inherently involve a calculation of more than just visual similarity, but rather a model of the higher-level relationships between objects. The primary commercial application for such technology is in recommending an item to a user on the basis of the other items they have already showed interest in. Such systems are of considerable economic value, and are typically built by analysing meta-data, reviews, and previous purchasing patterns. By introducing into these systems the ability to examine the appearance of the objects in question we aim to overcome some of their limitations, including the 'cold start' problem [28, 41].

Figure 1: A query image and a matching accessory, pants, and a shirt.

The problem we pose inherently requires modeling human visual preferences. In most cases there is no intrinsic connection between a pair of objects, only a human notion that they are more suited to each other than are other potential partners. The most common approach to modeling such human notions exploits a set of hand-labeled images created for the task. The labeling effort required means that most such datasets are typically relatively small, although there are a few notable exceptions. A small dataset means that complex procedures are required to extract as much information as possible without over-fitting (see [2, 5, 22] for example). It also means that the results are unlikely to be transferable to related problems. Creating a labeled dataset is particularly onerous when modeling pairwise distances because the number of annotations required scales with the square of the number of elements.

We propose here instead that one might operate over a much larger dataset, even if it is only tangentially related to the ultimate goal. Thus, rather than devising a process (or budget) for manually annotating images, we instead seek a freely available source of a large amount of data which may be more loosely related to the information we seek. Large-scale databases have been collected from the web (without other annotation) previously [7, 34]. What distinguishes the approach we propose here, however, is the fact that it succeeds despite the indirectness of the connection between the dataset and the quantity we hope to model.

1.1 A visual dataset of styles and substitutes

We have developed a dataset suitable for the purposes described above based on the Amazon web store. The dataset contains over 180 million relationships between a pool of almost 6 million objects. These relationships are a result of visiting Amazon and recording the product recommendations that it provides given our (apparent) interest in the subject of a particular web page. The statistics of the dataset are shown in Table 1. An image and a category label are available for each object, as is the set of users who reviewed it. We have made this dataset available for academic use, along with all code used in this paper to ensure that our results are reproducible and extensible.[1] We label this the *Styles and Substitutes* dataset.

The recorded relationships describe two specific notions of 'compatibility' that are of interest, namely those of *substitute* and *complement* goods. Substitute goods are those that can be interchanged (such as one pair of pants for another), while complements are those that might be purchased together (such as a pair of pants and a matching shirt) [23]. Specifically, there are 4 categories of relationship represented in the dataset: 1) 'users who viewed X also viewed Y' (65M edges); 2) 'users who viewed X eventually bought Y' (7.3M edges); 3) 'users who bought X also bought Y' (104M edges); and 4) 'users bought X and Y simultaneously' (3.4M edges). Critically, categories 1 and 2 indicate (up to some noise) that two products may be substitutable, while 3 and 4 indicate that two products may be complementary. According to Amazon's own tech report [19] the above relationships are collected simply by ranking products according to the cosine similarity of the sets of users who purchased/viewed them.

Note that the dataset does not document users' preferences for pairs of images, but rather Amazon's estimate of the set of relationships between pairs objects. The human notion of the visual compatibility of these images is only one factor amongst many which give rise to these estimated relationships, and it is not a factor used by Amazon in creating them. We thus do not wish to summarize the Amazon data, but rather to use what it tells us about the images of related products to develop a sense of which objects a human might feel are visually compatible. This is significant because many of the relationships between objects present in the data are not based on their appearance. People co-purchase hammers and nails due to their functions, for example, not their appearances. Our hope is that the non-visual decision factors will appear as uniformly distributed noise to a method which considers only appearance, and that the visual decision factors might reinforce each other to overcome the effect of this noise.

1.2 Related work

The closest systems to what we propose above are content-based recommender systems [18] which attempt to model each user's preference toward particular types of goods. This is typically achieved by analyzing metadata from the user's previous activities. This is as compared to collaborative recommendation approaches which match the user to profiles generated based on the purchases/behavior of other users (see [1, 16] for surveys). Combinations of the two [3, 24] have been shown to help address the sparsity of the review data available, and the cold-start problem (where new products don't have reviews and are thus invisible to the recommender system) [28, 41]. The approach we propose here could also help address these problems.

There are a range of services such as Jinni[2] which promise content-based recommendations for TV shows and similar media, but the features they expoit are based on reviews and meta-data (such as cast, director etc.), and their ontology is hand-crafted. The Netflix prize was a well publicized competition to build a better content-based video recommender system, but there again no actual image analysis is taking place [17]. Hu et al. [9] describe a system for identifying a user's style, and then making clothing recommendations, but this is achieved through analysis of 'likes' rather than visual features.

Content-based image retrieval gives rise to the problem of bridging the 'semantic-gap' [32], which requires returning results which have similar semantic content to a search image, even when the pixels bear no relationship to each other. It thus bears some similarity to the visual recommendation problem, as both require modeling a human preference which is not satisfied by mere visual similarity. There are a variety of approaches to this problem, many of which seek a set of results which are visually similar to the query and then separately find images depicting objects of the same class as those in the query image; see [2, 15, 22, 38], for example. Within the Information Retrieval community there has been considerable interest of late in incorporating user data into image retrieval systems [37], for example through browsing [36] and click-through behavior [26], or by making use of social tags [29]. Also worth mentioning with respect to image retrieval is [12], which also considered using images crawled from Amazon, albeit for a different task (similar-image search) than the one considered here.

There have been a variety of approaches to modeling human notions of similarity between different types of images [30], forms of music [31], or even tweets [33], amongst other data types. Beyond measuring similarity, there has also been work on measuring more general notions of compatibility. Murillo et al. [25], for instance, analyze photos of groups of people collected from social media to identify which groups might be more likely to socialize with each other, thus implying a distance measure between images. This is achieved by estimating which of a manually-specified set of 'urban tribes' each group belongs to, possibly because only 340 images were available.

Yamaguchi et al. [40] capture a notion of visual style when parsing clothing, but do so by retrieving visually similar items from a database. This idea was extended by Kiapour et al. [14] to identify discriminating characteristics between different styles (hipster vs. goth for example). Di et al. [5] also identify aspects of style using a bag-of-words approach and manual annotations.

A few other works that consider visual features specifically for the task of clothing recommendation include [10, 13, 20]. In [10] and [13] the authors build methods to parse complete outfits from single images, in [10] by building a carefully labeled dataset of street images annotated by 'fashionistas', and in [13] by building algorithms to automatically detect and segment items from clothing images. In [13] the authors propose an approach to learn relationships between

[1] http://cseweb.ucsd.edu/~jmcauley

[2] http://jinni.com

Category	Users	Items	Ratings	Edges
Books	8,201,127	1,606,219	25,875,237	51,276,522
Cell Phones & Accessories	2,296,534	223,680	5,929,668	4,485,570
Clothing, Shoes & Jewelry	3,260,278	773,465	25,361,968	16,508,162
Digital Music	490,058	91,236	950,621	1,615,473
Electronics	4,248,431	305,029	11,355,142	7,500,100
Grocery & Gourmet Food	774,095	120,774	1,997,599	4,452,989
Home & Kitchen	2,541,693	282,779	6,543,736	9,240,125
Movies & TV	2,114,748	150,334	6,174,098	5,474,976
Musical Instruments	353,983	65,588	596,095	1,719,204
Office Products	919,512	94,820	1,514,235	3,257,651
Toys & Games	1,352,110	259,290	2,386,102	13,921,925
Total	20,980,320	5,933,184	143,663,229	180,827,502

Table 1: The types of objects from a few categories in our dataset and the number of relationships between them.

clothing items and events (e.g. birthday parties, funerals) in order to recommend event-appropriate items. Although related to our approach, these methods are designed for the specific task of clothing recommendation, requiring hand-crafted methods and carefully annotated data; in contrast our goal is to build a general-purpose method to understand relationships between objects from large volumes of *unlabeled* data. Although our setting is perhaps most natural for categories like clothing images, we obtain surprisingly accurate performance when predicting relationships in a variety of categories, from recommending outfits to predicting which books will be co-purchased based on their cover art.

In summary, our approach is distinct from the above in that we aim to generalize the idea of a visual distance measure beyond measuring only similarity. Doing so demands a very large amount of training data, and our reluctance for manual annotation necessitates a more opportunistic data collection strategy. The scale of the data, and the fact that we don't have control over its acquisition, demands a suitably scalable and robust modeling approach. The novelty in what we propose is thus in the quantity we choose to model, the data we gather to do so, and the method for extracting one from the other.

1.3 A visual and relational recommender system

We label the process we develop for exploiting this data a *visual and relational recommender system* as we aim to model human visual preferences, and the system might be used to recommend one object on the basis of a user's apparent interest in another. The system shares these characteristics with more common forms of recommender system, but does so on the basis of the appearance of the object, rather than metadata, reviews, or similar.

2. THE MODEL

Our notation is defined in Table 2.

We seek a method for representing the preferences of users for the visual appearance of one object given that of another. A number of suitable models might be devised for this purpose, but very few of them will scale to the volume of data available.

For every object in the dataset we calculate an F-dimensional feature vector $\mathbf{x} \in \mathbb{R}^F$ using a convolutional neural network as described in Section 2.3. The dataset contains

notation	explanation
$\mathbf{x}_i$	feature vector calculated from object image i
F	feature dimension (i.e., $\mathbf{x}_i \in \mathbb{R}^F$)
r_{ij}	a relationship between objects i and j
$\mathcal{R}$	the set of relationships between all objects
$d_\theta(\mathbf{x}_i,\mathbf{x}_j)$	parameterized distance between $\mathbf{x}_i$ and $\mathbf{x}_j$
$\mathbf{M}$	$F \times F$ Mahalanobis transform matrix
$\mathbf{Y}$	an $F \times K$ matrix, such that $\mathbf{Y}\mathbf{Y}^T = \mathbf{M}$
$\mathbf{D}^{(u)}$	diagonal user-personalization matrix for user u
$\sigma_c(\cdot)$	shifted sigmoid function with parameter c
$\mathcal{R}^*$	$\mathcal{R}$ plus a random sample of non-relationships
$\mathcal{U},\mathcal{V},\mathcal{T}$	training, validation, and test subsets of $\mathcal{R}^*$
$\mathbf{s}_i$	K-dimension embedding of $\mathbf{x}_i$ into 'style-space'

Table 2: Notation.

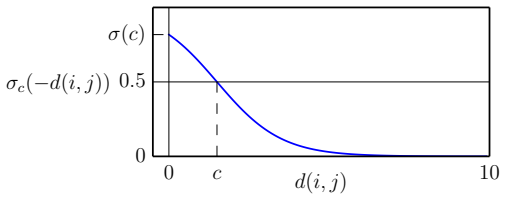


Figure 2: Shifted (and inverted) sigmoid with parameter $c = 2$.

a set $\mathcal{R}$ of relationships where $r_{ij} \in \mathcal{R}$ relates objects i and j. Each relationship is of one of the four classes listed above. Our goal is to learn a parameterized distance transform $d(\mathbf{x}_i,\mathbf{x}_j)$ such that feature vectors $\{\mathbf{x}_i,\mathbf{x}_j\}$ for objects that are related ($r_{ij} \in \mathcal{R}$) are assigned a lower distance than those that are not ($r_{ij} \notin \mathcal{R}$). Specifically, we seek $d(\cdot,\cdot)$ such that $P(r_{ij} \in \mathcal{R})$ grows monotonically with $-d(\mathbf{x}_i,\mathbf{x}_j)$.

Distances and probabilities: We use a shifted sigmoid function to relate distance to probability thus

$$P(r_{ij} \in \mathcal{R}) = \sigma_c(-d(\mathbf{x}_i,\mathbf{x}_j)) = \frac{1}{1 + e^{d(\mathbf{x}_i,\mathbf{x}_j)-c}}. \quad (1)$$

This is depicted in Figure 2. This decision allows us to cast the problem as logistic regression, which we do for reasons of scalability. Intuitively, if two items i and j have distance $d(\mathbf{x}_i,\mathbf{x}_j) = c$, then they have probability 0.5 of being related; the probability increases above 0.5 for $d(\mathbf{x}_i,\mathbf{x}_j) < c$, and

decreases as $d(\mathbf{x}_i,\mathbf{x}_j) > c$. Note that we do not specify c in advance, but rather c is chosen to maximize prediction accuracy.

We now describe a set of potential distance functions.

Weighted nearest neighbor: Given that different feature dimensions are likely to be more important to different relationships, the simplest method we consider is to learn which feature dimensions are relevant for a particular relationship. We thus fit a distance function of the form

$$d_{\mathbf{w}}(\mathbf{x}_i,\mathbf{x}_j) = \|\mathbf{w} \circ (\mathbf{x}_i - \mathbf{x}_j)\|_2^2, \tag{2}$$

where $\circ$ is the Hadamard product.

Mahalanobis transform: (eq. 2) is limited to modeling the visual similarity between objects, albeit with varying emphasis per feature dimension. It is not expressive enough to model subtler notions, such as which pairs of pants and shoes belong to the same 'style', despite having different appearances. For this we need to learn how different feature dimensions *relate* to each other, i.e., how the features of a pair of pants might be transformed to help identify a compatible pair of shoes.

To identify such a transformation, we relate image features via a *Mahalanobis distance*, which essentially generalizes (eq. 2) so that weights are defined at the level of pairs of features. Specifically we fit

$$d_{\mathbf{M}}(\mathbf{x}_i,\mathbf{x}_j) = (\mathbf{x}_i - \mathbf{x}_j)\mathbf{M}(\mathbf{x}_i - \mathbf{x}_j)^T. \tag{3}$$

A full rank p.s.d. matrix $\mathbf{M}$ has too many parameters to fit tractably given the size of the dataset. For example, using features with dimension $F = 2^{12}$, learning a transform as in (eq. 3) requires us to fit approximately 8 million parameters; not only would this be prone to overfitting, it is simply not practical for existing solvers.

To address these issues, and given the fact that $\mathbf{M}$ parameterises a Mahanalobis distance, we approximate $\mathbf{M}$ such that $\mathbf{M} \simeq \mathbf{Y}\mathbf{Y}^T$ where $\mathbf{Y}$ is a matrix of dimension $F \times K$. We therefore define

$$\begin{aligned} d_{\mathbf{Y}}(\mathbf{x}_i,\mathbf{x}_j) &= (\mathbf{x}_i - \mathbf{x}_j)\mathbf{Y}\mathbf{Y}^T(\mathbf{x}_i - \mathbf{x}_j)^T \\ &= \|(\mathbf{x}_i - \mathbf{x}_j)\mathbf{Y}\|_2^2. \end{aligned} \tag{4}$$

Note that all distances (as well as their derivatives) can be computed in $O(FK)$, which is significant for the scalability of the method. Similar ideas appear in [4, 35], which also consider the problem of metric learning via low-rank embeddings, albeit using a different objective than the one we consider here.

2.1 Style space

In addition to being computationally useful, the low-rank transform in (eq. 4) has a convenient interpretation. Specifically, if we consider the K-dimensional vector $\mathbf{s}_i = \mathbf{x}_i\mathbf{Y}$, then (eq. 4) can be rewritten as

$$d_{\mathbf{Y}}(\mathbf{x}_i,\mathbf{x}_j) = \|\mathbf{s}_i - \mathbf{s}_j\|_2^2. \tag{5}$$

In other words, (eq. 4) yields a low-dimensional embedding of the features $\mathbf{x}_i$ and $\mathbf{x}_j$. We refer to this low-dimensional representation as the product's embedding into 'style-space', in the hope that we might identify $\mathbf{Y}$ such that related objects fall close to each other despite being visually dissimilar. The notion of 'style' is learned automatically by training the model pairs of objects which Amazon considers to be related.

2.2 Personalizing styles to individual users

So far we have developed a model to learn a *global* notion of which products go together, by learning a notion of 'style' such that related products should have similar styles. As an addition to this model we can personalize this notion by learning for each individual user which dimensions of style they consider to be important.

To do so, we shall learn personalized distance functions $d_{\mathbf{Y},u}(\mathbf{x}_i,\mathbf{x}_j)$ that measure the distance between the items i and j *according to the user* u. We choose the distance function

$$d_{\mathbf{Y},u}(\mathbf{x}_i,\mathbf{x}_j) = (\mathbf{x}_i - \mathbf{x}_j)\mathbf{Y}\mathbf{D}^{(u)}\mathbf{Y}^T(\mathbf{x}_i - \mathbf{x}_j)^T \tag{6}$$

where $\mathbf{D}^{(u)}$ is a $K \times K$ diagonal (positive semidefinite) matrix. In this way the entry $\mathbf{D}^{(u)}_{kk}$ indicates the extent to which the user u 'cares about' the k^{th} style dimension.

In practice we fit a $U \times K$ matrix $\mathbf{X}$ such that $D^{(u)}_{kk} = \mathbf{X}_{uk}$. Much like the simplification in (eq. 5), the distance $d_{\mathbf{Y},u}(\mathbf{x}_i,\mathbf{x}_j)$ can be conveniently written as

$$d_{\mathbf{Y},u}(\mathbf{x}_i,\mathbf{x}_j) = \|(\mathbf{s}_i - \mathbf{s}_j) \circ X_u\|_2^2. \tag{7}$$

In other words, X_u is a personalized *weighting* of the projected style-space dimensions.

The construction in (eq. 6 and 7) only makes sense if there are *users* associated with each edge in our dataset, which is not true of the four graph types we have presented so far. Thus to study the issue of user personalization we make use of our rating and review data (see Table 1). From this we sample a dataset of triples (i,j,u) of products i and j that were both purchased by user u (i.e., u reviewed them both). We describe this further when we outline our experimental protocol in Section 4.1.

2.3 Features

Features are calculated from the original images using the Caffe deep learning framework [11]. In particular, we used a Caffe reference model[3] with 5 convolutional layers followed by 3 fully-connected layers, which has been pre-trained on 1.2 million ImageNet (ILSVRC2010) images. We use the output of FC7, the second fully-connected layer, which results in a feature vector of length $F = 4096$.

3. TRAINING

Since we have defined a probability associated with the presence (or absence) of each relationship, we can proceed by maximizing the likelihood of an observed relationship set $\mathcal{R}$. In order to do so we randomly select a negative set $\mathcal{Q} = \{r_{ij} | r_{ij} \notin \mathcal{R}\}$ such that $|\mathcal{Q}| = |\mathcal{R}|$ and optimize the log likelihood

$$\begin{aligned} l(\mathbf{Y},c|\mathcal{R},\mathcal{Q}) = &\sum_{r_{ij}\in\mathcal{R}} \log(\sigma_c(-d_{\mathbf{Y}}(\mathbf{x}_i,\mathbf{x}_j))) + \\ &\sum_{r_{ij}\in\mathcal{Q}} \log(1 - \sigma_c(-d_{\mathbf{Y}}(\mathbf{x}_i,\mathbf{x}_j))). \end{aligned} \tag{8}$$

Learning then proceeds by optimizing $l(\mathbf{Y},c|\mathcal{R},\mathcal{Q})$ over both $\mathbf{Y}$ and c which we achieve by gradient ascent. We use (hybrid) L-BFGS, a quasi-Newton method for non-linear optimization of problems with many variables [21]. Likelihood

[3] bvlc_reference_caffenet from caffe.berkeleyvision.org

(eq. 8) and derivative computations can be naïvely parallelized over all pairs $r_{ij} \in \mathcal{R} \cup \mathcal{Q}$. Training on our largest dataset (Amazon books) with a rank $K = 100$ transform required around one day on a 12 core machine.

4. EXPERIMENTS

We compare our model against the following baselines:

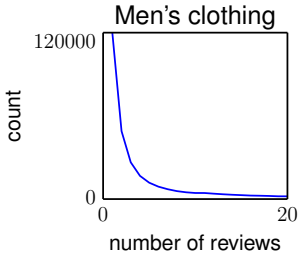
We compare against *Weighted Nearest Neighbor (WNN)* classification, as is described in Section 1.3. We also compare against a method we label *Category Tree (CT)*; CT is based on using Amazon's detailed category tree directly (which we have collected for Clothing data, and use for later experiments), which allows us to assess how effective an image-based classification approach *could* be, if it were perfect. We then compute a matrix of coocurrences between categories from the training data, and label two products (a,b) as 'related' if the category of b belongs to one of the top 50% of most commonly linked categories for products of category a.[4] Nearest neighbor results (calculated by optimizing a threshold on the ℓ_2 distance using the training data) were not significantly better than random, and have been suppressed for brevity.

Comparison against non-visual baselines As a non-visual comparison, we trained topic models on the reviews of each product (i.e., each document d_i is the set of reviews of the product i) and fit weighted nearest-neighbor classifiers of the form

$$d_{\mathbf{w}}(\theta_i, \theta_j) = \|\mathbf{w} \circ (\theta_i - \theta_j)\|_2^2, \tag{9}$$

where θ_i and θ_j are topic vectors derived from the reviews of the products i and j. In other words, we simply adapted our WNN baseline to make use of topic vectors rather than image features.[5] We used a 100-dimensional topic model trained using Vowpal Wabbit [8].

However, this baseline proved not to be competitive against the alternatives described above (e.g. only 60% accuracy on our largest dataset, 'Books'). One explanation may simply be that is is difficult to effectively train topic models at the 1M+ document scale; another explanation is simply that the vast majority of products have few reviews. Not surprisingly, the number of reviews per product follows a power-law, e.g. for Men's Clothing:



This issue is in fact exacerbated in our setting, as to predict a relationship between products we require both to have reliable feature representations, which will be true only if *both* products have several reviews.

Although we believe that predicting such relationships using text is a promising direction of future research (and one we are exploring), we simply wish to highlight the fact that there appears to be no 'silver bullet' to predict such relationships using text, primarily due to the 'cold start' issue that arises due to the long tail of obscure products with little text associated with them. Indeed, this is a strong argument in favor of building predictors based on visual features, since images are available even for brand new products which are yet to receive even a single review.

[4]We experimented with several variations on this theme, and this approach yielded the best performance.

[5]We tried the same approach at the word (rather than the topic) level, though this led to slightly worse results.

Category	method	substitutes: buy after viewing	substitutes: also viewed	complements: also bought	complements: bought together
Books	WNN	66.5%	62.8%	63.3%	65.4%
	$K = 10$	70.1%	68.6%	69.3%	68.1%
	$K = 100$	71.2%	69.8%	71.2%	68.6%
Cell Phones and Accessories	WNN	73.4%	66.4%	69.1%	79.3%
	$K = 10$	84.3%	78.9%	78.7%	83.1%
	$K = 100$	85.9%	83.1%	83.2%	87.7%
Clothing, Shoes, and Jewelry	WNN	·	77.2%	74.2%	78.3%
	$K = 10$	·	87.5%	84.7%	89.7%
	$K = 100$	·	88.8%	88.7%	92.5%
Digital Music	WNN	60.2%	56.7%	62.2%	53.3%
	$K = 10$	68.7%	60.9%	74.7%	56.0%
	$K = 100$	72.3%	63.8%	76.2%	59.0%
Electronics	WNN	76.5%	73.8%	67.6%	73.5%
	$K = 10$	83.6%	80.3%	77.8%	79.6%
	$K = 100$	86.4%	84.0%	82.6%	83.2%
Grocery and Gourmet Food	WNN	·	69.2%	70.7%	68.5%
	$K = 10$	·	77.8%	81.2%	79.6%
	$K = 100$	·	82.5%	85.2%	84.5%
Home and Kitchen	WNN	75.1%	68.3%	70.4%	76.6%
	$K = 10$	78.5%	80.5%	78.8%	79.3%
	$K = 100$	81.6%	83.8%	83.4%	83.2%
Movies and TV	WNN	66.8%	65.6%	61.6%	59.6%
	$K = 10$	71.9%	69.6%	72.8%	67.6%
	$K = 100$	72.3%	70.0%	77.3%	70.7%
Musical Instruments	WNN	79.0%	76.0%	75.0%	77.2%
	$K = 10$	84.7%	87.0%	85.3%	82.3%
	$K = 100$	89.5%	87.2%	84.4%	84.7%
Office Products	WNN	72.8%	75.0%	74.4%	73.7%
	$K = 10$	81.2%	84.0%	84.1%	78.6%
	$K = 100$	85.9%	87.2%	85.8%	80.9%
Toys and Games	WNN	67.0%	72.8%	71.7%	77.6%
	$K = 10$	75.8%	78.3%	78.4%	80.3%
	$K = 100$	77.1%	81.9%	82.4%	82.6%

Table 3: Accuracy of link prediction on top-level categories for each edge type with increasing model rank K. Random classification is 50% accurate across all experiments.

4.1 Experimental protocol

We split the dataset into its top-level categories (Books, Movies, Music, etc.) and further split the Clothing category into second-level categories (Men's, Women's, Boys, Girls, etc.). We focus on results from a few representative subcategories. Complete code for all experiments and all baselines is available online.[6]

For each category, we consider the subset of relationships

[6]http://cseweb.ucsd.edu/~jmcauley/

Category	method	substitutes also viewed	complements also bought	complements bought together
Baby	CT	77.1%	70.5%	80.1%
	WNN	83.0%	87.7%	81.7%
	$K = 10$	92.2%	92.7%	91.5%
	$K = 100$	94.6%	94.3%	93.3%
Boots	CT	75.0%	72.7%	74.2%
	WNN	83.9%	85.6%	84.7%
	$K = 10$	93.0%	94.9%	95.4%
	$K = 100$	94.6%	96.8%	96.4%
Boys	CT	81.9%	77.3%	83.1%
	WNN	85.0%	87.2%	87.9%
	$K = 10$	94.4%	94.1%	93.8%
	$K = 100$	96.5%	95.8%	95.1%
Girls	CT	83.0%	76.2%	78.7%
	WNN	83.3%	86.0%	84.8%
	$K = 10$	94.5%	93.6%	93.0%
	$K = 100$	96.1%	95.3%	94.5%
Jewelry	CT	50.1%	49.5%	51.1%
	WNN	81.2%	81.6%	75.8%
	$K = 10$	89.6%	89.3%	82.8%
	$K = 100$	89.1%	91.6%	86.4%
Men	CT	88.2%	78.4%	83.6%
	WNN	86.9%	78.4%	82.3%
	$K = 10$	91.6%	89.8%	92.1%
	$K = 100$	92.6%	93.3%	95.1%
Novelty Costumes	CT	79.1%	76.3%	81.5%
	WNN	80.1%	74.1%	76.0%
	$K = 10$	86.3%	86.6%	85.0%
	$K = 100$	89.2%	90.0%	89.1%
Shoes and Accessories	CT	81.3%	78.1%	90.4%
	WNN	75.4%	80.2%	77.9%
	$K = 10$	89.7%	90.4%	93.5%
	$K = 100$	92.3%	94.7%	96.2%
Women	CT	86.8%	79.1%	84.3%
	WNN	78.8%	76.1%	80.0%
	$K = 10$	88.9%	87.8%	91.5%
	$K = 100$	90.4%	91.2%	94.3%

Table 4: Accuracy of link prediction on subcategories of 'Clothing, Shoes, and Jewelry' with increasing rank K. Note that 'buy after viewing' links are not surfaced for clothing data on Amazon.

Figure 3: Examples of closely-clustered items in style space (Men's and Women's clothing 'also viewed' data).

from $\mathcal{R}$ that connect products within that category. After generating random samples of non-relationships, we separate $\mathcal{R}$ and $\mathcal{Q}$ into training, validation, and test sets (80/10/10%, up to a maximum of two million training relationships). Although we do not fit hyperparameters (and therefore do not make use of the validation set), we maintain this split in case it proves useful to those wishing to benchmark their algorithms on this data. While we did experiment with simple ℓ_2 regularizers, we found ourselves blessed with a sufficient overabundance of data that overfitting never presented an issue (i.e., the validation error was rarely significantly higher than the training error).

To be completely clear, our protocol consists of the following:

1. Each category and graph type forms a single experiment (e.g. predict 'bought together' relationships for Women's clothing).

2. Our goal is to distinguish relationships from non-relationships (i.e., link prediction). Relationships are identified when our predictor (eq. 1) outputs $P(r_{ij} \in \mathcal{R}) > 0.5$.

3. We consider *all* positive relationships and a random sample of non-relationships (i.e., 'distractors') of equal size. Thus the performance of a random classifier is 50% for all experiments.

4. All results are reported on the test set.

Results on a selection of top-level categories are shown in Table 3, with further results for clothing data shown in Table 4. Recall when interpreting these results that the learned model has reference to the object images only. It is thus estimating the existence of a specified form of relationship purely on the basis of appearance.

In every case the proposed method outperforms both the category-based method and weighted nearest neighbor, and

Figure 4: A selection of widely separated members of a single K-means cluster, demonstrating an apparent stylistic coherence.

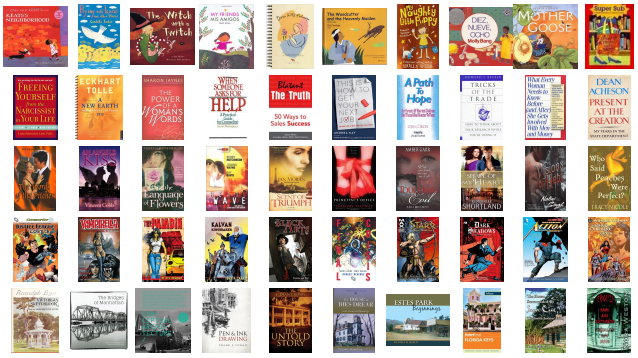


Figure 5: Examples of K-means clusters in style space (Books 'also viewed' and 'also bought' data). Although 'styles' for categories like books are not so readily interpretable as they are for clothes, visual features are nevertheless able to uncover meaningful distinctions between different product categories, e.g. the first four rows above above appear to be children's books, self-help books, romance novels, and graphic novels.

Figure 6: Navigating to distant products: each *column* shows a low-cost path between two objects such that adjacent products in the path are visually consistent, even when the end points are not.

Figure 7: A 2-dimensional embedding of a small sample of Boys clothing images ('also viewed' data).

the increase from $K = 10$ to $K = 100$ uniformly improves performance. Interestingly, the performance on compliments vs. substitutes is approximately the same. The extent to which the $K = 100$ results improve upon the WNN results may be seen as an indication of the degree to which visual similarity between images fails to capture a more complex human visual notion of which objects might be seen as being substitutes or compliments for each other. This distinction is smallest for 'Books' and greatest for 'Clothing Shoes and Jewelery' as might be expected.

We have no ground truth relating the true human visual preference for pairs of objects, of course, and thus evaluate above against our dataset. This has the disadvantage that the dataset contains all of the Amazon recommendations, rather than just those based on decisions made by humans on the basis of object appearance. This means that in addition to documenting the performance of the proposed method, the results may also be taken to indicate the extent to which visual factors impact upon the decisions of Amazon customers. The comparison across categories is particularly interesting. It is to be expected that appearance would be a significant factor in Clothing decisions, but it was not expected that they would be a factor in the purchase of Books. One possible interpretation of this effect might be that customers have preferences for particular genres of books and that individual genres have characteristic styles of covers.

4.2 Personalized recommendations

Finally we evaluate the ability of our model to personalize co-purchasing recommendations to individual users, that is we examine the effect of the user personalization term in (eqs. 6 and 7). Here we do not use the graphs from Tables 3 and 4, since those are 'population level' graphs which are not annotated in terms of the individual users who co-purchased and co-browsed each pair of products. Instead for this task we build a dataset of co-purchases from products that users have *reviewed*. That is, we build a dataset of tuples of the form (i,j,u) for pairs of products i and j that were purchased by user u. We train on users with at least 20 purchases, and randomly sample 50 co-purchases and 50 non-co-purchases from each user in order to build a balanced dataset. Results are shown in Table 5; here we see that the addition of a user personalization term yields a small but significant improvement when predicting co-purchases (similar results on other categories withheld for brevity).

5. VISUALIZING STYLE SPACE

Recall that each image is projected into 'style-space' by the transformation $\mathbf{s}_i = \mathbf{x}_i\mathbf{Y}$, and note that the fact that it is based on pairwise distances alone means that the embedding is invariant under isomorphism. That is, applying rotations, translations, or reflections to $\mathbf{s}_i$ and $\mathbf{s}_j$ will preserve their distance in (eq. 5). In light of these factors we perform k-means clustering on the K dimensional embedded coordinates of the data in order to visualize the effect of the embedding.

Figure 3 shows images whose projections are close to the centers of a set of selected representative clusters for Men's

Category	method	accuracy
Men's clothing	CT	84.8%
	WNN	84.3%
	$K = 10$, no personalization	90.9%
	$K = 10$, personalized	93.2%
Women's clothing	CT	80.5%
	WNN	80.8%
	$K = 10$, no personalization	87.6%
	$K = 10$, personalized	89.1%

Table 5: Performance of our model at predicting copurchases with a user personalization term (eqs. 6 and 7).

Figure 8: Outfits generated by our algorithm (Women's outfits at left; Men's outfits at right). The first column shows a 'query' item that is randomly selected from the product catalogue. The right three columns match the query item with a top, pants, shoes, and an accessory, (minus whichever category contains the query item).

and Women's clothing (using a model trained on the 'also viewed' graph with $K = 100$). Naturally items cluster around colors and shapes (e.g. shoes, t-shirts, tank tops, watches, jewelery), but more subtle characterizations exist as well. For instance, leather boots are separated from ugg (that is sheep skin) boots, despite the fact that the visual differences are subtle. This is presumably because these items are preferred by different sets of Amazon users. Watches cluster into different color profiles, face shapes, and digital versus analogue. Other clusters cross multiple categories, for instance, we find clusters of highly-colorful items, items containing love hearts, and items containing animals. Figure 4 shows a set of images which project to locations that span a cluster.

Although performance is admittedly not outstanding for a category such as books, it is somewhat surprising that an accuracy of even 70% can be achieved when predicting book co-purchases. Figure 5 visualizes a few examples of style-space clusters derived from Books data. Here it seems that there is at least some meaningful information in the cover of a book to predict which products might be purchased together—children's books, self-help books, romance novels, and comics (for example) all seem to have characteristic visual features which are identified by our model.

In Figure 6 we show how our model can be used to navigate between related items—here we randomly select two items that are unlikely to be co-browsed, and find a low cost path between them as measured by our learned distance measure. Subjectively, the model identifies visually smooth transitions between the source and the target items.

Figure 7 provides a visualization of the embedding of Boys clothing achieved by setting $K = 2$ (on co-browsing data). Sporting shoes drift smoothly toward slippers and sandals, and underwear drifts gradually toward shirts and coats.

6. GENERATING RECOMMENDATIONS

We here demonstrate that the proposed model can be used to generate recommendations that might be useful to a user of a web store. Given a query item (e.g. a product a user is currently browsing, or has just purchased), our goal is to recommend a selection of other items that might complement it. For example, if a user is browsing pants, we might want to recommend a shirt, shoes, or accessories that belong to the same style.

Here, Amazon's rich and detailed category hierarchy can help us. For categories such as women's or men's clothing, we might define an 'outfit' as a combination of pants, a top, shoes, and an accessory (we do this for the sake of demonstration, though far more complex combinations are possible—our category tree for clothing alone has hundreds of nodes). Then, given a query item our goal is simply to select items from each of these categories that are most likely to be connected based on their visual style.

Specifically, given a query item $\mathbf{x}_q$, for each category $\mathcal{C}$ (represented as a set of item indices), we generate recommendations according to

$$\underset{j \in \mathcal{C}}{\operatorname{argmax}} P_{\mathbf{Y}}(r_{qj} \in \mathcal{R}), \tag{10}$$

i.e., the minimum distance according to our measure (eq. 4) amongst objects belonging to the desired category. Examples of such recommendations are shown in Figures 1 and 8, with randomly chosen queries from women's and men's clothing. Generally speaking the model produces apparently reasonable recommendations, with clothes in each category usually being of a consistent style.

7. OUTFITS IN THE WILD

An alternate application of the model is to make assessments about outfits (or otherwise combinations of items) that we observe 'in the wild'. That is, to the extent that the tastes and preferences of Amazon customers reflect the zeitgeist of society at large, this can be seen as a measurement of whether a candidate outfit is well coordinated visually.

To assess this possibility, we have built two small datasets of real outfits, one consisting of twenty-five outfits worn by the hosts of *Top Gear* (Jeremy Clarkson, Richard Hammond, and James May), and another consisting of seventeen 'before' and 'after' pairs of outfits from participants on the television show *What Not to Wear* (US seasons 9 and 10). For each outfit, we cropped each clothing item from the image, and then used *Google*'s reverse image search to help identify images of similar clothing (examples are shown in

Figure 9: Least (top) and most (bottom) coordinated outfits from our Top Gear dataset. Richard Hammond's outfits typically have low coordination, James May's have high coordination, and Jeremy Clarkson straddles both ends of the coordination spectrum. Pairwise distances are normalized by the number of components in the outfit so that there is no bias towards outfits with fewer/more components.

Figure 9).

Next we rank outfits according to the average log-likelihood of their pairs of components being related using a model trained on Men's/Women's co-purchases (we take the average so that there is no bias toward outfits with more or fewer components). All outfits have at least two items.[7] Figure 9 shows the most and least coordinated outfits on *Top Gear*; here we find considerable separation between the level of coordination for each presenter; Richard Hammond is typically the least coordinated, James May the most, while Jeremy Clarkson wears a combination of highly coordinated and highly uncoordinated outfits.

A slightly more quantitative evaluation comes from the television show *What Not to Wear*: here participants receive an 'outfit makeover', hopefully meaning that their made-over outfit is more coordinated than the original. Examples of participants before and after their makeover, along with the change in log likelihood are shown in Figure 10. Indeed we find that made-over outfits have a higher log likelihood in 12 of the 17 cases we observed ($p \simeq 7\%$; log-likelihoods are normalized to correct any potential bias due to the number of components in the outfit). This is an important result, as it provides external (albeit small) validation of the learned model which is independent of our dataset.

8. CONCLUSION

We have shown that it is possible to model the human notion of what is visually related by investigation of a suitably large dataset, even where that information is somewhat tangentially contained therein. We have also demonstrated that the proposed method is capable of modeling a variety of visual relationships beyond simple visual similarity. Perhaps what distinguishes our method most is thus its ability to model what makes items *complementary*. To our knowledge this is the first attempt to model human preference for the appearance of one object given that of another in terms of more than just the visual similarity between the two. It is almost certainly the first time that it has been attempted directly and at this scale.

We also proposed visual and relational recommender systems as a potential problem of interest to the information retrieval community, and provided a large dataset for their training and evaluation. In the process we managed to figure out what not to wear, how to judge a book by its cover, and to show that James May is more fashionable than Richard Hammond.

[7] Our measure of coordination is thus undefined for a subject wearing only a single item, though in general such an outfit would be a poor fashion choice in the opinion of the authors.

References

[1] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *TKDD*, 2005.

[2] G. Carneiro, A. B. Chan, P. J. Moreno, and N. Vasconcelos. Supervised learning of semantic classes for image annotation and retrieval. *IEEE Trans. on PAMI*, 2007.

[3] W. Chu and S.-T. Park. Personalized recommendation on dynamic content using predictive bilinear models. In *WWW*, 2009.

[4] M. Der and L. Saul. Latent coincidence analysis: A hidden variable model for distance metric learning. In *NIPS*, 2012.

[5] W. Di, C. Wah, A. Bhardwaj, R. Piramuthu, and N. Sundaresan. Style finder: Fine-grained clothing style detection and retrieval. In *CVPR Workshops*, 2013.

[6] C. Doersch, S. Singh, A. Gupta, J. Sivic, and A. A. Efros. What makes paris look like paris? *SIGGRAPH*, 2012.

[7] J. Hays and A. A. Efros. Im2gps: estimating geographic

Figure 10: Contestants in *What Not to Wear*. Original outfits (top), 'made-over' outfits (bottom), and the change in log-likelihood (δ) between the components of the old and the new outfits (positive δ denotes an increase in coordination).

information from a single image. In *CVPR*, 2008.

[8] M. D. Hoffman, D. M. Blei, and F. Bach. Online learning for latent dirichlet allocation. In *In NIPS*, 2010.

[9] D. J. Hu, R. Hall, and J. Attenberg. Style in the long tail: Discovering unique interests with latent variable models in large scale social e-commerce. In *KDD*, 2014.

[10] V. Jagadeesh, R. Piramuthu, A. Bhardwaj, W. Di, and N. Sundaresan. Large scale visual recommendations from street fashion images. *arXiv:1401.1778*, 2014.

[11] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv:1408.5093*, 2014.

[12] X. Jin, J. Luo, J. Yu, G. Wang, D. Joshi, and J. Han. Reinforced similarity integration in image-rich information networks. *IEEE Trans. on KDE*, 2013.

[13] Y. Kalantidis, L. Kennedy, and L.-J. Li. Getting the look: Clothing recognition and segmentation for automatic product suggestions in everyday photos. In *ICMR*, 2013.

[14] M. H. Kiapour, K. Yamaguchi, A. C. Berg, and T. L. Berg. Hipster wars: Discovering elements of fashion styles. In *ECCV*. 2014.

[15] W. Kong, W.-J. Li, and M. Guo. Manhattan hashing for large-scale image retrieval. In *SIGIR*, 2012.

[16] Y. Koren and R. Bell. Advances in collaborative filtering. In *Recommender Systems Handbook*. Springer, 2011.

[17] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 2009.

[18] M. S. Lew, N. Sebe, C. Djeraba, and R. Jain. Content-based multimedia information retrieval: State of the art and challenges. *ACM TOMCCAP*, 2006.

[19] G. Linden, B. Smith, and J. York. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing*, 2003.

[20] S. Liu, J. Feng, Z. Song, T. Zhang, H. Lu, C. Xu, and S. Yan. Hi, magic closet, tell me what to wear! In *ACM Conference on Multimedia*, 2012.

[21] Y. Liu, W. Wang, B. Lévy, F. Sun, D.-M. Yan, L. Lu, and C. Yang. On centroidal Voronoi tessellation – energy smoothness and fast computation. *ACM Trans. on Graphics*, 2009.

[22] A. Makadia, V. Pavlovic, and S. Kumar. A new baseline for image annotation. In *ECCV*. 2008.

[23] A. Mas-Colell, M. Whinston, and J. Green. *Microeconomic Theory*. Oxford University Press, 1995.

[24] P. Melville, R. J. Mooney, and R. Nagarajan. Content-boosted collaborative filtering for improved recommendations. In *AAAI*, 2002.

[25] A. C. Murillo, I. S. Kwak, L. Bourdev, D. Kriegman, and S. Belongie. Urban tribes: Analyzing group photos from a social perspective. In *CVPR Workshops*, 2012.

[26] Y. Pan, T. Yao, T. Mei, L. Houqiang, C.-W. Ngo, and Y. Rui. Click-through-based cross-view learning for image search. In *SIGIR*, 2014.

[27] P. Peng, L. Shou, K. Chen, G. Chen, and S. Wu. The knowing camera 2: Recognizing and annotating places-of-interest in smartphone photos. In *SIGIR*, 2014.

[28] A. Schein, A. Popescul, L. Ungar, and D. Pennock. Methods and metrics for cold-start recommendations. In *SIGIR*, 2002.

[29] B.-S. Seah, S. Bhowmick, and A. Sun. Prism: Concept-preserving social image search results summarization. In *SIGIR*, 2014.

[30] A. Shrivastava, T. Malisiewicz, A. Gupta, and A. A. Efros. Data-driven visual similarity for cross-domain image matching. *ACM Trans. on Graphics*, 2011.

[31] M. Slaney, K. Weinberger, and W. White. Learning a metric for music similarity. In *International Conference of Music Information Retrieval*, 2008.

[32] A. W. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-based image retrieval at the end of the early years. *IEEE Trans. on PAMI*, 2000.

[33] D. Spina and J. Gonzalo. Learning similarity functions for topic detection in online reputation monitoring. In *SIGIR*, 2014.

[34] A. Torralba, R. Fergus, and W. T. Freeman. 80 million tiny images: A large data set for nonparametric object and scene recognition. *IEEE Trans. on PAMI*, 2008.

[35] L. Torresani and K. chih Lee. Large margin component analysis. In *NIPS*, 2007.

[36] M. Trevisiol, L. Chiarandini, L. Maria Aiello, and A. Jaimes. Image ranking based on user browsing behavior. In *SIGIR*, 2012.

[37] C.-C. Wu, T. Mei, W. Hsu, and Y. Rui. Learning to personalize trending image search suggestion. In *SIGIR*, 2014.

[38] H. Xia, P. Wu, S. Hoi, and R. Jin. Boosting multi-kernel locality-sensitive hashing for scalable image retrieval. In *SIGIR*, 2012.

[39] K. Xu, H. Li, H. Zhang, D. Cohen-Or, Y. Xiong, and Z.-Q. Cheng. Style-content separation by anisotropic part scales. *ACM Trans. on Graphics*, 2010.

[40] K. Yamaguchi, M. H. Kiapour, and T. L. Berg. Paper doll parsing: Retrieving similar styles to parse clothing items. In *ICCV*, 2013.

[41] K. Zhou, S.-H. Yang, and H. Zha. Functional matrix factorizations for cold-start recommendation. In *SIGIR*, 2011.