2016 IEEE 16th International Conference on Data Mining

Learning Compatibility Across Categories for Heterogeneous Item Recommendation

Ruining He, Charles Packer, Julian McAuley
Department of Computer Science and Engineering
University of California, San Diego
Email: {*r4he, cpacker, jmcauley*}*@cs.ucsd.edu*

***Abstract*—Identifying relationships between items is a key task of an online recommender system, in order to help users discover items that are functionally complementary or visually compatible. In domains like clothing recommendation, this task is particularly challenging since a successful system should be capable of handling a large corpus of items, a huge amount of relationships among them, as well as the high-dimensional and semantically complicated features involved. Furthermore, the human notion of “compatibility” to capture goes beyond mere similarity: For two items to be compatible—whether jeans and a t-shirt, or a laptop and a charger—they should be similar in some ways, but systematically different in others.**

In this paper we propose a novel method, *Monomer*, to learn complicated and heterogeneous relationships between items in product recommendation settings. Recently, scalable methods have been developed that address this task by learning similarity metrics on top of the content of the products involved. Here our method relaxes the *metricity* assumption inherent in previous work and models multiple localized notions of ‘relatedness,’ so as to uncover ways in which related items should be systematically similar, and systematically different. Quantitatively, we show that our system achieves state-of-the-art performance on large-scale compatibility prediction tasks, especially in cases where there is substantial heterogeneity between related items.

***Keywords*-Recommender Systems; Visual Compatibility; Metric Learning**

I. Introduction

Identifying and understanding relationships between items is a key component of any modern recommender system. Knowing which items are ‘similar,’ or which otherwise may be substitutable or complementary, is key to building systems that can understand a user’s context, recommend alternative items from the same style [10], or generate bundles of items that are compatible [14, 17, 28].

Typically, identifying these relationships means defining (or otherwise learning from training data) an appropriate distance or similarity measure between items. This is appropriate when the goal is to learn some notion of ‘equivalence’ between items, e.g. in order to recommend an item that may be a natural alternative to the one currently being considered. However, identifying such a similarity measure may be insufficient when there is substantial *heterogeneity* between the items being considered. For example, the characteristics that make clothing items, electronic components, or even romantic partners compatible exhibit substantial heterogeneity: for a pair of such items to be compatible they should be systematically similar in some ways, but systematically different in others.

Recently, a line of work has aimed to model such heterogeneous relationships, e.g. to model co-purchasing behavior between products based on their visual appearance or textual descriptions [16, 17, 26]. In spite of the substantial heterogeneity in the data used for training and the complexity of the models used, these works ultimately follow an established metric-learning paradigm: (1) Collect a large dataset of related (and unrelated) items; (2) Propose a parameterized similarity function; and (3) Train the parameterized function such that related items are more similar than non-related items. Such metric-learning approaches can be incredibly flexible and powerful, and have been used to identify similarities between items ranging from music [22] to members of the same tribe [7]. Such methods work to some extent even in the presence of heterogeneity, since they learn to ‘ignore’ dimensions where similarity should not be preserved. But we argue that ignoring such dimensions discards valuable information that ought to be used for prediction and recommendation.

In this paper, we propose a novel method, *Mixtures of Non-Metric Embeddings for Recommendation*, or *Monomer* for short, to identify relationships between items in product recommendation settings. In particular, we relax the metricity assumption present in recent work, by proposing more flexible notions of ‘relatedness’ while maintaining the same levels of speed and scalability. Specifically, we aim to overcome the following limitations of previous work: (1) The similarity measures learned by previous approaches inherently project categories as clusters into a metric space (albeit potentially via a complex embedding), since an item is ultimately more similar to those from the same category than others. This means that cross-category recommendations can only be made by exploiting an explicit category tree (e.g. ‘find the shoes nearest to these jeans’). Not only do such approaches require explicit category labels, but they are also subject to any noise or deficiencies in the category data. Our method can make cross-category recommendations without any dependence on the presence (or quality) of explicit category information. (2) Other assumptions made by metric learning approaches are also too strict for recommendation: an item is not necessarily compatible with itself (identity), nor are the types of relationships we want to learn

2374-8486/16 $31.00 © 2016 IEEE
DOI 10.1109/ICDM.2016.65

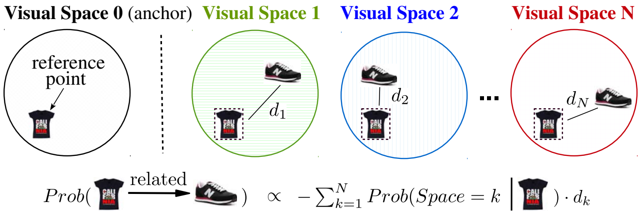

Figure 1: Illustration of the high-level ideas of *Monomer*. The query item (a t-shirt) is embedded into visual space 0 (the anchor space) whose position we superimpose into the other spaces. The potential match (a shoe) is embedded to N visual spaces and within each of them Euclidean distance between the pair is computed. Finally, the mixtures-of-experts framework is adopted to model the relative importance of the different components w.r.t. the given query.

necessarily symmetric (e.g. a spare battery is a good add-on item for a laptop, but not vice-versa). Other assumptions hidden in previous approaches (such as transitivity) may also be too strict, e.g. an iPhone is dissimilar from a Surface, though both are related to an iPad. Our approach is flexible enough to capture such complex and non-metric relationships. (3) Previous approaches learn a single 'global' notion of relatedness, neglecting any 'local' notions that could be equally important. In contrast, we capture multiple (and possibly competing) notions of 'relatedness' simultaneously. This is also key to generating *diverse sets* of recommendations. E.g. a shirt may be compatible with (a) a similar shirt from a different brand, (b) a similar shirt with a different color, (c) a complementary pair of pants, or (d) a complementary pair of shoes. By learning 'relatedness' as a *mixture* of multiple competing notions, our method can handle diverse sets of recommendations naturally.

We include experiments with textual features in the extended version of our paper [9]. Code and data are at https://sites.google.com/a/eng.ucsd.edu/ruining-he/.

II. Related Work

Item-to-Item Recommendation. Identifying relationships among items is a fundamental part of many real-world recommender systems. Such methods may be based on collaborative filtering, as in *Amazon's* own solution [14]. Latent-factor approaches aim to model user-item relationships in terms of low-dimensional factors, such that 'similar' items are those with close embeddings. See (e.g.) [1, 13] for surveys, and [14], [10], and [28] for specific systems that make use of *Amazon*, *Etsy*, and *Ebay* data.

Of more interest to us are systems that predict item-to-item relationships based on the *content* (e.g. images/text/metadata) of the items themselves. Various systems have been proposed to address specific settings, e.g. to identify relationships between members of 'urban tribes' [20], tweets [23], text [2, 4], or music [22]. Several methods have also been used to model visual data [3, 8, 12, 21, 27], though typically in settings where the metric assumption is well-founded (e.g. similar image retrieval).

Our work follows recent examples that aim to model co-purchase and co-browsing relationships, using a recently introduced dataset from *Amazon* [16, 17, 26], yet substantially relaxes the model assumptions to allow for more complex relationships than mere similarity between items.

Metric (and non-metric) Learning. Outside of the recommendation scenarios considered here, learning the features that describe relationships between objects is a vast topic. Typically, one is given some collection of relationships between items (i.e., a training set), and the goal is then to identify a (parameterized) function that can be tuned to fit these relationships, i.e., to assign observed relationships a higher likelihood or score than non-relationships. State-of-the-art methods identify hidden variables that describe relationships among items [7, 24], e.g. by factorizing the matrix of links between items [18]. Again, the main contributions we hope to make over such approaches are (1) to relax the assumption of metricity, and (2) to allow for multiple notions of 'relatedness' to compete and interact. While a few approaches have recently been proposed to learn non-metric relationships (e.g. [5]), we are unaware of any that allow for the scale of the data (thousands of features, millions of items and relationships) that we consider.

III. The Model

Formally, we are given a dataset $\mathcal{D}$ comprising a large corpus of 'items' and the pairwise relationships $\mathcal{R}$ between items from different subcategories, i.e., if $(x, y) \in \mathcal{R}$ then (1) item x and y are related, and (2) x and y are not from the same subcategory (e.g. a shirt and a matching pair of pants). Such cross-category recommendations highlight the ability of our model to generate recommendations between heterogeneous pairs of items, and matches the training instance selection approach from [26]. A high-dimensional (F-d) feature vector f_x associated with each item x is also provided (we mainly consider the feature vector extracted from item images with a neural network [17]). We seek a scalable method to model such relationships with a set of parameterized transform functions $d(x, y)$ such that related objects ($(x, y) \in \mathcal{R}$) are assigned higher probabilities than non-related ones ($(x, y) \notin \mathcal{R}$).

A. Preliminaries

Mahalanobis Transform. To model subtle notions like 'compatibility' from raw visual features, we need expressive transformations that are capable of modeling the relationships between pairs of items. We follow the approach from [17]: there, a *Mahalanobis Distance* is used to measure the distance (or 'dissimilarity') between items within the feature space. Let $\mathbf{M}$ denote the matrix that parameterizes the Mahalanobis Distance, then the distance between an item pair (x, y) is defined by $d_{\mathbf{M}}(x, y) = (f_x - f_y)^T \mathbf{M}(f_x - f_y)$, where f_x and f_y are the features vectors of x and y respectively. Although such an approach defines a distance function (and therefore suffers from the issues we are hoping to address), we use this method as a building block and ultimately relax its limitations.

Mixtures-of-Experts. Mixtures of experts (MoEs) are a classical machine learning method to aggregate the predictions of a set of 'weak' learners, known as experts [11]. What is particularly elegant about this approach is that it allows each learner to 'focus' on classifying instances about which it is relevant (i.e., expert), without being penalized for making misclassifications elsewhere.

For regression tasks such as the one we consider, each learner (denoted by l) outputs a prediction value $Pred_l(X)$ for the given input X. These predictions are then aggregated to generate the final prediction by associating weighted 'confidence' scores with each learner. Here we are interested in probabilistically modeling such confidences to be proportional to the expertise of the learners:

$$\underbrace{Pred(X)}_{\text{final prediction}} = \sum_l \overbrace{P(l|X)}^{\text{confidence in } l\text{'s expertise}} \cdot \underbrace{Pred_l(X)}_{l\text{'s prediction}}. \tag{1}$$

In our model, each 'expert' corresponds to a single notion of 'relatedness' between items. Thus, for a given pair of items that are potentially related, we can determine (a) which notions of relatedness are relevant for these items ($P(l|X)$); and (b) whether or not they are related according to that notion ($Pred_l(X)$). These two functions are learned jointly, such that the model automatically uncovers multiple notions of 'relatedness' simultaneously.

B. Model Specifics

1) Low-rank Mahalanobis Metric: Considering the high dimensionality of the visual features we are modeling (F = 4096), learning a full rank positive semi-definite Mahalanobis matrix $\mathbf{M}$ is neither computationally tractable for existing solvers nor practical given the size of the dataset.

Recently it was shown in [17] that a low-rank approximation of a Mahalanobis matrix works very well on visual datasets for the tasks considered in this paper. Specifically, the $F \times F$ Mahalanobis matrix is approximated by $\mathbf{M} \approx \mathbf{EE^T}$, where $\mathbf{E}$ is an $F \times K$ matrix and $K \ll F$. Then the distance between a pair (x, y) is calculated by

$$d_{\mathbf{E}}(x, y) = (f_x - f_y)^T \mathbf{E}\mathbf{E}^T (f_x - f_y) = ||\mathbf{E}^T f_x - \mathbf{E}^T f_y||_2^2. \tag{2}$$

This can be viewed as *embedding* the high-dimensional feature space (F-d) into a much lower-dimensional one (K-d) within which the Euclidean distance is measured. Note that the low rank property reduces the number of model parameters and increases the training efficiency significantly.

2) Multiple, Non-Metric Embeddings: There are two key limitations from using a low-rank Mahalanobis embedding approach like the one above. First, it can capture only a single set of dimensions (or the 'statistically dominant reason') that determines whether two given items are related or not, while there might be multiple relevant reasons. This drives us to use a group of embeddings, parameterized by N matrices $\mathbf{E}_1, \ldots, \mathbf{E}_N$ each with dimensionality $F \times K$ for the prediction task, with each capturing a different set of factors or 'reasons' that items may be related.

Another limitation of the single Mahalanobis embedding method, or more generally any metric-based method, is that it assumes that the closest neighbor of a given item is always itself, which is inappropriate for our task of placing many different categories of items close to the target. To overcome this shortcoming, we propose an anchor embedding (denoted by $\mathbf{E}_0$, again, with dimensionality $F \times K$) to learn the feature mappings in a non-metric manner.

In our model, $\mathbf{E}_0$ projects item x to a *reference point* $\mathbf{E}_0^T f_x$ in the corresponding space, referred to as the anchor space as it will be used as the basis for further comparisons. Next, embeddings $\mathbf{E}_k$ (for $k = 1, 2, \ldots, N$) map the potential match y and correspond to a particular notion of relatedness, such that $\mathbf{E}_0^T f_x$ will be close to $\mathbf{E}_k^T f_y$ (for some k) if x and y are related. That is, the predicted distance by the k-th learner is

$$d_k(x, y) = ||\overbrace{\mathbf{E}_0^T f_x}^{x\text{'s position in the anchor space}} - \underbrace{\mathbf{E}_k^T f_y}_{y\text{'s position in the k-th 'pseudo' space}}||_2^2. \tag{3}$$

For clarity, we call the N spaces defined by $\mathbf{E}_k$ ($k > 0$) 'pseudo' spaces as all distance calculations are still performed within the anchor space. The above definition supports learning directed relationships as the model is not required to be symmetric; but, it is flexible enough to learn symmetric (or even metric) embeddings if such structures are exhibited by the data.

3) Probabilistic Mixtures of Embeddings: Now we introduce how we aggregate the predictions from different embeddings. Given an item pair (x, y), we build our model upon the MoE framework to learn a probabilistic gating function to 'switch' among different embeddings. Considering our asymmetric setting where the query item x in the pair is used as the reference point, we model the probability that the k-th embedding is used for the given pair (x, y) with a softmax formulation:

$$P(k|\underbrace{(x, y)}_{\text{the given item pair}}) = \overbrace{P(k|x)}^{\text{only depends on } x} = \frac{\exp(\mathbf{U}_{:,k}^T f_x)}{\sum_i \exp(\mathbf{U}_{:,i}^T f_x)}, \tag{4}$$

where $\mathbf{U}$ is a new $F \times N$ parameter matrix with $\mathbf{U}_{:,k}$ being its k-th column. Briefly, the idea is to compute the probability distribution over the N learners given the characteristics of the 'pivot' item x. Note that our formulation is efficient as it only introduces a small number of parameters given that N is usually a small number (4 or 5 in our experiments).

Finally, our model calculates the 'distance' of an item pair (x, y) by the probabilistic expectation:

$$d(x, y) = \sum_{k}^{N} P(k|(x, y)) \cdot d_k(x, y). \tag{5}$$

Note that our 'distance' definition is a *non-metric* method as it only preserves the non-negativity and is relaxing the symmetry, identity, and triangle inequality properties.

C. Learning the Model

With the 'distance' function defined above, we model the probability that a pair is related by a shifted sigmoid function (in a way similar to [17]):

$$P((x, y) \in \mathcal{R}) = \sigma_c(-d(x, y)) = \frac{1}{1 + \exp(d(x, y) - c)}. \tag{6}$$

Next, we need to randomly select a negative set of relationships $\bar{\mathcal{R}}$. To this end, we use a procedure from [19] which randomly rewires the positive set in such a way that (1) the degree sequence of items is preserved and (2) each negative pair consists of items from two categories.

Then we proceed by fitting the parameters by maximizing the log-likelihood of the training corpus:

$$\begin{aligned} \widehat{\Theta} = \arg\max_{\Theta} \mathcal{L}(\mathcal{R}, \bar{\mathcal{R}}|\Theta) = \sum_{(x,y) \in \mathcal{R}} \log(P((x, y) \in \mathcal{R})) \\ + \sum_{(x,y) \in \bar{\mathcal{R}}} \log(1 - P((x, y) \in \mathcal{R})) + \Omega(\Theta), \end{aligned} \tag{7}$$

where Θ is the parameter set $\{\mathbf{E}_0, \mathbf{E}_1, \ldots, \mathbf{E}_N, \mathbf{U}, c\}$, and $\Omega(\Theta)$ is an $\mathcal{L}_2$-regularizer to avoid overfitting. Since N and K are *small* numbers (see Section IV), the log-likelihood as well as the derivatives can be computed efficiently.

Monomer is learned with L-BFGS [15], a quasi-Newton method for non-linear optimization of problems with a large number of variables. Log-likelihood and the full derivative computations can be naïvely parallelized over all training pairs $(x, y) \in \mathcal{R} \cup \bar{\mathcal{R}}$. This means the optimization can easily benefit from multi-threading and even parallelization across multiple machines (e.g. [6]). Note that *Monomer* and the single-embedding method share the same time complexity when using the same amount of embedding parameters.

IV. Experiments

In our experiments, we adopt the dataset from *Amazon* recently introduced by [17]. We focus on five large top-level categories under the category tree rooted with 'Clothing Shoes & Jewelry', i.e., Men's, Women's, Boys', Girls', and Baby's Clothing & Accessories. See [9] for dataset statistics. For each of the above categories, we experiment with two important types of relationships: 'users who bought *x* also bought *y*,' and 'users who viewed *x* also viewed *y*,' denoted as 'also_bought' and 'also_viewed' respectively for brevity. Such relationships are a key source of data to learn from in order to recommend items of potential interest to customers. Ground-truth for these relationships is also introduced in [17], and are originally derived from co-purchase and co-browsing data from *Amazon*.

Recall that our objective is to learn heterogeneous relationships so as to support cross-category recommendation. Across the entire dataset, such relationships are noisy, sparse, and not always meaningful. To address issues of noise and sparsity to some extent, it's sensible to focus on the relationships within the scope of a particular top-level category, e.g. Women's Clothing, etc. We then consider relationships between '2$^{\text{nd}}$-level' categories, e.g. women's shirts, women's shoes, etc.

Our evaluation protocol is as follows. A single experiment consists of a specific category (e.g. Men's Clothing) and a graph type (e.g. 'also_bought'). For each experiment, the relationships ($\mathcal{R}$) and a random sample of non-relationships ($\bar{\mathcal{R}}$, see Section III-C) are pairs of items connecting different subcategories of the category we are experimenting on. For each experiment, we use an 80/10/10 random split of the dataset ($\mathcal{R} \cup \bar{\mathcal{R}}$) with the training set being at most two million pairs. Our goal is then to predict the relationships and non-relationships correctly, i.e., link prediction. For all methods, the *validation* set is used for tuning the regularization hyperparameters, and finally the learned models are evaluated on the *test* set in terms of misclassification rate.

All experiments were performed on a single machine with 64GB memory and 8 cores. Our largest experiment required around 40 hours to train, though most took only a few hours.

A. Comparison Methods

Weighted Nearest Neighbor (WNN): This method uses a weighted Euclidean distance in the raw feature space to measure similarity between items: $d_w(x, y) = \|w \circ (f_x - f_y)\|_2^2$. Here $\circ$ is the Hadamard product and w is a weighting vector that is learned from the data.

Category Tree (CT): This method computes a matrix of co-occurrences between subcategories from the training data. Then a pair (x, y) is predicted to be positive if the subcategory of y is one of the top 50% most commonly connected subcategories to the subcategory of x.

Low-rank Mahalanobis Transform (LMT): LMT [17] is a state-of-the-art method for learning visual similarities among different items (possibly between categories) on large-scale datasets. LMT learns a *single* low-rank Mahalanobis embedding matrix to embed all items into a low-dimensional space. It predicts the links between a given pair based on the Euclidean distance within the embedded space (i.e., Eq. (2)).

Mixtures of Non-metric Embeddings (*Monomer*): Our method learns a mixture of low-rank embeddings to uncover groups of underlying reasons that explain the relationships

Table I: Test errors of the link prediction task using visual features (4096-d) on clothing categories of the *Amazon* dataset. The best performing method in each case is boldfaced. Lower is better.

Dataset	Graph	(a)	(b)	(c)	(d)	% impr.
		WNN	CT	LMT	*Monomer*	d vs. c
Men	*also_bought*	34.95%	47.71%	9.20%	**6.48%**	30%
	also_viewed	18.98%	47.40%	6.78%	**6.58%**	3%
Women	*also_bought*	30.50%	49.73%	11.52%	**7.87%**	32%
	also_viewed	20.50%	49.48%	7.90%	**7.34%**	7%
Boys	*also_bought*	31.16%	46.02%	8.80%	**5.71%**	35%
	also_viewed	21.52%	46.22%	6.72%	**5.35%**	20%
Girls	*also_bought*	31.10%	47.63%	8.33%	**5.78%**	31%
	also_viewed	22.36%	46.43%	6.46%	**5.62%**	13%
Baby	*also_bought*	37.26%	48.01%	12.48%	**7.94%**	36%
	also_viewed	30.89%	47.72%	11.88%	**9.25%**	22%
Avg.		27.92%	47.64%	9.00%	**6.79%**	22.9%

between items. It measures the 'distance' (or dissimilarity) between items in a non-metric manner (i.e., Eq. (5)).

Ultimately, our baselines are designed to demonstrate that (a) the raw feature space is not directly suitable for learning the notions of relationships (WNN); (b) using category metadata directly and not using other features (CT) results in relatively poor performance; and that (c) our proposed model is an improvement over the state-of-the-art method on our task (LMT).

B. Performance & Quantitative Analysis

Error rates on the test set for all experiments are reported in Table I. It has been shown by [17] that LMT can achieve better accuracy when using a reasonably large number of embedding dimensions (K). Therefore in all cases we choose K large enough such that LMT obtains the best possible (validation) performance. For fair comparison, in all cases we try to compare LMT and *Monomer* under the same total number of model parameters.

For experiments on 'also_bought' relationships, LMT uses $K = 100$ dimensions and *Monomer* uses $K = 20$ and $N = 4$. While for experiments on 'also_viewed' relationships, K is set to 50 for LMT and $K = 10$ and $N = 4$ for *Monomer*. We make a few observations to explain and understand our findings as follows: (1) WNN is particularly inaccurate for our task. We also observed relatively high training errors of this method for most experiments. This confirms our conjecture that raw similarity is inappropriate for our task, and that in order to learn the relationships across (sub)categories, some sort of expressive transforms are needed for manipulating the raw features. (2) The counting method (CT) performs considerably worse than other methods. This reveals that the predictive information used by the other models goes beyond the categories of the products, i.e., that the image-based models are learning relationships between finer-grained attributes. (3) *Monomer* outperforms LMT significantly for all experiments, especially for the harder task of predicting co-purchase dyads. In addition, all models perform better at predicting 'also_viewed' than 'also_bought' relationships. This is reasonable since intuitively items that are "also viewed" indeed tend to share more common characteristics compared to the "also bought" scenario. The greater heterogeneity between training pairs in the latter task makes it comparatively harder to address.

C. Visualization of the Embeddings

Next, we proceed by demonstrating the embeddings learned from our largest dataset, Women's Clothing, by *Monomer*. We take the same model trained on co-purchase relationships from the previous subsection and visualize it in Figure 2. We show each of the 5 visual spaces by a 2-d visualization with t-SNE [25]. Images are a random sample of size 50,000 from the Women's Clothing dataset and projected (using the learned embedding matrices) to each visual space to demonstrate the underlying structure.

As analyzed earlier, each embedding is capturing a specific notion of relatedness that explains the relationships of pairs of items in the corpus. In other words, it means that the nearest neighbors in each of the N 'pseudo' spaces should be related to the query according to the specific notion captured. Therefore those neighbors should be recommended as potential matches to the query item, as shown by the example in Figure 2. For the query image (a t-shirt) in this example, *Monomer* recommends bundles of similar t-shirts, pants, shoes, and accessories that resemble the query in terms of patterns (e.g. space 1), colors (e.g. space 2), and more generally 'styles' (e.g. spaces 3 and 4).[1] Such matching between a query image and nearby items in alternate spaces directly facilitates the task of recommending visually consistent outfits, where modeling and understanding the visual compatibility across categories is essential.

V. Conclusion

In this paper, we presented *Monomer*, a method to model heterogeneous relationships for item-to-item recommendation tasks. We noted that existing methods for item-to-item recommendation suffer from a few limitations when dealing with heterogeneous data, due mainly to their reliance on metricity or 'nearest-neighbor' type assumptions. To overcome these limitations, our method made use of 'mixtures' of non-metric embeddings, which allows us to relax the identity, symmetry, and triangle inequality assumptions of existing metric-based methods. The proposed scalable approach generates diverse and cross-category recommendations effectively that capture more complex relationships than mere visual similarity. Quantitatively, *Monomer* achieves state-of-the-art results at link prediction tasks using co-purchase and co-browsing dyads from *Amazon*.

[1] The second patch actually contains a few men's clothing items due to data deficiency—an intrinsic problem suffered by *Amazon*.

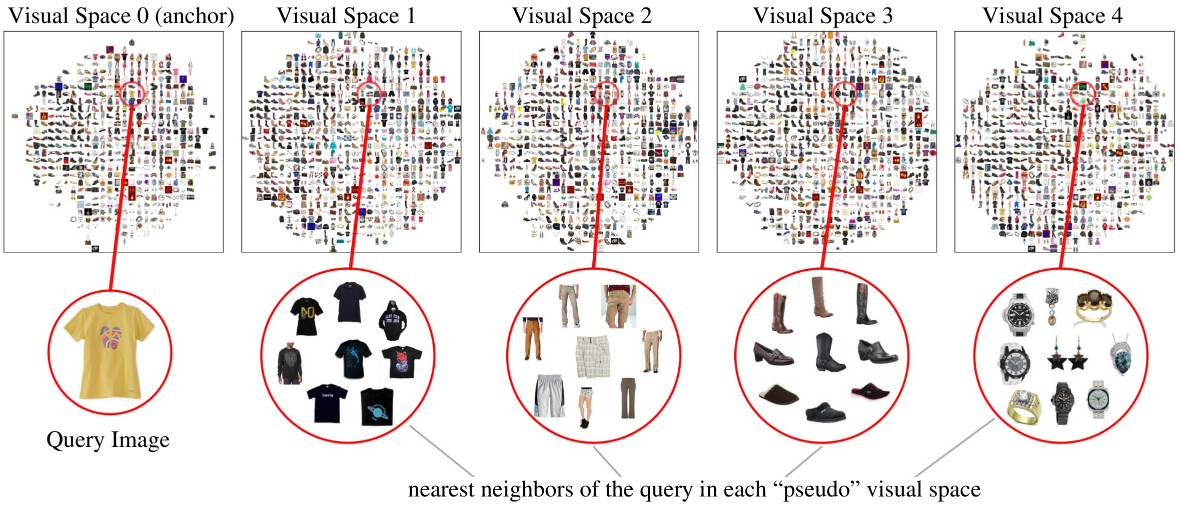

Figure 2: Visualization of *Monomer* trained on Women's Clothing for 'also_bought' prediction. According to our 'distance' function (i.e., Eq. (5)), *Monomer* recommends the nearest neighbors of the query within each visual space, based on the associated 'reasons' learned from data. Note that each visual space exhibits different category 'clusters' at the query image's location, allowing us to recommend *diverse* sets of items from the most closely-related categories.

Acknowledgments. This work is supported by NSF-IIS-1636879, and donations from Adobe, Symantec, and NVIDIA.

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] R. Balasubramanyan and W. Cohen. Block-LDA: Jointly modeling entity-annotated text and entity-entity links. In *SDM*, 2011.

[3] 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.

[4] J. Chang and D. Blei. Relational topic models for document networks. In *AISTATS*, 2009.

[5] S. Changpinyo, K. Liu, and F. Sha. Similarity component analysis. In *NIPS*, 2013.

[6] W. Chen, Z. Wang, and J. Zhou. Large-scale l-bfgs using mapreduce. In *NIPS*, 2014.

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

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

[9] R. He, C. Packer, and J. McAuley. Learning compatibility across categories for heterogeneous item recommendation. *arXiv:1603.09473*, 2016.

[10] 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.

[11] R. Jacobs, M. Jordan, S. Nowlan, and G. Hinton. Adaptive mixtures of local experts. *Neural computation*, 1991.

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

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

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

[15] 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 Transactions on Graphics*, 2009.

[16] J. McAuley, R. Pandey, and J. Leskovec. Inferring networks of substitutable and complementary products. In *KDD*, 2015.

[17] J. McAuley, C. Targett, Q. Shi, and A. van den Hengel. Image-based recommendations on styles and substitutes. In *SIGIR*, 2015.

[18] A. K. Menon and C. Elkan. Link prediction via matrix factorization. In *ECML*, 2011.

[19] R. Milo, N. Kashtan, S. Itzkovitz, M. E. J. Newman, and U. Alon. On the uniform generation of random graphs with prescribed degree sequences. *Arxiv:0312028*, 2003.

[20] 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.

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

[22] M. Slaney, K. Weinberger, and W. White. Learning a metric for music similarity. In *ICMIR*, 2008.

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

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

[25] L. van der Maaten. Accelerating t-SNE using tree-based algorithms. *JMLR*, 2014.

[26] A. Veit, B. Kovacs, S. Bell, J. McAuley, K. Bala, and S. Belongie. Learning visual clothing style with heterogeneous dyadic co-occurrences. In *ICCV*, 2015.

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

[28] J. Zheng, X. Wu, J. Niu, and A. Bolivar. Substitutes or complements: another step forward in recommendations. In *EC*, 2009.