CSE255 Assignment 1
Improved image-based recommendations for what not to wear dataset

Prabhav Agrawal and Soham Shah

23 February 2015

1 Introduction

We are interested in modeling the human perception of similarity. To model this human notion of what objects go well with each other, we can not rely only on image similarity. Image features have been used to capture relationship between objects that represent this human notion on the "What Not to Wear" dataset [3].

We extend the cited model in two fundamentally different ways. First, we include different distance functions between image features to capture relationship between images of products that are visually different but related to each other (e.x. white shirt and blue jeans). Secondly, we exploit the category information present as meta-data to find relationships between products (e.x.product-pair belonging to categories *boys* $->$ *badminton shoes* $->$ *Nike* and *boys* $->$ *badminton shoes* $->$ *Adidas* tend to be related). Category information in this dataset is hierarchical but our model only considers number of matching product categories between a product pair.

In the next subsections, we describe the dataset, predictive tasks and relevant literature. Section 2 describes the features and the proposed models. In Section 3, we report the results and evaluate the performance of each model. Section 4 describes a post-model exploratory data analysis to confirm our intuition in designing the models. We finally discuss the further extensions and analyses of proposed models in Section 5.

1.1 Dataset

We use the What Not to Wear (WNW) dataset which contains over 180 million relationships among 6 million objects from the Amazon web store. The original data was obtained by visiting the web store and recording the product recommendations given by Amazon. Amazon's tech reports mentions that recommendations are made based on the cosine similarity of the set of users that bought/viewed the product. An important point to note is that the data does not exactly represent user's preferences for pairs of products, but rather it represents Amazon's estimate of what products are "similar". Therefore, we use it to find out what information it tells about images of related products so that we can model the human notion of visual relationship.

An image and a set of hierarchical category labels are associated with each product. Further, a non-complete set of meta-data such as reviews, brand, and price information is also included. For every pair of products (X,Y), there are 4 types of directed relationships recorded in the dataset.

1. users who viewed X and also viewed Y - "also-viewed"
2. users who viewed X and eventually bought Y - "buy after viewing"
3. users who bought X and also bought Y - "also-bought"

4. users who bought X and Y simultaneously - "bought-together"

The relationships as defined in [3] describe two specific notion or substitute goods and complement goods. Substitute products are those that can be interchanged (ex: two different pairs of jeans) whereas complement goods generally go well together (ex: an iPhone and an iPhone scratch guard). "also viewed" and "buy after viewing" relationships tend to represent substitute goods, whereas "also-bought" and "bought-together" tend to represent complement goods.

The category information associated with a product is a path from the top level categories which is broad to the lower level specific categories.

- Shoes
  - Sports Shoes
    - Badminton Shoes

Each product can contain multiple category paths associated with it, each representing a hierarchy as shown above. We construct a category tree to represent the category information. For our experiments, we only consider products in "boys" category and relationships as "also-viewed" and "also-bought" graphs due to limited computation power and time.

1.2 Predictive Tasks

Our aim is to model visual human perception using images of objects along with the category information. The prediction task is predicting whether two products are related or not. It falls under the broad category of link-prediction in graphs.

The developed system can be subsequently used to recommend products based on the current product that the user is browsing. There are existing recommendation systems which make similar prediction based on meta-data or on visual information. Our model, as described in Section 2 aims to improve the model of *Julian et al.* [3], by modifying the distance function for a pair of products. Therefore, we use it as a baseline model for evaluating performance of our model. We divide the dataset into train, validation and test set and evaluate performance based on link prediction for products-pairs in the test set. Table 2 compares performance of baseline MA model from [3], with our models on the Boys dataset.

1.3 Relevant Literature

Content based recommendation systems typically use meta-data from users previous activities to make predictions. Collaborative recommendation systems match user to profiles of users with common interests and makes predictions based on profile information. Often a mix of the two are used to over the cold start problem where initially no data is available to make any predictions. Our models use visual information and category meta-data which is available even for new products and thus addresses the cold start problem.

Netflix prize was a competition aimed to make content-based video predictions, but the major difference was there was no image analysis taking place. Yamaguchi et al. [2] capture a notion of visual style when parsing clothes, but do so by retrieving visually similar items from a database.

Julian et al. [3] tries to "generalise the idea of a visual distance measure beyond measuring only similarity". The novelty here lies in the data, the quantity being modeled and how they model it from the data. The authors model human visual preferences rather than modelling visual similarity between objects. The MA model in Section 2 describes their model.

We extend this idea with our MMA model (in Section 2), to capture relationships between objects that are visually different (e.x. white shirt and black pants are visually different but often co-purchased together.). We include the "category" meta-data in our models and compare its performance with baseline MA model. We then design models which combine both types of information (images and category data) and then evaluate their performance.

2 Model

In this section, we present a description of the features and the model that we use.

2.1 Feature Description

We use two types of features for our model.

- **Image features**: We use the Caffe deep learning framework [1] to calculate features from the original images. In particular, we used a Caffe model with 3 fully connected layers and 5 convolutional layers, which is trained on a dataset of 1.2 million ImageNet images. We obtain a feature vector of length $F = 4096$ from the output of second fully-connected layer.
- **Category features**: We have hierarchical information of all the categories associated with each product. We find all the unique categories present in the dataset (e.x. in Boys, also-viewed and then prune the categories based on the counts of products associated with each categories. We then try to use this information in form of two different features.
  - **Category Intersect (c_{ij})**: From exploratory data analysis, we found that related products tend to intersect in more categories as compared to unrelated products. c_{ij} denotes the number of categories in between object i and j intersect.
  - **Category Vector (C_i)**: C_i has same dimensions as the set of pruned categories. For an object i, it has $1s$ in indices where correspond to the categories present in i and $0s$ in the remaining indices.

2.2 Model Description

We define our notation in Table 1 Our aim is to create a model which represents human notion of visual

Notation	Description
x_i	feature vector for object image i
F	dimension of image feature vector x_i
r_{ij}	relationship between objects i and j
R	set of relationship between all objects
$d_\theta(x_i, x_j)$	parametrized distance between x_i and x_j
M	$F \times F$ Mahalanobis transform matrix
U	a $F \times K$ matrix for approximating M
V	a $F \times K$ matrix for approximating M
c_{ij}	Number of categories which intersect between object i and j
λ	Parameter learnt for c_{ij}
N_{cat}	Number of categories present after pruning
PC	Pruning cut-off in terms of product count for a category
C_i	Category vector for object i with dimension N_{cat}
γ	Parameter vector learnt for C_i
$\sigma_c(.)$	shifted sigmoid function with parameter c
$R*$	R plus random set of non-related objects
U, V, T	training, validation and test subsets of $R*$

Table 1: Notation description

relationship between a pair of objects. We design a model keeping in mind that the it needs to scale to the volume of data.

For each object, an $F-$ dimensional feature vector $x \in \Re^F$ is calculated using Caffe framework in Section 2.1. The category features C_i, C_j and c_{ij} and calculated using the category information present in the

data. In the dataset, we also have a set R of relationships where each $r_{ij} \in R$ denotes that objects i and j are related. We learn a parametrized distance function $d(x_i, x_j)$ such that feature vectors x_i, x_j for objects which are related ($r_{ij} \in R$) are assigned a lower distance than the ones which are unrelated ($r_{ij} \notin R$)][3]. Or in other words,

$$P(r_{ij} \in R) \propto -d(x_i, x_j, C_i, C_j, c_{ij}) \tag{1}$$

We use a shifted sigmoid function to map distance to probability as:

$$P(r_{ij} \in R) = \sigma_c(-d(x_i, x_j, C_i, C_j, c_{ij})) = \frac{1}{1 + exp(d(x_i, x_j, C_i, C_j, c_{ij}) - c)} \tag{2}$$

The intuition behind this is for two items i and j:

- If $d = c$, probability i and j are related $= 0.5$
- If $d > c$, probability i and j are related < 0.5
- If $d < c$, probability i and j are related > 0.5

The parameter c is also learnt by the model to maximize the log-likelihood. We will now discuss the distance function that we used in our analysis:

2.2.1 Distance Functions

- **Mahalanobis Approximation (MA)** [3]
  This distance function is proposed by *Julian et. al.* The authors argue that a Mahalanobis transform captures information about how different feature dimensions relate to each other and the distance function is defined as:

  $$d_M(x_i, x_j) = (x_i - x_j)M(x_i - x_j)^T \tag{3}$$

  A full-rank matrix M requires about a million parameters to fit, therefore a low rank approximation U of dimension $F \times K$ is used, such that $M \simeq UU^T$.

  $$\begin{aligned} d_U(x_i, x_j) &= (x_i - x_j)UU^T(x_i - x_j)^T \\ &= ||(x_i - x_j)U||_2^2 \end{aligned} \tag{4}$$

  For K=1, the above model becomes a weighted nearest neighbour (WNN) model with a distance function as $d_\theta(x_i, x_j) = \theta^T(x_i - x_j)$. We use this MA model for K=1 as our baseline model and compare its performance with different distance functions.

- **Modified Mahalanobis Approximation (MMA)**
  MA model is able to capture the visual similarity between products (e.x. a blue shirt of one brand related to blue shirt of other brand). But it might not be able to represent the relationship between products which are visually different (e.x. a white shirt matches with blue jeans). Based on this intuition, we modify the distance function as:

  $$d_{U,V}(x_i, x_j) = ||(x_i - x_j)U||_2^2 - ||(x_i - x_j)V||_2^2 \tag{5}$$

- **Single Category Parameter (SCP)**
  Based on exploratory analysis, the distance is learnt as a function of number of categories items in i and j that intersect. Only the pruned categories are considered. The distances is defined as:

  $$d_\lambda(c_{ij}) =_{ij} \tag{6}$$

- **Category Vector (CV)**
  Category vectors C_i and C_j are constructed for both items i and j. Only the pruned categories are considered for feature vector representation. The distance function is defined as:

  $$d_\gamma = \gamma^T(C_i - C_j) \tag{7}$$

We now propose models which combine both the image features and category features. The motivation behind this is that we should be able to obtain a better model by combining both types of information.

– **MA + SCP** The distance function for this model is defined as:

$$d_{U,\lambda}(x_i, x_j, c_{ij}) = ||(x_i - x_j)U||_2^2 + \lambda * c_{ij} \tag{8}$$

– **MA + CV** The distance function for this model is defined as:

$$d_{U,\gamma}(x_i, x_j, C_i, C_j) = ||(x_i - x_j)U||_2^2 + \gamma^T (C_i - C_j) \tag{9}$$

– **MMA + SCP** The distance function for this model is defined as:

$$d_{U,V,\gamma}(x_i, x_j, c_{ij}) = ||(x_i - x_j)U||_2^2 - ||(x_i - x_j)V||_2^2 + \gamma^T (C_i - C_j) \tag{10}$$

3 Results and Discussion

From the WNW dataset, we perform our experiments on "Boys" category with only 100K edges. We consider two types of relationship graphs - "also-viewed" and "also-bought". For the models involving matrix U, V as parameters, we have selected $K = 1$. This has been done to reduce the computation time. We report the link prediction accuracies of the models discussed for train, validation and test set in Table 2.

Model	**Also-bought (Compliments)**			**Also-viewed (Substitutes)**		
	Train	Validation	Test	Train	Validation	Test
MA	81.2%	73.2%	73.3%	85.6%	80.0%	79.4%
MMA	87.3%	77.2%	77.2%	89.2%	79.4%	79.3%
SCP	75.5%	76.5%	76.1%	80.9%	81.1%	81.5%
CV	55.8%	54.5%	53.4%	51.2%	50.2%	49.9%
MA + SCP	82.9%	78.4%	77.9%	90.1%	86.4%	86.7%
MA + CV	82.8%	75.5%	75.6%	86.5%	80.8%	80.1%
MMA + SCP	79.2%	74.6%	74.4%	87.2%	83.2%	83.6%

Table 2: Accuracies of link prediction on subcategory 'Boys' of 'Clothing' category for K=1

We will give the explanation for the results. We will talk about the performance on the test set when comparing models. We see that MMA model outperforms MA model on "also-bought" graph because it is able to capture the relationship between visually different products which are related to each other. We do not observe such improvement in case of "also-viewed" graph because related objects in "also-viewed" category are visually similar (substitutes).

For category based features, we have set the pruning count to be 0 i.e. N_{cat} before pruning and after pruning is equal. We selected prune-count value based on the best performance on the validation set (as shown in Figure 1). We observe that the performance decreases with increase in PC for both CV and SCP models. This is due to the loss of category information with increased pruning. We had expected the accuracy to increase with the pruning in the beginning because of removal of noisy/junk categories. We do not observe this experiments because both models are not affected by noise in category information.

The CV model is not able to perform as expected. This parameter γ vector might not be getting trained properly because of few categories with significant product counts (only 62 categories with product-count $>$ 1000 for also-bought dataset). We also tried to prune the categories based on product-counts, but we were not able to obtain any significant improvement in the performance because of useful category information getting lost (i.e categories which contain related product-pairs have low counts)

One interesting aspect to note that SCP model is one of the best models with just one parameter and

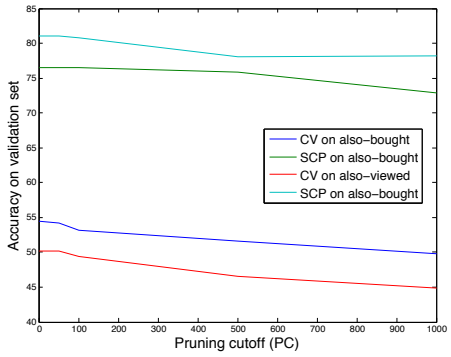

Figure 1: Pruning cutoff parameter (PC) vs accuracy on validation set

one feature. This supports our results from exploratory data analysis that related products intersect in larger number of categories than unrelated ones.

We also see that category based models perform better on "also-viewed" data rather than "also-bought" data. The products in "also-viewed" (substitutes) are more likely of being in the same category than the products in "also-bought" graph (compliments).

We use observe that MMA+SCP model performs worse than MA+SCP model. This can be due to the reason that related products which are visually different belong to different categories. In other words, minimizing distance w.r.t to image features leads to increase in distance w.r.t category features and vice versa.

Overall, we see that MA+SCP model performs the best among all the models. Also in terms of computation in training and prediction, there is not much overhead introduced on top of baseline model because we have added only one more parameter.

4 Post-Model Exploratory Data Analysis

We found that pairs of products corresponding to positive edges (related products) shared 6.7 product categories on an average where as the negative edges (unrelated products) shared 3.9 product categories on average. This led to design of SCP model, which learns a parameter based on the number of common categories.

Boys dataset has 54677 products and 1462 product categories, but only 62 categories had more than 1000 products belonging to them. Thus we tried varying the pruning cutoff(PC) among {0,50,100,500,1000} for the models based on category information.

We propose that MMA model should be able to model the visual difference in relationships as opposed to MA model. We consider the Euclidean distance between image features as a metric for visual similarity. Figure 2 shows how the accuracy of both models on test set for "also-bought" graph varies when we consider only the related objects which are atleast x distance apart. We conclude that for higher values of distance threshold (x), MMA model performs better because it is able to capture the relationship between visually different objects.

5 Future Scope

We will also try to run the experiments on higher K values and also for data belonging to other categories according to the computation resources available. We can try to model the category information in different ways. We should further exploit the hierarchical nature of category information associated with each product.

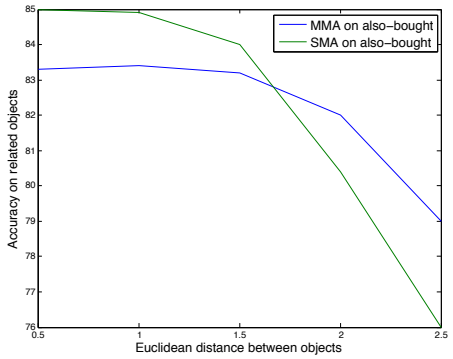

Figure 2: Accuracy on related objects which are x (value on X axis) distance apart

The distance function for category features can have parameters for number of categories intersecting at each level. One other way can be to include a correlation parameter for each category pair in the distance between two items i and j and we will have to learn a matrix of dimension $nCat \times nCat$. We can also try to exploit the review information present in the dataset. We can also build a feature representation of the review text using a library such as "word2vec" and then learn distance between two products as a function of the their text feature representations.

References

[1] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

[2] M. H. Kiapour K. Yamaguchi and T. L. Berg. Paper 940 doll parsing: Retrieving similar styles to parse cloth- 941 ing items. In *ICCV*, 2013.

[3] Julian Mcauley. What not to wear: Image-based recommendations on style and substitutes. In *Anonymous CVPR submission*, 2014.