Predicting Purchase Behavior using Visually Generated Product Gallery Networks

Andrew Zhai
andrewz@stanford.edu
Stanford University

ABSTRACT

Modern e-commerce recommendation systems recommend users products through purchase prediction off of historical purchase data. This signal however has limitations as new and long tail products have little to no such signal to exploit. One signal however that influences user purchase behavior, especially in verticals such as fashion, is visual.

In this paper we explore how visual similarity and object detection can together be used to predict fashion purchase behavior without using any purchase network based features. We formulate the problem as a network inference problem through creating a network consisting of product and gallery images with product-product edges, derived using visual similarity, and product-gallery edges, derived using a combination of object detection and visual similarity. We evaluate our approach through triplets sampled from the Amazon purchase relationship.

1. INTRODUCTION

Collaborative filtering approach based on co-purchase history (Amazon) or co-placement statistics (Pinterest) have shown great success in user conversion and engagement. Such systems however face difficulties such as 1) cold-start situations – for example, if a new product just entered into the database and 2) rich gets richer phenomenon – existing products in the recommendation system are shown more due to strengthening of links from user engagement, preventing relevant but new products from being shown. We conjecture however that such purchase behavior, especially for verticals such as fashion, can be modeled through using visual signals. In this work, we focus on the fashion vertical. In particular, we look to predict the purchase link relationships of Amazon fashion products using the Amazon Product dataset introduced by [7] [6].

When looking for signals to model purchase behavior, we need to have a signal that can model both *substitute* and *complementary* relationships as shown in Figure 1. For example, given a user has purchased a black leather backpack, he/she may want to purchase another bag like the current one to *substitute* the current bag. Also given that the user bought this backpack, he/she may be looking for shoes that complement the current bag. Intuitively, we can see that

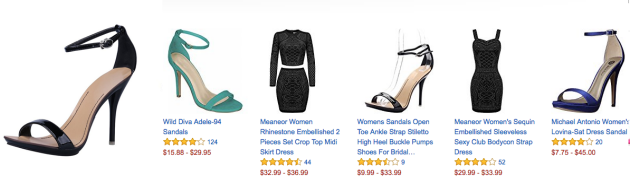

Figure 1: Amazon recommendation system contains both substitutes and complementary products.

visually similarity should be a strong motivating signal for product substitution purchases as visual similarity tries to find products that are very similar to the current product visually. Though complementary product purchase behavior does not have the same intuition, we believe that by combining object detection with visual similarity, we can model this complementary behavior.

To see how we can use visual signals to model complementary behavior, we define two types of images: **product** and **gallery**. We define product images as the images shown to users for purchasing such as those in Figure 1. We define gallery photos as images professionally created to illustrate how *multiple* products can be composed together to present an aesthetic expression of style such as those in Figure 2. We see that gallery photos naturally encode complementary relationships between product images. As such, by using object detection to find the objects within gallery images and connecting the gallery objects with the amazon images through visual similarity, we intuitively can model complementary relationships. One natural way of representing such complementary behavior is through a product-gallery network.

Though we can see how visual signals can be used to model *substitute* and *complementary* relationships, one problem with such signals is noise as the reliability of embedding modeling for visual similarity and object detection are still active areas of research. We however are motivated for this approach because of two reasons: 1) due to the recent wide adoption and improvement of deep learning methods, object detection and visual similarity have significantly improved in robustness in the recent years 2) due to the explosive growth and availability of online photos from sources such as Flickr, Google Images, and Pinterest, we can gather product and gallery photos at scale. With both better methods for visual signals and data at scale, we conjecture that the aggregate statistics will be reliable enough for us to do fashion purchase prediction reliably. If possible, these visual signals can be used not only to give engaging product recommendations in cold start scenarios, but more significantly, will allow *anyone* to create a large scale product purchase recommendation system without access to proprietary user

ACM ISBN 978-1-4503-2138-9.
DOI: 10.1145/1235

purchase data.

In Section 2, we describe related works to our current work. In Section 3, we describe the dataset we will use for our product and gallery images, the methods to extract visual features, and the evaluation dataset we use to measure how well we are at predicting fashion purchase behavior. In Section 4, we describe how we construct our product gallery network through the use of visual features and product and gallery images along with visualizations of our network to show complementary relationships being formed. In Section 5, we describe our evaluation and methods we use to do fashion link prediction with our visual network against the Amazon product relationships. In Section 6, we describe our results.

2. RELATED WORKS

The traditional setup of the link prediction problem of a certain network focuses on methods of utilizing the same network, but at a particular snapshot time t to predict the edges that will occur in the network at a future snapshot of time t' [5] [4]. There are other works however that relax this definition by removing the time component [1] [6]. In these works, the network is divided into train and test nodes where the training nodes and edges amongst these nodes are used as training data to learn parameters for some link prediction method and the test nodes and edges are used to evaluate the methods. Our work addresses the link prediction problem similar to the later.

Link Prediction with Network features: One of the most robust features to do link prediction is the network structure itself. Previous works [1] [6] [5] [4] present methods that utilize this network structure either by itself or with the additional of additional metadata to approach the link prediction behavior. Our method however differs as we explore only visual features and therefore features that are independent of the network structure.

Link Prediction with External features: Besides utilizing the network features, there have been previous work that focuses on external features [11] [7] [10]. In [11], Zhang et al. present an approach to purchase prediction on eBay using network information from other social networks such as Facebook. This approach however still relies on proprietary information such as the Facebook social network information which is not accessible to the general public at scale. Our work attempts to use public information (simply images) for purchase link prediction.

Link Prediction with Visual Features: Continuing along with the external features previous works, the area that is most similar to us are link prediction methods that rely solely on visual information. In [7] and [10], visual features are used for link prediction on the Amazon co-purchase data. For example, Veit et al. [10] proposed to learn style-compatible feature embedding from the Amazon co-purchase data and apply it to product recommendation. Though the problem is very similar to ours, our approach differ in that previous approaches rely heavily on expensive-to-obtain and often proprietary co-visitation statistics as supervised training data. The reason for this is that these previous approaches learn the complementary feature space directly from co-purchase triplets sampled from the Amazon co-purchase data. Our approach aligns more with semi-supervised methods as we use existing gallery images as sources of complementary data. Though we do rely on training triplets, we do so to tune the hyperparameters of the product gallery network creation and so much less data is required.

3. DATASETS

3.1 Amazon

Figure 2: We observed two categories of gallery photos – the first category is a professionally made model-shoot such as shown-room or runway images as shown on the left, and the second category contains scrap-book style image with products pieces together by fashion/design hobbyists as shown on the right. This work uses both.

In order to evaluate how well we can predict fashion purchase link relationships, we need a ground truth data source that encodes such information. As such we look towards the Amazon Product Data [7] [6] and specifically restrict ourselves within the "Clothing, Shoes and Jewelry" category to target the fashion vertical where we believe visual signals are significantly involved in the purchase behavior. In this dataset, we have data on 1,503,384 products where most products contain an image along with relationships to other products. This dataset contains four such relationships: "also bought", "also viewed", "bought together", and "buy after viewing". For our study, we restrict ourselves to the "also bought" and "bought together" relationships which best describe co-purchase behavior. We describe how we use this dataset as the *product* images to create the product-gallery network in Section 4 and the evaluation triplets in Section 6.

3.2 Pinterest

In order to get a collection of gallery images, we scrape Pinterest within the Men's Fashion and Women's Fashion categories and accept an image as a gallery image if our object detector detects at least 3 distinct object types within the image. The distinct objects type constraint ensures that we will obtain images with multiple distinct objects (shoes, bags, skirts, ...) instead of images with only a single object or multiple objects of the same type (multiple shoes) which would not encode complementary data, our primary motivation for utilizing the gallery images. From our scrape, we result with ~200K gallery images with a total of ~900K objects.

3.3 Visual Features

For our task, we involve two types of visual models. The first type of visual model is an image embedding model that takes an image and transforms it into an embedding space. In this embedding space, visual similarity can be computed through simple distance functions such as Euclidean Distance. The model we use for this is the VGG16 image classification model [9]. Though this is an image classification model, we take the FC6 intermediate features of this model which have been shown to work well as embeddings [2]. For efficiency, we binarize the FC6 features and use Hamming Distance for visual similarity as per [3].

The second type of visual model we use is object detection. Given an image, an object detector will return the objects within

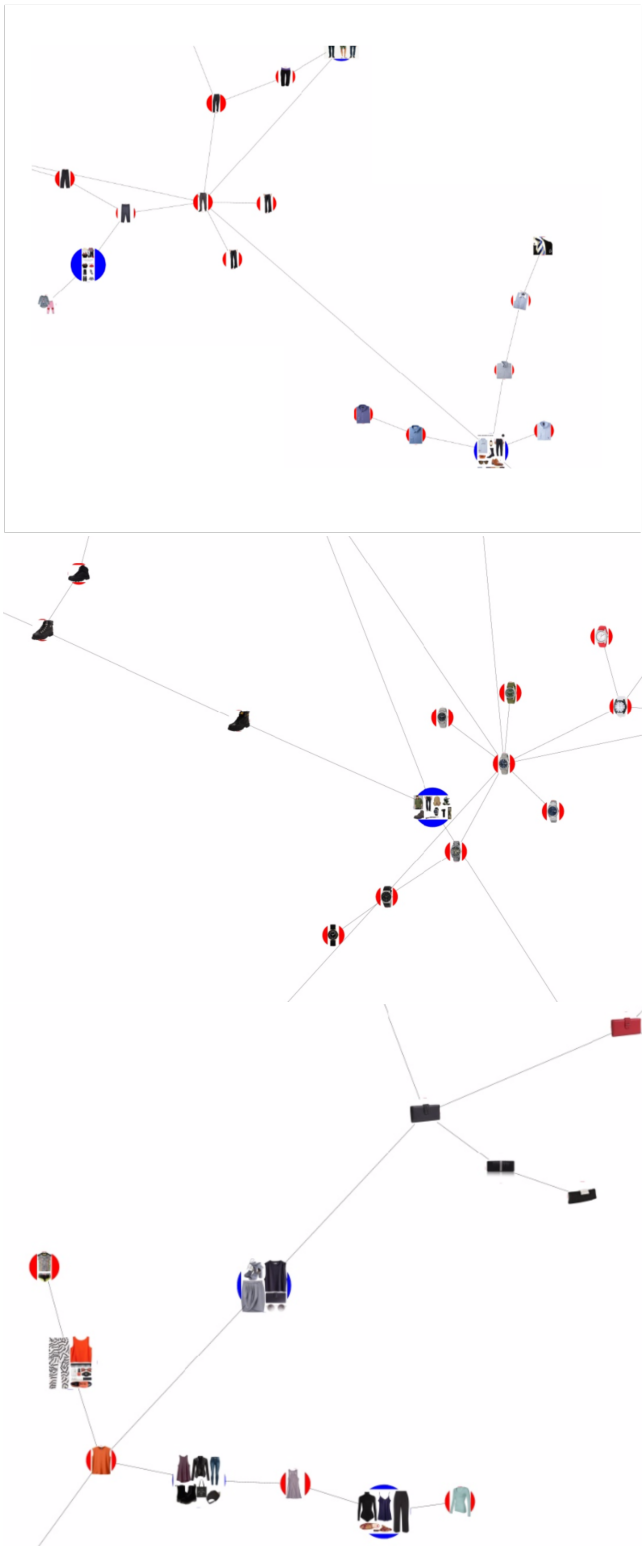

Figure 3: Minimum Spanning Tree visualization of our Product-Gallery + Product-Product network. Red circled images are the product images and blue circled images are the gallery images. For the top image, we see that the gallery image connects dress pants and shirt complements together. For the middle image, we see that the gallery image connects shoe and watch complements together. For the bottom image, we see that the gallery image connects handbags with women's tops.

the image where each object is defined by a bounding box, label, and score. Specifically we use the Faster-RCNN object detector [8] with a detection threshold of 0.7 to return high confident fashion objects.

4. NETWORK DEFINITION

This section describes how we use the Amazon product images and Pinterest gallery images to create our visually generated product gallery networks. Our networks consist of two types of nodes (product and gallery) along with two types of undirected edges (product-product and product-gallery). For our experiments, we maintain two networks, one for training and one for testing. The difference in the two networks is soley the product images in the network as the test network contains only the amazon product images that we are evaluating while the training network contains the rest of our considered amazon product images. We currently use the training network to tune hyperparameters for inference only as described in Section 5 however in the future, we plan on using it to also tune the network construction hyperparameters.

4.1 Nodes

When combined, we have a total of ~200K product nodes split into the training and test networks. Starting with the ~1.5 million Amazon products, we created an undirected network with these nodes and connected them through the real "also bought" and "bought together" (co-purchase) relationships in an undirected manner. Then, we ran an iterative algorithm to generate a 10-core network where we ensured that every node left contains at least 10 co-purchase edges, an attempt to reduce the noisy co-purchase relationships in our dataset. This results in the ~200K product nodes. For testing, we randomly sampled 10K product nodes while the rest are used in the training network. Our exact evaluation task is defined in Section 5.

Another type of node that we have in our visual network are gallery nodes. These are simply the 200K gallery images we scraped from Pinterest.

4.2 Edges

We generate two types of undirected edges for our visual network: product-product edges (intuitively a proxy for the substitute relationship) and product-gallery edges (intuitively a proxy for the complementary relationship). In both cases, we generate edges by running approximate K-nearest neighbors using a set of visual features of queries and a set of visual features as the database. In particular, we use hierarchical clustering trees with hamming distance on our binarized visual features which is shown to do well in [3].

To generate the product - product edges, for each of the product images in the network, we generate N nearest neighbors from a database consisting of the same set of product images. We remove self-edges from consideration to result in N unique product - product edges. For our initial investigation, we hardcoded $N = 20$. For the test network, this results in 200K edges

To generate the product - gallery edges, for each of the product images, we generated M nearest neighbors from a database consisting of 900K gallery objects. Note that for each object our detector outputs, we extract a visual embedding for the object by running our image embedding model on the crop of the image defined by the object bounding box. As such here, we are doing product to gallery object visual similarity. For our initial investigation, we hardcoded $M = 20$. For the test network, this results in 200K edges as well.

In future works, we plan on learning N and M by optimizing

for the training set of the real Amazon co-purchase links with our training network.

4.3 Visualization

We visualize a minimum spanning tree version of our test visual network in Figure 3 where red nodes are product images and blue are gallery images. The minimum spanning tree allows us to generate a planar view of our network which is very useful to reduct the noisy edges. In the visualization, we can see instances of the substitute and complementary behaviors through the visual edges. We can see from the visualization that the product-product edges cluster substitute looking products together such as the shoe, shirt, pants, and watch clusters. We also see that blue gallery nodes connect these substitute clusters together to form complementary relationships. For example, the dress shirts are connected to dress pants in the top example of Figure 3.

5. EVALUATION AND METHODS

5.1 Evaluation

From the random 10K sampled Amazon product nodes as described in the previous Section, we generate all possible triplets from the co-purchase ("also bought" and "bought together") relationships between the product nodes. The triplet sampling method we used is as follows: Given a co-purchase network formed with the 10K product nodes and the ground truth co-purchase edges between these 10K nodes, each positive pair in the triplet are direct neighbors in this network while each negative pair is an anchor image with a randomly sampled negative image guaranteed to not be the immediate neighbor of anchor. From this approach, we result in 12,292 triplets.

For evaluation, we measure how well our methods can be used to correctly predict the 12,292 triplet relationships. In particular, each of our methods defines a metric D such that smaller distances means more likely to purchase together. As such given a triplet (a, p, n) where a = anchor, p = positive, and n = negative, we correctly predict this triplet if

$$D(a, p) < D(a, n)$$

We report our results through *Precision* and *Recall*. Measuring Recall is important for methods where D cannot be applied to either (a, p) or (a, n). This can for example happen in shortest path based methods where nodes a and p are disconnected and nodes a and n are also disconnected. Recall essentially measures what fraction of triplets can the given method be applied on while Precision measures what fraction of applicable triplets do we correctly predict.

5.2 Methods

All our methods define a metric $D(a, b)$ such that smaller values means more likely to co-purchase. We explore the following methods:

5.2.1 Baseline

The baseline method sets $D(a, b)$ = Hamming distance of VGG16 binarized FC6 embeddings of a and b. This baseline disregards our visual network and instead directly uses the visual similarity to predict purchase relationships.

5.2.2 Shortest Path length

We set $D(a, b)$ to be the shortest path length between a and b in our visual network. Intuitively short paths in the visual network should mean two products are very related to each other in either the substitute or complementary relationships. As such, shorter paths should mean more likely to purchase.

A triplet (a, p, n) is not applicable for this method if no shortest paths exist for both (a, p) and (a, n).

Method	Precision	Recall
VGG16 only	0.872437	1
VGG16 Unweighted Edges Shortest Path	0.817767654	1
0 Degree Co-Occurrence	0.9612286	0.1615685
3 Degree Co-Occurrence	0.86877644	1
VGG16 + 3 Degree Co-Occurrence	**0.882443866**	1

Table 1: Fashion co-purchase link prediction results

5.2.3 0-Degree Co-Appearance

Given $G_0(x)$ = the set of gallery nodes that product node x is connected to, let us defined degree 0 co-appearance as the Jaccard similarity between sets $G_0(a)$ and $G_0(b)$:

$$Co_0(a, b) = \frac{G_0(a) \cap G_0(b)}{G_0(a) \cup G_0(b)}$$

In this method, we intuitively believe that two product nodes a and b are related to each other more if they appear together often in gallery nodes. This measurement is motivated by the success of co-appearance based methods in modern recommendation systems such as ones at Pinterest. Because Co_0 is a score where higher values means more related, we define $D(a, b) = -Co_0(a, b)$

A triplet (a, p, n) is not applicable for this method if both $Co_0(a, p)$ and $Co_0(a, n)$ are 0.

5.2.4 K-Degree Co-Appearance

We extend the previous method for more recall by considering the K degree neighbor product nodes of a given product node. Given product node a, 1-degree neighbors are defined as product nodes that are immediate product-product neighbors of a and 2-degree neighbors are defined as product nodes within two edge lengths away from product node a. We extend $G_0(x)$ such that $G_k(x)$ is defined as the set of gallery nodes that the set of product nodes at degree k are connected to. The relationship between G_k, Co_k and D are the same as previously mentioned.

5.2.5 Feature Combination

Here we combine D metrics of previous methods in a linear combination as follows:

$$D(a, b) = D_1(a, b) + \alpha D_2(a, b) + ...$$

The weight parameters are tuned on the training network with triplets generated from product nodes *not* used in evaluation.

5.2.6 Future

In the future, we plan on exploring more methods such as Personalized Pagerank which can scale better than shortest distance and co-appearance based methods. An intuition we have to improving results may be to add more data to our visual networks which may require faster methods.

6. RESULTS

We present our results in Table 1. With the baseline alone, we are able to achieve 87.24% on our triplets evaluation dataset. This

is interesting in that it shows that a large amount of co-purchase behavior in fashion can be explained by *substitute* purchase behavior modeled solely with visual similarity.

The "VGG16 Unweighted Edges Shortest Path" method described in table was evaluated by taking only the product-product edges of the test visual network and running the shortest path length method on the network. We ran this evaluation to get a sense for the information loss from both having a threshold on the number of visually similar neighbors allowed and encoding the visually similar neighbor weights as unweighted instead of using the visually similar hamming distance information. We can see that due to the two sources of information loss, this method performs worse than the baseline as the precision is only 81.78%.

The "0 Degree Co-occurrence" method is able to achieve a very high precision at 96.12% with the trade off however of recall. When considering only the set of gallery images directly connected to a given product node, we see that the resulting co-appearance statistics is sparse. This is not surprising as we used $10K$ amazon product nodes in our test set to connect to $900K$ gallery images. Motivated by the high precision of this method, we looked into the more general K-degree co-appearance method. We show the results for the 3-degree method as it was the smallest K such that the Recall was 1. We see that we are able to achieve a precision of 86.88%, much higher than the shortest path method based solely on the product-product network. We however are not able to beat our baseline with this result.

With the motivation that the "VGG16 only" method best encodes the "substitution" relationship and that the 3-Degree Co-Occurrence method should encode some "complementary" relationship, we looked to combine the two metrics through "Feature Combination". With tuning of our weights on the training network and training triplets dataset, we were able to get a precision of **88.22%** on our evaluation triplets dataset. This is promising as we are able to do better than the baseline which intuitively directly encodes "substitution" relationships.

We visualize some of our successes in Figure 4 and some of our failures in Figure 5. We see in our successful cases that not only are we able to correctly predict "substitute" relationships which involve items that look visually similar, we are also able to correctly predict "complementary" relationships. When looking at the failed examples however, we clearly see that even though some "complementary" relationships are successfully predicted, most of the failed examples are "complementary" relationship based triplets. One future work of ours is to revisit our network construction phase and rethink how we define edges and connections between product to gallery images. For example, instead of a fixed N or M for the number of nearest neighbors, instead base the number of connections on a threshold of the hamming distance between neighbors. Another area to explore is to make the product-gallery edges directed. Have the product to gallery edges be defined by the current product-gallery edges and derive the gallery to product edges by using gallery objects as queries against a database of product images. Our motivation for this is when viewing the node distribution of our test visual network, we see that some gallery nodes receive as many as 160 edges with product gallery nodes showing that few gallery objects map to many product images. This can hurt methods such as shortest path length and co-appearance as these popular gallery nodes connect many product images together and essentially can possibly act as noisy shortcuts. By making product gallery edges direct, we can limit the number of out going edges from gallery nodes and hopefully result in better shortest path and co-appearance based results.

Figure 4: Examples of successful triplet prediction. In the first triplet, we see a successful complementary prediction

Figure 5: Examples of failed triplet prediction. Majority of the failed examples are complements which shows that more work is needed to better encode complementary information

7. CONCLUSIONS

To be written in final version.

8. REFERENCES

[1] L. Backstrom and J. Leskovec. Supervised random walks: Predicting and recommending links in social networks. In *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining*, WSDM '11, pages 635–644, New York, NY, USA, 2011. ACM.

[2] 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 preprint arXiv:1408.5093*, 2014.

[3] Y. Jing, D. Liu, D. Kislyuk, A. Zhai, J. Xu, and J. Donahue. Visual search at pinterest. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining (SIGKDD)*.

[4] A. M. Kameshwar Chinta, Kevin Clark. Supervised link prediction in bipartite networks.

[5] D. Liben-Nowell and J. Kleinberg. The link-prediction problem for social networks. *J. Am. Soc. Inf. Sci. Technol.*, 58(7):1019–1031, May 2007.

[6] J. McAuley, R. Pandey, and J. Leskovec. Inferring networks of substitutable and complementary products. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794. ACM, 2015.

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

[8] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Neural Information Processing Systems (NIPS)*, 2015.

[9] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[10] A. Veit*, B. Kovacs*, S. Bell, J. McAuley, K. Bala, and S. Belongie. Learning visual clothing style with heterogeneous dyadic co-occurrences. In *International Conference on Computer Vision (ICCV)*, Santiago, Chile, 2015. *The first two authors contributed equally.

[11] Y. Zhang and M. Pennacchiotti. Predicting purchase behaviors from social media. In *Proceedings of the 22nd international conference on World Wide Web*, pages 1521–1532. ACM, 2013.