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 as we explore different variations of the visual network consisting of product and gallery images. In our experiments, we motivate and explore three types of edges: product-product edges, derived using visual similarity, product-gallery edges, consisting of whole product images connecting to gallery objects derived through object detection and visual similarity, and also gallery-gallery edges, connecting objects through visual similarity. We evaluate our approach through triplets sampled from the Amazon purchase relationship.

1. INTRODUCTION

Collaborative filtering approaches 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 [8] [7].

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 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.

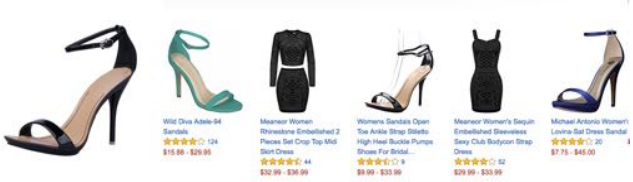

Figure 1: Amazon recommendation system contains both substitutes and complementary products.

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 en-

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

gaging 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 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] [7]. 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] [7] [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 [12] [8] [11]. In [12], 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 [8] and [11], visual features are used for link prediction on the Amazon co-purchase data. For example, Veit et al. [11] 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

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.

3.1 Amazon

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 [8] [7] 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 visual networks 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 2 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 in two datasets. The first dataset we explored consists of ~200K gallery images with a total of ~900K objects which we use to build the PP_PG network as shown in Figure 3. Later to scale our network, we collected ~1M gallery images with a total of ~3.5M objects to build the PG_GG network. These networks and their motivations are described in more detail in Section 4.

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

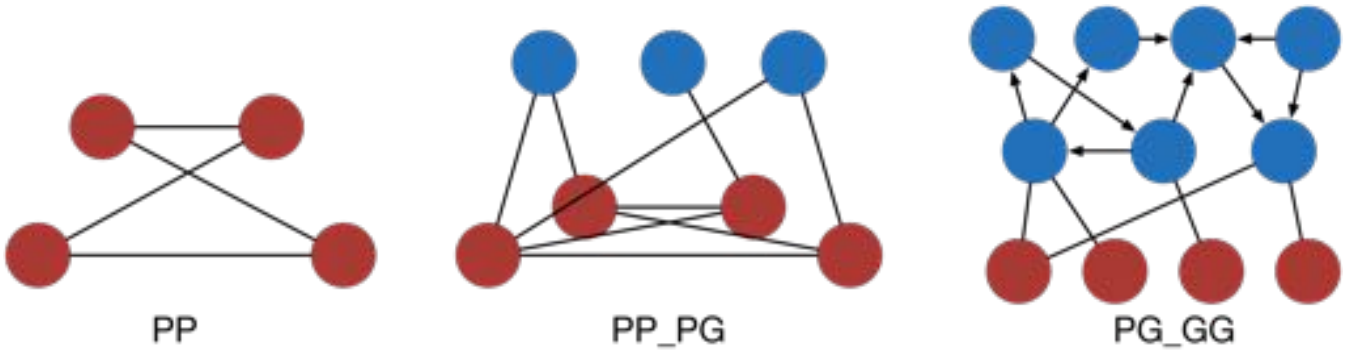

Figure 3: Visualization of the three types of visual networks we explore. Red nodes represent product images while blue nodes represent gallery images.

is the VGG16 image classification model [10]. 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 the image where each object is defined by a bounding box, label, and score. Specifically we use the Faster-RCNN object detector [9] 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. We explore three types of visual networks: Product-Product (PP), Product-Product + Product-Gallery (PP_PG), and Product-Gallery + Gallery-Gallery (PG_GG) as shown in Figure 3. For our experiments, we maintain two networks of each type, 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

4.1.1 Product

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.

4.1.2 Gallery

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

4.2 Edges

Overall we generate three types of edges for our visual network: product-product, product-gallery, and gallery-gallery. In all 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 investigation, we hardcoded $N = 20$.

To generate the product-gallery edges, for each of the product images, we generated M nearest neighbors from a database consisting of 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 investigation, we hardcoded $M = 20$.

Finally to generate the gallery-gallery edges, a similar process is used except the query to the database consisting of gallery objects is also a gallery object. We use all gallery objects as queries and extract L nearest neighbors per query. One additional constraint we place on the edges generated is that the object to object link must be of the same object type (the object detector will output a type per object). This strengthens the reliability of these edges, one motivation for using these types of edges. For our investigation, we hardcode $L = 1$ as method generates $L *$ (# of gallery images) edges.

In future works, we plan on learning N, M, and L by optimizing for the training set of the real Amazon co-purchase links with our training network.

4.3 Network Types

We have developed multiple network types motivated by iterations of our experiment results. These network types are shown in Figure 3.

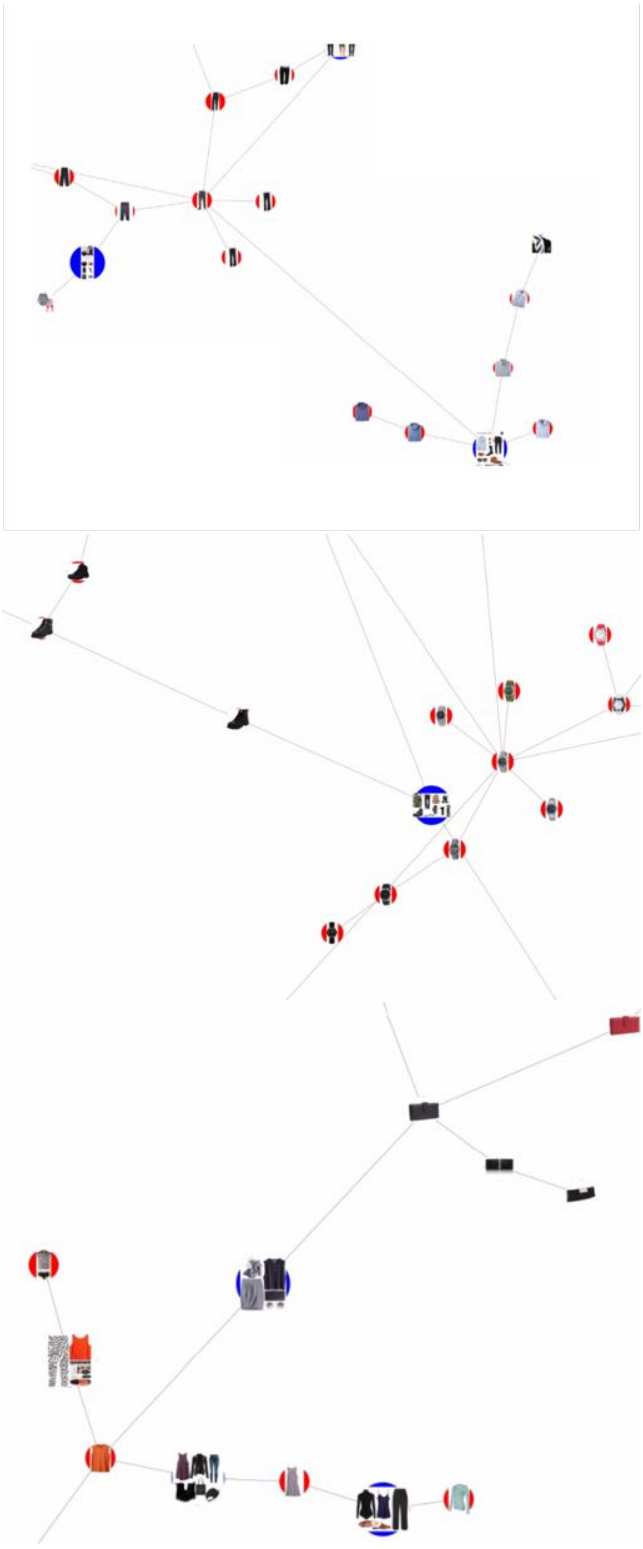

Figure 4: Minimum Spanning Tree visualization of our PP_PG network. Red circled images are the product images and blue circled images are the gallery images. For the top image, we see that the gallery image connects dress pants and shirt complements together. For the middle image, we see that the gallery image connects shoe and watch complements together. For the bottom image, we see that the gallery image connects handbags with women's tops.

4.3.1 Product-Product (PP) Network

This network consists only of undirected product-product edges to be used only for baseline purposes. Product-product links allow only for visual similarity between the amazon products, a proxy for the *substitute* relationship. As such we do not expect this network to perform well since no *complementary* relationships are modeled.

4.3.2 Product-Product + Product-Gallery (PP_PG) Network

This network is the initial version of our visual network that encodes both *substitute*, through product-product links, and *complementary*, through product-gallery links, relationships. This network uses the 900K objects version of the gallery dataset. We describe explicitly which experiments we run with this network in Section 6.

We visualize a minimum spanning tree version of our test PP_PG visual network in Figure 4 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 4.

4.3.3 Product-Gallery + Gallery-Gallery (PG_GG) Network

This network is motivated from qualitative evaluation of the experiment results from the PP_PG network . In particular, this network attempts to improve the robustness of the visually similar edges, an issue visualized in Figures 5 and 6, by introducing gallery-gallery edges and increasing the number of gallery objects used by using the larger gallery dataset we collected. One advantage of this network is that the *complementary* relationship is not encoded through a series of product to gallery to product to gallery and so on relationships which we see are noisy in Figure 6 partially due to the bottleneck of needing to match to 10K product images. Instead the relationship is encoded through gallery-gallery edges which are higher quality due to both the larger gallery object database and because we can also add the additional constraint that the edge objects must have the same object type.

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 (PP,PP_PG)

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).

5.2.3 0-Degree Co-Appearance (PP_PG)

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 (PP_PG)

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 Personalized Page Rank (PG_GG)

As the gallery-gallery edges in the PG_GG significantly increase the size of the network, the methods presented previously do not scale well. For example, the shortest path complexity through Dijkstra's algorithm is $O(|E|+|V|\log|V|)$ worst case and $|E|$ here is much larger. Though the co-appearance based methods do not depend on the number of gallery-gallery edges, the recall will suffer as we increase the number of gallery nodes.

A method that is independent to the size of the network is personalized pagerank which allows us to measure how important a node is to a query set of nodes. We use personalized pagerank as implemented in [6] in the following manner:

Given the PG_GG network and a pair of product nodes (a, b), $D(a, b) = -\sum_{g \in GP(b)} PPR(GP(a), g)$. Here $GP(a)$ returns the set of gallery nodes that product node a is connected to and PPR computes the personalized page rank value. We ensure that each product node is connected to an equal number of gallery nodes to be fair.

5.2.6 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.

Method	Precision	Recall
VGG16 only	0.87243	1
Shortest Path (PP)	0.81776	1
Shortest Path (PP_PG)	0.74544	1
0 Degree Co-Occurrence (PP_PG)	0.96122	0.16156
3 Degree Co-Occurrence (PP_PG)	0.86877	1
VGG16 + 3 Degree Co-Occurrence (PP_PG)	**0.88244**	1
Personalized Pagerank (PG_GG)	0.81938	0.01846

Table 1: Fashion co-purchase link prediction results

6. RESULTS

6.1 Baseline

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.

6.2 Shortest Path (PP,PP_PG)

We ran the shortest path method on the PP network 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%.

Running the shortest path method on the PP_PG method does even worse as the precision is only 74.54%. Given that PP vs PP_PG is the inclusion of the product-gallery edges, we suspect that the gallery images are acting as shortcuts more for the negative examples than for the positive. We visualize product-gallery edges in Figure 6 and see that product-gallery edges are fairly noisy which could be the reason for the drop in precision performance.

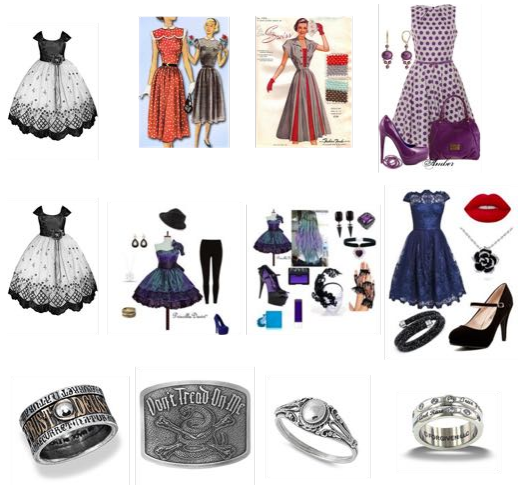

Figure 5: In the top two rows, we see that the quality of the product-gallery links are heavily dependent on the size of the gallery database. Given the product image on the left, we visualize the gallery images of the top-3 matching objects. The top image is generated using our 900K objects index. The bottom image is generated using a much larger index of 800M objects (we do not ever use this index since the scale is too large for us to handle currently). In the same format, we see in the third row an example of how the product-product links are noisy as well as the gothic ring query is matched to products that are either too feminine or not even a ring.

6.3 Co-Occurrence (PP_PG)

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.

6.4 Baseline + Co-Occurrence (PP_PG)

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 7 and some of our failures in Figure 8. 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.

6.5 Personalized Pagerank (PG_GG)

As mentioned in the previous sections, we developed the PG_GG network in response to the previous results in order to strengthen the visual connections by scaling up the number of gallery images and using gallery-gallery edges which we believe are more robust due to the same object type constraint and the larger gallery database index. We believe that the visual connections are an issue as the visualizations of Figures 5 and 6 clearly show that the visual connections lack the subtle style and semantic meaning that we believe is necessary to correctly predict triplets such as the second row triplets of Figure 8.

Due to the scalability issues of the co-occurrence and shortest path based methods, we look into using personalized page rank as implemented in [6]. As shown in Table 1, the personalized pagerank method resulted in a precision of 81.94% and a recall of 1.85%. Though this is disappointing, we have identified two areas of improvement to amend. Because of our choice of setting $L = 1$ as described in Section 4.2 due to scalability concerns, our resulting network is not well connected. In particular, we see that the GG part of the PG_GG network (where the pagerank is run) contains 735,136 weakly connected components. This may explain the low recall as our setting of L results in a network where the random walker cannot easily reach parts of the network simply due to the lack of connections. As such an immediate change would be to increase the L parameter. Another method we could investigate to improve recall and possibly precision is increasing the number of product-gallery edges per product node. Though we previously have worried that product-gallery edges are noisy as seen in Figure 5, we also see that scaling the number of gallery nodes as we did in the PG_GG network improves the reliability of the product-gallery edges. By having more nodes in the query set for the personalized pagerank algorithm, we hope to reduce the influence that a few bad product-gallery edges could have while also increasing recall as the query set covers more of the PG_GG network and therefore allowing us to teleport to different parts of the PG_GG network.

7. CONCLUSIONS

Overall we have shown that by using our best visually generated product gallery network, we are able to predict amazon purchase behavior relatively well at 88.24%. Though we have seen that majority of the prediction is due to visual similarity modeling the *substitute* relationship well, we have nonetheless shown that visual signals are valuable as we can use publically available images and models to replicate the behavior of a co-purchase recommendation system.

We will continue to investigate this work as we wonder how scaling the visually generated network to hundreds of millions of gallery nodes could affect the evaluation quality of our network. Furthermore though we have shown through our evaluation that our network does well in replicating Amazon purchase behavior, we are curious in seeing the actual recommendations that our visually generated network can produce.

8. ACKNOWLEDGEMENTS

I'd like to thank Yushi Jing, Michael Feng, and Yiming Jen for helping to brainstorm the initial investigation of this work.

9. REFERENCES

Figure 6: Visualization of the shortest path between from anchor to positive (up) and anchor to negative (bottom) when only using the product-gallery edges of the PP_PG network. We can see how the product-gallery edges are very noisy as shown for example by the holster and shoe edge.

Figure 7: Examples of successful triplet prediction. In the first triplet, we see a successful complementary prediction

Figure 8: Examples of failed triplet prediction. Majority of the failed examples are complements which shows that more work is needed to better encode complementary information

[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] P. Lofgren, S. Banerjee, and A. Goel. Personalized pagerank estimation and search: A bidirectional approach. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*, pages 163–172. ACM, 2016.

[7] 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.

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

[9] 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.

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

[11] 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.

[12] 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.