Deciphering Fashion Sensibility Using Community Detection

Sagar Arora*, Amber Madvariya*, Divya Alok, Sumit Borar
Myntra Designs, Pvt. Ltd
Bengaluru, India
{sagar.arora,amber.madvariya,divya.alok,sumit.borar}@myntra.com

ABSTRACT

Myntra is one of the leading fashion e-commerce portal in India. As a leading fashion e-tailer with high repeat rates, it is incumbent on us to understand our users better over time and provide an unparalleled fashion buying experience. In order to do that effectively, it is imperative to understand fashion tastes of an individual that underpins an individual’s fashion choices and use it to enhance the e-store experience. In the first part of the paper we have described the methodology to encode fashion tastes in a product relationship graph (using the clickstream data) and it’s usage in an application leading to better user engagement. The latter part details on partitioning of the graph using Louvain Algorithm and creation of fashion sensibilities which can be thought of as commonly occurring fashion tastes over our user cohorts. We show that graph communities are able to capture the user’s fashion taste better than typical content based homogeneous communities. As a validation of the approach, we would be testing the time invariance of fashion sensibilities over our users.

1. INTRODUCTION

E-commerce is often considered to have infinite virtual shelf space; serving millions of products to customers. Specifically, the advent and growth of fashion industry pose striking challenges to render relevant products to user. The rapid pace at which industry is growing, with a huge catalogue of products necessitates personalisation. Users tend to browse significantly large set of products before purchasing; especially due to the fact that their fashion need state is not as concrete and specific as in other domains like movies, books, electronics etc. Hence, personalisation becomes an important lever to cater to diverse users’ need, allowing for better product discovery and customer experience.

In Myntra, there are around 1.5M distinct products in the catalogue, and are rapidly increasing everyday. On an average 50 products are browsed before the first click. Everyday millions of users visit the portal and have hundreds of millions of list impressions1. This raises a challenge to decipher user’s fashion sensibilities to enable better, efficient and faster product discovery via personalisation and drive conversions.

*Both authors have contributed equally
1Cumulative number of views on the catalogued products

A fashion product is usually described in terms of various attributes like fabric, neck, styling, pattern, shape, hemline etc. However, these attributes often fail to describe users’ implicit fashion sensibility (aka taste) in entirety. Considering women-dresses as an example, any attribute based approach would tend to discover homogeneous communities of different shape of dresses (eg A-line, maxi, bodycon, skater, boat etc) on a broader level; shape being the most important feature of a dress. On Myntra, we have 26 distinct dress shapes. Across all the dress shoppers on Myntra, we have observed that 74% users shop for more than or equal to 4 distinct dress shapes. The Figure 1 shows the actual distribution of dress shoppers across distinct dress shapes. Hence, a typical attribute based approach of understanding sensibilities does not suffice. In fact, sensibilities should be defined in terms of broader fashion communities as shown in Figure 7.

Our approach hinges on discovering relationships between products using millions of user sessions. We model these relationships as a graph with products as vertices and use both the explicit signals (purchases) and implicit signals (product clicks, add to carts etc.) to compute the edge weights between vertices. This product relationship graph is then partitioned into multiple communities, using Louvain Algorithm [9]. A user’s fashion sensibility is then computed in terms of these discovered communities. We define fashion sensibility as a real valued vector describing affinities of a user to different product communities (clusters).

2. RELATED WORK

There have been studies to capture the products’ relationships using various content based approaches [2] or collaborative filtering [11]. [8] describes how it is possible to infer networks of substitutable and complementary products using product text, ratings, reviews, specifications etc. [3] uses multi-modal features of fashion products to recommend outfits to users using functional tensor factorization method. There has been a separate body of work trying to capture fashion aesthetics using computer vision. [5]. Content based similarity measures rely on a curated taxonomy to represent a particular content. Maintaining and curating this taxonomy system becomes an unscalable task over time. On the other hand, collaborative filtering algorithms use user signals instead of a taxonomy to compute similarity between products. [15] involves using user purchases to create the product similarity graph. The challenge in using only user purchase data to compute product similarity in fashion e-commerce is that it is sparse and erratic; mainly due to the
short life cycle of products and the breadth of the catalogue available.

Our earlier work [1] focussed on decoding the fashion context using word embeddings [10]. This considers session\(^a\) as a proxy to context and embed all fashion attributes, products and even sessions to a common space. The approach used was to consider each session as a document and the attributes of all products clicked in the session as words. Skip gram model is trained to learn embeddings of all products' attributes; which are further aggregated (centroïd) to learn embeddings of all products and sessions. Our another work [12] further attempts to model the impulsive behaviour of users in fashion ecommerce by employing a deep Gated Recurrent Neural Network. While both our previous body of work focused more on learning the embeddings and using those to decode the user’s context in-session (using only the current session’s context), this work focuses on deciphering the long-term and broad fashion sensibility of users. This work uses the entire user’s history on platform to model the fashion sensibility:

\[^a\]Session refers to products clicked by a user within a 30 minute window.

Let \( U \) be the set of all users, \( P \) be the set of all products, and \( A \) be the set of all categories (eg Men-Tshirts, Men-Jeans, Women-Tops, Women-Dresses etc.). Further, let \( m \) be the number of users and \( n \) be the number of products.

Let \( E_{up} = \{E_c, E_b, E_w, E_p\} \) be the set of all possible events (user-products interactions) where \( E_c = \) click event, \( E_b = \) add to bag event, \( E_w = \) add to wish list event and \( E_p = \) purchase event. It is worth mentioning that we just consider the highest priority for a given user and product with priorities defined as:

\[
E_c < E_b < E_w < E_p
\]  

We also define the importance score\(^b\) of an event \( e \) as

\[
w_e = \begin{cases} 
1, & \text{if } e = E_c \\
\frac{\sum E_c}{E_b}, & \text{if } e = E_b \\
\frac{\sum E_c}{E_w}, & \text{if } e = E_w \\
\frac{\sum E_c}{E_p}, & \text{if } e = E_p 
\end{cases}
\]  

\[^b\]The importance scores are computed using past 1 month data of the platform.

3. METHODOLOGY

This section describes the approach to create the product relationship graph, carve out communities from it and use these communities to decipher fashion sensibilities of users as shown in figure 2.

Hereafter, we will use the following terminology.

\[^a\]Session refers to products clicked by a user within a 30 minute window.

\[^b\]The importance scores are computed using past 1 month data of the platform.

Figure 1: Percentage of users by distinct shapes of dresses bought

Figure 2: System design showing various components and data flow

3.1 Product Relationship Graph

We construct the undirected weighted graph \( G = (V, E) \) with \( V = P \) and \( E \) be the edges denoting the similarity between two products. Considering \( e_{ij} \) as the interaction event of user \( U_i \) with product \( P_j \), \( U_i \) is represented by a vector \( u_i \) of length \( n \) such that

\[
u_{ij} = \begin{cases} 
w_e, & \text{if } e_{ij} \in E_{up} \\
0, & \text{otherwise} \end{cases}
\]  

We create a sparse user-product matrix of size \( m \times n \) where each row represents a user and each column represents a product. On this matrix, we apply non-negative matrix factorization [6] to learn the latent \( d \)-dimensional embeddings for each user (\( v_{ui} \)) and product (\( v_{pj} \)). We use these latent embeddings to calculate the edge weight between vertices in the graph. The edge weight between products \( p_i \) and \( p_j \) is then defined as \( E_{ij} = v_{pi} \cdot v_{pj} \) where \( E_{ij} \in E \) 

The product relationship graph \( G \) formed after calculating edge weights contains edges across all categories in \( A \) i.e. there are edges both within and across categories. However, it is observed that intra-category edge weights are much higher than inter-category edge weights.

3.2 Community Detection

For the purpose of finding communities within the graph, we remove all inter category edges and preserve only intra category edges. Hence, the graph \( G \) after pruning, splits into sub-graphs \( G_i = (V_i, E_i) \) with each sub-graph representing a category and \( V_i \) represents the set of products within that category.

Now, for each subgraph \( G_i \), we run louvain algorithm to maximise the modularity [9] and discover communities \( S = \)
\{1, 2, \ldots, S_i\} \) for the \(i^{th}\) category. The resulting partitioning ensures that each product gets mapped to only one category.

### 3.3 Fashion Sensibility

Let \( C = \bigcup S \) be the set of communities across all categories. We define fashion sensibility of user \( U_i \) as a real valued vector \( w = (w(c_1), w(c_2), \ldots, w(c_n)) \) with each dimension referring to long term fashion affinity to corresponding community. The affinity is calculated as cumulative sum of user’s events corresponding to a given community weighted by an exponential time decay factor (half life as 1 year). Considering all events \( e = \{E_{ij}\} \) as the interactions of each user \( U_i \) with products of community \( C_j \), fashion sensibility \( F \) of \( U_i \) (for community \( C_j \)) is computed as

\[
F_j = \sum_{e \in U_i, C_j} w(e) \times e^{-\frac{t}{\ln(2)}} \quad (4)
\]

\( t \) denotes the days elapsed since the event occurred.

### 4. EXPERIMENTS AND RESULTS

For all our experiments, we use the clickstream data of users for the last 1 year. We consider 100M sessions aggregated per user across 1.5M distinct products from 100 distinct categories. To calculate the values of \( w_{E_0}, w_{E_a}, w_{E_w}, \text{and } w_{E_p} \), we aggregate the value of \( E_0, E_a, E_w, E_p \) over data from the past month. From these values and equation 2, we have \( w_{E_0} = 1, w_{E_a} = 11.45, w_{E_w} = 26.17 \text{ and } w_{E_p} = 66.15 \).

#### 4.1 Product Relationship Graph

We use the entire 100M sessions to create the product graph \( G \). The graph has 1.5M vertices corresponding to each product and 93.7M edges. We evaluated the graph qualitatively based on the inputs from our stylists and also used the graph for recommending similar products on our platform as shown in Figure 3.

---

**4.1.1 Qualitative Results**

This section shows nearest products to different query products sorted by the edge weights (within different categories). For each query product, we consider the subgraph \( G_i \) of the required category and inspect the nearest products. Figure 6(a) shows similar dresses and tops to a fashion-forward edgy dress. Users who prefer an off shoulder dress would also prefer cropped tops. Figure 6(b) shows similar jeans and shirts to a distressed and torned jeans. The similar shirts are either denim/biker shirts or printed. Both the examples reflect that users share a common fashion sensibility across categories.

**4.1.2 Similar Products A/B Test**

For the A/B test, we render the similar products on the product details page of the query product as shown in Figure 3. We test the similar products generated by product relationship graph against the similar products generated by attribute based similarity as the baseline approach wherein each product is represented in terms on its attributes and the relationship is defined in terms of its cosine similarity with other products. We render similar products generated by the product relationship graph \( G \) to randomly sampled 50% of our total traffic and render attribute based similar products to the rest of the traffic (both test and control set have 1M users). We recorded the number of list impressions and the number of clicks for both the approaches over a period of three weeks. Figure 4 shows the CTR (click through rate defined as ratio of clicks to impressions) recorded during this test period for both the approaches averaged over each day of the week. From the test, we observe that graph based similar products show an improvement of 58.4% in average CTR over attribute based similar products with a p-value[14] of 6.6*10^{-6}.

---

**4.2 Community Detection**

We computed 100 subgraphs \( G_i \), corresponding to distinct categories from the product relationship graph \( G \). On each subgraph, we ran Louvain algorithm to compute the communities. The Table 1 shows the structural properties of subgraph, final modularity and number of communities discovered for top categories. We also got the communities validated by fashion stylists. Each community was tagged on various dimensions like product sensibility (basic, moderate fashion quotient, high fashion affinity to corresponding community. The affinity is calculated as cumulative sum of user’s events corresponding to a given community weighted by an exponential time decay factor (half life as 1 year). Considering all events \( e = \{E_{ij}\} \) as the interactions of each user \( U_i \) with products of community \( C_j \), fashion sensibility \( F \) of \( U_i \) (for community \( C_j \)) is computed as

\[
F_j = \sum_{e \in U_i, C_j} w(e) \times e^{-\frac{t}{\ln(2)}} \quad (4)
\]

\( t \) denotes the days elapsed since the event occurred.

---

**Figure 4**: Average CTR over a period of 3 weeks

---

**Figure 3**: Similar Products Feature on Platform
Table 1: Communities of Top Article Types

<table>
<thead>
<tr>
<th>Category</th>
<th>No. of nodes</th>
<th>No. of edges</th>
<th>No. of Communities</th>
<th>Graph Modularity</th>
<th>No. of Attribute Clusters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Men-Tshirts</td>
<td>111590</td>
<td>9985252</td>
<td>16</td>
<td>0.3998</td>
<td>14</td>
</tr>
<tr>
<td>Men-Shirts</td>
<td>102220</td>
<td>16459493</td>
<td>11</td>
<td>0.4015</td>
<td>8</td>
</tr>
<tr>
<td>Men-Jeans</td>
<td>27740</td>
<td>6014500</td>
<td>7</td>
<td>0.3009</td>
<td>9</td>
</tr>
<tr>
<td>Men-Casual Shoes</td>
<td>36003</td>
<td>5729011</td>
<td>9</td>
<td>0.3602</td>
<td>6</td>
</tr>
<tr>
<td>Women-Dresses</td>
<td>44909</td>
<td>12934782</td>
<td>6</td>
<td>0.2908</td>
<td>5</td>
</tr>
<tr>
<td>Women-Tops</td>
<td>88536</td>
<td>19058796</td>
<td>10</td>
<td>0.3456</td>
<td>7</td>
</tr>
<tr>
<td>Women-Kurtas</td>
<td>76494</td>
<td>19329855</td>
<td>10</td>
<td>0.2802</td>
<td>8</td>
</tr>
<tr>
<td>Women-Heels</td>
<td>39042</td>
<td>4214382</td>
<td>6</td>
<td>0.2743</td>
<td>4</td>
</tr>
</tbody>
</table>

Table 2: Sensibilities for Top Article Types

<table>
<thead>
<tr>
<th>Category</th>
<th>Average Communities per User</th>
<th>Average Clusters per User</th>
<th>Jaccard Index (Product Graph)</th>
<th>Jaccard Index (Attributes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Men-Tshirts</td>
<td>2.892</td>
<td>4.135</td>
<td>0.3411</td>
<td>0.0988</td>
</tr>
<tr>
<td>Men-Shirts</td>
<td>2.362</td>
<td>3.560</td>
<td>0.4690</td>
<td>0.1786</td>
</tr>
<tr>
<td>Men-Jeans</td>
<td>1.964</td>
<td>2.766</td>
<td>0.5017</td>
<td>0.3087</td>
</tr>
<tr>
<td>Men-Casual Shoes</td>
<td>2.065</td>
<td>1.848</td>
<td>0.4969</td>
<td>0.3264</td>
</tr>
<tr>
<td>Women-Dresses</td>
<td>2.661</td>
<td>3.317</td>
<td>0.6394</td>
<td>0.3629</td>
</tr>
<tr>
<td>Women-Tops</td>
<td>3.127</td>
<td>5.526</td>
<td>0.4917</td>
<td>0.1923</td>
</tr>
<tr>
<td>Women-Kurtas</td>
<td>2.962</td>
<td>3.686</td>
<td>0.4247</td>
<td>0.3446</td>
</tr>
<tr>
<td>Women-Heels</td>
<td>2.036</td>
<td>2.139</td>
<td>0.6104</td>
<td>0.3178</td>
</tr>
</tbody>
</table>

4.3 Fashion Sensibility

To compute fashion sensibility, we consider 12 months of activity for each user i.e. from 1 January 2016 to 31 December 2016. To analyze the variation in fashion sensibility of users, we split this 12 month period into 4 quarters for each user. After filtering users who have ordered less than 2 products in each quarter, we are left with 350k users from an initial randomly sampled set of 2M users. We compute users’ sensibility towards each community within a specific category (e.g. Men-Jeans). Figure 5 shows the distribution of count of distinct communities with non zero sensibilities across users in Men-Jeans. We observe that 74.1% of users have a sensibility in less than or equal to 3 distinct communities. Also from Table 2, where the second column represents the number of communities with non zero sensibility averaged across users for different categories and column three represents the same metric for attribute based clusters, it is evident that user sensibilities towards product relationship graph based communities are more cohesive as compared to attribute based communities, with the exception of Men-Casual Shoes. Hence, graph based communities are able to capture the broad fashion taste of users as compared to attribute based clusters.

In another experiment to analyse the coherency in users’ sensibilities across time, we compute the set of communities with non zero sensibilities for every quarter (for a specific category). We denote the set of communities for each quarter as \(\{Q_1, Q_2, Q_3, Q_4\}\) and calculate the jaccard index[4] between them for each user, using the following equation

\[
J = \frac{|Q_1 \cap Q_2 \cap Q_3 \cap Q_4|}{|Q_1 \cup Q_2 \cup Q_3 \cup Q_4|}
\]  

We repeat the above experiment for attribute based groups and compare the jaccard index obtained from both the approaches. The fourth and fifth column of Table 2 lists the jaccard index for each article type averaged across users for both product graph communities and attribute based clus-
Figure 6: Comparison of adjacent products to a) Edgy, fashion forward and off-shoulder dress and b) Distressed and Torn Jeans

(a) Classics
- MANGO
- MANGO
- VEROMODA
- ESPRIT
- MARKS & SPENCER
- VEROMODA

(b) Quirky
- DESIGUAL
- MANGO
- DESIGUAL
- HARPA
- DOROTHYPERKINS
- DOROTHYPERKINS

(c) Trend-Setters
- FOREVER21
- FOREVER21
- FOREVER21
- RARE
- FARALLEY
- AND

(d) Sophisticated
- ESPRIT
- ESPRIT
- MARKS & SPENCER
- MARKS & SPENCER
- MANGO
- MARIECLAIRE

(e) Experimental
- REDSISTER
- SUO
- COVERSTORY
- FOREVER21
- MS TAKEN
- PEPEJEANS

Figure 7: Sample Communities from Women-Dresses
ters. From the table, it is evident that the jaccard index for product graph based communities is higher than attribute based clusters in each category. Hence, graph based communities result in sensibilities which are more coherent w.r.t time; hence allowing better insights into long term user behaviour.

5. CONCLUSIONS
We showed how we construct the product relationship graph using collaborative filtering on the entire user history of all users. The product graph was used in a similar recommendation product setting and it showed improvements in key customer metrics. We also showed how typical content based approaches fail to discover long term fashion sensibilities. Instead, the product relationship graph when partitioned using Louvain algorithm forms more coherent communities aligned to long term fashion sensibilities. We also learnt the fashion sensibilities of users across article types and demonstrated that sensibilities are quite coherent w.r.t. time.

6. REFERENCES