

# Fashionista: A Fashion-aware Graphical System for Exploring Visually Similar Items

Ruining He\*, Chunbin Lin\*, Julian McAuley  
University of California, San Diego  
La Jolla, California, U.S.A.  
{r4he, chunbinlin, jmcauley}@cs.ucsd.edu

## ABSTRACT

To build a fashion recommendation system, we need to help users retrieve *fashionable items* that are visually similar to a particular query, for reasons ranging from searching alternatives (i.e., substitutes), to generating stylish outfits that are visually consistent, among other applications. In domains like clothing and accessories, such considerations are particularly paramount as the visual appearance of items is a critical feature that guides users' decisions. However, existing systems like *Amazon* and *eBay* still rely mainly on keyword search and recommending loosely consistent items (e.g. based on co-purchasing or browsing data), without an interface that makes use of *visual information* to serve the above needs.

In this paper, we attempt to fill this gap by designing and implementing an image-based query system, called *Fashionista*, which provides a graphical interface to help users efficiently explore those items that are not only visually similar to a given query, but which are also *fashionable*, as determined by visually-aware recommendation approaches. Methodologically, *Fashionista* learns a low-dimensional visual space as well as the evolution of fashion trends from large corpora of binary feedback data such as purchase histories of *Women's Clothing & Accessories* from *Amazon*, which we use for this demonstration.

## Keywords

Fashion Trends; Visualization; Recommendation

## 1. INTRODUCTION

With the proliferation of e-commerce and online recommendation services, users increasingly rely on intelligent systems to explore large corpora of items, in order to discover products to purchase, movies to watch, news to read, etc. When it comes to domains like clothing recommendation—where visual factors play a key role—users' decisions depend

\*Both authors contributed equally to this work.

largely on the visual attractiveness (or 'fashionability') of the products in question. Therefore we argue that an interactive, visually-aware interface for exploring items can be of significant value in such domains.

Visual similarity entails specific notions of 'visual compatibility' between items, such as substitutable and complementary products [1]. Substitutes are similar items that are interchangeable with one another, e.g. one pair of shoes for another. In contrast, complements are items that can visually 'go together,' such as a t-shirt and a matching pair of jeans. We hope to help users retrieve and explore items that are potential substitutes or complements for items they are currently considering. Retrieving visual substitutes can help users to discover items that may be preferable in terms of their price, popularity, brand, or even items which are considered more fashionable by the community. Alternately, retrieving visually complementary items can help users to find items with consistent styles, and to generate sets of items (i.e., 'outfits').

The major challenges for a system to provide such support lie in the ability to (1) use complex similarity measures that are capable of capturing human notions of visual similarity (i.e., what humans consider to be substitutable and complementary), and (2) index, query, and retrieve visually similar items from a large corpus (e.g. millions of items) efficiently.

Furthermore, this problem is challenging as fashions are continually evolving over time, such that temporal information is key to determining what is currently considered 'fashionable.' This means that in addition to modeling large corpora of users' interactions with items, we must also model their temporal dynamics in order to capture the notion of 'fashion trends.' Our main goal is to address these challenges and design a system (which we title *Fashionista*) to provide a user-friendly graphical interface to help users find *fashionable* items that are visually similar to a given query.

In contrast to current approaches, existing systems like *Amazon's* own recommendation interface<sup>1</sup> don't help users to retrieve visually consistent items, though they do recommend items that have been frequently bought/viewed together, which may be correlated with visual compatibility. Alternately, existing image search engines (like *Google Images*<sup>2</sup>) are based either on keyword search or traditional image similarity comparison; however they are not optimized specifically to capture human notions of 'visual style,' i.e., they are not trained on users' interactions with items as ours is. To the best of our knowledge, none of these existing sys-

<sup>1</sup><http://www.amazon.com/>

<sup>2</sup><https://images.google.com/>

tems provides an interface that can help users evaluate the fashionability of items as *Fashionista* does.

Methodologically, *Fashionista* is based on state-of-the-art visual models that are able to learn the temporal evolution of fashion trends from large corpora of binary feedback data with images and timestamps [1, 2, 3]. For visualization and efficient querying, all items in the corpus are embedded into a low-dimensional visual space (‘style space’) such that nearby items (i.e., neighbors) have been evaluated similarly (in terms of appearance) by the users in the corpus. More specifically, our contributions are summarized as follows:

- We propose a demonstration of a system, *Fashionista*, which provides a novel graphical interface for exploring visually similar items within an automatically learned visual space.
- *Fashionista* learns and visualizes the fashionability evolution of each item during the lifetime of the corpus, which can help users to find currently ‘stylish’ items.
- *Fashionista* operates on large, real-world corpora of items. For demonstration, we use the *Women’s Clothing & Accessories* dataset from *Amazon* [1], which consists of over 0.6 million items, 1.8 million users, and 3.2 million user-item interactions spanning 11 years.

This rest of the paper is organized as follows. We first introduce *Fashionista*’s architecture and interface in Section 2, followed by demonstrations of multiple use-cases in Section 3. We discuss related works in Section 4 and conclude in Section 5.

## 2. OVERVIEW OF FASHIONISTA

In this section we give an overview of the architecture of *Fashionista*, after which we discuss the unique features of our user interface.

### 2.1 Architecture

The system architecture of *Fashionista* is presented in Figure 1. On the server side, an *Index Builder* creates in-memory indices on the system’s ‘knowledge’, i.e., the visual space and fashion trends produced by the *Fashion Learner*. On the client side, a query is created by the *Query Generator* with the help of the *Autocompletion Manager*, which provides a list of candidate queries. Once a complete query is submitted from the client, the *Fashionability Retriever* and the *Nearest Neighbor (NN) Searcher* are triggered to obtain the query item’s fashionability values and its k-NNs. In particular, to obtain k-NNs, *Fashionista* first looks up the category inverted index to retrieve a list of candidate items. Afterwards, *Fashionista* builds a heap of items whose fashionability scores are greater than  $\alpha$  and the top-k items are popped off the heap. Those results are then sent to the *Fashionability Visualizer* and the *Visual Space Explorer* for visualization.

### 2.2 User Interface

The interface for *Fashionista* is shown in Figure 2, which consists of three parts: the query generator, fashion trend exhibitor, and visual space explorer (corresponding to ①, ②, and ③ in Figure 2, respectively).

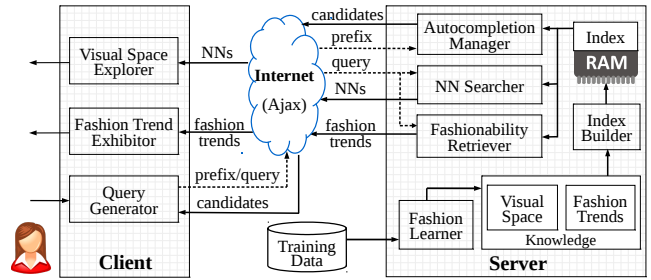


Figure 1: Architecture of *Fashionista*.

#### 2.2.1 Query Generator

A query generated by the *Fashionista* query generator consists of four parts: (1) a list of categories chosen using the category filter component; (2) an item identifier input by the user, which can either be typed directly or chosen from a list of candidate queries provided by *Fashionista*’s auto-completion manager (or selected through navigation); (3) a number  $k$ , which is selected from a drop-down list; and (4) a threshold value  $\alpha$ , which is set by sliding the window in the slider.

Query autocompletion [4] helps users to identify candidate queries when only prefixes are input. It significantly improves the efficiency of query generation since users are only required to input a few characters. In *Fashionista*, the server maintains a *trie* for all item identifiers in its main memory. Every time a new character is entered, an Ajax request is sent to the server which looks up the *trie* and returns a list of candidate items with a matching prefix. In addition, the server also returns the corresponding icon images for the candidate items to provide a preview in the item id input text box.

#### 2.2.2 Fashion Trend Exhibitor

The fashion trend exhibitor surfaces a plot of the fashionability scores of the query image from the year of 2006 to 2014, based on the model from [2]. Technically, *Fashionista* maintains an inverted index for each item to index its fashionability scores. For example, the fashionability trend in Figure 2 tells the user that the shoe (with item id *B004V7858C*) is gaining fashionability in recent years.

#### 2.2.3 Visual Space Explorer

The visual space explorer visualizes the k-NNs of the query image within the visual space, after filtering out items not from the selected categories or with fashionability scores lower than  $\alpha$ . The k-NNs are displayed as a heat map where the color of each point indicates the fashionability of the item,<sup>3</sup> as shown in ③ in Figure 2. When the spot is clicked, detailed information like brand, price, rating, etc. of the clicked item is displayed on a product description panel.

Users can explore this visual space by (1) zooming in/out and dragging the visual space explorer, and (2) clicking the ID of an item on its product description panel to issue a new query. Figure 3 shows an example where we zoom in the area around the query item *B004V7858C*. As we can see from this figure, the nearest neighbors retrieved by *Fashionista* share similar visual style with the query. Assuming the user is

<sup>3</sup>Warmer color means greater fashionability score.

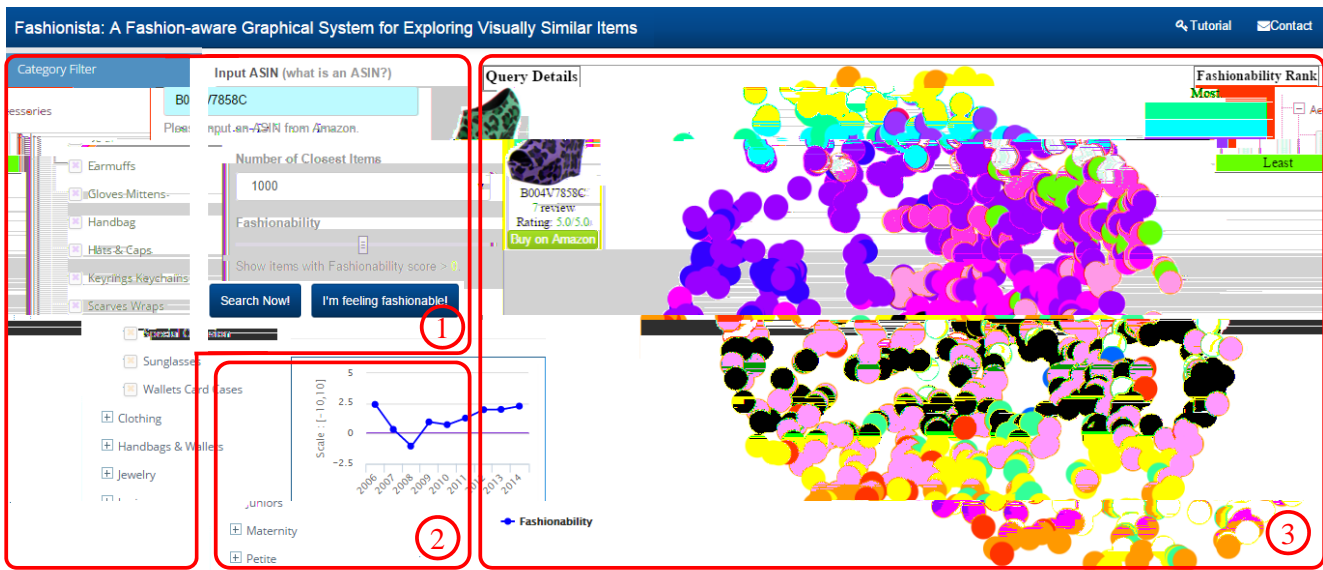


Figure 2: A screenshot of *Fashionista*. ① is the query generator with four components: I. category filter (supporting multiple selections); II. item id input text box; III. drop-down list accepting a number of NNs to return; and IV. a slider to filter out unfashionable items. ② is the exhibitor showing the fashion trend of the query item. ③ is the visual space explorer with zoom-in/out, drag and rotation features. Demo link: <http://132.239.95.211:8080/demowww/index.jsp>.

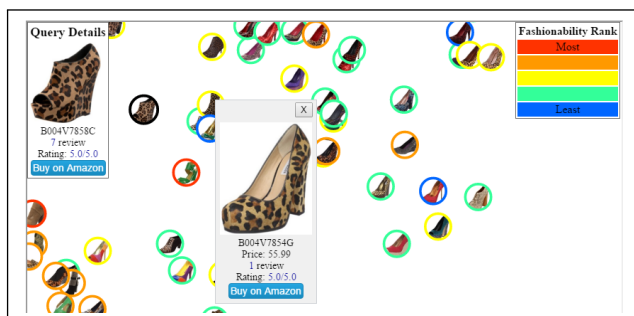


Figure 3: A screenshot of the zoom-in effect of the visual space explorer. Note that (1) the detailed information of each item can be obtained by clicking its image; and (2) the color of the circled border of each image indicates the item’s fashionability rank. The ranking color bar can be seen at the top-right corner.

attracted to this neighbor, they can query this item simply by clicking its ID.

### 2.3 Fashion Learner

*Fashionista* is built on top of state-of-the-art visual models that are capable of learning both the dimensions of people’s visual preferences as well as their temporal dynamics, i.e., fashion trends.

**Training data.** To capture the temporal dynamics of the general tastes of the public, our model directly learns from large datasets from *Amazon*, introduced by [1]. This dataset temporally spans almost one decade. Training data consists

of triplets of the form  $(user, item, timestamp)$  denoting one purchase entry. Additionally, one image for each item is used to extract high-level visual features from a pretrained Deep Convolutional Neural Network (Deep CNN) [5], fed as input to the model. This is a One-Class Collaborative Filtering setting [6] where our model estimates users’ *fashion-aware* personalized ranking functions based on their past feedback.

**Model Specifics.** Our model produces a *visually-aware* preference predictor that extends standard Matrix Factorization by modeling *visual* dimensions and non-visual (i.e., latent) dimensions simultaneously. Here the basic idea is to uncover low-dimensional decision factors to explain user’s historical purchase activities. To account for fashion dynamics, our model further captures the following phenomena: (1) items gradually gain/lose ‘attractiveness’ in different visual dimensions over time; (2) users weigh visual dimensions differently as time goes by; and (3) each user’s visual preferences also evolve with time. Our model uses ‘global’ structures to capture the first two phenomena which describe visual dynamics at the community level (i.e., fashion evolution) as well as ‘local’ structures to account for personal evolution. See our papers for further details [2, 3]. Ultimately, our model uncovers visual dimensions and fashion dynamics by predicting users’ behaviors as accurately as possible.

Finally, the output of the model includes (1) a projection of items into an uncovered ‘visual space’ where items with similar styles—in terms of the way their appearance is evaluated—are embedded to nearby positions, and (2) fashionability evolution of each item during the time span of the dataset, among other features. To visualize the visual space (10 dimensional in our case) on the user interface, *Fashionista* further embeds it in two dimensions with t-SNE [7].

### 3. DEMONSTRATION & USE CASES

In this section, we demonstrate different use-cases and real-world scenarios where our system might be used to help a user to explore the visual space. Each of these use-cases can be naturally handled by *Fashionista*, and to the best of our knowledge can not be handled by existing systems.

#### 3.1 Querying for fashionability advice

Rachel is hesitant to purchase a dress on *Amazon*. She personally likes it but is not sure whether it is consistent with the current ‘fashion zeitgeist,’ i.e., its fashionability. She searches for the dress on *Fashionista*. *Fashionista* demonstrates the fashionability evolution of the dress’s appearance in the past decade, and Rachel finds that it’s gaining popularity in recent years. Armed with this information, she decides to purchase the dress immediately.

#### 3.2 Searching substitutes with similar styles

Lucy finds a pair of shoes (e.g. *B004V7858C*) on *Amazon*, but wants to compare them against alternatives that are similar in appearance, but potentially have a preferable brand, price, or rating (etc.). Among items that are ‘frequently bought together’ (as recommended by *Amazon*), Lucy finds they tend to vary too much in appearance, or otherwise are not visually attractive. Using *Fashionista*, she quickly retrieves hundreds of visually similar items and finds a shoe of the same style but with a preferable brand.

#### 3.3 Finding complements for outfit generation

Angelina is herself a fashionista who just bought a beautiful t-shirt on *Amazon*, and she wants to find some pants and shoes that can go together with it, that is, to generate an outfit with a consistent visual style. She uses *Fashionista*’s category filter to limit the search results to be within the two categories under consideration. According to the query conditions, the system retrieves the nearest neighbors of the t-shirt, according to the learned visual ‘style space,’ each represented with different colors denoting their visual popularity. Angelina explores these items and finds several with high fashionability scores that match the style of the t-shirt.

#### 3.4 I’m feeling fashionable!

Unlike the above scenarios, Christina doesn’t know what she really wants, but just intends to explore some fashionable items. She clicks the ‘I’m feeling fashionable!’ button on *Fashionista*, and the system returns a randomly selected hotspot in the visual space that currently has a high popularity (fashionability). She decides to focus on fashionable coats, using the provided category filter. Some attractive coats are surfaced, and Christina decides to buy one of them using the redirection links.

#### 3.5 Querying for statistical fashion trends

Jennifer is a fashion designer who cares about the trends of contemporary fashion. Using *Fashionista*, she quickly checks the distribution of popular appearances over hundreds of thousands of clothing and accessory items. She zooms into some areas that interest her the most and observes the corresponding fashion trend evolution during the past decade. She identifies certain trends that are likely to gain popularity in the near future and decides to design products that fit them.

### 4. RELATED WORK

Most relevant to our work are existing e-commerce systems like *Amazon*, *eBay*,<sup>4</sup> *Walmart*,<sup>5</sup> etc. In spite of the wide successes achieved by such systems, they are not designed for exploration of visually relevant items. Although they are able to recommend items that are frequently bought or viewed together, they are unable to tease-apart the underlying reasons including functionality, complementarity, visual compatibility, and so forth. This limitation makes it difficult for these systems to address the task of recommending visually consistent items, nor do they provide an interface for efficient exploration within a visual space as we do in this work.

There are also image search engines like *Google Images* that are either based on keywords or image similarity comparison. Such systems differ from ours in two key ways. First, such systems are general-purpose and thus are not suitable for the specific applications considered here. Second, our model maps items into a low-dimensional visual space such that nearby items have been evaluated similarly (by users) in terms of their appearance. This improves upon traditional image similarity comparison methods by performing comparisons only on those dimensions that are discovered to be relevant to people’s decisions.

### 5. CONCLUSION AND FUTURE WORK

In this paper, we built a demo system, *Fashionista*, with a user-friendly graphical interface to search and explore visually similar and fashionable items. Experimentally, we found that *Fashionista* can index and search large-scale real-world corpora efficiently and effectively. In the future, we hope to enable our system to automatically generate outfits according to *personalized* visual preferences and fashion trends. This will be feasible once we are able to observe users’ interactions with *Fashionista*.

### 6. REFERENCES

- [1] J. J. McAuley, C. Targett, Q. Shi, and A. van den Hengel, “Image-based recommendations on styles and substitutes,” in *SIGIR*, 2015.
- [2] R. He and J. McAuley, “Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering,” in *WWW*, 2016.
- [3] R. He and J. McAuley, “VBPR: visual bayesian personalized ranking from implicit feedback,” in *AAAI*, 2016.
- [4] C. Lin, J. Lu, T. W. Ling, and B. Cautis, “Lotus: a position-aware xml graphical search system with auto-completion,” in *ICDE*, 2012.
- [5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *NIPS*, 2012.
- [6] R. Pan, Y. Zhou, B. Cao, N. N. Liu, R. Lukose, M. Scholz, and Q. Yang, “One-class collaborative filtering,” in *ICDM*, 2008.
- [7] L. van der Maaten, “Accelerating t-SNE using tree-based algorithms,” *JMLR*, 2014.

<sup>4</sup><http://www.ebay.com/>

<sup>5</sup><http://www.walmart.com/>