Discovering Latent "Style": An Approach to Recommending and Matching Clothes

Andrew Zeng
Department of Computer Science
Princeton University
andrewz@princeton.edu

Urvashi Uberoy
Department of Computer Science
Princeton University
uuberoy@princeton.edu

Khyati Agrawal
Department of Computer Science
Princeton University
khyatia@princeton.edu

Abstract

This report explores the extent to which a machine learning model can be a personal stylist by exploring the effectiveness of various topic models in discovering style and recommending clothes. Using the Deep Fashion dataset, we explored two areas: finding latent "looks" in a collection of tops and bottoms through "topic" modeling techniques (Latent Dirichlet Allocation, Non-negative Matrix Factorization, Restricted Bolztmann Machine), and using these latent looks to quantify style similarity between clothes and recommend outfits. We evaluated the effectiveness of our models using reconstruction error as the metric. Additionally, we qualitatively analyzed top weighted semantic attributes and representative clothes images for a "topic" to identify the signature style of that topic. We predicted "bottoms" to match "tops" and vice versa based on distance in the discovered latent space. We qualitatively and quantitatively evaluated our matches against random matches. NMF had the best reconstruction error among the three methods we applied, and our predicted matches had an average accuracy of 58% compared to the baseline accuracy for random matches of 49%.

1 Introduction

1.1 Problem Overview

The fashion industry is a $1.2 trillion USD industry. Computer vision applications in fashion that can successfully models fashion concepts such as style, compatibility, and stylishness will transform the way consumers shop for clothing and the way clothing brands analyze trends. Recent computer vision research in fashion includes advances in parsing clothing, recognizing clothing, attributes, and styles, matching real world clothing images to catalogs, and recommending clothing.

An important challenge for computer vision is to learn how to capture the style-coherent similarity between outfits, i.e quantifying how much two pieces of clothing share a latent "look" or "style". Finding latent styles in clothing items can help consumers browse diverse outfits with similar style or help create a recommendation system that suggests new clothing items that are close to a user's personal style. It is often difficult to identify if a particular piece of clothing fits an intended "style". For example, while dressing for an interview, it can be helpful for an individual to measure how well a top fits the "professional" style. This can be done by measuring similarity between the clothing item in question and other clothing items falling under the "professional" category. This can also be done by matching a set of attributes to a "style". (For example, if the top is a "button-down" with a "collar", it is likely to fit the "professional" criteria). Discovering relationships between attributes of clothing (such as fabric, cut, print, color) and intended styles (such as formal, business casual, summery) can enable individuals to shop "smarter" and save money by looking for versatile clothing items (i.e. items that can fall under multiple "styles" or "topics."

The data used for this project is the Category and Attribute Prediction Benchmark subset of the DeepFashion dataset [6]. The original purpose of this data is to train and evaluate the performance of clothing category and attribute prediction models. It contains 289,222 clothes images each annotated with a binary vector of 1000 semantic clothing attributes.

Given this matrix of binary data, we want to see how well various topic model methods can discover latent styles using the text metadata of clothes images. Using the topics we learn, we then use the latent topic representation of the clothes to recommend compatible clothes for a given top or bottom clothing item.

1.2 Approach Overview

Unlike previous work, we are not interested in supervised broad style classification (labeling an outfit as one of a few categories such as athletic, preppy, hipster). [3] Instead, the "style" under consideration for this project is latent fine-grained trends (groups of attributes). The topic model methods we used are Latent Dirichlet Allocation, Non-negative Matrix Factorization, and Bernoulli Restricted Boltzmann Machine. To use topic models for our purposes, we represent a clothing item as a "document", visual attributes (e.g., abstract print, belted floral, fuzzy knit) as "words", and a discovered style as a "topic".

1.3 Results Overview

The latent "topics" from our models were able to capture latent "styles" for clothing images. The reconstruction error for topic extraction using the NMF model (using k = 10 components) was lower than the "root sum of squared distances" from K-Means clustering (with k = 10 clusters) confirming the presence of exploitable latent structure in the images of clothes. Qualitatively, we observed that the topics successfully weighed attributes to represented specific styles such as "formal", "boho", "professional", "casual" etc. We found that NMF had lower reconstruction error on the training and test sets compared to other models. Additionally we found that, predicting "top" to "bottom" matches based on similarity between latent "styles" found by NMF is better than performing random matching.

2 Related Work

Fashion Attributes

Human understandable attributes such as *floral, striped, knit* are very important in analyzing fashion images. There has been prior work that explores ways to identify attribute labels for clothing images, including training convolutional neural networks (CNNs) to predict various different types of attributes such as *pattern, shape, clothing article, color* [1]. Improvements in attribute prediction would greatly benefit any unsupervised models created for learning latent style. For this project, because we did not have the time or resources to create our own attribute classifiers using neural networks, we chose to use a dataset that is pre-annotated with 1000 clothing attributes.

Clothing recommendations

There has been previous work that matches clothing from the real world to clothing in store catalogs [2]. This work has mostly been supervised learning with labeled pairs of matching clothing items. For our project we look at unsupervised approaches that can be more easily applied to large sets of clothing image data without manual labeling. In addition, we try to match/recommend clothing items that are compatible in terms of style rather than matching by visual attribute similarity.

Models of style, compatibility, fashionability

A previous project called The Hipster Wars defined 5 style categories (*Hipster, Goth, Preppy, Pinup, Bohemian*) and identified these styles using body part keypoint patches. [3] Similar to this work, we want to see if we can capture some notion of style; however, we believe style is better captured as a larger set of discoverable latent factors, which is why we turn to topic modelling for our approach.

Topic Models

Topic models were first created in the natural language processing field. The most well-known topic model is Latent Dirichlet Allocation (LDA), which uses multinomial distributions to model the generation of documents comprised of words. [4] Another algorithm that can be used for topic modelling is Non-negative Matrix Factorization (NMF) which is applicable for this project because our data is binary. [5] By using semantic visual attributes for topic modelling, we are able to get interpretable latent topics.

3 Methods

3.1 Dataset Used: Deep Fashion Attribute Prediction Benchmark [6]

The Deep Fashion Attribute Prediction Benchmark contains 289,222 images, each of which is labeled with 1000 descriptive attributes as a binary vector. Each image is identified as a "lower-body", "upper-body" or "full-body" article of clothing. The 1000 descriptive attributes are from 5 different categories: "texture"(abstract, animal, dots), "fabric" (lace, leather, acid), "shape"(crop, midi, asymmetric), "part" (back lace, draped)and "style" (art, babe, island, kiss). The advantage of using this dataset is that it is extremely robust and pre-annotated with ample features for each image. Furthermore, the division of images into upper-body and lower-body provides a good starting point for setting constraints during matching, allowing us to match only upper-body items with lower-body ones.

3.2 Data preprocessing

The first step in preprocessing the data was to separate the tops and the bottoms from the full-body clothing. Once that was done, we were left with around 200,000 images. To make our dataset more manageable, we took 100,000 random samples from the set of tops and bottoms, and split these into 80,000 samples for training set and 20,000 samples for testing.

3.3 Topic Modelling

We decided to use unsupervised learning for this project because we can *discover* underlying components style instead of manually defining a set of styles and trying to classify clothes images as a certain style. Furthermore, in fashion, styles are constantly changing so styles we might define now may not be relevant in 10 years. Topic models also allow us to describe clothes as a mixture of styles instead of a single style. Building topic models on semantic visual attributes yields interpretable latent topics, unlike using pixel data.

3.4 Baseline method

As a baseline comparison for our various topic models we decided to run K-means clustering on the data to see a simple method like this would be able to identify style-coherent clusters. We compare the root squared error (not RMSE) from K-Means with the reconstruction error of our models. For evaluating predicted matches, we used randomly generated matches as a baseline.

3.5 Techniques Used

To identify latent styles, we used the following topic modelling methods on the data:

1. **Latent Dirichlet Allocation (LDA)**
   Hyperparameters: Number of latent variables, learning rate
   Cross validated in sets [5, 10, 15] and [0.5, 0.7, 0.9] respectively

   LDA is a widely used "generative, probabilistic" model used for topic modelling, especially for documents, where each document is represented as a bag of words. Although LDA works best in practice with word counts or non-negative integers as values for the features, it can also be applied to one-hot encoded feature sets (our data). We chose LDA because we could draw an analogy between "document" (clothing item), "words" (visual attributes) and "topics" (hidden styles). Another advantage of LDA is that it outputs a distribution over "topics" for each sample which helps identify versatile clothes that fall under different styles.

2. **Non-negative Matrix Factorization (NMF)**
   Hyperparameters: Number of components [5, 10, 15]

   NMF is a linear algebra algorithm that factorizes a non-negative input matrix $\mathbf{X}$ into two non-negative matrices $\mathbf{W}$ and $\mathbf{H}$. We chose NMF because our data was all non-negative (binary). The derived component $\mathbf{W}$ and weight matrices $\mathbf{H}$ of the decomposition are also non-negative, which allowed us to interpret clothing items as a combination of the style topics. NMF also naturally produces sparse representations. This is appropriate for our version of topic modeling because a single clothing item generally does not contain a large number of styles.

3. **Bernoulli Restricted Boltzmann Machine (bRBM)**
   Hyperparameters: Learning rate, Number of training iterations, Number of latent variables
   Learning rate cross validated in [0.1, 0.01, 0.001] and training iterations in [20, 40, 80]

   The bRBM is a "generative" stochastic neural network that has binary "hidden variables" and binary "visible variables". Since all visible features are binary, our data fits the assumptions of this model. Moreover, finding "binary" hidden variables are appropriate for our task considering that can a clothing item can belong to a latent style (1) or not (0). We used the "restricted" model because it is reasonable to assume that there are no connections between hidden units, as we are trying to capture distinct styles. We also wanted to try a neural network approach to topic modelling, and due to resource and time constraints, we chose the single-layer bRBM.

To tune the hyperparameters to obtain the optimum number of components, we used SciKitLearn's GridSearchCV function that finds the optimal values for each hyperparameter through cross-validation and returns the parameters that yield the best model performance (based on metrics like log-likelihood and model perplexity). For LDA, "10" was the best value for number of latent variables, and to directly compare the performance of the three models, we used "10" to be our number of latent variables for all the models. Intuitively, because our goal was to model latent styles, 10 hidden variables seemed appropriate.

3.6 Evaluation Metrics Used

To evaluate the three models we used reconstruction loss as our primary metric. Reconstruction loss is a measure of the difference between true visible features and reconstructed visible features. We calculated the reconstructed error differently for the three models mathematically described below:

1. **NMF:**
$$E_{\text{reconstruction}} = \text{Frobenius norm}(\mathbf{X}_{train} - \mathbf{WH})$$
   - W = Transformed data according to the fitted NMF model.
     Dimensions: **[number of samples, number of hidden variables]**
   - H = Factorization matrix.
     Dimensions: **[number of hidden variables, number of features]**
2. **LDA:**
$$\mathbf{R} = \underset{\mathbf{A}}{\text{argmin}} ||\mathbf{W}\mathbf{A}^T - \mathbf{B}^T||^2$$
$$E_{\text{reconstruction}} = \text{Frobenius norm}(\mathbf{X}_{train} - \mathbf{R})$$
   where
   - $\mathbf{W}$ represents the distribution over attributes (words) for each the latent style (topic). Dimensions: **[number of hidden variables, number of visible features]**
   - $\mathbf{B}$ is the transformed matrix, it represents the distribution over "topics" for the "documents". Dimensions: **[number of samples, number of hidden variables]**
   - $\mathbf{R}$ is the reconstructed matrix.
     Dimensions: **[number of samples, number of visible features]** as desired.
   - $\mathbf{X}_{train}$ is the target matrix, i.e. the input matrix we reduced to latent features.
     Dimensions: **[number of samples, number of visible features]**

3. **bRBM:** For sampling values of the visible nodes, the bRBM uses the following formula
$$\mathbf{R}^T = \sigma(\mathbf{W}^T\mathbf{B}^T + \mathbf{b}_v)$$
$$E_{\text{reconstruction}} = \text{Frobenius norm}(\mathbf{X}_{train} - \mathbf{R})$$
   where
   - $\mathbf{W}$, $\mathbf{B}$, $\mathbf{R}$ and $\mathbf{X_{train}}$ are the same as in LDA

- b_v is a model parameter representing the biases of the visible units,
  Dimensions: **[number of visible features, 1]**

Since the LDA and bRBM are "generative" models, we additionally compare the "pseudo-log likelihood" (averaged over the samples) of the bRBM model to the compare "approximate-log likelihood" (averaged over the samples) of LDA.

For evaluating our matching algorithm, we randomly sampled a set of 50 tops and 50 bottoms. For these 100 items we predicted the best match (nearest neighbor) using the L2 distance between the hidden features of our NMF model. Additionally, for a "top", we generated a random match from the "bottoms" set and vice versa. For each of these 100 items we had a random match and a predicted match, without knowing which of the two matches was our predicted match and which was the random one, we labeled the combined outfits (the query item with each match) as "good" or "bad" using our own judgement. After labeling, we separated the "predicted" and "random" matches, and calculated the percentage of "good" matches predicted by the two algorithms. We adopted this approach because we wanted to evaluate our outfit recommendations but did not have the time and resources to train a good classifier to grade outfits. We realize that our human evaluation approach is not quite robust. Given more time and resources we would find multiple people who have a good sense of fashion and ask them to grade a much larger set of outfit recommendations and average the results. The evaluation we present here is only a preliminary one.

4 Results

4.1 Latent Structure

Figures 1, 2 and 3 show sample topics for the training data found by RBM, LDA and NMF respectively. Figure 4 shows baseline clustering done with K-Means. For each topic, a list of the top 5 attributes as well as some sample images from the topic are included.

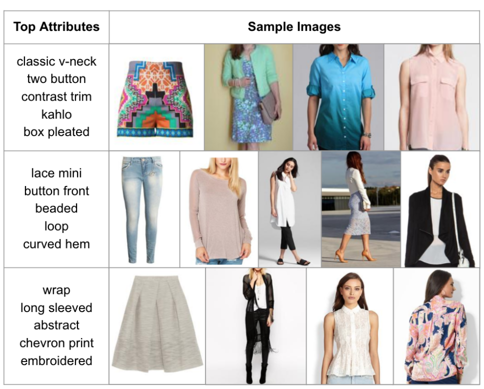

Fig 1: Sample Topics with RBM

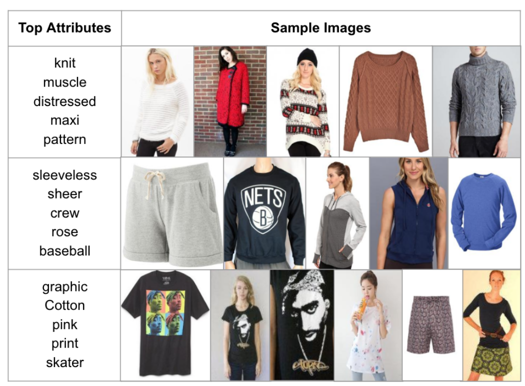

Fig 2: Sample Topics with LDA

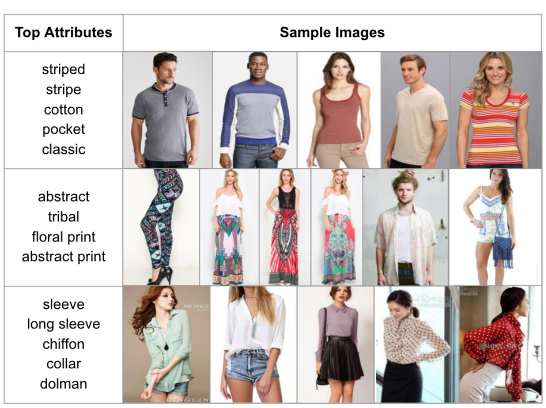

Fig 3: Sample Topics with NMF

Fig 4: Sample Topics with K-Means

From Figure 1, we observe that RBM extracted interesting styles. Topic 1 prioritizes attributes like "kahlo" prints that have bold "contrast" and an artistic look. Topic 2 has features contributing to a "Smart Casual" look while Topic 3 is a more "Formal" look.

Figure 2 shows that LDA identified a "winter" look in topic 1. In topic 2, the attributes "crew", "baseball" and "sleeveless" accompanied by representative images suggest a comfortable "Athleisure" style. Topic 3 consists of "graphic" and "printed" tees suggesting a "cool" look that could work well for both casual or party-wear.

Although Figure 3 only represents 3 topics identified by NMF, attributes and images from all 10 topics found by NMF are included in the Appendix for reference. The first topic in Figure 3 weighs attributes like "striped", "cotton", "pocket", "classic" and the associated images resemble a "casual" comfortable look. Topic 2 represent a "boho" look with "abstract", "tribal" and "floral" prints. The "chiffon", "collar" and "long sleeve" attributes in topic 3 signal a "professional" or "smart casual" appearance of these clothes.

From Figure 4 we see that the clusters found in K-Means are not really representative of a particular style. The clothes in cluster 1 do not seem to have obvious "stylistic" similarity. Topic 2 clusters "animal print" clothing, but that is more of one specific attribute rather than a group of shared attributes. Topic 3 has no discernible similarity except perhaps the loose fitting of the clothes.

4.2 Quantitative Results

Table 1 contains the reconstruction error for each model across both the train and test sets, calculated as per the methods described in Section 3.6. The root sum of squared distances between samples and their cluster means for K-Means clustering was **468.9708** on the training set. These metrics show that latent feature "extraction" is better for clustering this data than the baseline K-Means. One limitation of the reconstruction error as a metric is that the objective function for NMF minimizes this error, and the objective of KMeans minimizes sum of squared distances from means, but the objective functions for LDA and bRBM maximize the data log likelihood. This can be the reason for the lower error for K-Means compared to LDA and bRBM. Since reconstruction error is not the best metric for evaluation of LDA and bRBM we also use the log likelihood scores.
Table 2 shows that the log likelihoods for LDA are marginally higher than those for RBM which may be because of the binary constraint on the hidden variables.

Model	Train Error	Test Error
LDA	503.6761	252.5018
NMF	446.912	224.267
RBM	498.4291	249.8609

Table 1: Reconstruction error across train and test sets for each model

Model	Train log likelihood	Test log likelihood
RBM	-17.8777	-20.4311
LDA	-18.894	-19.3302

Table 2: Log likelihoods across train and test sets

4.3 Predicted Matches

Figure 5 highlights sample queries and matches generated for each of the three models for that query. For these matches, Euclidean distance between points in the latent space was used as the condition for matching.

As seen in the top queries in Figure 5, LDA generates less reliable matches than NMF and RBM. This is especially seen in the example where the query top is a men's sweater (the first row), and LDA matches it with a long skirt, whereas RBM and NMF find a more compatible match by pairing it with sweatpants. For women's tops, however, it is seen that viable matches are predicted for all models. For example, the query of a white tank top could be matched with athletic shorts (LDA), or printed skirts (NMF and RBM). In general, the matches are more consistent for women's clothing than men's, perhaps because our dataset is imbalanced in terms of having a much higher quantity of women's clothes than men's clothes. If this matching algorithm were to be incorporated in a clothing recommendation application, a hard constraint could be set to only match men's clothing with men's clothing. We could not set this constraint because the clothes were not labeled with gender, and the attributes relating to material and cut could be applicable to both men's and women's clothing and so were not helpful in determining the gender of a clothing item.

Fig 5: Sample queries and their predicted matches

Figure 6 shows some query tops and their corresponding predicted and randomly-matched bottoms. This gives a sense of the baseline we are using in determining whether an outfit is good or bad. While the predicted bottoms are consistent with the style of the top in terms of color and style, the random bottoms are mismatched and therefore constitute a bad outfit (like the pink skirt paired with the blue blazer). Moreover for query 1, the predicted match perfectly matches the top with the bottom the model is wearing. We observed that it is harder to match clothing items like "blazers", "cardigans" because these items have fewer attributes and convey a weaker sense of style.

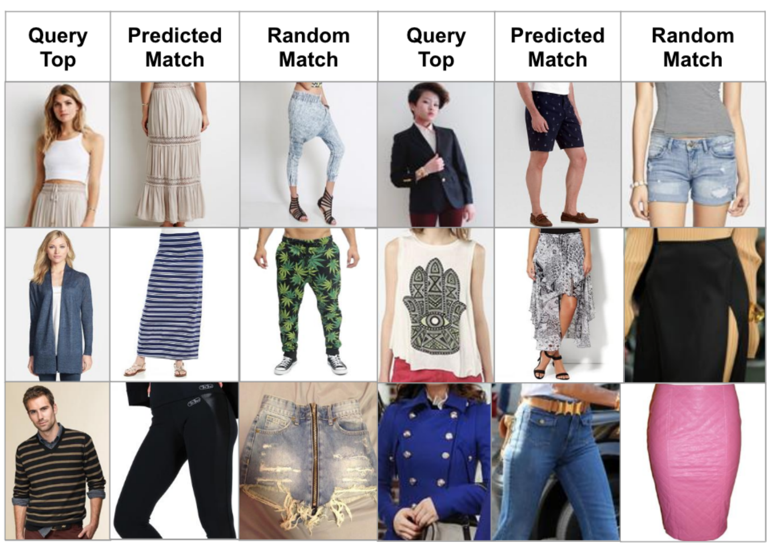

Fig 6: Predicted versus Random Matches

Table 3 highlights the performance of predicted matches against a baseline obtained by generating random matches between articles of clothing as described in Section 3.6. The predicted matches for tops have a higher ratio of good matches than the baseline - 0.64 versus 0.48. For the bottoms, however, the predicted matching performs slightly worse than the baseline. Our algorithm is not able to perform well for recommending matches for bottoms because it is harder to identify a latent styles in bottoms. Bottoms also have fewer attributes in general, making it difficult to predict matches based on similarities in style. For all the 100 clothing examples, our algorithm predicted good matches 58% of the time, and random matching worked around 50% of the time. It is important to keep in mind that these accuracies are preliminary evaluation on a very small subset of data. Nevertheless they indicate that outfit recommendations from a topic model has the potential to outperform a random baseline.

Matching	Ratio of good matches for tops	Ratio of good matches for bottoms	Ratio of good matches for all clothing items
Predicted Match	0.64	0.48	0.58
Random Match	0.48	0.5	0.49

Table 3: Comparison of our matching algorithm with random matching

5 Discussion

5.1 Conclusion

The main focus of this project was to analyze how well unsupervised topic modelling methods are able to discover latent styles for clothing using semantic visual attributes. Out of the three methods we used, NMF had the lowest reconstruction error, but we realize that this does not indicate that the other 2 topic modeling methods are not appropriate for modelling style. Interesting latent styles were discovered by “LDA” and “NMF” models, including “athleisure”, “professional”, “bohemian”, “casual” and “gothic” looks. The models effectively found relationships between attributes (for example, attributes like “cotton”, “pocket”, ”classic” were highly weighted for the topic resembling a “casual” look in Figure 3). The matches predicted by “finding the similarity in latent style(s)” between two clothing items using the L2 distance performed better than the baseline “random” matching algorithm.

5.2 Limitations

Because we used a dataset that was pre-annotated with attributes, it is not possible to use other clothes images with our models to determine their latent style embeddings. Therefore it makes it difficult for us to use other datasets with style labels (labeled by humans) to see how well our discovered styles align with human perceived styles. If we had more time and computing resources, we could have trained multiple semantic visual attribute classifiers using convolutional neural networks and segmentation networks using the image to attribute information in DeepFashion [6]. With these classifiers to predict attributes, our unsupervised methods would could be easily applied to any dataset of clothing images, saving significant manual effort.

Another limitation of our approach is that we only model style for single clothing items. The concept of gestalt is important for style. Individual items may not directly represent a style; instead it is the composition of multiple clothing items that defines a look/style. To incorporate this idea, we could apply topic modelling methods on a dataset of complete outfits (images of tops and bottoms). We would also have to localize attributes to include attributes such as *linen shirt* and *linen pants* instead of just including *linen*.

5.3 Future Work

There are many applications for accurate style-coherent representations of clothes images obtained using topic models. In this project we looked at basic recommendation of similar style images given a query image by finding nearest neighbors in the latent topic space. To determine better recommendations we could incorporate filters such as sex, clothing category, age, and price data.

Our approach to modelling style allows us to mix fashion styles based on user preferences. A user could identify some subset of styles that they prefer and query for outfits that are a blend of those styles. Another application for the latent style embeddings is determining “personal” style by summarizing the styles present in a person’s wardrobe.

Acknowledgments

Professor Barbara Engelhardt
Matt Myers

References

[1] Hsiao, Wei-Lin, and Kristen Grauman. ”Learning the latent look: Unsupervised discovery of a style-coherent embedding from fashion images.” 2017 IEEE International Conference on Computer Vision (ICCV). IEEE, 2017.

[2] M. H. Kiapour, X. Han, and S. Lazebnik. Where to buy it: Matching street clothing photos in online shops. In ICCV, 2015.

[3] M. H. Kiapour, K. Yamaguchi, A. Berg, and T. Berg. Hipster wars: Discovering elements of fashion styles. In ECCV, 2014.

[4] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. JMLR, 2003.

[5] Lee, Daniel D., and H. Sebastian Seung. ”Learning the parts of objects by non-negative matrix factorization.” Nature 401.6755 (1999): 788.

[6] http://mmlab.ie.cuhk.edu.hk/projects/DeepFashion.html

[7] https://fashion-gen.com

[8] https://github.com/xthan/polyvore-dataset

[9] https://scikit-learn.org/stable/index.html

6 Appendix

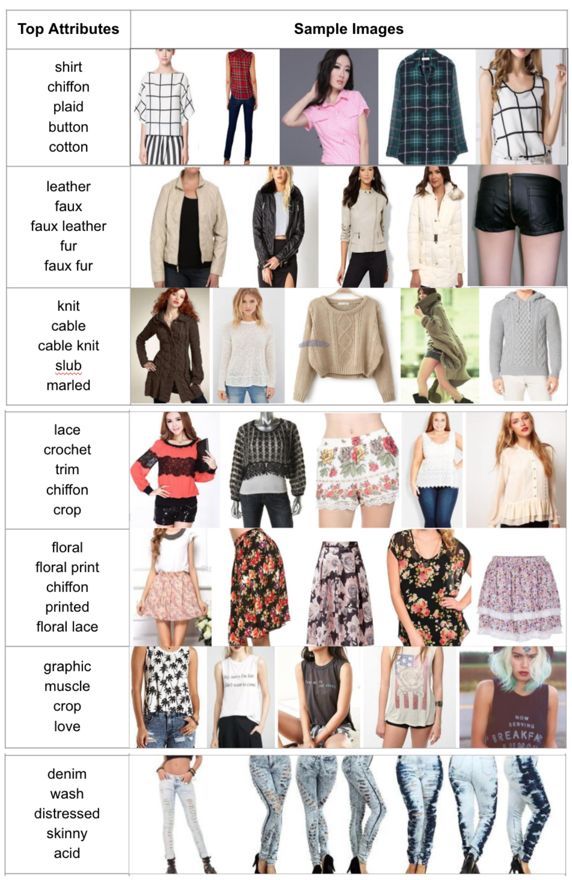

Fig 1: Other topics for NMF