Recommending Outfits from Personal Closet

Pongsate Tangseng*[1], Kota Yamaguchi[2], and Takayuki Okatani[1,3]

[1]Tohoku University, Sendai, Japan
[2]CyberAgent, Inc., Tokyo, Japan
[3]RIKEN Center for AIP, Tokyo, Japan

Abstract

We consider the outfit grading problem for outfit recommendation, where we assume that users have a closet of items and we aim at producing a score for an arbitrary combination of items in the closet. The challenge in outfit grading is that the input to the system is a bag of item pictures that are unordered and vary in size. We build a deep neural network-based system that can take variable-length items and predict a score. We collect a large number of outfits from a popular fashion sharing website, Polyvore, and evaluate the performance of our grading system. We compare our model with a random-choice baseline. The performance of our model achieves 84% in both accuracy and precision, showing our model can reliably grade the quality of an outfit. We also built an outfit recommender on top of our grader to demonstrate the practical application of our model for a personal closet assistant.

1. Introduction

There have been growing interests in applying computer vision to fashion, perhaps due to the rapid advancement in computer vision research [2, 6, 8, 13, 14, 18, 20–23]. One of the popular fashion application is item recommendation [3, 4, 7, 9], where the objective is to suggest items to users based on user's and/or society's preference. Computer vision is used in various fashion applications such as e-commerce and social media. Recently, Amazon announced their automatic style assistant called "Echo Look™". Although the underlying mechanism is not published, emerging commercial applications confirm the ripe of computer vision applications in fashion.

Measuring the quality of outfit is essential in building fashion recommendation system. In this paper, we consider the problem to grade arbitrary combination of items (Fig 1). Outfit recognition has been studied in the past. Previous works in outfit evaluation can be divided into two groups based on the input format: an outfit as a full-body image as in [6, 15, 16, 24], as a set of images of items [3, 7], or a combination of both [9]. However, outfits can have arbitrary numbers of items. For examples, in some day, one might prefer a combination of a jacket, a t-shirt, and jeans, while in the other day she might want to wear only a dress. A machine learning system should be able to accept variable numbers of items yet produce a consistent score for any size of combinations.

Figure 1: Given an arbitrary number of items, our goal is to evaluate the quality of the outfit combination.

In this paper, we consider an outfit as a bag of fashion items and utilize deep neural networks to produce a score for a fixed-length representation of outfits. Our outfit representation is an ordered concatenation of item embeddings from convolutional networks, and we express non-existent items by mean item representation to deal with variable length input. Unlike style recognition [2, 6, 16, 24], we take item images in isolation, not worn items, as seen on e-commerce sites or catalogs. For evaluating our model, we collect a large number of outfit data from a popular fashion website, Polyvore. Our Polyvore409k dataset consists of 409,776 sets of clothing items from 644,192 unique items. The dataset forms a large bipartite graph of items and outfits. For evaluation, we develop an approach to partition the dataset into training and testing sets such that there is no overlapping nodes and edges between the sets. Last but not least, we also build an outfit recommender that takes clothing items as the input and recommends best outfits from

*tangseng@vision.is.tohoku.ac.jp

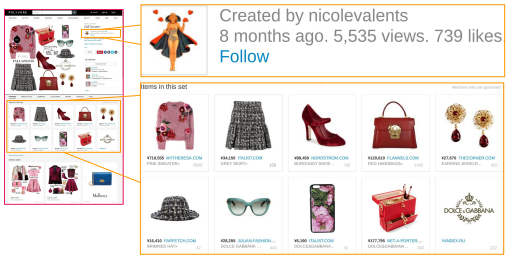


Figure 2: Example of Polyvore set and items.

those items to show the usefulness of the system in solving a real-world problem. We summarize our contributions in the following:

1. We build Polyvore409k dataset containing 409,776 outfits and 644,192 items. Every outfit covers the entire body with variable numbers of items.
2. We propose an outfit grader that produces a score for fashion outfits with a variable number of items. Our empirical study shows that the grader achieves 84% of accuracy and precision in Polyvore409k dataset.
3. We demonstrate that our outfit grader can build a recommendation system that suggests good outfits from a pool of items.

2. Polyvore409k dataset

There have been several datasets proposed for fashion applications in the past. Some have multiple items as an outfit in one image [8,15,21,23], others have single clothing item per image [3, 4, 7, 12], or combination of both [9, 10]. Although [3, 7, 17] used dataset with combinations of images as outfits and each item has its own image, the datasets are not publicly available. Therefore, we decide to collect suitable outfit data.

We collect Polyvore409k dataset from the fashion-based social media website `polyvore.com`. Each outfit, or "set" in Polyvore's terminology, consists of a title, items in the set, a composed image, and behavioral data such as likes and comments from other users. Fig 2 shows a Polyvore set and items. We plan to release the dataset to the public.

Data cleansing The sets sometimes contain non-clothing items. We trim items not containing one of 121 clothing terms such as jacket in the name, and also categorize items into one of six parts: outer, upper-body, lower-body, full-body, feet, or accessory. We collect only sets that contain at least one clothing item. We crawl 576,653 unique clothing sets with 969,071 items. From the collected data, we identify outfits that have at most one item per category (e.g., two dresses are rejected) with an exception for accessories that can appear at most three times. Outfits that do not cover the entire body, e.g. missing lower body, are removed. After filtering, we obtain 409,776 valid outfits consisting of 644,192 unique items. There are 212,623 outfits that have at least one like in Polyvore409k, which we use as positive samples.

Algorithm 1: Disjoint Set Sampling

input : All outfits $\mathbf{O}$
output: Set A, B, and C containing outfits such that items in outfits in A is not in B and vice versa

$A \leftarrow B \leftarrow C \leftarrow \emptyset$;
$A \cup \{\mathbf{O}_0\}$;
for $i \leftarrow 1$ **to** $|\mathbf{O}|$ **do**
  $O \leftarrow \mathbf{O}_i$;
  $items_A \leftarrow$ items in outfits in A;
  $items_B \leftarrow$ items in outfits in B;
  $items_O \leftarrow$ items in O;
  $sec_{AO} \leftarrow intersection(items_A, items_O)$;
  $sec_{BO} \leftarrow intersection(items_B, items_O)$;
  if $|sec_{AO}| > 0$ *and* $|sec_{BO}| > 0$ **then** $C \cup \{O\}$;
  else if $|sec_{AO}| > 0$ **then** $A \cup \{O\}$;
  else if $|sec_{BO}| > 0$ **then** $B \cup \{O\}$;
  else
    if $|A|/2 > |B|$ **then** $B \cup \{O\}$;
    else $A \cup \{O\}$;
  end
end

Table 1: Number of unique items in each outfit part

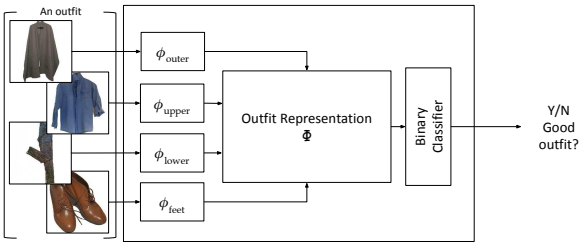
Part	Outer	Upper	Lower	Full	Feet	Accessory
Train	11,168	21,760	16,287	11,523	26,574	60,760
Test	6,656	12,744	11,089	8,871	17,564	37,988

Table 2: Number of samples in train and test partition

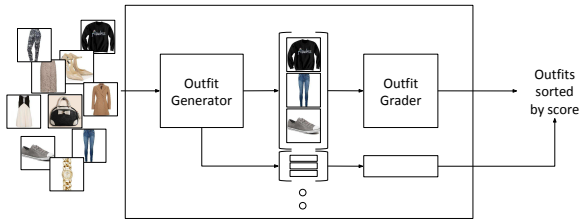
Number of samples	Train	Test
Positive samples	66,434	26,813
Negative samples	132,868	53,626
Total samples	199,302	80,439
Positive negative ratio	1:2	1:2
Average items per sample	4.75	4.48

Sampling evaluation data from a bipartite graph The set-item relationship constitutes a bipartite graph, where nodes are outfits or items, and edges represent inclusion relationship. For performance evaluation using Polyvore409k, we have to split the bipartite graph such that the training and testing splits do not share any item or outfits. We use Algorithm 1 to separate training and testing splits.

Negative samples We create negative samples as follows. For each positive sample, we create two identical samples as negative samples. Then, we replace items in those two negative samples with random items of the same parts from the same train/test item pool. This method guarantees the disjoint set property between training and testing sets. It



(a) Outfit Grader



(b) Outfit recommender

Figure 3: Overview of the proposed system

also prevents artificial bias between positive and negative samples in term of outfit composition, because the distribution of number of items and existences of outfit parts in samples are preserved. Table 1 shows the number of items in each outfit part. Table 2 shows the numbers of positive and negative samples in each split.

3. Outfit grading and recommendation

We formulate outfit grading as a binary classification problem. Given an outfit $O \equiv \{x_{\text{outer}}, x_{\text{upper}}, \cdots, x_{\text{accessory3}}\}$, where x_{part} is an item image, the goal is to learn a mapping function: $F : O \mapsto y$ to predict the outfit validity $y \in \{0, 1\}$. Once we learn the mapping F, we are able to sort arbitrary combinations of items according to the prediction score for the recommendation.

The challenge is how to represent an outfit O with a variable number of items. Luckily, the number of visible items is limited even though an outfit can contain a variable number of items. Therefore, we assign items into one of the six categories and concatenate the item representations to produce the outfit representation. Fig 3 shows our grader and recommender. Our grader takes a bag of images and convert them to feature representations, then concatenates the individual features according to the item's category to produce the fixed-length representation. We describe details below.

Item representation We convert the image of each item in the outfit to a feature representation $\phi_{\text{part}}(x_{\text{part}})$, using a convolutional network. In this paper, we use ImageNet-pretrained ResNet-50 [1], and extract the 2048 dimensional embedding from "pool 5" layer as an item representation. We extract features for 5 item parts and up to 3 accessories. For missing parts, we give a mean image to obtain features.

Outfit representation After we extract features from each item, we concatenate all features in the fixed order to form an outfit representation $\Phi(O) \equiv [\phi_{\text{outer}}, \phi_{\text{upper}}, \cdots, \phi_{\text{accessory2}}]$. Note that we allow accessories to appear multiple times in the outfit, and we simply concatenate all the accessory features ignoring the order. Outfits with less than 3 accessories get mean images as well to the other part. We have 5 item parts and 3 accessories per outfit, resulting in a 16,384 dimensional representation as the outfit representation.

Scoring outfits From the outfit representation Φ, we learn a binary classifier and predict a score. We utilize a multi-layer perceptron (MLP) to learn the mapping function. In this paper, we use an MLP consisting of one 4096-d linear layer followed by batch normalization and rectified linear activation (ReLU) with dropout, and 2-d linear layer followed by soft-max activation to predict a score. We use multinomial logistic loss to learn the parameters of the grading model.

4. Experiments

4.1. Experimental setting

We learn the grading model from the training split of Polyvore409k dataset, and evaluate the binary classification measures (accuracy and precision) on the testing split. The performance is measured against the ground truth. In this paper, we report the performance of our model without fine-tuning the parameters of the convolutional network for the item feature extraction. We implement the neural network using Caffe framework [5]. We train the model for 400,000 iterations using stochastic gradient descent with momentum, where the initial learning rate and momentum are set to 10^{-4} and 0.9, respectively.

4.2. Grading performance

The model achieves 84.51% accuracy and 83.66% in precision in Polyvore409k. Given the positive–negative ratio of 1:2, the result suggests that our model can successfully separate randomly created, machine-made outfits (negative samples) from human-made outfits (positive samples). Fig 4 shows top 4 positive and negative samples from our predictions. Qualitatively, preferred outfits contain consistent colors between items, whereas low-scoring outfits tend to have less common visual elements between items.

4.3. Outfit recommendation

We build a prototype system to make an outfit recommendation from a personal closet based on our grader. We

Figure 4: Four best (top row) and worst (bottom row) outfits judged by our outfit grader

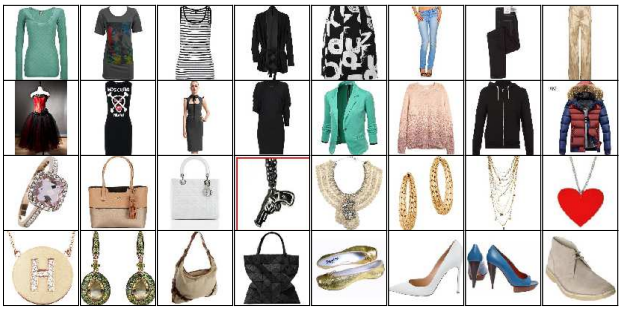


Figure 5: Items used in outfit recommendation experiments

randomly select 4 items per outfit part from items in testing partition of the dataset. The selected items are shown in figure 5. We create outfits in 3 configurations: (1) outer layer with upper- and lower-body, (2) only upper- and lower-body, (3) full-body only. Each of these configurations also includes feet part and up to 3 accessories. We randomly create outfits according of each configuration from items in the item pool, then we use our outfit grader to grade the outfits. Figure 6 shows four recommended outfits for each configuration based on the score from our model. Although formal human evaluation is still in progress, our preliminary human evaluation indicates strong preference for our recommendation compared to a random baseline.

5. Conclusion and future works

We study the outfit grading problem to build a personal outfit recommendation application. We collect a large clothing dataset that consists of over 600K clothing items and

(a) Outer + upper + lower + feet + accessories

(b) Upper + lower + feet + accessories

(c) Full + feet + accessories

Figure 6: Recommended outfits for each configuration.

over 400K outfits, and use the dataset to learn and evaluate the outfit grader and recommender. Given a variable-number of items as an outfit, our outfit grader can give a score if the outfit looks good or not. Our experimental result shows that our model achieves at over 84% accuracy on testing samples. In addition, we demonstrate that our system can recommend outfits from user's personal collection of items in a closet.

In this work, we consider rather a simple concatenation approach to obtain a fixed-length representation for outfits. We wish to improve the model architecture by, for example, end-to-end learning, recurrent networks to model unordered bag of items, or introducing adversarial training to learn a recommendation model. For recommendation, an outfit generator would need to pick items from collections rather than generating pixels as commonly seen in generative adversarial networks [11]. Learning a set generator using deep neural networks seems a challenging topic to investigate in the future [19]. We also wish to design a thorough evaluation framework using human judgment in crowdsourcing.

Acknowledgements

This work was supported by JSPS KAKENHI JP15H05919 and JP16H05863.

References

[1] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778. IEEE, 2016. 3

[2] W.-L. Hsiao and K. Grauman. Learning the Latent "Look": Unsupervised Discovery of a Style-Coherent Embedding from Fashion Images. *arXiv*, 2017. 1

[3] Y. Hu, X. Yi, and L. S. Davis. Collaborative fashion recommendation: a functional tensor factorization approach. In *MM*, pages 129–138. ACM, 2015. 1, 2

[4] J. Huang, R. S. Feris, Q. Chen, and S. Yan. Cross-domain image retrieval with a dual attribute-aware ranking network. In *ICCV*, pages 1062–1070. IEEE, 2015. 1, 2

[5] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *MM*, pages 675–678. ACM, 2014. 3

[6] M. H. Kiapour, K. Yamaguchi, A. C. Berg, and T. L. Berg. Hipster wars: Discovering elements of fashion styles. In *ECCV*, pages 472–488. Springer, 2014. 1

[7] Y. Li, L. Cao, J. Zhu, and J. Luo. Mining fashion outfit composition using an end-to-end deep learning approach on set data. *TMM*, 2017. 1, 2

[8] S. Liu, J. Feng, C. Domokos, H. Xu, J. Huang, Z. Hu, and S. Yan. Fashion parsing with weak color-category labels. *TMM*, 16(1):253–265, 2014. 1, 2

[9] S. Liu, J. Feng, Z. Song, T. Zhang, H. Lu, C. Xu, and S. Yan. Hi, magic closet, tell me what to wear! In *MM*, pages 619–628. ACM, 2012. 1, 2

[10] Z. Liu, P. Luo, S. Qiu, X. Wang, and X. Tang. Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In *CVPR*, pages 1096–1104. IEEE, 2016. 2

[11] L. Ma, X. Jia, Q. Sun, B. Schiele, T. Tuytelaars, and L. Van Gool. Pose guided person image generation. *arXiv*, 2017. 4

[12] J. McAuley, C. Targett, Q. Shi, and A. Van Den Hengel. Image-based recommendations on styles and substitutes. In *SIGIR*, pages 43–52. ACM, 2015. 2

[13] J. Oramas and T. Tuytelaars. Modeling Visual Compatibility through Hierarchical Mid-level Elements. *arXiv*, 2016. 1

[14] E. Simo-Serra, S. Fidler, F. Moreno-Noguer, and R. Urtasun. A high performance crf model for clothes parsing. In *ACCV*, pages 64–81. Springer, 2014. 1

[15] E. Simo-Serra, S. Fidler, F. Moreno-Noguer, and R. Urtasun. Neuroaesthetics in fashion: Modeling the perception of fashionability. In *CVPR*, pages 869–877. IEEE, 2015. 1, 2

[16] E. Simo-Serra and H. Ishikawa. Fashion style in 128 floats: joint ranking and classification using weak data for feature extraction. In *CVPR*, pages 298–307. IEEE, 2016. 1

[17] K. Vaccaro, S. Shivakumar, Z. Ding, K. Karahalios, and R. Kumar. The Elements of Fashion Style. *UIST*, 2016. 2

[18] A. Veit, B. Kovacs, S. Bell, J. McAuley, K. Bala, and S. Belongie. Learning Visual Clothing Style with Heterogeneous Dyadic Co-occurrences. *ICCV*, page 27, 2015. 1

[19] O. Vinyals, S. Bengio, and M. Kudlur. Order matters: Sequence to sequence for sets. *arXiv*, 2015. 4

[20] S. Vittayakorn, K. Yamaguchi, A. C. Berg, and T. L. Berg. Runway to realway: Visual analysis of fashion. In *WACV*, pages 951–958. IEEE, 2015. 1

[21] K. Yamaguchi, M. Hadi Kiapour, and T. L. Berg. Paper doll parsing: Retrieving similar styles to parse clothing items. In *ICCV*, pages 3519–3526. IEEE, 2013. 1, 2

[22] K. Yamaguchi, M. H. Kiapour, L. E. Ortiz, and T. L. Berg. Parsing clothing in fashion photographs. In *CVPR*, pages 3570–3577. IEEE, 2012. 1

[23] W. Yang, P. Luo, and L. Lin. Clothing co-parsing by joint image segmentation and labeling. In *CVPR*, pages 3182–3189. IEEE, 2014. 1, 2

[24] L.-F. Yu, S. K. Yeung, D. Terzopoulos, and T. F. Chan. Dressup!: outfit synthesis through automatic optimization. *TOG*, 31(6):134, 2012. 1