2022

Fashion Compatibility Prediction Using Ensemble Learning

Nathan Utzman
nrutzman@mix.wvu.edu

Fashion Compatibility Prediction Using Ensemble Learning

Nathan Utzman

Thesis submitted to the
Benjamin M. Statler College of Engineering and Mineral Resources
at West Virginia University
in partial fulfillment of the requirements for the degree of

Master of Science
In
Electrical Engineering

Xin Li, Ph.D., Chair
Bin Liu, Ph.D.
Natalia Schmid, Ph.D.

Lane Department of Computer Science and Electrical Engineering

Morgantown, West Virginia
2022

Keywords: Fashion Compatibility, Ensemble Learning, Deep Learning

ABSTRACT

Fashion Compatibility Prediction Using Ensemble Learning

Nathan Utzman

Fashion is important both financially and for self-expression. There are many tasks in the fashion domain which can be addressed with artificial intelligence. The task of fashion compatibility prediction is to determine how well a set of items work together to form an outfit. Two main tasks are typically used to evaluate the performance of a fashion compatibility prediction model – Outfit Compatibility Prediction and Fill in the Blank.

In this work, a compatibility prediction model, which is based on the graph autoencoder, is evaluated. This same model is then used in a homogeneous ensemble learning approach, proposed to improve the compatibility prediction performance. This ensemble learning approach does not outperform the baseline. Finally, several potential approaches are introduced which may be of interest to future researchers.

Acknowledgements

I would like to thank my research advisor and committee chair Dr. Xin Li for his patience, guidance, and support throughout my studies. I would also like to thank Dr. Bin Liu for his frequent advice as my research co-advisor and committee member. I am also grateful to Dr. Natalia Schmid for her advice and feedback as a committee member.

I would like to thank my parents, family, and friends for their persistent support. Finally, I would like to thank my fiancée Caroline for her love and kindness, and our cat Eloise for all of the help studying.

Contents

List of Figures

List of Tables

Chapter 1
Introduction

1.1 Motivation

Fashion is an important form of self-expression. It is also important financially, with a global market of more than $3 trillion and about 2% of the world's Gross Domestic Product. [1] describes many tasks in the fashion domain which have been addressed using artificial intelligence methods.

1.1.1 Fashion Domain Tasks

1.1.1.1 Fashion Detection

Fashion detection tasks deal with identifying fashion items in a given image. The task of landmark detection seeks to detect key points of fashion items. This is similar to the task of the same name in biometrics applications and pose estimation, but fashion items have a less predictable structure than the face and body. Landmark detection is useful for image segmentation. The goal of fashion parsing is to label the fashion item by category (e.g. skirt, t-shirt, sandals). Item retrieval attempts to find visually similar items within a dataset, given an item of interest.

1.1.1.2 Fashion Analysis

The task of attribute recognition is to assign labels to a fashion item. For example, an item may have the color label "red", the material label "cotton", and the pattern label "checkered". Style learning is more general, and predicts which style an item belongs to, such as casual or hipster. The final task in this group is popularity prediction. This task predicts future fashion trends based on sources such as social media and blogs.

1.1.1.3 Fashion Synthesis

Tasks in the fashion synthesis group augment or create entirely new images of fashion items. The style transfer task is used to virtually try-on fashion items and makeup. In pose transformation, an image of a person wearing given fashion items is augmented such that the subject is in a desired target pose. For example, an image of the subject sitting may be augmented such that they are standing with their arms crossed. Finally, physical simulation predicts the dynamic movement of a fashion item in relation to the subject's body.

1.1.1.4 Fashion Recommendation

The goal of tasks in this group is to make quality recommendations to users. Personalized recommendation suggests outfits to users based on their previous interactions or preferences. Hairstyle suggestion uses facial features to suggest a hairstyle for a user. Finally, fashion compatibility prediction scores how well items work together to form an outfit.

1.2 Problem Statement

In this work, we focus on the task of compatibility prediction (briefly described in 1.1.1.4). In this task, an outfit is defined as an unordered collection of fashion items. There are two main tasks which are used to evaluate compatibility prediction models, described in the following sections.

1.2.1 Compatibility Prediction

An outfit is presented to the recommender system. The task is to predict the compatibility of the items as a whole, with compatibility scores ranging from 0 to 1. A more compatible outfit will have a higher score. Figure 1.1 shows a sample outfit with its compatibility score. Since fashion is subjective, compatible outfits are based on datasets which are created with some user interaction with outfits. For example, the dataset gathered in [2] is based on user click actions with items and outfits, and the Amazon dataset [3] is based on product recommendations to users.

Figure 1.1: A sample outfit from the POG dataset with an example of a compatibility score.

1.2.2 Fill in the Blank (FITB)

In this task, one item from the outfit is removed. The remaining items in the outfit are presented to the recommender system as an incomplete outfit. The removed item is then shuffled with a number of items randomly selected from the dataset. Following [4], we use 3 randomly selected items, resulting in 4 candidate items for each FITB question. The system must then correctly select the removed item from the candidate items.

1.2.3 Increasing Task Difficulty

There exist two modifications which increase the difficulty of the fashion compatibility prediction tasks described above.

1.2.3.1 Resampled Dataset

In many datasets used for compatibility prediction, each item is given a category identification number which corresponds to a category of clothing item (e.g. skirts, t-shirts, sandals). The *resampled* modification proposed by [5] affects the two tasks slightly differently.

For the compatibility prediction task, the outfit is modified to ensure that it contains no two items from the same category. In the FITB task, candidate items are selected from items with the same category label as the removed item.

1.2.3.2 Subset Dataset

The *subset* modification was proposed in [4]. Each outfit is sampled to contain only 3 clothing items. This makes both tasks more difficult because the system has less information to make its decision.

1.3 Contributions

In this work, we attempt to improve performance on the compatibility prediction task.

1. We reproduce and verify the results published in [4].
2. We propose a decision-level ensemble learning approach using the model described in [4] as a baseline.
3. We briefly explore a second potential ensemble learning approach using [6] to initialize the feature vectors used in [4].

1.4 Outline

In chapter 2, we explore our proposed approach using homogeneous ensemble learning at the decision level. We briefly explore the baseline model and outline our proposed approach to improve compatibility prediction performance.

In chapter 3, we explore areas which may be of interest to future researchers in the area of compatibility prediction. In particular, we explore a potential approach for compatibility prediction using a heterogeneous ensemble approach.

In chapter 4, we summarize the conclusions of our work.

Chapter 2

Decision-Level Homogeneous Ensemble Learning

2.1 Introduction to Ensemble Learning

Ensemble learning is the combination of multiple models to improve prediction performance. A homogeneous ensemble is constructed of component models which share the same architecture. In a heterogeneous ensemble, the component models have different architectures.

There are many techniques used for ensemble learning, including bagging, boosting, and stacking. However, we will be focusing on decision-level fusion. In this technique, the outputs of the component models are combined such that the component models do not interact until their decisions have been made. There are several approaches for combining model decisions.

Unweighted averaging is one of the most popular approaches and can be applied to either the outputs of the component models or the predicted probabilities of the classes. Majority voting is similar to unweighted averaging, but instead treats each model output as a vote. The decision with the majority of votes is selected as the ensemble's choice. [7]

2.2 Related Work

[8] proposes a session-based personalized recommendation approach which combines a global and local recommender. The global recommender captures the user's overall preference, while the local recommender selects items which match the user's current intention.

[9] proposes a method for the task of item retrieval while ensuring outfit compatibility. This method generates an outfit based on a target sentence. The items which are returned match the query while also constructing a compatible outfit.

[10] introduces multiple approaches for the compatibility prediction task. Fashion item type, context, and style are utilized to predict compatibility scores.

2.3 Methods and Procedure

2.3.1 Datasets

Datasets for fashion compatibility prediction generally have a number of common attributes. The core of the dataset in most cases is a list of outfits, which are composed of a number of fashion items. Each outfit has an identifier, as does each item. Additionally, each item has an image and a category (e.g. skirt, t-shirt, blazer).

2.3.1.1 Polyvore Dataset

The Polyvore dataset [11] is a frequently used dataset for fashion compatibility prediction, created from data on Polyvore.com. It contains 21,889 outfits, which are split for training (17,316), validation (1,497), and testing (3,076). Each outfit has a unique identification number and a collection of items. Each item has a unique identification number, a category identification number, and an image. In the Polyvore training set, each outfit has between 4 and 8 items, with an average of 6.63 items per outfit.

2.3.1.2 Personalized Outfit Generation (POG) Dataset

The POG dataset, created by Chen *et al.* [2], contains over 1 million outfits. Similar to the Polyvore dataset, the POG dataset provides outfits as collections of items with unique identification numbers, item categories, and item images. In addition, the POG dataset includes user-outfit and user-item interaction data. This user data is useful for the task of personalized recommendations, but we will not be taking advantage of it. Each outfit in the POG dataset has between 4 and 9 items, with an average of 4.27 items.

Figure 2.1: A sample outfit from the POG dataset.

2.3.2 Baseline Model

For this approach, we begin by verifying the published results of the CAVCP model [4], particularly on the FITB task. The CAVCP model is based on the graph-autoencoder and uses a graph to represent item-item relationships. Each node in the graph contains a 2048-dimensional feature vector representing the node item. These feature vectors are initialized by passing item images through a ResNet-50 [12] which was pre-trained on the ImageNet dataset [13].

The encoder of the CAVCP model is implemented as a Graph Convolutional Network, which aggregates information from adjacent nodes. The decoder is a distance function whose weights are learned during training.

During evaluation of the CAVCP model, a variable *k* is introduced to control the amount of contextual information that is utilized from neighboring nodes. When *k* is 0, the item embedding is based only on that item's features. As *k* increases, the information from more neighborhood nodes is incorporated into the item embedding. The *k*-neighborhood of a given node is the set of *k* nodes around the node found by a breadth-first-search.

For this work, the CAVCP model was trained on the Polyvore dataset with the hyperparameters set to the default published values. To verify the published results, the trained model was evaluated on the Polyvore test dataset. Additionally, the model was evaluated with larger values of *k* to investigate the expected performance increase.

2.3.3 Decision-Level Homogeneous Ensemble Learning

In this approach, we propose a homogeneous ensemble learning approach using multiple instances of the CAVCP model and majority voting to improve FITB task performance. The nature of the FITB task allows decision-level majority voting to conveniently take place. The trained model from the baseline approach is reused, but the value of *k* is varied during evaluation for each voting model. The FITB predictions are stored for each voting model, then the ensemble prediction is determined.

During evaluation, the final prediction was determined to be correct if more than half of the models in the ensemble correctly predicted the answer. Selecting an odd number of models ensured that there was no tie between the votes of the component models.

2.4 Experimental Results

Table 2.1: Verifying CAVCP published results evaluated on the Polyvore dataset.

	Fill in the Blank		Compatibility Prediction	
	Original	Resampled	Original	Resampled
k=0	0.6638	0.5010	0.8530	0.7688
k=3	0.9574	0.9272	0.9961	0.9892
k=15	**0.9685**	**0.9408**	**0.9983**	**0.9948**
k=0 (subset)	0.6167	0.4802	0.6859	0.6389
k=3 (subset)	0.7942	0.6886	0.8862	**0.9881**
k=15 (subset)	**0.8859**	**0.8202**	**0.8926**	0.8873

Table 2.2: FITB Performance with large *k* values using the baseline CAVCP model.

	Polyvore	POG
k=0	0.4802	0.3156
k=1	0.5881	0.4319
k=3	0.6886	0.4581
k=7	0.7207	0.4656
k=15	0.8202	0.4625
k=20	0.8443	0.4697
k=30	0.8664	0.4641
k=40	0.8693	0.4706
k=50	**0.8700**	**0.4725**
k=100	0.8696	0.4700

Table 2.3: Majority Voting results evaluated on the Polyvore dataset.

	Original	Resampled	Subset	Resampled & Subset
k=0	0.6638	0.5010	0.6167	0.4802
k=1	0.9269	0.8817	0.6847	0.5881
k=3	0.9574	0.9272	0.7942	0.6886
k=7	0.9613	0.9298	0.8173	0.7207
k=15	**0.9685**	**0.9408**	**0.8859**	**0.8202**
k=[1, 7, 15]	0.9633	0.9356	0.8345	0.7500
k=[0, 1, 3, 7, 15]	0.9551	0.9233	0.8046	0.7035

Table 2.4: Majority Voting results evaluated on the POG dataset.

	Resampled & Subset
k=0	0.3156
k=1	0.4319
k=3	0.4581
k=7	0.4625
k=15	**0.4656**
k=[1, 7, 15]	0.4547
k=[0, 1, 3, 7, 15]	0.4341

Table 2.1 shows that our baseline model trained on the Polyvore dataset performed similarly to the published results in [CAVCP] were confirmed. Our results also confirm that the CAVCP model performance in both tasks improves as the neighborhood size *k* increases. The only exception to this was in the compatibility prediction task with both resampling and subset modifications, but this did not appear as a pattern in any of our other results.

Table 2.2 shows the baseline CAVCP performance as *k* continues to increase. It is clear that the prediction score increases the most when *k* is small. As *k* gets large, the marginal increase in performance trails off. In our results, the *k* with the highest performance in both datasets is *k*=50. More evaluation is needed to determine if performance begins to decrease at a certain value of *k*.

Tables 2.3 and 2.4 show that the proposed Majority Voting approach was unable to outperform the baseline models which it consists of. Across both datasets, the model with the best FITB performance was the baseline model with the highest *k* value.

In addition, Table 2.4 shows that the CAVCP baseline model trained on the Polyvore dataset did not generalize well to the POG dataset. This may be due to the amount of time between the gathering of the two datasets. The Polyvore dataset was released in 2017, while the POG dataset was released in 2019.

2.5 Summary

In this chapter, we investigated the CAVCP model for compatibility prediction and verified the published results. We proposed a homogeneous ensemble learning approach which was unable to outperform the baseline.

Chapter 3

Future Work

There is still room for improvement on the compatibility prediction task. In this chapter, we will discuss several ideas that may be explored further to achieve this improvement. In particular, we will discuss a different ensemble learning approach to improve the CAVCP model by initializing the input feature vectors using a different model.

3.1 Self-Supervised Learning Feature Initialization

3.1.1 Learning Type-Aware Embeddings for Fashion Compatibility

The SVAL model, proposed in [6], addresses the problem of fashion compatibility prediction using self-supervised learning and contrastive loss. This work shows that fashion compatibility requires a model to learn color and texture features, and ignore features based on item shape. To achieve this, SVAL uses three pretext tasks to learn features more useful for compatibility prediction: histogram prediction, shapeless local patch discrimination, and texture discrimination.

3.1.2 Proposed Methods

In this section, we will describe a partially explored approach to improve compatibility prediction by combining the SVAL and CAVCP models. The CAVCP model initializes the representation for each fashion item using a ResNet-50 model [12] pretrained on ImageNet [13]. However, this pretrained model was designed for object recognition and classification. Therefore, we proposed to use the SVAL model to initialize the feature vectors instead.

To achieve this, the SVAL model would be trained on the Polyvore dataset as described in [6]. This trained model would then be used to create the initial feature vectors for each fashion item in the dataset.

Using the newly created fashion item representations, the CAVCP model could be trained and evaluated using the same procedure as for the baseline CAVCP model. This approach could

be evaluated on both the Polyvore and POG datasets, and both the Compatibility Prediction and Fill in the Blank tasks.

3.1.3 Preliminary Results

The SVAL portion of the proposed model was trained on the Polyvore dataset, and the feature vectors were created to pass to the CAVCP portion. The CAVCP portion of the model was then trained and evaluated using those feature vectors. During training, the model performed similarly to the baseline. However, during evaluation, the resulting scores were similar, if not slightly lower, than those from the baseline (shown in Table 2.1). Due to time constraints, the quantitative training and evaluation performance of this model is unavailable currently. We hope to obtain results after the publication of this document.

3.1.4 Summary

Although no conclusive results are available, it is believed that further fine-tuning of the SVAL portion of the model may be able to improve fashion compatibility prediction performance when compared to the CAVCP baseline. It was demonstrated that the addition of the SVAL model to the input pipeline of the CAVCP model is possible. Further exploration of the SVAL model is needed before this proposed approach should be disregarded.

3.2 Fashion Task Combination

Personalized recommendation, closely related to compatibility prediction, is a task in which user preferences and past actions are used to recommend fashion items or outfits to an individual user. [14] combines these two tasks by using a hierarchical graph to represent fashion items, outfits, and users. In this approach, the latent representations for outfits are propagated from the relevant item representations. Similarly, the latent user representations are propagated from relevant outfit representations. It may be possible to combine other tasks in the fashion domain in order to improve performance.

Chapter 4

Conclusions

In this work, we introduced the many tasks in the fashion domain which can be addressed with artificial intelligence. An ensemble learning model was proposed to improve fashion compatibility prediction performance. The Decision-Level Ensemble Learning approach was not able to outperform the baseline, but other ensemble learning methods may be viable. Several other related problems and approaches were also briefly discussed.

References

[1] W. H. Cheng, S. Song, C. Y. Chen, S. C. Hidayati, and J. Liu, “Fashion meets computer vision: A survey,” *arXiv [cs.CV]*, 2020.

[2] W. Chen *et al.*, “POG: Personalized outfit generation for fashion recommendation at alibaba iFashion,” in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining - KDD ’19*, 2019.

[3] J. McAuley, C. Targett, Q. Shi, and A. van den Hengel, “Image-based recommendations on styles and substitutes,” in *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2015.

[4] G. Cucurull, P. Taslakian, and D. Vazquez, “Context-aware visual compatibility prediction,” in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[5] M. I. Vasileva, B. A. Plummer, K. Dusad, S. Rajpal, R. Kumar, and D. Forsyth, “Learning type-aware embeddings for fashion compatibility,” *arXiv [cs.CV]*, 2018.

[6] D. Kim, K. Saito, S. Mishra, S. Sclaroff, K. Saenko, and B. A. Plummer, “Self-supervised visual attribute learning for fashion compatibility,” *arXiv [cs.CV]*, 2020.

[7] M. A. Ganaie, M. Hu, A. K. Malik, M. Tanveer, and P. N. Suganthan, “Ensemble deep learning: A review,” *arXiv [cs.LG]*, 2021.

[8] J. Li, P. Ren, Z. Chen, Z. Ren, T. Lian, and J. Ma, “Neural Attentive Session-based Recommendation,” in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, 2017.

[9] K. Li, C. Liu, and D. Forsyth, “Coherent and controllable outfit generation,” *arXiv [cs.CV]*, 2019.

[10] A. Singhal, A. Chopra, K. Ayush, U. Patel, and B. Krishnamurthy, “Towards a unified framework for visual compatibility prediction,” in *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2020.

[11] X. Han, Z. Wu, Y. G. Jiang, and L. S. Davis, “Learning fashion compatibility with bidirectional LSTMs,” in *Proceedings of the 25th ACM international conference on Multimedia*, 2017.

[12] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[13] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and L. Fei-Fei, “ImageNet: A large-scale hierarchical image database,” in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[14] X. Li, X. Wang, X. He, L. Chen, J. Xiao, and T. S. Chua, “Hierarchical fashion graph network for personalized outfit recommendation,” in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2020.