Interactive Machine Learning Framework for Outfit Scoring System

Dana Aubakirova, Dorin Doncenco, Zhe Huang, Heidi Sainte-Catherine

***Abstract*—Machine learning algorithms have the potential to revolutionize the fashion industry, improving efficiency, sustainability, and customer experience. However, the traditional learning process involves training algorithm-specific models with a substantial amount of data, and with little or no feedback from the user. Such a "big data" based strategy is sometimes unrealistic for applications where processing is very expensive or difficult, or where personal experience and preferences play a significant role, such as fashion industry. Furthermore, expert knowledge can be very valuable in the process of building intelligent fashion outfit scoring system. In this paper, we propose a new interactive machine learning framework for outfit rating system based on user experience and personal fashion preferences. Our approach allows dynamic user feedback in different forms, such as data selection, data labeling, and data correction. In particular, this approach can significantly reduce the amount of data required for training an accurate model, and can embed personal preferences through the direct interaction and achieve personalized results.**

***Index Terms*—interactive machine learning, Marcelle, web-applicaton, design, fashion, outfit scoring system**

I. Introduction

The recent surge of online fashion communities such as Polymore, Pinterest and Youtube videos have greatly helped spreading fashion trends and fashion tips. Most of these advice given are targeting entire communities and not specific individuals which all have different backgrounds and preferences [1]. In result, the recommendations lack of personalization and do not fit as well as it could the needs of the users. Moreover, the existing personalized systems require a massive amount of data in order to give accurate predictions. We want to find a way to blend personal experiences and preferences with the current fashion trends by using only a small sample of data. To achieve this, we propose an interactive machine learning fashion assistant that will rate the outfit chosen by the user. The user will be able to input his feedback and contribute to the learning process of the assistant. Our design could fit the needs of an app designer who wants to build robust models with limited data or of an machine learning developer build better models through a given interaction technique. Our contributions are the following:

- We build a reactive machine learning pipeline that considers users feedback and improves through a given interactive technique. This facilitates more informed decisions in the future outfit choice.
- Our system can be customized to suit individual preferences and styles, providing users with personalized ratings that take into account their unique tastes.
- By merging recent advancements in machine learning and a modular open source toolkit Marcelle for programming interactive machine learning applications, we were able to eliminate the need for training on massive amounts of data to achieve desired performance.

II. Related works

Recently, there has been a lot of interest in using an intelligent fashion analysis, such as clothing recognition, parsing, retrieval, and recommendation, due to the huge market potential in the fashion industry. Previous research on fashion recommendation mainly focused on the retrieval framework, which involves recommending fashion items that go well with a given query [2]–[5]. However, these works only studied compatibility between two objects and did not consider personalization. Some researchers have explored the use of deep networks for metric learning, which involves learning feature representations using CNN and measuring the distance between objects [6]. Others have applied deep learning to recommender systems, such as by jointly performing deep representation learning and collaborative filtering or learning a unified feature representation for both users and images [7].

However, these works only considered recommendation of individual items and did not address the issue of compatibility between items from different categories. Our work follows the ideas of the both kinds of works merging the concept of compatibility and personalization through merging the interactive part with standard machine learning approaches. To the best of our knowledge, this is the first Interactive Machine Learning Franework for the Outfit Rating System that takes into account the compatibility between items and personal preferences of the user.

III. Proposed Design

A. System Overview

Our goal is to develop a new interactive and iterative learning technique built on top of an MLP regressor, so that the user can interact with the machine learning model dynamically to provide feedback and to incrementally improve the performance of the model. Among different forms of user feedback available, such as knowledge input, feature selection, and parameter setting, we will concentrate mainly on adding the subset of training data samples, where the score might be based on the user's experience and preferences and/or professional expertise. By doing so, we hope to achieve maximal improvement in the model's performance and personalization while minimizing the number of additional training points required. To summarize, the problem can be defined as follows:

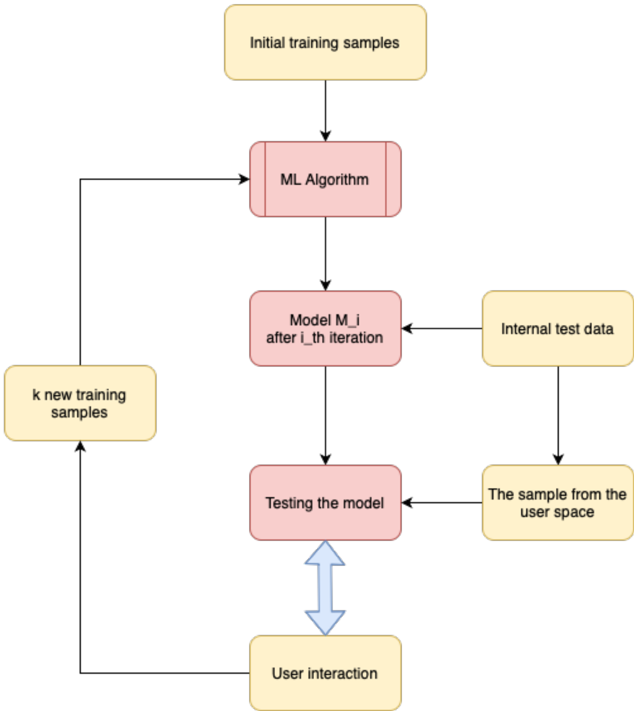

Fig. 1. A structural flowchart of the interactive machine learning system.

Assuming that we have a feature space F for a machine learning algorithm, a starting training set of $X = \{x_1, x_2, x_3, ..., x_n\} \subset F$*, and an internal test set* $Y = \{y_1, y_2, y_3, ..., y_n\} \subset F$*, we introduce a user space U that contains user-defined attributes, in our case scored images based on personal preferences and experience. These attributes must meet two criteria: they must have the same format as the original dataset and they must have at least 2 samples that have similar attributes [8].*

Let $Z = \{z_1, z_2, z_3, ..., z_n\} \subset U$ *be a set represented in the user space U, and let S be the range of application values (scores) for the learned model* $M_0(y) : F \rightarrow S$ *using the initial training set* X.

Our objective is to identify a set of k *new data points (where* k *is a fixed constant),* $X' \subset F$*, such that* X' *fulfill certain user-defined conditions of attribute values in* U*. These conditions are defined interactively by testing the model on the set Z in the user space U. By adding new training samples, the learned model* $M_1(y)$ *becomes an improved model compared to the original model* $M_0(y)$*. This process can be repeated iteratively until the model's performance is satisfactory and meets the user's expectations.*

B. User Space and User Interactions

We define two phases of interaction that makes the learning process unique and personalized: 1) interaction scheme that improves the model performance based on personal preferences; 2) the personalization scheme based on user's experience. In other words, we firstly aim to improve the model performance and then focus on building personalized autonomous fashion assistant, by shifting the purpose of interaction from teaching to incorporating user specific experience and asking for the model's feedback.

1) Phase I. Model Improvement Scheme: To improve the model, the user aims to add new training samples according to the following principle:

- Define the sample similar to the sample being tested, with the low score in case the model scored this testing sample higher than the user expected. Pass the new sample along with the sample on which model was tested, having at least 2 images in the new training set.
- Define the sample similar to the sample being tested, with the high score in case the model scored this testing sample lower than the user expected. Pass the new sample along with the sample on which model was tested, having at least 2 images in the new training set.

2) Phase II. Personalization Scheme: We base this scheme on the underlying assumption that machine learning based computer vision models can identify the features that remain invisible to human eye [6], which enables them to make weighted decision by identifying the correlation between robust features in comparison to the human assessment, which is limited to visual perception and personal preferences. To personalize the model we define the following scheme:

- The user incorporates the successful samples associated with a particular experience, e.g. the outfit that received many positive reviews and feedback, into the model's training set keeping the record of the good outfits.
- The user incorporates the unsuccessful samples associated with a particular experience, e.g. the outfit that was very uncomfortable, into the model's training set keeping the record of the bad outfits.

We believe this scheme would eventually result in a personalized fashion assistant which takes into account the examples from the user's real-life experience, and in the testing phase, when the user is contemplating about the choice of the outfit, can spot the similarities between the underlying features of the new testing samples [9] with the previous samples and output the score which is based on the user's experience. Basically, it would help the user to assess the probability of the particular sample being as much successful as it was in the previous user's experience. Our main assumption is that ML model can spot the features that remain invisible to human eyes, and find the correlation between the samples in the training set.

IV. Implementation

We used the recent advancements of the traditional machine learning models and a modular open source toolkit Marcelle [10] to build our application. Marcelle is a flexible and modular open-source toolkit that allows for the programming of interactive machine learning applications. It is designed around the concept of components that embed both computation and interaction, which can be combined to create reactive machine learning pipelines and customized user interfaces.

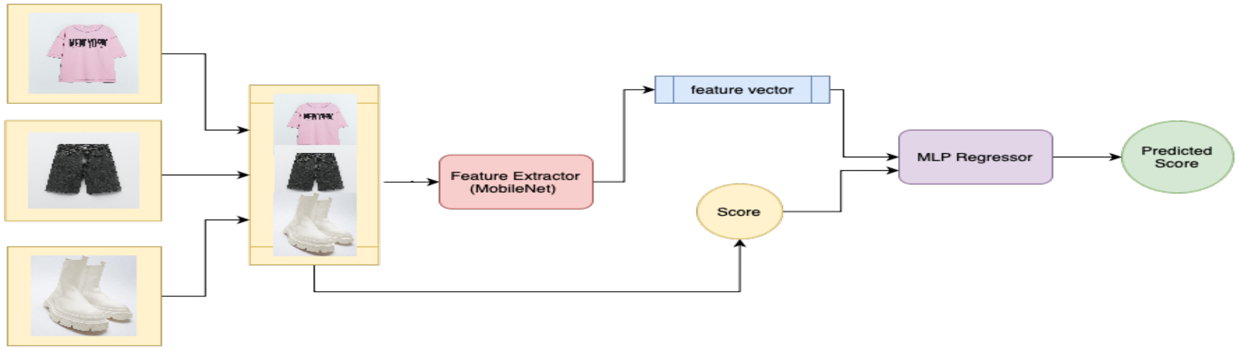

Fig. 2. Machine Learning Pipeline.

This architecture facilitates rapid prototyping and extension of machine learning applications. Additionally, Marcelle provides adaptable data stores that promote collaboration between machine learning experts, designers, and end-users.

A. Image Processing

Firstly, we acquired the images of three groups: UpperBody, LowerBody and Shoes. We tried to keep the combination of images of all scores from 1 to 10. The given input of three images are resized and concatenated into one single image vertically for feature extraction phase.

B. Feature Extraction

The concatenated input is passed to the feature extractor, we used MobileNet [11] which is implemented as a component in Marcelle toolkit. More specifically, for feature extraction, the .process() method was used to get the embeddings from an input image. MobileNet is a type of convolutional neural network (CNN) architecture that is designed to be efficient and lightweight, making it suitable for use on mobile and embedded devices. We extracted high-level features and obtained the vector with a corresponding score that can be further used for MLP regressor.

C. MLP Regression

In the final stage of the pipeline, we pass the feature vector and corresponding score to the Multi-Layer Perceptron (MLP) regressor, which is also implemented in Marcelle. It consists of multiple layers of interconnected nodes or neurons that process input data to produce a continuous output value. In our case, we used 2-layer perceptron with 32 nodes each. It works by processing input data through multiple layers of interconnected nodes, with each node applying an activation function to the weighted sum of the outputs from the previous layer. The weights and biases of the nodes are learned during the training phase, and the final output value is produced by the output layer.

V. Test Phase

After training the model, we passed the images from the internal test set that we described in the Proposed Design Section. In Phase I, in case the predicted scores were higher than expected, we label the test sample according to our perception and pass it to the training set, along with the similar new sample with the same score, which was not present in the internal test set and train set. The similar logic was applied for the outfits that were scored lower than expected. We repeat the steps in Phase I until the model predicts the score that aligns our perception. In Phase II, we tried to pass the images of clothes from our everyday life that received positive feedback from others and that we liked the most, and similarly the outfits which were uncomfortable to wear with corresponding lower score. We expect the model to identify the high-level features that are invisible to human perception and identify whether the new outfit that is passed to the network can be assessed based on the previous experience.

Evaluation

We want to assess the ability of our model to make correct predictions on participants' preferences based their feedback. To evaluate the effectiveness of the predictions, we propose to conduct a study with 12 participants. The participants should have different preferences concerning their clothing choices. To make sure of that, we first present them with 5 sets of 3

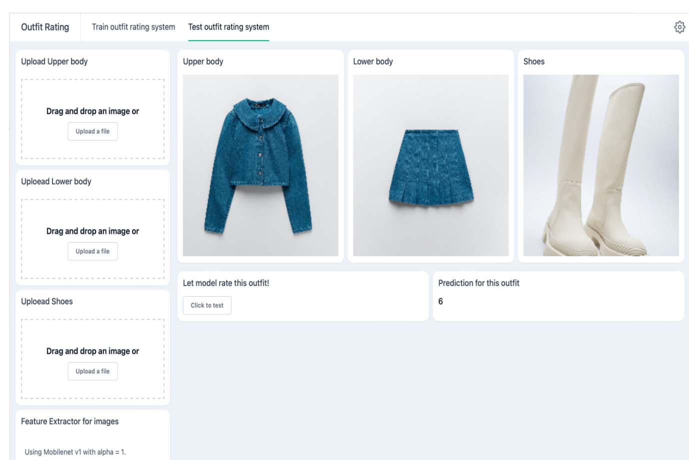

Fig. 3. Testing outfit rating system

different outfit styles and let them select the preferred one. Then, each participant will be given the same set of 10 outfits of different styles and will be asked to rate them depending on their tastes. The model will be trained with this sample of 10 outfits and the outfits will be labelled with the ratings given by the participant. A different model will be trained for each participant to keep the personalized aspect. In the final phase, participants and models will have to grade 10 new outfits. We compare the ratings given by the participant and the ones given by the model trained on his preferences. Our goal is to see whether there is any statistically significant differences between the ratings of each. If not, this means that our model is effective: with limited data, our model is able to make accurate predictions.

CONCLUSION

In this work, we addressed the problem of personalized outfit recommendation by proposing the interactive machine learning framework with two phase interactive scheme. Our reactive machine learning pipeline can be customized to suit individual tastes, making more informed decisions for future outfit choices. By utilizing recent advancements in machine learning and a modular open source toolkit, we were able to eliminate the need for extensive training data to achieve desired performance. Overall, our proposed system offers a promising solution for app designers and machine learning developers who want to build robust models through interactive techniques that prioritize personalization and individual preferences. While we were able to show that the 1st phase of the interactive scheme improves the model and adapts to the personal preferences of the user, it remains to be seen whether personalized recommendations actually lead to more satisfactory outcomes for users in the scenario described in phase 2. Because the 2nd phase of the interaction needs more time for evaluation and over-time assessment. Therefore, future research could evaluate the impact of personalized recommendations on user satisfaction and engagement.

REFERENCES

[1] J. Zhu, Y. Yang, J. Cao, and E. C. F. Mei, "New product design with popular fashion style discovery using machine learning," in *Artificial Intelligence on Fashion and Textiles: Proceedings of the Artificial Intelligence on Fashion and Textiles (AIFT) Conference 2018, Hong Kong, July 3–6, 2018*. Springer, 2019, pp. 121–128.

[2] S. Liu, J. Feng, Z. Song, T. Zhang, H. Lu, C. Xu, and S. Yan, "Hi, magic closet, tell me what to wear!" in *Proceedings of the 20th ACM international conference on Multimedia*, 2012, pp. 619–628.

[3] T. Iwata, S. Watanabe, and H. Sawada, "Fashion coordinates recommender system using photographs from fashion magazines," in *Twenty-Second International Joint Conference on Artificial Intelligence*, 2011.

[4] V. Jagadeesh, R. Piramuthu, A. Bhardwaj, W. Di, and N. Sundaresan, "Large scale visual recommendations from street fashion images," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2014, pp. 1925–1934.

[5] A. Veit, B. Kovacs, S. Bell, J. McAuley, K. Bala, and S. Belongie, "Learning visual clothing style with heterogeneous dyadic co-occurrences," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 4642–4650.

[6] S. Bell and K. Bala, "Learning visual similarity for product design with convolutional neural networks," *ACM transactions on graphics (TOG)*, vol. 34, no. 4, pp. 1–10, 2015.

[7] H. Wang, N. Wang, and D.-Y. Yeung, "Collaborative deep learning for recommender systems," in *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, 2015, pp. 1235–1244.

[8] H. Li, S. Fang, S. Mukhopadhyay, A. J. Saykin, and L. Shen, "Interactive machine learning by visualization: A small data solution," in *2018 IEEE International Conference on Big Data (Big Data)*. IEEE, 2018, pp. 3513–3521.

[9] J. Wang, Y. Song, T. Leung, C. Rosenberg, J. Wang, J. Philbin, B. Chen, and Y. Wu, "Learning fine-grained image similarity with deep ranking," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 1386–1393.

[10] J. Françoise, B. Caramiaux, and T. Sanchez, "Marcelle: composing interactive machine learning workflows and interfaces," in *The 34th Annual ACM Symposium on User Interface Software and Technology*, 2021, pp. 39–53.

[11] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.