Predicting Clothing Fit

Amarpreet Singh
1002513764
Department of Computer Engineering
University of Toronto
amarkbr.singh@mail.utoronto.ca

Charles Horn
1006958721
Institute of Aerospace Science
University of Toronto
charlie.horn@mail.utoronto.ca

Abstract

In the current age of e-commerce, customer experience relies even more heavily on product fit recommendations. Without fitting rooms, online stores often rely on industry standards, manufacturers' specifications, and customer data to recommend the right fit for each customer. There is inherent uncertainty in each of these information sources, and this can lead to a high return rate and lower profits. We build on the model and dataset provided in [5] and provide a predictive framework for size prediction. Our model is able to achieve better results than the previously recorded state of the art in [3]. Additionally, we test the effect of including additional product/customer attributes in classification, and report those results.

1 Dataset and Related Work

For this project we have chosen to predict the fit of clothing items for online retailers and customers. This problem has been investigated in recent years due to the explosion of e-commerce. The dataset has been provided by Misra et al. [1]. Unlike most data used to research this problem, this dataset has been made public. Research in this area is often carried out by private retailers, using their own data [1] [3].

Some research tends to focus on explicitly learning customer/product biases in order to score a customer/product pair [1]. Other approaches fuse observable and latent data to eventually maximize the log-likelihood of product/size pair for a customer [2]. With both of these approaches, the purpose is to recommend a size to a customer given a customer/product pair. This is a useful result for customers browsing items in an online store.

However, both of these approaches have an underlying goal of deciding, given a customer/product/size, will the item fit. This is the question we aim to answer in this paper, by building upon the methods in [2].

By performing an ablation study, we highlight the importance of the latent features. Our evaluation agrees with the results in [3], and re-affirms that the underlying semantics of text reviews give sufficient information for reliable fit prediction.

2 Data Analysis

The dataset provided by Misra et al. [1] contains measurements of clothing fit from two online clothing retailers: ModCloth and RentTheRunway. Data from both retailers provide user feedback on purchased items. We begin by considering the features in the item feedback data of each retailer. To use both datasets consistently, we only analyze features that are common to both ModCloth and RentTheRunway.

Submitted for assessment as a final project for "CSC 2515: Introduction to Machine Learning"

ModCloth: *Category, Fit, Review Text, Item Id, Height, Size, Review Summary, Bra Size, Cup Size, Bust, Shoe Size, Show Width, Hips, Length, User Id, Waist, Quality, User Name.*

RentTheRunway: *Category, Fit, Bust Size, Review Text, Rating, Item Id, Height, Size, Rented For, Body Type, Age, Review Summary, User Id, Weight, Review Date.*

Common: *Category, Fit, Review Text, Item Id, Height, Size, Review Summary, User Id.*

In both datasets, each item can receive multiple reviews from different customers. The dataset contains items and reviews from ModCloth and RentTheRunway. ModCloth contributes 5850 items and 192544 reviews, whereas RentTheRunway contributes 1378 items and 82790 reviews.

Clothing Fit: Among all the common features in our datasets, we seek to predict the ’fit’ of an item based on the review of a customer. We identified three types of fit in our datasets: ’large’, ’small’ and ’fit’. We determined that the clothing was ’fit’ for 73% of users, ’small’ for 14% of customers, and ’large’ for 13% of customers.

Customer Height: We observe that customers taller than 1.6435 meters, which is the mean height of customers whose clothing ’fit’ them, categorized their clothing as ’small’ for them. Customers shorter than the mean height of 1.6435 meters categorized their clothing as ’large’ for them. It seems like shorter customers find their clothing items to be large for them, and larger customers find their clothing to be small for them.

Clothing Size: We observe that customers that tried clothing sizes greater than 12.04, which is the mean clothing size that ’fit’ the customers, categorized their clothing as ’small’. Customers that tried clothing sizes slightly smaller than 12.04 categorized their clothing as ’large’ for them. Spikes in the plots suggests that the manufacturers produce clothes of fixed sizes.

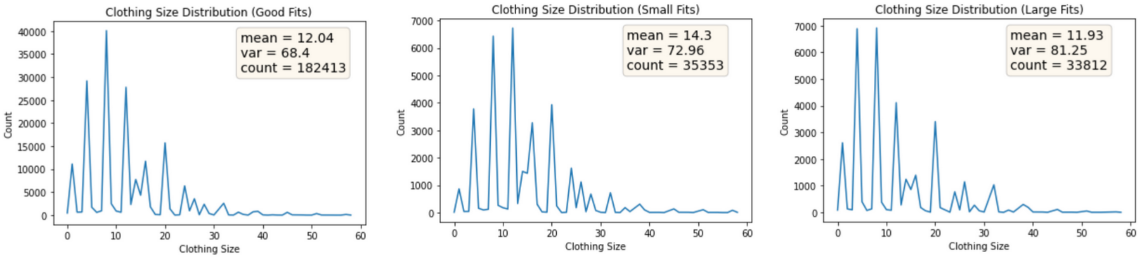


Figure 1: Clothing size distributions across fits

Review Text: Customers seem to give positive reviews regardless of their clothing fit. However, customer reviews are more negative if their clothing was ’small’ for them. Customers seem to tolerate purchasing clothes that are ’large’ for them. Percentage of negative sentiment reviews was 2.35% for ’good’ fits, 6.1% for ’large’ fits, and 12.05% for ’small’ fits.

Clothing Category: We observed 62 unique clothing categories ranging from ’frock’ to ’tunic’. However, majority of the reviews associated to ’small’ and ’large’ fits are directly for the ’dress’ clothing category. 33% of all non ’fit’ clothing fall under the ’dress’ category. All other clothing categories uniformly take the remaining reviews, leaving us with relatively little information gain.

3 Predictive Task and Evaluation

The predictive task that we are studying on our dataset is a classification task to predict the fit of a clothing item based on the review. The review comprises of the common features described in section 2 above, such as ’Review Summary’ and ’Size’, and the fit of the clothing is either ’fit’, ’small’ or ’large’. Our baseline for comparison will be the ’Majority Class’ method described by Baier [4], where the majority class ’fit’ is always predicted, and the accuracy is measured using Micro-F1 scores (multiclass classification) [4]. The baseline accuracy is 0.6892 on the ModCloth dataset, and accuracy is 0.7396 on the RentTheRunway dataset [4].

Revisiting the common features discussed in section 2, 'Customer Height' and 'Clothing Size' relay the same information - that larger (taller) customers often find their clothing to be 'small', and that smaller (shorter) customers often find their clothing to be 'large'. Since variation in the 'Customer Height' is very small compared to variation in 'Clothing Size', we ignore 'Customer Height' in our analysis. 'Review Text' is useful to identify 'small' fits, and the 'dress' 'Clothing Category' gives us higher confidence that we may be considering a non 'fit' clothing item.

Our model will comprise of a pre-trained ULMFiT network [5], followed by a custom classifier layer that is adjusted to make use of the aforementioned common feature observations. The 'Review Text' and 'Summary' features will be pre-processed before being input into the ULMFiT network by removing all special characters and single characters. Multiple spaces will be replaced with a single space, and all of the characters will be converted to lowercase.

4 Model

In researching the models to predict product fit, we found a few competitors that have very recently become the leaders in this area, because processing customer reviews to predict fit is a relatively new domain. In [4], the author compares multiple methods for predicting fit on the same datasets that we discuss here in this paper. Table 1 includes the results of that comparison as well as our own.

The ULMFiT [5] model is the most effective at predicting product fit. For this reason, we have adopted this model for our own experiment. The primary components of our model follow.

4.1 Pre-trained Language Model

Given an incomplete segment of text, Recurrent Neural Networks (RNNs) are effective at predicting the next word. However, they do not explicitly store each word in memory and run it through a typical neural network. Instead, they learn to develop a latent representation of the given text data. For this reason, a trained RNN becomes a perfect tool to encode text content. The ULMFiT process uses a model that was pretrained on Wikipedia text content to encode text samples.

4.2 Language Model Tuning

The English language structure is often consistent regardless of context. However, there are some nuances to each context that can change what would be considered the optimal latent representation for a latent representation of the text. For this reason, we can improve upon the pre-trained network by tuning the weights with our own training data. By referencing [6], ULMFiT assumes that different layers capture different types of information. Consequently, we tune each layer one at a time, with different learning rates. The authors of [5] empirically found that an effective learning rate adjustment schedule was to determine an ideal learning rate for the head layer, and use the following equation to set future learning rates: $lr^{i-1} = lr^{i}/2.6$.

4.3 Classifier Model Tuning

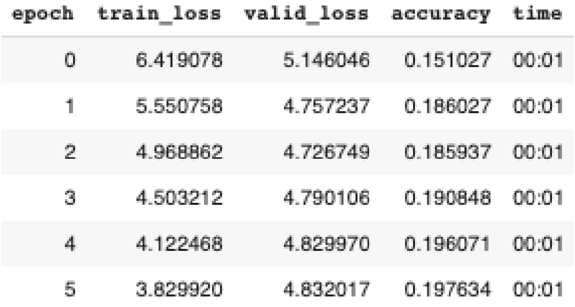
After tuning the language model, we input the latent feature representation into a separate classifier network. Since transfer learning was originally validated in the image classification domain, [5] followed standard computer vision practices and uses a classifier with two layers. Each layer contains some level of dropout and batch normalization with a ReLU activation function. To fine tune this network we use the method of gradual unfreezing. This is similar to how we did fine tuning for the Language Model, but for every layer i, every layer $j > i$ is also unfrozen. This gradual fine tuning scheme has been shown to prevent the model from forgetting the previously learned structure of the current context [5].

4.4 Additional Classifier

To test if 'summary' and 'reviewText' contained the full semantics of a customer/product fit, we also included product size and category (identified as relevant in previous section) as inputs to our classifier. These values were concatenated with the outputs of the classifier in Section 4.3, and fed through a trained logistic regression classifier. The results for this comparison can be seen in Table 1.

4.5 Assessment

Our model leveraged transfer learning to reduce training time. We considered training a RNN from scratch as in [2], but chose not to due to the compute cost. We encountered over-fitting while fine-tuning the language model. Figure 2 shows that as iterations increased, often the training loss would decrease, but the validation results would increase.



epoch	train_loss	valid_loss	accuracy	time
0	6.419078	5.146046	0.151027	00:01
1	5.550758	4.757237	0.186027	00:01
2	4.968862	4.726749	0.185937	00:01
3	4.503212	4.790106	0.190848	00:01
4	4.122468	4.829970	0.196071	00:01
5	3.829920	4.832017	0.197634	00:01

Figure 2: Iterations of optimizing a layer of the Language Model

5 Results and Conclusions

After loading the pre-trained model, fine tuning the language model, re-using the encoder in our classifier, and fine tuning the classifier, we get the results shown in Table 1 when running predictions on our previously unseen test set. The comparison to our baseline results can be seen in Table 1. Since [4] chose to compare models with the Micro-F1 score, we have reported this as well. However, this metric is a poor choice for this dataset because it is not evenly distributed, and the micro F1 score is essentially equivalent to accuracy. For this reason, we also present the F1 score per class, and the Weighted F1 Score. When compared to the baseline models, our choice of structure described in Section 4 performed better than all models compared in [4].

Considering that our model choice outperformed the model described in [5], this could indicate that the ’Slanted Triangular Learning Rate’ (described in [5] and omitted in our model) is not required for this task. On top of what was tested in [5] and [4], we have added an additional classifier network to the output, and included individual product/user features to see if they improved results, or if the text representations are sufficient to understand the fit semantics. We can see that the additional features improved the individual F1 scores for the under-represented classes, but overall the changes to the Micro-F1 and Weighted-F1 scores are negligible. We conclude that the contents of the review text and summary are necessary and sufficient to make accurate fit predictions.

Table 1: Results of our models compared to the baseline models from [3]

Method	F1 Small	F1 Fit	F1 Large	Micro-F1 score	Weighted-F1
Majority Class	-	-	-	0.68	-
TF-IDF LR	-	-	-	0.7899	-
Mean GloVe LR	-	-	-	0.7124	-
ULMFit Fine-Tuned	-	-	-	0.8269	-
Our Model	0.6354	0.9024	0.6354	**0.8423**	0.8524
Our Model w/ added features	0.6368	0.9004	0.6454	0.8403	0.8518

References

[1] Rishabh Misra, Mengting Wan, Julian McAuley *Decomposing fit semantics for product size recommendation in metric spaces*. RecSys, 2018

[2] G. Mohammed Abdulla and Sumit Borar *Recommending Product Sizes to Customers*. RecSys, 2017

[3] Vivek Sembium, Rajeev Rastogi, Atul Saroop and Srujana Merugu *Recommending Product Sizes to Customers* RecSys, 2017

[4] Stephan Baier *Analyzing Customer Feedback For Product Fit Prediction* RecSys, 2019

[5] Jeremy Howard and Sebastian Ruder *Universal Language Model Fine-tuning for Text Classification* RecSys, 2018

[6] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson *How transferable are features in deep neural networks?* arXiv, 2014