Predicting Clothing Fit Size Using a single Latent Variable

Luis Diaz

December 3, 2019

ABSTRACT

Because of the rise of online shopping it is important to predict clothes that will fit properly so that both consumers and companies can benefit. However, it is hard to predict clothes that fits correctly due to discrepancies that arise in labeling clothes and an imbalance of labels. This assignment focuses on predicting whether clothes will predict properly using a single latent variable in both the user and the items combined with other observed features.

I. Introduction

With the increase in online shopping, buying clothes online can be risky. This is because there is always a possibility that the clothes bought will not fit properly. If clothes do not fit then customers then need to either go through a "return" and "re-order" process, which could lead to a waste of money and time. Customers might even choose to not re-purchase the clothes which means a lost sale for the company. And for some shopping sites there can be no refund policy and the customer are either forced to throw away the clothes or try to resell the ill-fitting clothes. To make matters worse, a lack of a universal standard for clothes makes the process more difficult just to get the right fit. For example, across different countries and regions sizes can differ like a "small" size in Europe is different from a "small" size in the US. Additionally, differences can even exist in sizing standards between different brands within a single country. For example, across different brands sizes differ like a "small" Hollister shirt does not necessarily have the same fit as a "small" Tommy Hilfiger shirt. Such issues have substantially hampered the expansion of online shopping for years.

In this assignment I will look at data from the online retailer, Modcloth. The site allows customers to place feedback on multiple attributes including how the clothes fit them. I will look at fit predictions and see if I can predict whether clothes will fit a user or not using user's measurements. I will then assume a single latent variable for a customer and product and combine it with observable features to see if it improves prediction.

Literature Review

Much of the research into product size recommendation is new. The Modcloth dataset used was collected previously in a research publication that investigated the problem by decomposing Fit Semantics for product size recommendations in Metric Spaces [1]. In that research article two clothing datasets were used in learning fit semantics and a metric learning approach that took the user biases and item biases to maximize the ROC-AOC score of 5 different models with prototyping. They used user ids, item ids, sizes and categories to make their predictions. The results in this paper showed that more latent factors and a metric space performed better in predicting correct clothing fit sizes with an improvement of up to 18%. My work differs in that they used complex latent factor models and my models are a lot simpler. I also only used one Latent Variable while they used multiple latent variables and metric learning approaches.

Other recent research includes approaches that uses skipgram models to learn latent features and gradient boosting to predict preferred product sizes for the user [2]. The data gathered in their analysis was generated from the online clothing platform, Myntra. The data they used in their analysis is a lot larger than the one used here as they had over a million transactions and around 360,000 users and had more features while this one had more limited features. The

results of this approach also showed that their latent feature models performed well in predicting clothing fits.

Also, there is further research that uses Bayesian logit on Amazon shoe datasets and generated datasets that shows improvement in fit predictions [9]. All these recent studies are on the cutting edge to try and improve the customer experience to improve fit predictions using different techniques. When I conduct my analysis, I will compare my results to the results of this research to see if I can also see an improvement in predicting the fit type.

II. Data

Data used in this assignment is from the online retailer Modcloth. The dataset has data on customer feedback on various things including bust, bra size, hip size, review summaries, etc. Many of the variables were omitted because they were not used in my analysis. Table 1 describes some of the variables used in my approach.

Name of Variable	Descriptions
Fit	Variable I am performing my prediction analysis on. Given as string and types are {small, fit, long}. One hot encoded into array
Height	Height of the user as a string. Converted into inches and integer
Quality	Integer value from 1-5. Reported by user as quality of clothes
Size	The reported size of clothes as an integer
Item_id	Integer value that represents the item. Used to make indexes for all items and user data of all the items they bought
User_id	User value that represents the user. Used to make indexes for all users and item data of all the users that bought the item

Table 1: Modcloth Variables used

Statistic	Descriptions		Notes
Total Transactions	82,790		
Unique Users	47958		
Unique Items	1378		If looking at different sizes, there are 5005 items
Fit Proportions	Fit	.686	
	Small	.158	
	Large	.157	
Customers with one Purchase	32029		If looking at different sizes, there are 31870 users
Items only bought once	316		If looking at different sizes, there are 2029 items

Table 2: Dataset Statistics

In Fit Proportions in Table 2 shows, we can see that the data is heavily skewed towards clothes that fit and there is less data on the other fit types. Figure 2.1 shows a better visualization of the distribution of the different fits. We can also see that there are also many users who only made one purchase and a lot of clothes that were only bought once. It is also important to distinguish that clothes of different sizes can be treated

as different items and so that I listed that difference in the Notes column in Table 2.

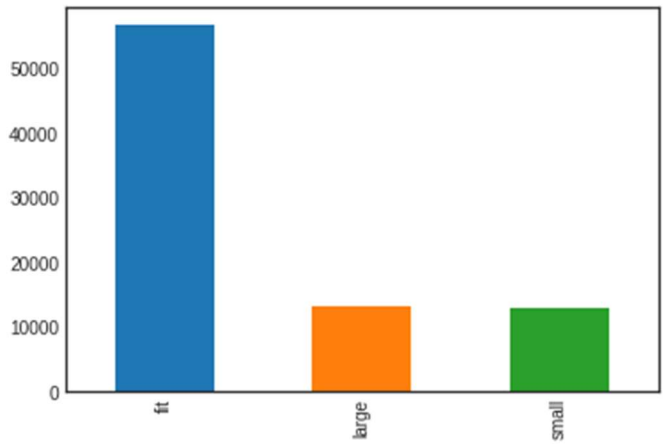

Figure 2.1: Fit size distribution

III. Methodology

Because this dataset is so unbalanced it is hard to make accurate predictions on whether clothes will fit. Starting out, I made looked at the features in the dataset and pick some out to use in a classifier. I decided to use the features quality, height, and size but there were some problems. Both quality and height had null values in the data (68 and 1107 null values respectively) so I decided to drop the rows with those values. The height column was also a problem because it was a string so with a little processing, I managed to turn into integer values in inches. And since the column I am trying to predict, fit, is also strings I one hot encoded them to use in my analysis. So, after picking out the variables and processing them I used them in a logistic regressor. Without using the balanced option, the logistic regressor got a 70% accuracy just by predicting that clothes will fit but it did not predict and clothes being large or small. This led me to try different metrics, like the balanced error rate, to judge the performance of the model and I decided to use the AUC (Area Under the Curve) score since it tells us how much the model is capable of distinguishing between classes, an important factor in this model. I then adjusted the model and added a balanced class weight to the logistic regression which decreased accuracy but improved the AUC score. I wanted to try out different features, so I then decided to use user ids, item ids, and size. For all these features none were null, so I did not have to drop any null values and I one hot encoded the values like earlier. I then made a balanced logistic regressor that used these new features and another regressor that used a combination of both different features from feature set 1[1] and feature set 2[2]. To optimize the values, I added a regularization pipeline to maximize the AUC score.

For my approach I used a single latent variable for each of the customers and items that I tried to implement in Algorithm 1 from [4]. I decided to do this because I believe it will increase the AUC due to it being able to capture the biases in the clothes and customers from their true sizes and I could differ from their approach by adding different features and building different baselines. This would allow me to give more accurate fit predictions that should outperform the baselines. To do this I tried to get the true sizes of the customers and items and I then used the learned features in a Logistic Regressor to classify the fit prediction. For this method we use the function $f_w(\mathrm{t}) = \mathrm{w}(s_i - t_j)$ as the scoring function for the model, where s_i and t_j are single latent variables and w is some weight. The goal of the algorithm is then to minimize the loss function

$$L(y_{ij}, f_w(s_i, t_j)) = \begin{cases} \max\{0, 1 - f_w(s_i, t_j) + b_2)\} & \text{if } y_{ij} = \texttt{Small} \\ (\max\{0, 1 + f_w(s_i, t_j) - b_2)\} & \\ + \max\{0, 1 - f_w(s_i, t_j) + b_1)\}) & \text{if } y_{ij} = \texttt{Fit} \\ \max\{0, 1 + f_w(s_i, t_j) - b_1)\} & \text{if } y_{ij} = \texttt{Large} \end{cases}$$

where b_1 and b_2 are threshold parameters with $b_2 > b_1$. Again, I one hot encoded the fit, and I optimized the model by testing the AUC score on the validation set with features from both feature set 1 and 2 and used the one with the highest score. I also built a regularization pipeline to get the highest AUC score and used gradient descent to minimize the loss function and maximize AUC.

A problem I had was that I initially tried to implement a Support Vector Machine. After I could not get it to work with the imbalanced data, I switched to a balanced logistic regressor. I also wanted to try and implement text features

[1] ***Feature Set 1:** {'quality', 'size', 'height'}

[2] ****Feature Set 2:** {'user_id, 'size', 'item_id

with the models, but I found to be difficult to extract the text and use it in this assignment.

ALGORITHM 1: Algorithm for computing customer and product true sizes.

ComputeTrueSizes(C, $\mathcal{P}$, $\mathcal{D}$, $\{c_j\}$)

1: **for each** product j, $t_j = c_j$.
2: $w = 1$; $b_1 = -1$; $b_2 = +1$.
3: **while** (not converged) and (numIterations < maxIterations) **do**
4: **for each** customer i, $s_i = \arg\min_{s_i}(\mathcal{L}|\{t_j\}, w, b_1, b_2)$.
5: **for each** product j, $t_j = \arg\min_{t_j}(\mathcal{L}|\{s_i\}, w, b_1, b_2)$.
6: $w, b_1, b_2 = \arg\min_{w,b_1,b_2}(\mathcal{L}|\{s_i\}, \{t_j\})$.
7: **end while**
8: **return** $\{\{s_i\}, \{t_j\}, w, b_1, b_2\}$.

Algorithm 1: Basic structure that is seen in [4]

IV. Experiments

4.1 Comparison Methods

For different comparisons I used the following models. I compared their performance of each using the AUC score. Training, validation and test sets are created using an 65:20:15 random split of the data

- **Model 1**: This model is a basic logistic regression with a balanced class weight to handle the imbalance in the data. This model has the input variables from Feature Set 1
- **Model 2:** This model is a basic logistic regression with a balanced class weight to handle the imbalance in the data. This model has the input variables from Feature Set 2
- **Model 3:** This model is a basic logistic regression with a balanced class weight to handle the imbalance in the data. This model has the input variables from models 1 and 2: Feature Set 1 and 2
- **Model 4 (Proposed method):** The Single latent variable model described in Section 3. Observed variables used in this are from both Feature set 1 and 2

Model	**1**	**2**	**3**	**4**
Avg AUC	.5104	.4932	.4931	.6507

Table 3: Performance of various models in term of Average AUC on test data

By Looking at Figure 3 we can see that adding the combined feature sets helped increase

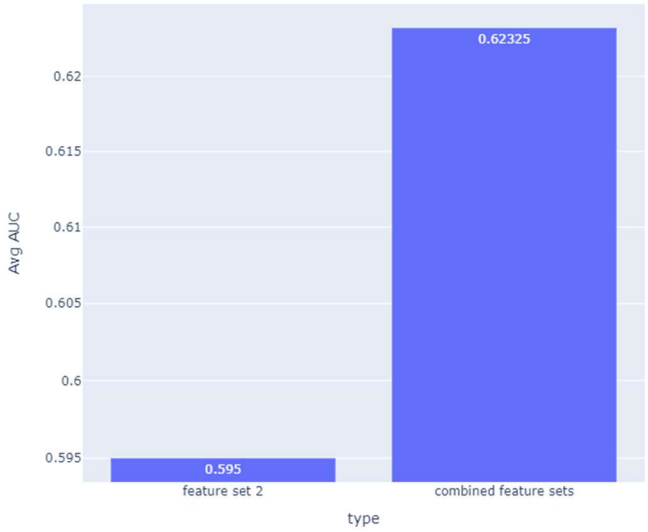

Figure 3: The Avg AUC on validation with just Feature set 2 vs. with both Feature Sets

the AUC by around 4.75% on the validation data. Looking at my results in Table 3 we can also see a huge improvement of 27.5% from model 1 to model 4. Interestingly we can see that models 2 and 3 faired worse than the first model which only had feature set 1.

V. Results/Conclusion

The recent need for online retailers to develop efficient ways of predicting accurate fit has grown to be more important recently. By predicting clothes that fits right the first time we can save both consumers and the companies a lot of money and time. Looking at recent studies, I saw that there is a lot of new work into trying to solve this prediction problem through use of various methods and latent factor models. My own research and findings matched those of all the articles [1,2,3,4] that used some sort of latent variables to increase the effectiveness of their models.

From my data analysis I saw that I was able to improve fit prediction using my proposed method of a single latent variable with observed variables. By looking at our dataset and finding interesting statistics we can see that our data is heavily skewed towards the fit category (Figure 2.1). Because of this imbalance using unbalanced classifiers like the basic logistic regression led to a high accuracy but had worse

scores in the Balanced error rate and Area under the Curve. This helped me decide to use Area Under the Curve as the performance metric for its use in distinguishing between classes. After this I was able to use some balanced logistic regression models to create as a baseline for my proposed method. My proposed method is the implementation of an algorithm that I saw in [4] and thought I can use with this dataset to combine with a latent variable and observed variables to deliver accurate fit predictions. I decided to use different features in the dataset to try and maximize my data and developed two different feature sets that I used to optimize my models. In the logistic regression baselines, I saw that feature set 1 (model 1) outperformed feature set 2 (model 2) and the combined feature sets (model 3) as seen as the results in Table 3. This was the opposite case when I ran the combined features in model 4 where the combined features performed a lot better that just using feature set 2 shown in Figure 3. I managed to optimize all the models a bit more by running a regularization pipeline and I managed to greatly improve model 4 by running gradient descent. After running my optimized algorithm, the method I proposed delivered a performance that outperformed my baselines and was 27.5% more accurate than model 1. This approach is less effective than models done in other research [1,2] but it still delivered an increased performance in recommending fit predictions.

I didn't manage to use the review text and implement feature text in my analysis like I wanted to because there was a lot of words and it would crash my computer. For the next time I would want to try and implement it somehow and see if it can improve my performance. It does not seem that the model overfit the data.

VI. References/Citations

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

[2] G Mohammed Abdulla and Sumit Borar. 2017. **Size Recommendation System for Fashion E-commerce. KDD Workshop on Machine Learning Meets Fashion** (2017).

[3] Vivek Sembium, Rajeev Rastogi, Lavanya Tekumalla, and Atul Saroop. 2018. **Bayesian Models for Product Size Recommendations. In Proceedings of the 2018 World Wide Web Conference on World Wide Web**.

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