# Clothing Fit Recommendation System


Eddie Tseng
UC San Diego
edtseng@eng.ucsd.edu

Hsiao-Chen Huang
UC San Diego
hsh030@eng.ucsd.edu



## Abstract

*We try to implement some models from both the class and some references for recommending product size fits* $\{Small, Fit, Large\}$ *to each customer. The outcome for a (user,item) pair is predicted based on the difference between customer and product true sizes. We implemented two models: First is the baseline model based on customers past product purchase and minimizes the loss to predict the true size in a specific category of the customer. Then, we predict if the product he/she tending to buy has the best fit. Second model is implementing latent factor model from reference[3]. Predicting the true size value of customer and product that minimize two loss function variants. The results show a decent improvement compared to baselines.*


## 1. Introduction

With different companies coming with different products and with different size values. For example, the physical size of Reebok, Nike and Adidas are not the same. Therefore, it is hard for customers to buy items with the right size confidently. Some customers even buy multiple sizes of a product and returning the rest which is wasteful and not wise. This is a huge problem for e-commerce such as Amazon and the others, they try to be able to automatically provide highly accurate sizing guidance about products to customers.

In the size recommendation problem, a customer implicitly provides the context of a desired product by viewing the detail page of a product and requires a recommendation for the appropriate size variant of the product. However, the main problems are: *Data Sparsity*, *Random start* and *Diversity*.

**Data Sparsity**: a majority of customers and products have very few purchases.

**Random start**: each new customer is a whole new data for the website. Thus it is hard for size recommendation.

**Diversity**: many users purchase items not only for themselves but also for their parents, friends or even strangers.

In this report, we try to use the methods we learned from the class for determining how a product of a certain size fits a customer, i.e., {Small, Fit, Large}. As for how we do it, we will explain later in the following sections.

The remainder of the report is organized as follows. After introducing data and analyze it in Section 2. we formally define the models we use for training in Section 2. In Section 4, we simplify the definitions to algorithms. Last but not least, the results and comparison in Section 5.

## 2. Data Analysis

### 2.1. Dataset

In this project, the data we choose is from the class website[2] with the following simple description. we liked to mention that there are actually two datasets, *RentTheRunway* and *ModCloth*. However, we will used *RentTheRunway* only in our project.

| | RentTheRunway |
|---|---|
| Number of users | 105,508 |
| Number of items | 5,850 |
| Number of transactions | 192,544 |

Table 1: Data Description

### 2.2. Dataset exploration

The from the figure1 below we can see the keys of the data and some data is missing.

| | age | body type | bust size | category | fit | height | item_id | rating | rented for | review_date | review_summary | review_text | size | user_id | weight |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 28.0 | hourglass | 34d | romper | fit | 5' 8" | 2260466 | 10.0 | vacation | April 20, 2016 | So many compliments! | An adorable romper! Belt and zipper were a lit... | 14 | 420272 | 137lbs |
| 1 | 36.0 | straight & narrow | 34b | gown | fit | 5' 6" | 153475 | 10.0 | other | June 18, 2013 | I felt so glamourous!!! | I rented this dress for a photo shoot. The the... | 12 | 273551 | 132lbs |
| 2 | 116.0 | NaN | NaN | sheath | fit | 5' 4" | 1063761 | 10.0 | party | December 14, 2015 | It was a great time to celebrate the (almost) ... | This hugged in all the right places! It was a ... | 4 | 360448 | NaN |
| 3 | 34.0 | pear | 34c | dress | fit | 5' 5" | 126335 | 8.0 | formal affair | February 12, 2014 | Dress arrived on time and in perfect condition. | I rented this for my company's black tie award... | 8 | 909926 | 135lbs |
| 4 | 27.0 | athletic | 34b | gown | fit | 5' 9" | 616682 | 10.0 | wedding | September 26, 2016 | Was in love with this dress !!! | I have always been petite in my upper body and... | 12 | 151944 | 145lbs |

Figure 1: Data description

There are many features for us to choose. Since we are predicting the size fit, the data we choose will be the *userId*, *itemId* and *size* to predict *fit*.

#### 2.2.1 Exploratory Data Analysis

We can already make few observations here, by looking at the head of the data:

1. There are missing values across the dataframe, which need to be handled.
2. Height column needs to be parsed for extracting the height in a numerical quantity, it looks like a string (object) right now.
3. Not so important, but some columns could do with some renaming- for removing spaces.

Firstly, we handle the naming of columns for ease-of-access in Python.

#### 2.2.2 Observations on Missing Data

From the table below we can see that out of 15 columns, only 8 columns have complete data. Quite a lot of data seems to be missing in *bust*, *body type* and *weight*. Fortunately, the data we want to choose, such as *category*, *fit*, *itemID*, *userID* and *size*.

| | total_missing | perc_missing |
|---|---|---|
| age | 960 | 1.159560 |
| body type | 14637 | 17.679671 |
| bust size | 18411 | 22.238193 |
| category | 0 | 0.000000 |
| fit | 0 | 0.000000 |
| height | 677 | 0.817732 |
| item_id | 0 | 0.000000 |
| rating | 82 | 0.099046 |
| rented for | 10 | 0.012079 |
| review_date | 0 | 0.000000 |
| review_summary | 0 | 0.000000 |
| review_text | 0 | 0.000000 |
| size | 0 | 0.000000 |
| user_id | 0 | 0.000000 |
| weight | 29982 | 36.214519 |

Table 2: Missing Data

#### 2.2.3 Data Visualizations

From figure2 below, we can easily conclude the dataset is **unbalanced**.

1. **Category plot**: This dataset has tremendous category and most of the items are in the same category(dress). Furthermore, many items related to a its own category which others are not related to.

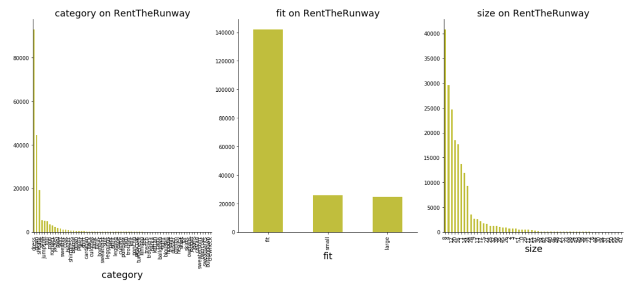


Figure 2: Data Visualizations

2. **Fit plot**: *Fit* is the label we want to predict, so if this plot shows that the data is unbalanced, we need to do some pre-processing for the data(balance).
3. **Size plot**: *Size* is the most weird one, each item have different size. And for each category, it has different measure method. So this plot cannot explain anything after we normalized it.

So, we normalized the dataset and comapre it.

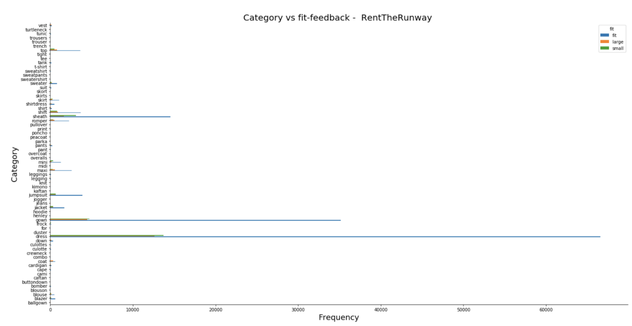


Figure 3: Category vs Fit

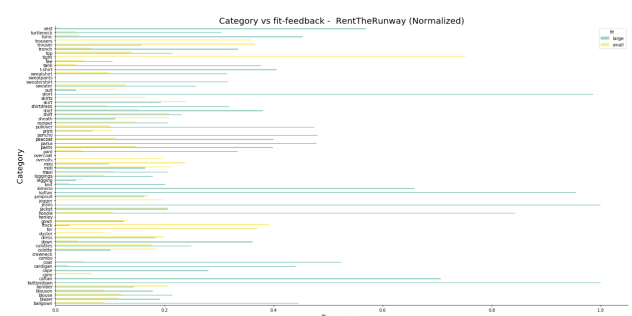


Figure 4: Category vs Fit(normalized)

Before normalized, the best fit category is *dress* and *gown*. After normalized, *shorts* and *jeans* categories usually have more returns due to large sizing and *top* usually has more returns due to small sizing.

## 3. Related Work

### 3.1. Problem Definition

A product in categories have multiple sizes. We will represent the product as *parent* product and the different size variations as *child* products. Denote the set of child products by **P** and the set of customer by **C**. For each transaction **D** has a triple of the form(customer, child product, return code), we denote it as $(i, j, y_{ij})$. Where $i$ is the customer making the purchase, $j$ is the purchased child product, and $y_{ij}$ is either Fit, Small or Large if the customer does not return the product.

Let $s_i$ denote the true size of customer $i$ and $t_j$ be the true size for child product $j$. Once we have the true sizes, we can predict the labels by calculate the difference between $s_i$ and $t_j$. We assume if the return label is **Fit**, this means $|s_i - t_j|$ must be small. On the other hand, if the return label is **Small**, $|s_i - t_j|$ must be large and $s_i > t_j$. Similarly, we will predict **Large** if $s_i < t_j$. Let $f_w(s_i, t_j) = w(s_i - t_j)$ be our linear model with weight parameter $w \geq 0$. Furthermore, let $L(y_{ij}, f_w(s_i, t_j))$ be the loss function. Finally, we try to minimize the loss function.

### 3.2. Model

The reference's model is closely related to paired comparison models such as the Thurstone model[5] and Bradley-Terry model[1]. The goal is to estimate a latent factor associated with the entities such that the difference in the latent factors in traditional paired comparison outcome.

### 3.3. Loss Function

Unlike general Logistic loss function, the paper used **Hinge loss**. This is because it can be optimized more efficiently.

$$L(y_{ij}, f_w(s_i, t_j)) = \begin{cases} \text{if } y_{ij} \text{ is small:} \\ max\{0, 1 - f_w(s_i, t_j) + b_2\} \\ \text{if } n \text{ is fit:} \\ max\{0, 1 + f_w(s_i, t_j) - b_2\} + \\ max\{0, 1 - f_w(s_i, t_j) + b_1\} \\ \text{if } n \text{ is large:} \\ max\{0, 1 - f_w(s_i, t_j) - b_1\} \end{cases} \tag{1}$$

Where $b_1$ and $b_2$ are the threshold parameters with $b_2 > b_1$ that split the model score line into three segments. The score is greater than $b_2$ corresponds to **Small**. The score is less than $b_1$ corresponds to **Large**.

## 4. Algorithms

### 4.1. Baseline

For our baseline solution, we try to use the information of user experiences. From items that was bought by a user, we calculate the mode of those sizes and consider it as the true size of the user. However, we found that the sizes of different item are formed in different standard. For instance, in the dataset, the size range of a dress is 0-57, but the size range of a shirt is 1-26. Therefore, we change to calculate the mode of sizes of items in the same category that was purchased by a user.

Follow the above baseline model, we can easily predict the user true size given an *userId* and the item *category*. If the corresponding *size* is equal to the user true size, our prediction of fitness is **Fit**. If the *size* is greater to the user true size, the prediction is **Larger**. If the *size* is lower to the user true size, the prediction is **Smaller**.

### 4.2. Latent Factor Model

The novel latent factor model[4] is proposed by Sembium *et al.* The algorithm which is shown in the following is consist of three phases:

**Phase 1: Computing true size values $\{s_i\}$ for customers**. To solve the convex optimal problem, for each $\{s_i\}$, we seek the size value that minimize the Hinge loss in formula (1). By searching size value in increasing order, when the slop of the loss becomes non-negative, we can consider the value $s_i$ is the true size value for the customer $i$.

**Phase 2: Computing true size values $\{t_j\}$ for products**. It is similar to the estimation on customer true size. For each $\{t_j\}$, we seek the size value of product $j$ that minimize the Hinge loss. Selecting the value $\{t_j\}$ while the slop of loss function becomes non-negative.

**Phase 3: Computing model parameters $\{w, b_1, b_2\}$**. After getting the true size of customers and products, we can find the parameter that minimize the loss $L$ by using gradient descent

For Phase 1, we focus on computing customer true sizes while keeping product true sizes and parameter value fixed. For Phase 2, we focus on computing product true sizes while keeping customer true sizes and parameter value fixed. For Phase 3, computing parameter value while keeping product and customer true sizes.

**Algorithm 1** Algorithm for computing customer and product true sizes.

**Require:** ComputeTrueSizes $(C, P, D, \{c_j\})$

**for** each product $j, t_j = c_j$ **do**
  $w = 1; b_1 = 1; b_2 = +1;$
**end for**
**while** (not converged) and (numIterations < maxIterations) **do**
  **for** each customer $i$ **do**
    $s_i = argmin_{s_i}(L|\{t_j\}, w, b_1, b_2);$
  **end for**
  **for** each customer $j$ **do**
    $t_j = argmin_{t_j}(L|\{s_i\}, w, b_1, b_2);$
  **end for**
  $w, b_1, b_2 = argmin_{w,b_1,b_2}(L|\{s_i\}, \{t_j\});$
**end while**
**return** $\{\{s_i\}, \{t_j\}, w, b_1, b_2\};$

# 5. Result

## 5.1. Evaluation Matrix

In our prediction result, we have three classes {**Small**, **Fit**, **Large**}. We found that the numbers of each classes in our dataset are different, so if we use the overall accuracy to evaluate our performance, it may not be precise. For example, there are about 70% of the data are in class **Fit**. If we predict all the data as **Fit**, we can still get a high accuracy as 70%, but it is unreasonable.

Instead, we use average class accuracy to show the performance of our models. Measuring the accuracy in each class can make sure we get good prediction in all the classes.

$$ACC = \frac{1}{3}\sum_{c}\frac{N_{y=c,\hat{y}=c}}{N_{y=c}}$$

where $N_{y=c,\hat{y}=c}$ is the number of correct prediction in class c, and $N_{y=c}$ is the total number class c.

## 5.2. Comparison

**Baseline**: The class average accuracy is 0.3208, the prediction of Small accuracy is 0.0329, the prediction of Fit accuracy is 0.1250 and the prediction of Large is 0.8045.

**Latent Factor Model**: From the baseline result, the average accuracy improved almost 30% Tab3. The main reason is the baseline did not really predict the true size of customer and product(we just use the data based on user experience). Furthermore, the small and large data are relatively small compare with fit data. Thus, it is hard for Baseline to predict small or large. However, Latent Factor model predict the customers and items true sizes, then predict the return value based on the true sizes.

From the dataset, there are many features which let us choose. Due to our observation, only **size** and **category** works well. **Body type** seems to have a great influences, but there is about 23% data is missing. Another feature might be the **review text**, to be honest, we should definitely add this feature. However, the time flies and we need more time to debug.

| | Ave | Small | Fit | Large |
|---|---|---|---|---|
| Baseline | 0.3208 | 0.0329 | 0.1250 | 0.8045 |
| Latent Factor | 0.6049 | 0.6316 | 0.5536 | 0.6297 |

Table 3: Accuracy Comparison

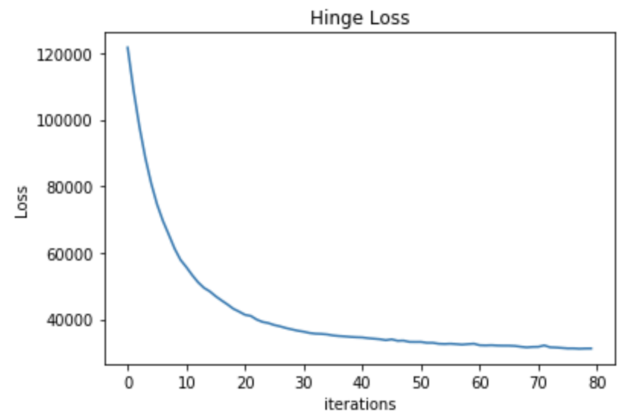


Figure 5: Hinge Loss

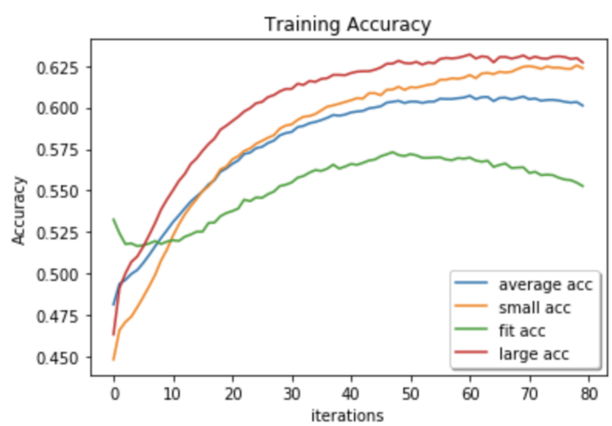


Figure 6: Accuracy

# 6. Future Work

## 6.1. Problem

Even though we improved the baseline accuracy from 30% to 60%. There are still a lot of works needed to be. As we mentioned from the Introduction, **Diversity** is the topic we need to deal with. In practice, a customer account may be shared by multiple individuals with different sizes

### 6.2. Future Work

A hierarchical clustering algorithm for identifying individuals is proposed by Sembium *et al.*. The algorithm is similar to Algorithm 1 except for each iteration, customer will be clustered based on items true sizes if the return value is Fit. The result should be better than the Algorithm 1 and Baseline.

## References


[1] R. A. BRADLEY and M. E. TERRY. Rank analysis of incomplete block designsthe method of paired comparisons. *Biometrika*, 39(3-4):324–345, 1952.

[2] jmcauley. Clothing fit data.

[3] R. Misra, M. Wan, and J. McAuley. Decomposing fit semantics for product size recommendation in metric spaces, 10 2018.

[4] V. Sembium, R. Rastogi, A. Saroop, and S. Merugu. Recommending product sizes to customers. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*, RecSys '17, pages 243–250, New York, NY, USA, 2017. ACM.

[5] L. L. Thurstone. The measurement of values. 1959.