Product Size Recommendation based on Latent Factor and Regression Model

ZHENGMING ZHANG[1], JINGSONG ZHU[2], AND LUYAN HAO[3]

Abstract:

While e-commence becomes popular in the past few years, the return rates and customer satisfaction gained much attention under the online fashion area. Due to size variations between across brands and the lack of standardization, fit prediction becomes critical and tricky, which has a significant influence on customer's rating feedback and reputation building for brands and platforms. Also, a proper recommendation of size could help contribute to a lower return rate. In our study, we focus on a dataset contributed by Rishabh Misra, Mengting Wan, and Julian McAuley in the study progress of [1]. We firstly conducted a brief analysis of the dataset and presented some interesting facts in section 1, following by a baseline predictive model for the size recommendation in section 2 with Latent Factor model. Part 3 gives an illustration of the modifications and improvements we made on the baseline model, including the feature processing progress, the Lasso and Ridge regression we implemented. In the following part 4, one could find information about related works and how our datasets and other similar datasets were used and studied, accompanied by an introduction of state-of-the-art methods in this fashion size prediction area. Section 5 exhibited the results of our models and gave an interpretative conclusion for the whole study.

1 Dataset Introduction and Exploratory Analysis

Basic statistics and properties of the dataset

Dataset: Clothing Fit Data
Dataset Name: Clothing fit data from the website RentTheRunway
Number of users: 105,508
Number of items (cloth leasing): 5850
Number of transactions: 192,544
Example data:


```
{"fit": "fit",
"user_id": "420272",
"bust size": "34d",
"item_id": "2260466",
"weight": "137lbs",
"rating": "10",
"rented for": "vacation",
"review_text": "An adorable romper! Belt and zipper were a little hard …",
"body type": "hourglass",
"review_summary": "So many compliments!",
"category": "romper",
"height": "5' 8",
"size": 14,
"age": "28",
"review_date": "April 20, 2016"}
```


Our dataset contains measurements of clothing fit from RentTheRunway, which is a unique online platform for clothes renting. This platform aims at female customers only. They provide different types of clothes for different occasions. After the transaction finished, our customers would have a chance to give feedback on the specific items they rent for this time.

Exploratory analysis and interesting facts about the dataset

The researchers ingested data from several categories focusing on the section of clothes. In our example data, this category is highlighted under the section: 'category'. Also, the self-reported data conclude the customer opinions in the following aspects: a plain-text review with a short review summary, ratings over 10, the overall fitting

[1] Mathematics Department in University of California, San Diego, A53290669, zhz010@ucsd.edu

[2] Mathematics Department in University of California, San Diego, A53288301 Jzhu2ucsd.edu

[3] Mathematics Department in University of California, San Diego, A53287411 luhao@ucsd.edu

feedback as ‘Fit’, ‘Small’ or ‘Large’. Moreover, the review provides information related to the customer’s personal body measurements. Weight, height, bust size, age, body type, and their regular clothes sizes are revealed through the review. In the end, since the different types of clothes will have different size measures, the researchers converted them all into a single numerical scale, preserving the order — for instance, a bra mark as ‘34D’ based on the bust size. At the same time, a mini dress may only have a size measure out of ‘S’, ‘M’, ‘L’, ‘XL’. Jeans have their own conventions for size, with a number scale of 22 - 34 usually. Researchers extracted them and processed them into an integer measure, which ensured the convenience and authenticity of the research.

Another interesting point worth attention is the following - this set of data is highly sparse, with most of the customers or items only got involved with one transaction. The reason for this phenomenon is quite apparent - since clothes rental is not very common for most people in their daily dress up. From our data, we see that the purpose of renting is for special occasions, like weddings, parties, work, or formal affairs. People need fancy, formal or non-daily clothes for these occasions. Therefore, when they are not willing to buy an expensive for one-time wear, clothes rental becomes their preference. Furthermore, these facts formed the high-sparseness feature for our dataset.

Motivation of our model study from the analysis

There are 192,544 data entries in the dataset. While our ultimate goal is to predict the right size for a given pair of user and item, we will only use data with feature “fit” labeled fit. Moreover, we only used the first 60,000 positive data points for simplicity. It is very “natural” to examine the scatter plots of the data in the first place. We start with selecting the features that could be potentially considered as numeric data, which are “age”,” rating”,” size”, “height,” and “weight”. Note that before plotting, we convert the “height” to be of in inches only. Figure 1 is the scatter plot matrix.

According to the density plots of “age” and “weight”, we can see that the data are concentrated around 40 inches and 130 lbs, respectively. Also, their distribution could be assumed as a normal distribution. However, the density plots of “height”, “rating” and “size” have some sharply rise and fall, which indicates the discontinuity of the features, we might apply one-hot encoding to them. Since our target feature is “size”, we need to examine the relation between it and the other features. Surprisingly, except for versus “weight” plot, which exits an obvious linear relation, the others do not show an explicit linear relation. In order to quantify the linear relation, we use the Pearson correlation coefficient *. The results are presented in Figure 2.

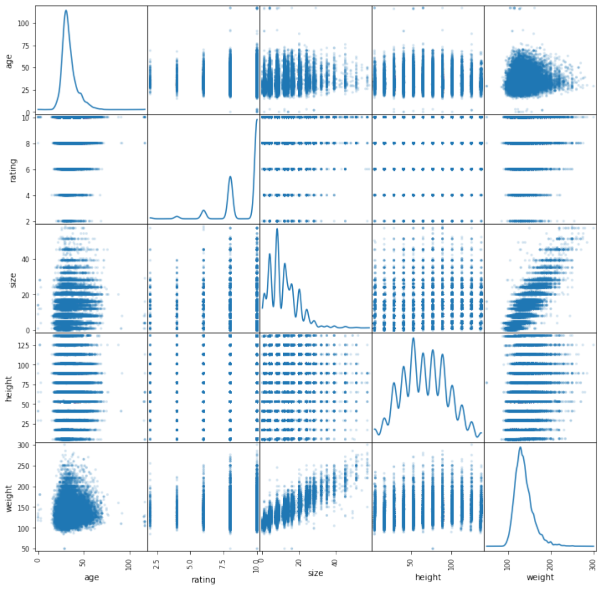

Figure 1. Scatter Plot Matrix (The X labels and Y labels represent that the feature is considered as X and Y

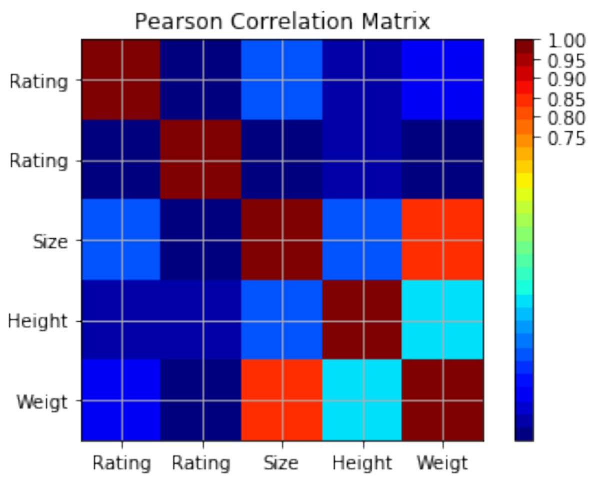

Figure 2. Pearson Correlation Matrix (We take the absolute value for the correlation coefficients, because we only care if

there exists a positive or negative linear relation, and directions do not matter)

We can see that our previous conjecture is correct, where "weight" relates to "size''. To take a glance at the categorical data, we graph the bar plot of "body type" and "rented for" in Figure 3.

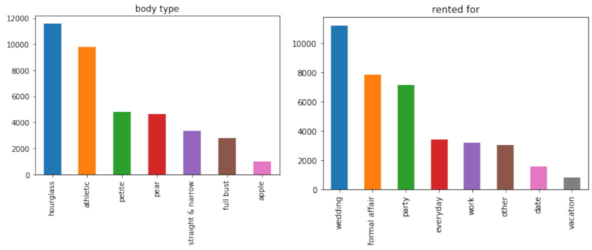

Figure 3. Bar Plots for "body type" and "rented for" respectively

2 Predictive Task Identification

Task definition

We define our predictive task in this assignment as recommending size from 0 to 58 for a given pair of "user_id" and "item_id". The reason we choose this task is that if we could precisely recommend a size of an item for a custom, or narrow down the possible size range, the custom will be unlikely to return the item to the merchant regarding size issue. The recommendation system not only helps the merchant to reduce costs but also helps customers shop more efficiently.

How to evaluate the model performance

The models will be evaluated using the mean squared error (MSE).

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \widehat{Y_i}\right)^2$$

where Y is the true label of the data point, and $\hat{Y}$ is our corresponding predicted size.

To make our predictions valid and examine the real performance of our model, we separate our data into a training set with 40000 data points, a validation set with 10,000 data points, and a test set with 10,000 data points. We trained our models on our training set, tuned regularization parameters on the validation set, test the final performance of each tuned model on the test set.

Feature selection and data processing

We are using the features described in the following Table 1. It is very natural to think that "age", "body type", "bust size", "height" and "weight" has influence on sizes chose. "item_id" and "user_id" could provide us with previous purchase information, and help us to construct the "image" of items and users.

Table 1

Field Name	Description	Usage
Age	Age of User	Continuous
Body Type	Body Type Selected by User	One-Hot
Bust Size	Bust Size of User	One-Hot
Category	Item Category	One-Hot
Height	Height in inches of User	One-Hot
Item ID	Unique Identification of Item	Transformed in Model
User ID	Unique Identification of User	Transformed in Model
Rating	Rating of Item	One-Hot
Review	Review Text of the Item	Sentimental Analysis
Weight	Weight of User	Continuous

Baseline model use for comparison and validation setting of the prediction

We use a Latent Factor model without intersection term as our baseline model. The size prediction is equal to the summation of global bias, item bias, and user bias.

$$f(u,i) = \alpha + \beta_u + \beta_i$$

This model could capture both the preference of the user and the bias term of the item. However, since the model only considers "user_id" and "item_id", it lacks degrees of freedom and depends on past purchase records. For new users and new items, the model will be useless, but there were no previous records to construct the biases. Even for exited users, the model could perform badly in some situations. For instance, the records of the users are too old, and the users experience a notable body transform through a certain period of time. To avoid over-fitting, we tuned the regularization term on the validation set showed in Figure 4. As a result, we choose 0.2 as the regularization parameter, because it attains minimum on validation set, which is 40.3463.

Model development ideas based on baselines and the discussion over the appropriateness of models

To improve the baseline model, though we could use Matrix Factorization model and Latent Factor model, they all share the same defect - they only rely on "user_id" and "item_id". The predicts on new items of new users will be highly unreliable, which is a professional term we call this the Cold Start issue. This problem could be identified by the MSE performance showed in Figure 5, which shows us clearly an unexpected performance of the Matrix Factorization model. Therefore, we choose ridge regression to include more features. It is reasonable to seek linear relations between size and user's body and item information. However, this regression model does not capture the preference of users and the bias of items. Therefore, in this way, our improved combined model would be much more appropriate when dealing with new users and new items as well as the lack of information of Latent Factor model. We can consider the predicts of the previous model as a new feature, which might contain the information the model lacks. Those models were implemented in the following section.

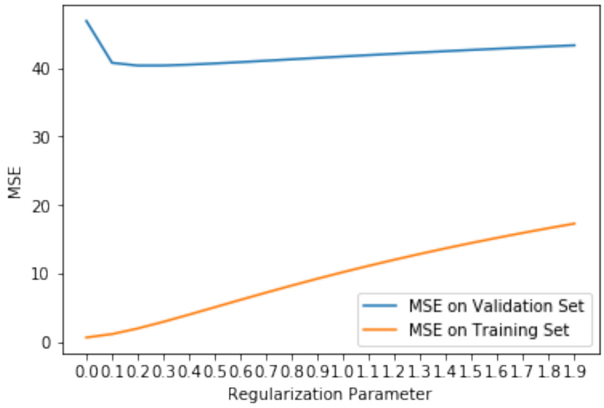

Figure 4. MSE for Baseline Model

3 Predictive Model Details

We begin with the setting in the form of 'user_id' and 'item_id'. Firstly, we use matrix factorization model. The model is specified below:

$$\mathrm{R} \approx \mathrm{P} \times \mathrm{Q}^{\mathrm{T}} = \widehat{\mathrm{R}}$$

where we have a set U of users, and a set D of items. Let R of size $|U| \times |D|$ be the matrix that contains all the sizes that the users have assigned to the items. In this way, each row of P would represent the strength of the associations between a user and the features. Similarly, each row of Q would represent the strength of the associations between an item and the features. To get the prediction of a size of an item d_j by u_i, we can calculate the dot product of their vectors:

$$\widehat{r_{ij}} = p_i^T q_j = \sum_{k=1}^{k} p_{ik} q_{kj}$$

To obtain P and Q, we use gradient descent. We also add a regularization term to avoid overfitting. The modified squared error is showed below:

$$e_{ij}^2 = \left(r_{ij} - \sum_{k=1}^{K} p_{ik} q_{kj} \right)^2 + \frac{\beta}{2} \sum_{k=1}^{K} (|P|^2 + |Q|^2)$$

where beta is our regularization term. However, the performance of the model is very bad. Figure 5 is our MSE plot. We can see that the MSE on the training set is close to zero while the MSE on the validation set is more than 100, and large regularization parameters do not decrease the difference intensively. Therefore, we abandoned this model and moved forward to other models.

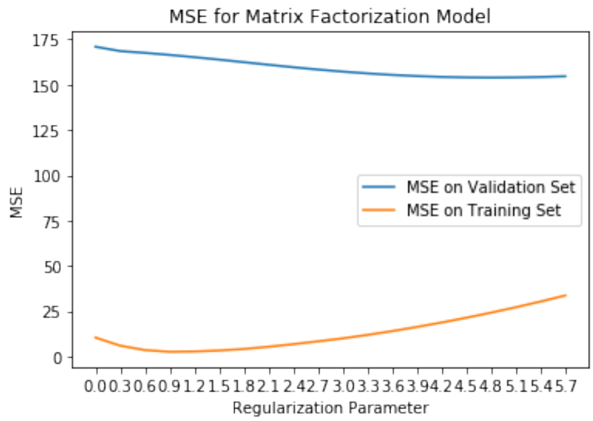

Figure 5. MSE for Matrix Factorization Model

For the previous baseline model, it sometimes performs surprising well. However, it manifests an essential problem. More concisely, we modified such model into the following form in order to find a better outcome:

$$f(user\ features, item\ features) = \{\varphi(user\ features), \theta_{user}\} + \{\varphi(item\ features), \theta_{item}\}$$

From the equation above, we find that it is a simple linear combination between user predictor and item predictor without any interactions. Therefore, for a single user, the model will always recommend the same size with the highest rating item. Therefore, we are trying to resolve this problem with additional interaction terms. The most common and popular model to solve this problem is Latent Factor Model.

Model 1: Latent Factor model

Latent Factors are some "unseen" variables that determines how a user rates a particular item, so if we intend to uncover these latent variables, the right procedure is to predict a rating with respect to certain user and item. For example, given user-item rating system, it is available to represent all the information in a matrix R just like:

	D1	D2	D3	D4
U1	5	3	-	1
U2	4	-	-	1
U3	1	1	-	5
U4	1	-	-	4
U5	-	1	5	4

Figure 6. Example of a Size Cross-Presenting Matrix

Hence, the task of predicting the missing rating can be regarded as filling in the blanks in above graph such that the values would be consistent with the existing ratings in the matrix. Generally, the Latent Factor Model can be expressed as followed:

$$f(u, i) = \alpha + \beta_u + \beta_c + p_u^t q_c$$

Then, we focus on how to figure out the latent-variables term $\gamma_u \gamma_i$. Here, we introduce matrix factorization. Actually, for any represented matrix R, there always is an approximation:

$$R \approx PQ^T = \hat{R}$$

where $P \in \mathbb{R}^{n*k}$, $Q \in \mathbb{R}^{m*k}$, and k is the number of latent variables. Then our main goal is to calculate the matrix P and Q. Here, we apply the famous descent gradient method to solve the problem of matrix factorization. The stepwise algorithm is as the following:

Algorithm of descent gradient matrix in matrix factorization

Step 1: Initial two matrices with some values

Step 2: while $e_{ij} \geq \delta$ do:

$$\hat{r}_{ij} = p_i^t q_j = \sum_{t=1}^{k} p_{ik} q_{kj}$$
$$e_{ij}^2 = (r_{ij} - \hat{r}_{ij})^2$$
$$p'_{ik} = p_{ik} + 2s_1 e_{ij} q_{kj}$$
$$q'_{kj} = q_{kj} + 2s_1 e_{ij} p_{ik}$$

Step 3: Reform matrix $\hat{R} \approx PQ^T = \sum_{t=1}^{k} p_{ik} q_{kj}$

The gradient descent method is a quite basic algorithm for factorizing a considerable size of matrix. Yet, in our Latent Factor model, it not only considers the interaction terms but also adds some biases terms that may give information about user and item features individually. Besides that, it also plausible to introduce a regularization to prevent overfitting. In this way, we express the algorithm of Latent Factor model showed in the following pseudo code. While in this algorithm, μ is the regularization parameter, s_1 is the step length and α can be considered as a global mean of rates.

Algorithm of Latent Factor Model

Step 1: Initial two matrices with some values

Step 2: Update the estimation of parameters

while $e_{ij} \geq \delta$ do:

$$\hat{r}_{ij} = p_i^t q_j = \sum_{t=1}^{k} p_{ik} q_{kj}$$
$$e_{ij}^2 = (r_{ij} - \hat{r}_{ij})^2$$
$$\beta'_{u_i} = \beta_{u_i} + s_1(e_{ij} - \mu\beta_{u_i})$$
$$\beta'_{c_j} = \beta_{c_j} + s_1(e_{ij} - \mu\beta_{c_j})$$
$$p'_{ik} = p_{ik} + s_1(2e_{ij} q_{kj} + \mu p_{ik})$$
$$q'_{kj} = q_{kj} + s_1(2e_{ij} p_{ik} + \mu q_{kj})$$

Step 3: Calculate the prediction:

$$rating = \alpha + \beta_u + \beta_c + \sum_{t=1}^{k} p_{ik} q_{kj}$$

Model 2: Lasso and Ridge regression model and optimization of the model

Firstly, to make our model more appropriate, we optimized the features we have. We categorized the variables in data set into 4 classes: continue variables, categorical variables, reviewing variables and handled variables. For continue variables such as height and weight, we standardized measured of height to all be inch and abstract the digit term from weight. For categorical variables such as body type, bust size and category, we treat them to be dummy variables. For reviewing variables, we apply unigram method to record the frequency of each word (throwing stop-words and punctuation mark), and choose the top 1000 highest-frequency word to be our variable, and the following is the most frequently appeared words' list:

Table 2

Word	Frequency
Dress	23382
Compliment	11274
Great	7763
Perfect	6951
Love	6946
Many	6854
Beautiful	6118

Intuitively, we can have a better view of the word frequency in the following charts:

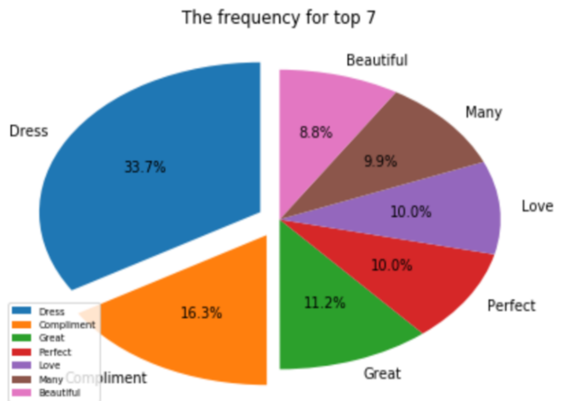

Figure 7. Bar Chart for Most Common Words

Figure 8. WordCloud Graph of Common Words in Reviews

Finally, in order to handle variables such as user_id and item_id, the result of Latent Factor model mentioned above has abstracted the information of these two variables, so we only need to replace original variables (user_id, item_id) by the result of our Latent factor model.

Lasso and Ridge Regression are introduced for the purpose of improving the prediction accuracy. Interpretability of regression models could be realized by altering the model fitting process to select only a subset of the provided covariates for use in the final model. Both of them impose a penalty term into the complexity of model to prevent overfitting. Our objective function can be expressed as:

$$min_\beta \{1/n \sum_{i=1}^{n}(y_i - x^t\beta)^2\}$$

subject to $||\beta||_i \leq t$

Notice that the only difference between Lasso and Ridge Regression is different measures of β in the constrain part $||\beta||_i \leq t$. Lasso has L_1 penalty while Ridge Regression has L_2. Geometrically, the process of these two regressions can be viewed as:

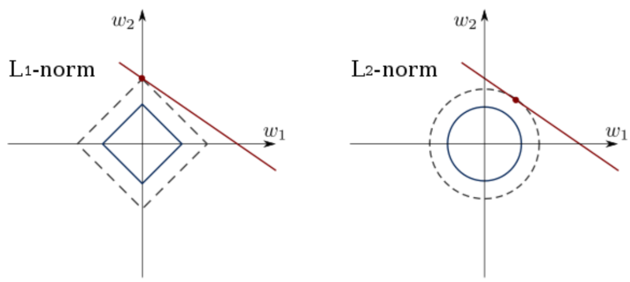

Figure 9. Illustration of Lasso and Ridge Regression

By the end of this section, our theoretical discussion over the multiple choice of models was presented. In the rest of this section we are going to discuss also the weakness and strength between the options we have so far, while the implementations and results of the above models were presented in section 5.

Failures, overfitting issues and our decision of the final model

The simple Latent Factor model did not have a very ideal performance. Actually, it becomes a failure due to the lack of information as we discussed before. Yet, instead of getting rid of this failed model, we actually decided to use it in another way. That is, we use it as a new feature into our following regression model. During our attempts of size prediction, the overfitting problem appeared in most of the cases. To resolve this conundrum, we thought through the Ridge regression and Lasso model. Our final model landed on the Ridge regression. The reason we choose this model is as following: Ridge regression preserves the information of variables in the progress of regression. Since Lasso will eliminated the variables that does not have a very significant influence on our output. However, this will result in an omitted variable bias. Since in our model, there is thousands of variables due to the one hot coding. And some of them may have a close contributed level, so we want to preserve the information insights from them instead of just eliminate them. In fact, our final results coincide with our inference, which could be found in section 5.

Strength and weakness of our model

The Latent Factor model considers the latent interaction between user features and item feature to address the problem of linearity which manifested in baseline model $f(u,i) = \alpha + \beta_u + \beta_i$ and it, actually, succeeds in showing the collaborated preference. However, the final results of Latent Factor model on test data set is not pretty good. The main reason here is that The Latent Factor model only considers two variables (user_id, item_id) ignoring other variables which lead to the information losses. In this way, we use Ridge regression and Lasso to incorporate the result of Latent Factor model to derive a more favorable estimation. In fact, the final model we choose is the mixture of Ridge regression, text mining and Latent Factor model which means it inherits the advantages of these model to some extent. More concisely, it not only considers the latent interaction between user and item, but also contain enough information to guarantee accuracy of prediction. Besides that, introducing a slight a square bias into the model is also conducive to variance reduction.

4 Related literatures and comparisons between studies

Where our dataset comes from

Our dataset was contributed by Rishabh Misra, Mengting Wan and Julian McAuley in the study progress of [1]. As mentioned before, they collected the data from RentTheRunway, an online website for female clothes rental. Since the area of size recommendation is recent compared to other general recommender systems, articles in this area were commonly new and novel. In fact, there were few articles proposed on this particular area so far [1, 2, 3, 4, 10]. Though in fashion area related to shoes and apparels there were some past study like [11] recommend items based on users' past interactions. Yet, they conducted researches about the taste preference instead of size-wise. As the popularization of the e-commerce, where trying on items are not available like usual shops, size preference gained more attention.

Other similar datasets have been studied

Similar datasets were used through all the size recommend problems. Amazon shoe datasets were collected for the study in [2], since all three researchers worked in Amazon while the research was done. The data contains transactions consisting of customer information, item information and return status. For a given (user, parent product) pair (where parent product refers to a particular piece of clothes while the child products are the different sizes of this piece of clothes), the model could recommend a child product with a specific size for this customer. In [3], engineers from Myntra Designs used the sales data collected from their own platform between Jan 2015 and Feb 2017. Same as most of fashion size recommendation study, their data was split categorically. The outstanding point is that their data was trained for different categories. This dataset is quite large, with only the 'Men Shirts' department provided over 300K users. But, some of the users may buy stuff for another person as gifts, so the size varies in a wide range. To identify those users whose account contains 'potential other users', they computed the mean and standard deviation of the purchases size for each account. Only those accounts with a standard deviation under the empirical threshold were included in their following study. And the use of these datasets was discussed in the following paragraph.

How our dataset and other similar datasets were used and introduction of state-of-the-art methods

In [1], researchers proposed a state-of-the-art framework in order to predict the product fit problem. To tackle this recent conundrum, our researchers employed two informative tools – the semantics coming from the users' feedback text and a metric learning technique to resolve the label imbalance issues. These tools would be the most distinct features that separates this paper with previous articles.

Based on these two instruments, the researchers implemented them into two different steps. The first step is developing a model to factorize the customers' plaintext fit feedback into semantical pieces, which could help the model capture a customer's fit preference on different product parts (like shoulders, chest, sleeves etc.) In this way, instead of only considering the overall size, the model could further learn from the fit preference in a more specific way. For example, a skinny user may have a usual size of '2' for all the tops, but she may need a '4' for her jeans because the long legs. Also, rompers may have a slim fit while the boyfriend style hoodies may intend to have an oversize fit. Specific demands based on the users' personal body shape could improve the model's performance to a large extent, as well as the products' particular style (which could conclude from users' feedback on the same item). In [1], this study applied an ordinal regression procedure to learn these words like 'shoulder', 'sleeves' so that the model could preserve the order of labels. Aiming at this fit semantics factorization, embedding over both customer and product into ordinal labels were implemented. Mapping the labels into a latent space by assigning a score to each transaction, which is indicative of the fitness of a corresponding product on a corresponding customer. The score/label for this transaction is ordinal as we mentioned before, with three statuses {'small', 'fit', 'large'}. Other than work [2], where the researchers cover the products' and users' overall size as features and developed an approach to predict the fitting label as a classifier, this embedding procedure improve the fitting prediction significantly using the same setting. In 2017, [2] proposed a Latent Factor model for recommending a size to customers only based on the sizes of the (user, item) pair. For each pair of (user, item) input, their model predicts a score outcome which is imply a linear function between the customer and product sizes. They implemented latent factor models. The latent factor models in [5, 6, 7] were based on matrix factorizations. At the same time, there are studies like [8, 9] also extended the previous work on Latent Factor model. However, studies focus on size recommendation are not very common. The Latent Factor model in [2] is novel since the problem researchers facing is a specific recommendation problem, and is not suitable for general solutions. The new model they developed is distinct from previous work and highly related to paired comparison models, which is pretty common in psycho-metrics for comparing choices or in sports ratings. Therefore, the study in [2] serves as a corner stone for the study in [1]. As the settings were inherited from [2], the models in [1] were highly aggregated and well-improved based on previous work. In [3], a novel recommender system was presented with the employing of Gradient Boosted classifier (GBC). It takes two vectors as input which implies both the parent product's and user's information collected from user's reviews. The study combined latent features learnt from word2vec (which is a group of related models to produce word embeddings) using the user's non-returned purchase and product content data, also the observed features from product catalogue. What is more, the latent feature vectors are generated using a skip gram instead of unigram or bigram we have seen. Finally, the performance of this combined model surpasses separated models in almost 7-8% higher precision and accuracy rate with 75.52% precision and 81.28% accuracy, which is greatly impressive.

Comparison between our results and previous works

Compared to the related articles, our study predicted the size number instead of label classifier as {'small', 'fit', 'large'}, which is commonly used in previous work. Of all the classifier study before, they measure the model's performance based on AUC, precision, and accuracy. Yet, in our study, the results come in the form of MSE on the size number, which could be found in the next conclusion section. Though the performance were presented and measured in a different form, our model has a similar predict outcome when compared to the work in [1] as the wrong fit prediction has a relatively small range.

5 Result and Conclusion

Performance of single Latent Factor model

We abstract two variables user's id and items' id from our data set, and divided this data set into 3 sets: training, validation and test. Firstly, we train the model with our training set and then utilizing validation set to choose the optimal number of latent variables k, the graph of $k\ vs\ MSE$ is in Figure 10.

From the Figure 10, we will choose $k = 5$ with lowest MSE to complete our model. Finally, we apply our Latent Factor Model into test data set and find the $MSE = 24.846261$. In fact, the MSE is so large that the Latent Factor model cannot be applied to predict individually. However, this model indeed reflects the preference of user and item to some extent, so, we consider using the result of rating as a variable to replace the original variable (user_id, item_id).

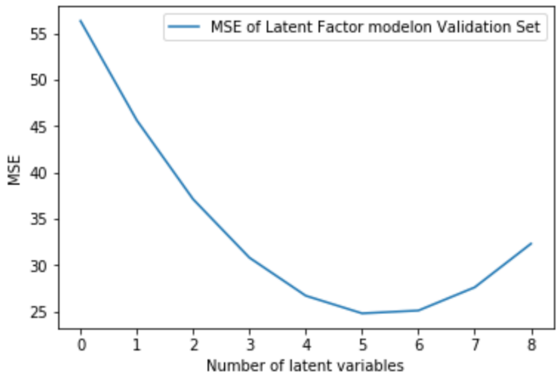

Figure 10. Latent Factor Model Tuning Graph

Performance of Lasso and Ridge regression model based on Text Mining and Latent Factors

Firstly, we consider three kinds of variables (continue, categorical, reviewing) to fit the model without information about user id and item id, and the result shows that $MSE = 18.698$ on test data set which is not so good. Therefore, we are going to incorporate the information in the result of Latent Factor model into our Lasso and Ridge Regression. Similarly, the penalty parameter can be chosen by the Figure 11.

From Figure 12, both of models improved a lot compared with the model mentioned above and the $MSE = 14.10156$ with corresponding penalty parameter $\alpha = 17$ for ridge model and the $MSE = 14.23041$ with corresponding penalty parameter $\alpha = 2$ for Lasso. Our final decision landed on Ridge Regression model because of its smallest MSE measure. Eventually, we perform our chosen model on test data set, and the test result shows that $MSE = 11.06978$. At the same time, our actual size is in the range of $(0,58)$. If we take the square root of the MSE result, roughly we only have a difference of 3 out of 58 in a single size prediction. We have to say this model's performance meet our expectation.

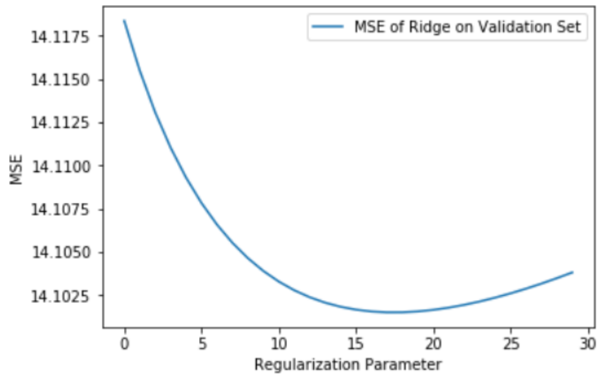

Figure 11. Ridge Regression Tuning Graph

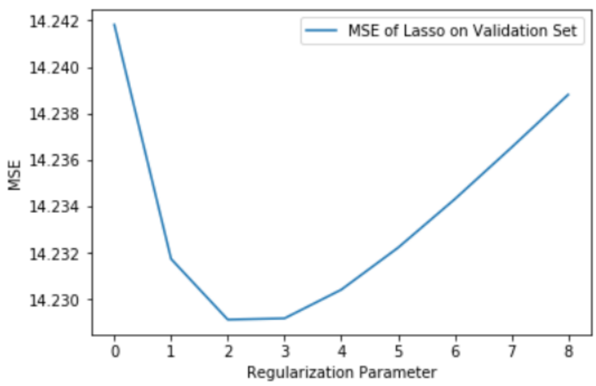

Figure 12. Lasso Tuning Graph

From the results, we could see that our size predict is quite close based on the MSE we have on the test set. Even we did not have it in the label form as mentioned before, the outcome reached our expectation. From the results, we could see that in our final model of Ridge Regression model, the feature that contributed the most is among the plain text in our customer review.

Parameter interpretation

To have a clear view of the features' contribution to our model, we listed the parameters with the largest absolute value and their relative variables.

From the Figure 13, "dress" under the category with the one hot coding contributes the most to our size result. On average, if the item is a dress, the sizes should be recommended larger for the customer. That is an understandable phenomenon since most of dresses is tight and have a smaller fitting tendency. The "34D" comes as the second indicative feature, because a large bust size would require a larger item. While the increment on "weight" have a very significant positive effect of the size recommendation. Words in the review text like "big" and "didnt" tend to have a negative influence on our size recommendation. That makes sense because the review has a "big" in it implies the actual size we recommend should be smaller.

	variable	estimation
0	dress	3.824637
1	34d	2.278401
2	weight	2.087047
3	great	-2.359122
4	big	-2.197106
5	didnt	-2.157989

Figure 13. Most Contributive Features and Relative Parameters

While some of our features contribute little to our fit prediction. For instance, "give" in the review text and "bold" have a parameter value close to 0.
We would say that our model did an impressive job when predicting the size for customers. Our baseline model did not work out very well because it only considered the features of user_id and item_id, which results in a deficiency of feature aspects. Also, it could not handle the issue of "cold Start", which means when we have a new user or new item comes in. Similarly, the failure of the simple Latent Factor model was due to the same reason. The features they considered were only the user_id and item_id, which eliminated some important variables.

Conclusion of the study

In the progress of our study, we chose the dataset of clothing fit data from thanks to the contribution of Rishabh Misra, Mengting Wan, and Julian McAuley. We proposed several predictive models where given a particular pair of input $(user, item)$, we recommended the "right" size for this user over a size range of $(0, 58)$. Specifically, we adopted a combination of Latent Factors and text mining techniques to process the features and decompose the review text into unigrams respectively. Then, we used them as features in our regression models as well as other features extracted from the user and item information. After a brief comparison of several models' performances separately, we found the Ridge Regression achieved the lowest MSE on our test set, which is computed by the squared difference between the actual size of user and the recommended one. Our best model achieved the lowest MSE of 11.06978 with selected features and Ridge regression. Quantitative and qualitative results on the dataset show the effectiveness of our proposed framework.

6 Reference

[1] Misra, R., Wan, M., & Mcauley, J. (2018). Decomposing fit semantics for product size recommendation in metric spaces. *Proceedings of the 12th ACM Conference on Recommender Systems - RecSys 18*. doi: 10.1145/3240323.3240398

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

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

[4] 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*.

[5] Ruslan Salakhutdinov and Andriy Mnih. Probablistic Matrix Factorization. *NIPS 2008*.

[6] Yehuda Koren, Robert Bell and Chris Volinsky. Matrix Factorization Techniques for Recommender Systems. *IEEE Computer 2009*.

[7] Asynchronous Distributed Matrix Factorization with Similar User and Item Based Regularization, Bikash Joshi, Franck Iutzeler, Massih-Reza Amini, *ACM Conference on Recommender Systems, Recsys 16*

[8] A Framework for Matrix Factorization based on General Distributions, by Josef Bauer and Alexandros Nanopoulos, ACM Conference on Recommender Systems, Recsys 14

[9] Modeling the Dynamics of User Preferences in Coupled Tensor Factorization by Dimitrios Rafailidis and Alexandros Nanopoulos, ACM Conference on Recom- mender Systems, Recsys 14

[10] Guigourès, R., Ho, Y. K., Koriagin, E., Sheikh, A.-S., Bergmann, U., & Shirvany, R. (2018). A hierarchical bayesian model for size recommendation in fashion. *Proceedings of the 12th ACM Conference on Recommender Systems - RecSys 18*. doi: 10.1145/3240323.3240388

[11] Y. Hu, X. Yi, and L. S. Davis, “Collaborative fashion recommendation: a functional tensor factorization approach,” in *Proceedings of the 23rd ACM international conference on Multimedia. ACM, 2015, pp. 129–138*.