# Comparing the Effectiveness of Different Imputation Methods

JORDAN ESIASON*, JUSTIN GEE*, JINGSHENG HUANG*, YINGYI LI*, AND CHANG YU*

May 1, 2019

**Abstract**

We extended the movie recommendation lab by adding an imputation step before model fitting. We compared the effectiveness of mean and GMM imputation methods when they are each followed by fitting imputed data with the same linear regression model. We evaluated performance by doing cross validation and prediction using the test set. Although no significant difference observed when each imputation method was combined with the downstream regression step, we established a framework for testing and comparing different imputation method and regression model fitting combinations using sparse data.

*Keywords:* Imputation, Gaussian Mixture Modeling, Mean Imputation, Missing Data

## 1 Background

Missing values in the input data matrix is a widespread problem in all different field since a lot of statistical or machine learning algorithms do not tolerate

*College of Natural Sciences, University of Massachusetts Amherst

such a situation. A commonly used approach is to remove the samples with missing values from the dataset. However this can cause further problems if the sample size is small, or the data is parsed, and important observations can be removed. Alternatively, imputation can be done, by replacing missing values with some reasonable guesses. Different guessing strategies are available, for instance, mean, random, last value carried forward, k-nearest neighbor, Gaussian mixture model.

Gaussian Mixture Modelling (GMM) is a very popular alternative to imputation methods such as mean imputation due to the fact that it optimizes several parameters of potential distributions within a data set independently [1]. The model assigns a weight $\phi_k$ (the probability that a point belongs to a given cluster), a mean $\mu_k$, and a standard deviation $\sigma_k$ to each individual cluster. This gives it the ability to fit Gaussian models to non-spherical clusters of data that cannot be easily visualized.

When building such a model, the parameters for each cluster in GMM are estimated with the expectation maximization (EM) algorithm, which is an iterative algorithm that finds the maximum likelihood estimators [4] [5]). It consists of an expectation step (E-step) where it calculate the likelihood ($\gamma_{ik}$) of each sample i belongs to the kth cluster:

$$\gamma_{ik} = \frac{\hat{\phi}_k \Theta(x_i|\hat{\mu}_k,\hat{\sigma}_k)}{\Sigma_{j=1}^{N} \hat{\phi}_j \Theta(x_i|\hat{\mu}_j,\hat{\sigma}_j)}$$

And a maximization step (M-step) where it maximizing the likelihood by updating the parameters accordingly:

$$\hat{\phi_k} = \Sigma_{i=1}^{N} \frac{\hat{\gamma_{ik}}}{N}$$
$$\hat{\mu}_k = \frac{\Sigma_{i=1}^{N} \hat{\gamma}_{ik} x_i}{\Sigma_{i=1}^{N} \hat{\gamma_{ik}}}$$
$$\hat{\sigma}_k = \frac{\Sigma_{i=1}^{N} \hat{\gamma}_{ik} (x_i - \hat{\mu}_k)^2}{\Sigma_{i=1}^{N} \hat{\gamma}_{ik}}$$

The algorithm iterates the E- and M- steps until convergence with tolerable error, or until desired number of iterations completed [4] [5]). By combining GMM in EM algorithm with other statistical model diagnostic methods, one can create well-fitting models for potentially unexpected clustering scenarios. In this study, we compared the effectiveness of mean and GMM imputation methods when they are each followed by fitting imputed data with the same linear regression model. Methodology (Methods are described in a reproducible way)

# 2 Methods

## 2.1 Dataset used and separation of training and test set

We used the dataset provided in the movie recommendation lab, which consisted of 31,620 rating events where 2,353 users rated a subset of 1,465 movies. We separated a random subset of 300 rating events from all data and saved separately as the test set for our regression model. The rest of 31,320 ratings were referred in the rest of the paper as the training set.

## 2.2 Summarizing characteristics of movies and users

We constructed the *movies* table, which for all movies that were in the database, regardless of whether they were rated in training set, test set or both, we calculated how many years had the movies been released. In addition, we also created 18 dummy variables (Action, Adventure, Animation, Childrens, Comedy, Crime, Documentary, Drama, Film-Noir, Horror, Musical, Mystery, Romance, Sci-Fi, Thriller, War and Western) to indicate which one or few genres a movie fell under. MovieID was the unique identifier for each tuple in this table.

We also constructed the *users* table, for all users using the originally provided age and gender information and the ratings in the training set. Each tuple in this table had a unique identifier userID, and the following characteristics summarized: for each of the 18 genres, how many movies had this user rated that fell into this genre (genre$_c$), and what is the average rating to these movies given by this user (genre$_{ar}$). If a user had ever rated any movie in any genre, then the corresponding genre$_c$ would be 0, and genre$_{ar}$ would be NaN (a true missing value). If a movie had three genre labels, then it will be counted three times, once each in the three corresponding genres. After the *users* table was fully constructed using the training dataset, we found the percent missing, mean and variance for each genre$_{ar}$. Imputation

All imputation was done on the *users* table. We only filled missing average ratings for the genres with 50% or less missing. The *users* table was either filled by solely applying mean imputation, or filled by applying GMM imputation followed by mean imputation.

## 2.3 Mean imputation

For each column to be imputed, we used pandas.DataFrame.mean() function to find the mean of existing values in the column. Then we replace the cells with NaN in that column with the calculated mean. The resulting dataframe was saved separately from the original.

## 2.4 GMM imputation

When selecting a number of clusters to attempt to fit the data (k-selection), three factors were considered: available data, Bayesian and Akaike Information Criterion (BIC/AIC), and processing power [2] [3]. First, an upper bound was set on k based on the amount of data available after excluding rows with NaN values in the column being imputed. Then BIC/AIC analysis was performed within these bounds to narrow possible values further. Finally, considerations were made for processing power.

In order to create the Gaussian Mixture Model, we used the sklearn.GaussianMixture($n$_components=5, covariance_type = full) function. The function sklearn.fit_predict(X, y=None) was used to fit the model, where X is array-like. Inputs for this function were modified by using the function numpy.vstack, which takes the columns of ratings from the list of genres used for imputation and formats the values into a 1 x (number of columns used * the number of users in the subset) array. Using the sklearn.fit_predict function, we get an array of Y clusters ranging from [0, $n$_components - 1] where Y is the number of users in the subset. A new array containing average values for each cluster was created using the functions sklearn.fit and sklearn.means_. Next, cluster averages were mapped to their corresponding clusters and movie ratings were imputed into the original dataframe *users* for each user based on their cluster assignment.

## 2.5 Fitting linear model and cross validation

The pandas.DataFrame.merge() function was used to merge each completed *users* table with the movie table and the *ratings* table. The *ratings* table was first left merge with the *users* table on userID, and then left merge with the *movies* table on movieID.

The LinearRegression() function from the sklearn.linear_model was used to fit the linear regression model. There were 24 independent variables in this model, including the average scores and counts for Comedy, Drama, and Action; the other fifteen genres, and gender as dummy variable; as well as the age of the users and how many years has the movie been released. The rating was the dependent variable.

The cross_validate() function from the sklearn.model_selection was used to evaluate the performance of the linear regression model by performing 10-fold cross validation. The scoring strategy used was neg_median_absolute_error. The average score of the cross validation was found for each fitted model.

### 2.6 Evaluating performance with the test set

The LinearRegression.predict() function from the sklearn.linear_model was used to predict the ratings for the test set with the two linear regression models that we obtained. The errors between predicted ratings and true ratings were calculated. The distribution of individual error was plotted and, the mean absolute error for predictions made with each regression model was also calculated.

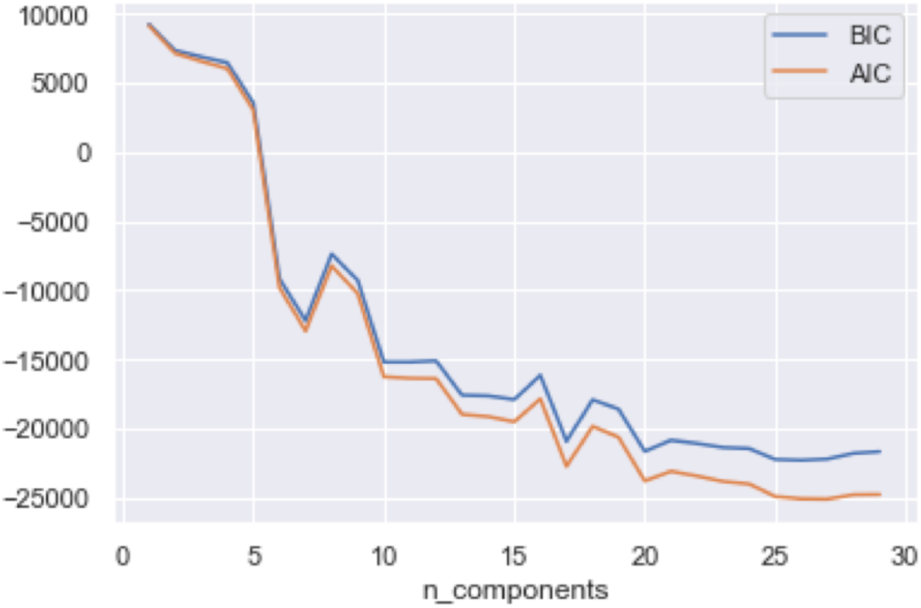
## 3 Results

When the *users* table was first built by using training data, we found three genres with less than 50% average rating values missing:Action$_{ar}$: 48.4%, Comedy$_{ar}$: 28.4% and Drama$_{ar}$: 30.0%. The other genres had percent missing varying from 54.6-93.1%.

After excluding NaN values from a given genre to impute we were left with about 1,200 data points in the more complete genres. A smallest useful sample size of 30 data points was chosen, resulting in the largest possible k of 40. Next, the BIC/AIC of various GMM models was analyzed and we determined that 5 clusters would be sufficient due to the significant drop in BIC/AIC values between 5 and 7 clusters (Figure 1). Improvements could be gained with 15-20 clusters, however, the number of test simulations that needed to be run became prohibitive.

*Figure 1: BIC/AIC analysis of n GMM clusters over the Action genre*

Before any imputation, 865 users out of 2,353 had all three genres of interest



with no missing value. GMM imputation was able to complete an additional 1,140 users profiles. Mean imputation was also able to replace any missing values in the dataframe. In general, mean imputation led to a decrease in the standard deviation of the variable, while GMM imputation might lead to slightly increased or decreased mean and increased variance (Table 1).

| | | Action_ar | Comedy_ar | Drama_ar |
|---|---|---|---|---|
| Before Imputation | Mean | 3.3617 | 3.3708 | 3.6539 |
| | Std. | 0.9063 | 0.9597 | 0.8917 |
| After Mean Imputation | Mean | 3.3617 | 3.3708 | 3.6539 |
| | Std. | 0.6505 | 0.8119 | 0.7458 |
| After GMM Imputation | Mean | 3.1979 | 3.3834 | 3.4313 |
| | Std. | 1.1445 | 1.0694 | 1.0930 |
| GMM followed by Mean Imputation | Mean | 3.1979 | 3.3834 | 3.4313 |
| | Std. | 1.1094 | 1.0342 | 1.0388 |

*Table 1. Descriptive statistics of variables before and after imputation.*

Both cross validation score and prediction error for the model built with dataset imputed with either mean or GMM imputation showed no significant difference. The average score of the cross validation for mean imputation was 0.6626 and was 0.6718 for GMM imputation. The mean absolute error for predictions with the mean imputation was 0.8161, and 0.7876 with the GMM imputation. The distributions of error for each

imputation method also showed no significant difference (Figure 2).

Both cross validation score and prediction error for the model built with dataset imputed with either mean or GMM imputation showed no significant difference. The average score of the cross validation for mean imputation was 0.6626 and was 0.6718 for GMM imputation. The mean absolute error for predictions with the mean imputation was 0.8161, and 0.7876 with the GMM imputation. The distributions of error for each imputation method also showed no significant difference (Figure 2).

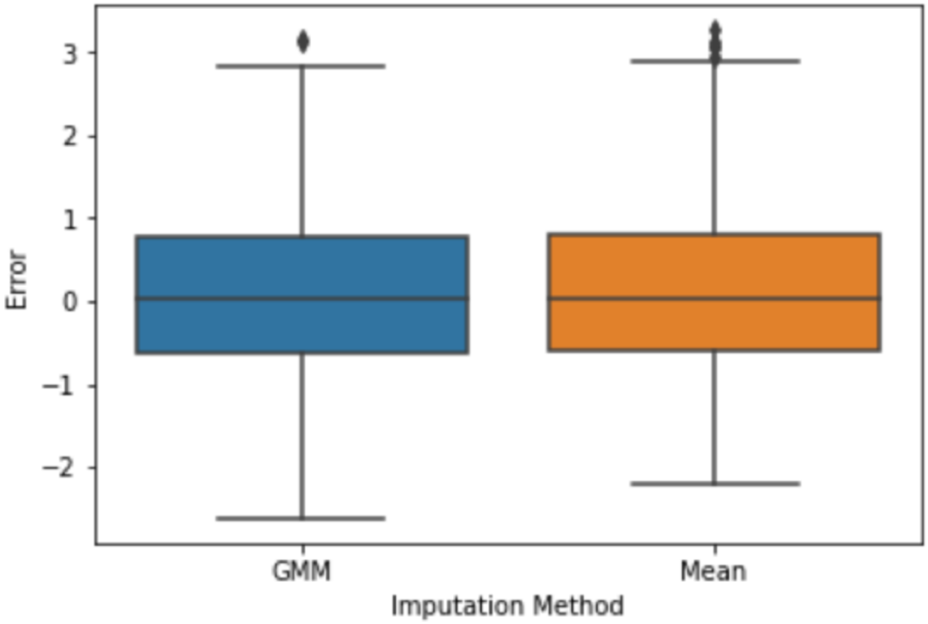


*Figure 2. Error distributions for predictions made with either model using the test set.*

## 4 Summary

Overall, it was a reasonable approach to summarize selected characteristics of the users and movies they had rated, and to use the characteristics as independent variables for the regression model, to predict rating scores, the outcome.

Mean imputation, in theory, could impute every missing value in the input matrix, regardless of what percent of this variable was missing, or the status of other variables. While GMM imputation, in our project, was able to impute the majority of the missing values and was limited by the algorithms requirement for some input data for clustering. Thus a combination of mean and GMM methods could be used when imputing input matrix with different degrees of completeness.

When each imputation method was used to replace missing values in the input matrix for the same linear regression model, we did not observe a significant difference in prediction accuracy for each fitted model. Both the cross validation scores and the prediction errors using the test set indicated that neither of the combinations outdid the other. One explanation was that both methods were replacing the missing values with some averages. Whether it was the average of the entire column or of a cluster, the imputed values were far less customized to a particular user when compared to values imputed with the k-nearest neighbor method, or other methods that aimed to make a guess that would be closer to the true value.

The project could be extended in different ways: other imputation methods could be used in combination with the linear regression model that included interaction terms or other regression models. Although no difference observed for the two chosen imputation methods, our project provided a framework for testing and comparing different imputation method and regression model fitting combinations using sparse data.

# References

[1] Ouyang, M. Welsh, WJ. Georgopoulos, P. *Gaussian mixture clustering and imputation of microarray data*, Bioinformatics, 20(6) (2004), pp. 917-23.

[2] Schwarz, G. *Estimating the Dimension of a Model*, Annals of Statistics, 6 (1978), pp. 461-464.

[3] McKenzie, P. Alder, M. *Selecting the optimal number of components for a Gaussian mixture model*, in Proceedings of 1994 IEEE International Symposium on Information Theory, Trondheim, Norway, 1994, pp. 393-

[4] Beale, E.M.L. Little, R.J.A. Missing values in multivariate analysis J. R. Statistical Soc., Series B, 37 (1975), pp. 129-145

[5] Hedderley, D. Wakeling, I. A comparison of imputation techniques for internal preference mapping using Monte Carlo simulation. Food Quality and Preference, 6 (1995), 281-297.