Multiple Environment Null Value Imputation With Mixed Effects Linear Models

Yonatan Davidov & Alon Jacoby
Department of Computer Science
Bar-Ilan University
Ramat Gan, Israel

Abstract

In a typical pipeline of data analysis, the first crucial phase is data preprocessing. One of the most common preprocessing tasks is missing value imputation. However, data may be sourced from multiple environments, each inducing different distributions of both the data itself, and the rate that each feature may be missing for any given sample. Therefore, typical methods for data imputation could be insufficient to generalize well during test time, when these rates cannot be estimated (for example in online inference). In this work, we propose a method for imputing missing values in a dataset, which is based on a mixed-effects linear model. Our method is designed to handle multiple environments. We show that our method outperforms existing imputation methods on a variety of datasets, including both synthetic and real-world datasets. We also demonstrate that our method is robust to the choice of hyperparameters and the number of environments, and that it generalizes well to unseen environments. The code for our method is available at https://github.com/alonj/Multiple-Environment-Imputation.

1 Introduction

We are interested in a method that can impute missing values in a dataset, when the data is sourced from multiple environments: Our goal is to improve the accuracy of null value imputation in cases where the distribution of data in some, or all of the features is conditioned on the identity of the cluster to which the sample belongs. Furthermore, there may be yet-unknown clusters in the test data, and the imputation method should be able to generalize to these.

Formally, we define the problem as learning the distribution of $P(X^i|Y = y)$ where X^i is the i-th feature in the dataset, and Y is the cluster to which the sample belongs. In such a scenario, simple imputation (disregarding information about the cluster) may not be sufficient, as it may not capture the distribution of the data in each cluster. In particular, such methods are inadequate if $\mathbb{E}(Y_{train}) \neq \mathbb{E}(Y_{test})$ or $Var(Y_{train}) \neq Var(Y_{test})$, as the shift in distribution is then conditioned on a latent variable that is otherwise not observed in the data and cannot be easily estimated in an online setting.

Ideally, an appropriate method for such imputation will achieve lower error rates, given a metric, on both in-sample examples as well as out-of-sample examples, and will be robust to the choice of hyperparameters and the number of clusters, compared to simple imputation.

In the following sections, we describe our method to impute in a multiple-environment setting (§3). We then present the data we chose to evaluate our method on, and the results of our experiments (§4). Additionally, we explore the generalizability aspect of our method compared to other baseline methods (§5). Finally, we discuss our results as well as some limitations (§6).

2 Related work

Tian et al. (2015) test four methods on six datasets (four simulated, two real-world data). The first is Multiple Agglomerative Hierarchical Clustering (MAHC), in which they iteratively cluster sample environments and complete null data using the respective features of other members of the cluster, directly. Second, in their Normal Distribution Model (NORM) they assume a distribution on multivariate normal distribution for each feature across environments, and estimate its mean vector and variance matrix using both Bayesian and non-Bayesian inference. In their Normal Regression Model (NRM) they estimate a regression model by constructing the design matrix as a wide matrix of each feature in each environment, for each sample. Finally, they also employ a Predictive Mean Matching (PMM) method for the choice of completion values for each missing value, by predicting also for non-missing values and choosing nearest values among those.

Resche-Rigon & White (2018) propose a two-stage approach in which imputation models are first fit for each cluster of the data, and then the models' decisions are merged using an additional model (an ensemble). They find that in such settings, it is important to consider heteroscedasticity and variation of means of the clusters: The distinct distribution of each cluster cannot be ignored in a general model for two-level data. This in particular motivates our work to consider a mixed-effects model for imputation.

3 Mixed-effects Linear Models for Imputation

A mixed-effects linear model is a generalization of a linear model that allows for both fixed and random effects. In our case, the fixed effects are the features of the dataset, and the random effects are the clusters to which the samples belong. We assume that the data is generated by the following model:

$$X^i = \beta_0 + \sum_{j=1}^{p} \beta_j X^j + \sum_{k=1}^{K} \gamma_k Y_k + \epsilon \tag{1}$$

Where X^i is the i-th feature in the dataset, Y_k is the k-th cluster, β_0 is the intercept, β_j are the coefficients of the fixed effects, γ_k are the coefficients of the random effects, and ϵ is the error term. We assume that the error term is normally distributed with mean 0 and variance σ^2. We can estimate the parameters of the model using the maximum likelihood method:

$$\hat{\beta}, \hat{\gamma} = \arg\max_{\beta,\gamma} \sum_{i=1}^{n} \log P(X^i | Y = y) \tag{2}$$

We can then use the estimated parameters to impute the missing values in the dataset. We can also use the estimated parameters to predict the cluster of a new sample, and use the estimated parameters to impute the missing values in the new sample.

In effect, estimating this regression model allows us to learn a different distribution of values for each environment (i.e, condition the β coefficients on the cluster). This is particularly useful when the distribution of the data is conditioned on the cluster. For clusters that are unseen during training, it means that the indicator design matrix is zeroed out, and the imputation is done using the fixed effects only. This allows the model to generalize to unseen clusters, as the fixed effects are learned from the entire dataset.

In practice, the model is estimated using the `statsmodels` package in Python, which employs the BFGS optimization algorithm by default to estimate the parameters of the model.

4 Experiments

In this section, we discuss the datasets we chose to evaluate our method on (§4.1), the experimental setup (§4.2), the baseline methods we compared our method to (§4.3), and the results of our experiments (§4.4). In particular, we design the experiment to investigate the generalizability of our method to unseen environments, compared to other baseline methods. We explore this aspect in more detail in §5.

4.1 Datasets

In the following experiments, we use a variety of datasets to evaluate our method. We use both synthetic and real-world datasets, to test the performance of our method in different settings. We utilize the six datasets in (Tian et al., 2015), of which four are synthetic and two are real-world data. Additionally, we use the `popularity2` dataset presented in (Hox et al., 2017), which simulates the effect of various factors on student popularity in a school setting.

4.2 Experimental setup

For each of the datasets described, we split it into a training set and a test set by randomly selecting 20% of environments to be in the test set (*i.e*, there is no overlap between the environments in the training and test sets). We then select μ_{train} and μ_{test} to be the means of the distribution of missing data ratios for each data split: For each environment in the training set, we remove $r \sim N(\mu_{train}, 0.1)$ of the data in the selected feature (likewise for the test set). We then impute the missing values in the training set using different models, and evaluate the performance of the imputation on both the train set and the test set. We repeat this process 30 times for each dataset, and report the average performance of the imputation method as well as the standard deviation (thus bootstrapping the mean and standard variation estimators of the rMSE).

In our experiments, the selection of μ_{test} is fixed to be at 10 uniform intervals in the range $[\mu_{train}, 0.4]$.

The metric we use to evaluate the performance of the imputation method is the root mean squared error (rMSE) between the imputed values and the true values:

$$rMSE(y, \hat{y}) = \sqrt{\frac{1}{n}\sum_i (y_i - \hat{y}_i)^2}$$

Where y is the true value, $\hat{y}$ is the imputed value, and n is the number of samples in the dataset. We also report the mean squared error of the imputed values on the test set, to evaluate the generalizability of the imputation method to unseen environments.

4.3 Baseline methods

We compare our method to the following methods:

Baseline (simple imputation): We impute the missing values in the dataset using the mean of the observed values in the feature.

KNN imputation: We impute the missing values in the dataset using the K-nearest neighbors imputation method, where the predicted value is the mean of the feature value for these neighbors.

Linear regression imputation: We impute the missing values in the dataset using a simple multivariate linear regression model, where the predicted value is the output of the regression model.

4.4 Results

We report the main results in Figure 1. We find that our method outperforms the baseline methods on all datasets in terms of generalization, with a lower rMSE difference between the train and test sets. However, it estimates the missing values worse than simple linear regression which uses the same explanatory variables. This can be explained by the fact that some of the data may be generated by similar distributions, while the complexity of the mixed-effects model may not be necessary as it would typically require more training samples. Therefore a simple regression model is sufficient in this case. However, the mixed-effects model is more robust to unseen environments, as it generalizes well to these environments, which allows users to obtain a better estimation of the performance of the model during test time. The full results for each dataset are reported in Appendix A.

5 Generalizability to unseen environments

In this section, we explore the generalizability of our method to unseen environments, compared to other baseline methods. We do so by varying the divergence between the missing-rate distributions of the train and test sets. Recall that the missing-rate distribution is defined as $r \sim N(\mu_i, 0.1)$, where μ_i is the mean of the missing-rate distribution for the i-th environment. The distributions imposed on the different data splits therefore differ only in their means. Thus, we define the difference between them as $\Delta(\mu) = \mu_{train} - \mu_{test}$. We vary the divergence by changing the value of μ_{test}. The results for Dataset 6 (Tian et al., 2015) are reported in Figure 2. The full results for each dataset are reported in Appendix B.

In some of the results, KNN and the simple imputation display either indifference to the divergence between the missing-rate distributions of the train and test sets, or a significant increase in error rate. Our method, on the other hand, is consistently as robust as simple linear regression to the divergence between the missing-rate distributions of the train and test sets when evaluated on this data, with the exception of datasets 3 and 4.

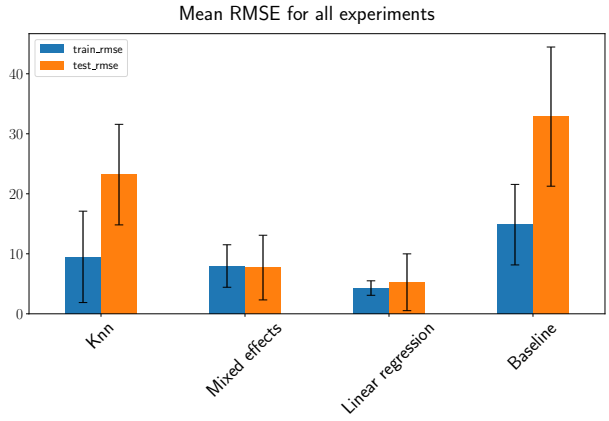


Figure 1: **rMSE of different imputation methods.** Results are reported for all datasets. Mixed-effects models are marginally worse than simple linear regression, but the error rate difference between train and test is lower. Other methods such as KNN and the simple imputation method have a significantly higher error rate on the test set.

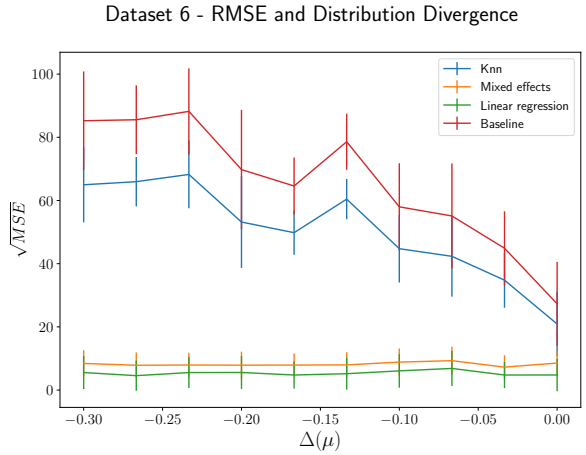


Figure 2: **rMSE as a function of $\Delta(\mu)$ of different imputation methods.** Results are reported Dataset 6 (Tian et al., 2015). Lower values of $\Delta(\mu)$ indicate a higher divergence between the missing-rate distributions of the train and test sets. Our method is as robust as simple linear regression to the divergence between the missing-rate distributions of the train and test sets when evaluated on this data.

One important aspect of the difference between simple linear regression and mixed-effects regression, is the observation that linear regression, if provided with cluster information, requires one of the clusters to be provided as a default cluster, in a dummy-variable embedding of the cluster IDs. Therefore, the default cluster is used to impute the missing values in the test set. This is a modelling choice that the user has to make, introducing inductive bias into the model. This differs from the behaviour of the mixed-effects model which takes as "default" the average of the fixed effects of all models, which is learned from the data.

6 Discussion and Limitations

In this work, we propose a method for imputing missing values in a dataset, which is based on a mixed-effects linear model. Our method is designed to handle multiple environments, and we show that it outperforms two other existing imputation methods on a variety of datasets, including both synthetic and real-world datasets. While we have not demonstrated that the method outperforms a simple linear regression, we are confident that it is more robust to unseen environments, and that it generalizes well to these environments. Furthermore, employing mixed-effects models in imputation tasks allows for several modelling choices which give the user more flexibility and control over the variables that are used to impute the missing values, as well as interactions between them although in our experiments we have fixed the explanatory variables to be only non-interactions, and non-polynomial.
One key limitation in our work is the assumption that the data is generated by a linear model. In practice, the data may be generated by a more complex model, and a linear model may not be sufficient to capture the distribution of the data. Furthermore, our datasets, though diverse, do not demonstrate a significant shift in the distribution of the data between different environments.

References

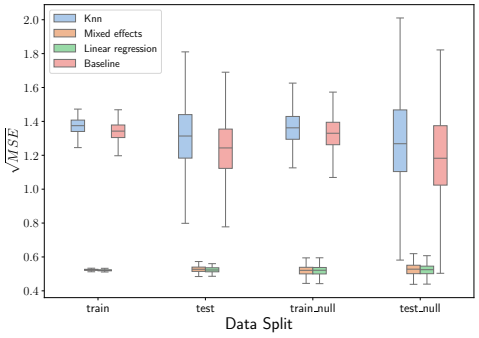
Joop Hox, Mirjam Moerbeek, and Rens Van de Schoot. *Multilevel analysis: Techniques and applications*. Routledge, 2017.

Matthieu Resche-Rigon and Ian R White. Multiple imputation by chained equations for systematically and sporadically missing multilevel data. *Statistical methods in medical research*, 27(6):1634–1649, 2018.

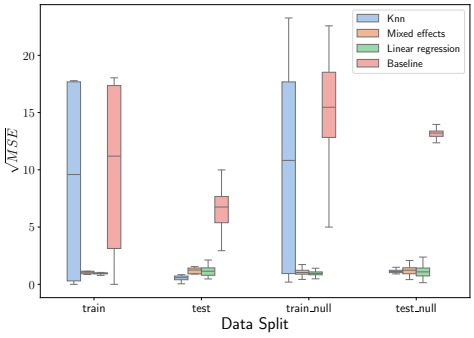
Ting Tian, Geoffrey J McLachlan, Mark J Dieters, and Kaye E Basford. Application of multiple imputation for missing values in three-way three-mode multi-environment trial data. *Plos one*, 10(12):e0144370, 2015.

A RMSE results for each dataset

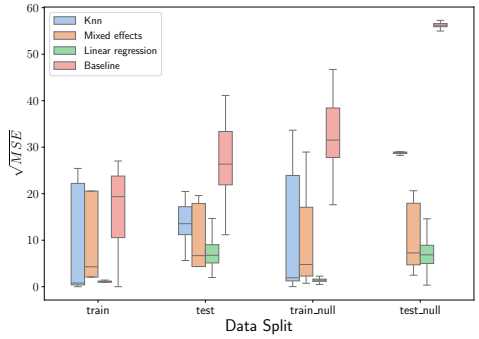
Dataset 1 - Error plot



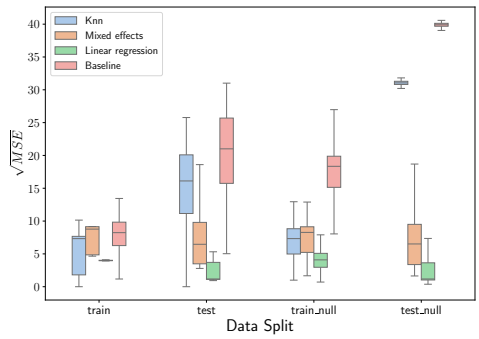
Dataset 2 - Error plot



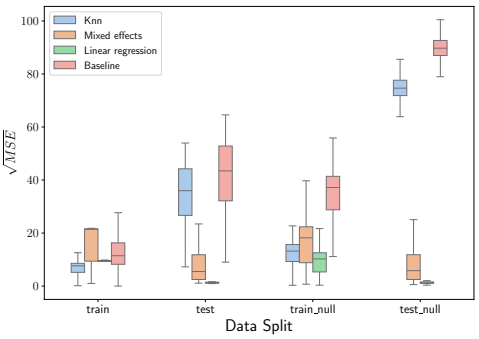
Dataset 3 - Error plot



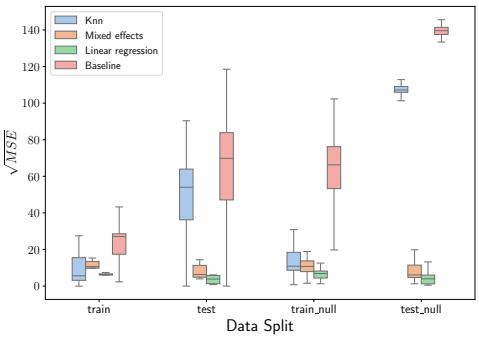
Dataset 4 - Error plot



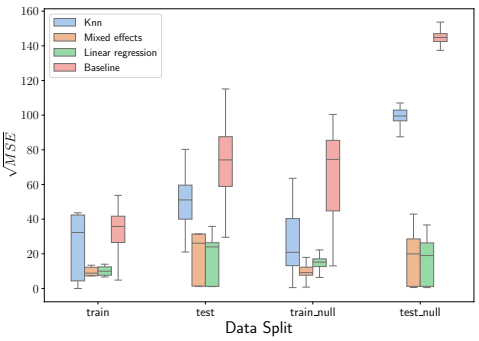
Dataset 5 - Error plot



Dataset 6 - Error plot

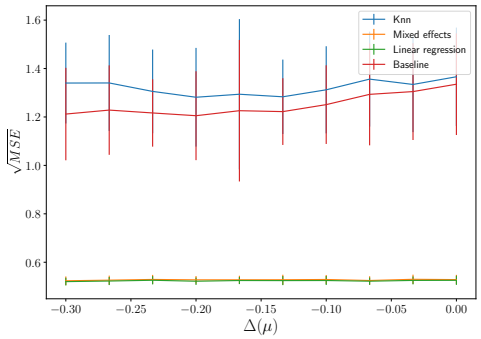


Dataset 7 - Error plot

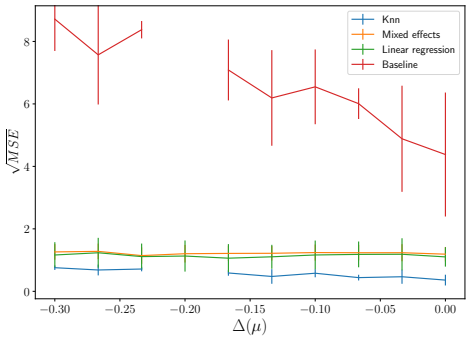


B Divergence results for each dataset

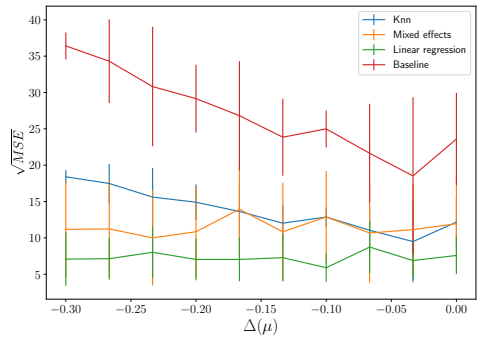
Dataset 1 - RMSE and Distribution Divergence



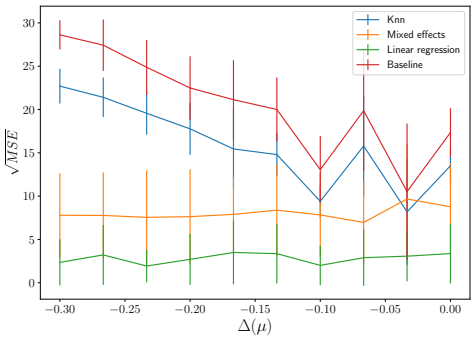
Dataset 2 - RMSE and Distribution Divergence



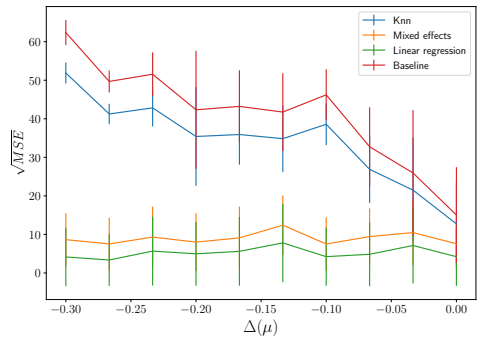
Dataset 3 - RMSE and Distribution Divergence



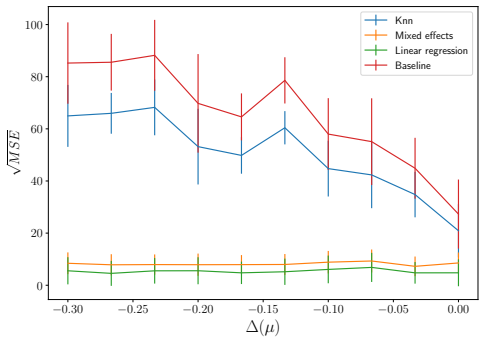
Dataset 4 - RMSE and Distribution Divergence



Dataset 5 - RMSE and Distribution Divergence



Dataset 6 - RMSE and Distribution Divergence



Dataset 7 - RMSE and Distribution Divergence