Missing data imputation and prediction

Antoine Ogier
Supervisor: Julie Josse
Academic supervisor: Geoff Nicholls

June 14, 2018

Contents

Acknowledgements

Introduction

Missing values in data is a prominent issue that has been much discussed in statistical literature. In the eighties, Donald Rubin devised many of the tools that are still used today to handle missing data: the expectation-maximisation algorithm, the definition of the three missing data patterns (MCAR, MAR, MNAR), multiple imputation. Two main approaches have been extensively researched to handle missing data: developing an ad-hoc algorithm that is capable of handling missing data itself (this is often based on the EM algorithm); or filling in the missing observations with imputed values.

However, there is in this regard a significant gap between the fields of statistical inference and machine learning: while the former has been actively developing and evaluating methods to handle missing data — mostly for parameter and confidence intervals estimation —, these methods are rarely used in the context of prediction, where replacing missing values with the mean of the observed data is standard practice.

This can be explained by the fact that machine learning practitioners enjoy having access to the whole range of standard algorithms they are familiar with, and are thus reluctant to use algorithms made for missing data rather than fill in the values. Conversely, filling by the mean is generally considered as 'good enough' for prediction purposes. Astonishingly in this regard, there is no thorough assessment of the way that missing values impact prediction performance, or comparison of imputation methods in this regard: whenever a new imputation method is published, a comparison with existing methods is usually conducted, but only regarding its performance for statistical inference. This means that we currently do not know whether it is worthwhile, when working on prediction, to turn to more elaborate (but also more computationally intensive) methods than mean imputation.

The goal of this work is to lay the groundwork for a review of imputation methods in the context of predictions. The final goal is twofold:

- Compare the predictive performance of some machine learning algorithms applied to datasets filled in with various imputation methods. This will be done both on real-world datasets and simulated ones with different missingness patterns.

- Investigate the relevance of multiple imputation methods (where multiple possible values of each missing observation are imputed) for prediction: in theory, having multiple imputed datasets gives us more

information on the certainty of the imputation (and so, of the resulting prediction).

However, conducting this investigation raises a major methodological issue related to the way that current imputation methods are implemented. We will start by addressing this issue before moving on to our investigation in itself.

1 Methodological investigation: the train-validation split

1.1 The issue with current imputation methods

To evaluate the predictive power of a model, the standard practice is to split the dataset in two parts: a training set and a validation set. The model parameters are learned by fitting on the training set. Then, an error metric is computed on the prediction of the same model for the validation set.

The idea for this is simple: we want to verify that the model has learned in a way that generalises to new data: if the validation performance is significantly lower than the training performance, it is a sign of overfitting. On the contrary, if the model is underfitted, both the training and validation error will be high. This validation method is an essential tool for model selection.

However, when trying to apply this methodology using imputation methods we are faced with a difficulty. Most of the imputation methods we will present in this work are iterative, which means that the imputation process alternates between computing new estimate for missing values, and estimating new parameters for some model based on the filled dataset. However, what happens if we want to separate the training and the imputation steps? This is necessary in order to perform the same imputation on new data (the validation set) after learning the imputation parameters on the training set.

Although this question is essential to machine learning applications, current implementations of imputation methods do not concern themselves with it. All of them consist of one main function which takes in a dataset with missing values and returns one or several filled in datasets. If we want to use those functions as-is, there are only two options:

- Run the imputation function separately on the training and the validation dataset. This is sound in terms of evaluation, because no data from the validation set can be used for training. However, on the other hand, we have no guarantee that the imputation will be performed with

the same parameters on both datasets. This is an important drawback because the predictive model will be trained on the first dataset and be applied to the second one with the same parameters. If they are filled in in a different way, the performance of the model is far from guaranteed. Note, however, that if both datasets are large and identically distributed, the imputation parameters should be roughly the same.

- Run the imputation function on both datasets joined together. This means that the parameters are learned on both datasets, which could completely undermine the performance evaluation by giving a validation loss that is too optimistic (since data was allowed to leak from the validation set). Still, no data from the response variable is used, so the impact may be rather small¡;

Of course, an ideal solution would be to rethink existing methods so that they can be used on separate training and validation data: though probably feasible in most cases, it would take a significant effort. This, it would be interesting to first assess whether using the aforementioned methods instead would impact the model very negatively. To that effect, we take a case where it is simple to perform the split and compare the three approaches in terms of performance.

1.2 An imputation method compliant with ML methodology

Amelia, a popular imputation package, uses a method that can easily be adapted to our purposes in the case of numeric data. If we model the whole dataset as drawn from a multivariate normal distribution, we can use the EM algorithm in order to estimate the parameters of that distribution. This leaves us with an estimated $\hat{\theta} = (\hat{\mu}, \hat{\Sigma})$. Then the missing values can be estimated for each observation by drawing the missing values conditionally on the observed ones.

In particular, it means that once $\hat{\theta}$ is estimated, it can be used to impute values on any new data without having to apply the EM estimation again. As a result, it is easy to estimate the parameters on just the training set and then impute the missing values of the validation set with those parameters.

We implemented this method.

1.3 Evaluation

In order to compare all three method, we performed a comparison of predictions on the Abalone dataset (a regression dataset where the goal is to

estimate the age of shells), which has no missing values. To that end we split the data at random into three datasets, which we call *Train*, *Test* and *Validation*. We proceed as follows:

- The Validation dataset is set aside with no further modification. In the rest of the data, we introduce some missing data completely at random in the covariates.
- We compute a first imputed from all of the remaining data, which we call X_{before}. It is then divided to form $X_{\text{before.train}}$ and $X_{\text{before.test}}$. These correspond to the data imputed before splitting.
- We split the data with missing values into Train and Test, and then impute the missing values separately on both of these to obtain $X_{\text{after.train}}$ and $X_{\text{after.test}}$ the data imputed after splitting.
- We split the data as previously, then we train an the imputation model on the Train data only, and use it to impute the missing data and obtain $X_{\text{corect.train}}$ and $X_{\text{correct.test}}$ the data imputed in the theoretically correct way.
- For each of those three pairs of datasets, a regression model in trained on the filled training data, and then used to perform predictions both the filled testing data and the Validation set with no missing values.
- We compare the mean squared error for the prediction in each case.

If the effects we mentioned previously are significant, we should observe the following:

- For the first dataset, information leak from the Testing dataset should mean that the testing error is lower than in the other cases, with higher Validation error.
- For the second dataset, if the Training dataset is filled differently from the Testing, the Testing error should be high.
- For the third dataset,