University of Oxford

MSc in Statistical Science

Final thesis

Missing data imputation for Haemorrhagic shock prediction

Author:
Antoine Ogier

Supervisor:
Pr. Julie Josse
(École polytechnique)
Pr. Geoff Nicholls
(University of Oxford)

September 2018

Abstract

Lorem ipsum dolot sit amet nunc cui Brexit.

Acknowledgements

Lorem ipsum dolot sit amet

Contents

Introduction

Main outline: Haemorrhagic shock is a condition that can be life-threatening but that has much higher survival rates if treated early. In addition, doctors tend to have a fairly bad record of detecting it. Because of this, we want to build a tool that predicts it based on measurement on trauma patients. For this we use Traumabase, a large patient-records database. Problem: lots of missing data in it. Solutions: use algorithms specifically made for missing data or impute missing data. Nice thing about imputation: once it is done, you can use any existing method -> here we only work on imputation (not so much the prediction part).

We present the data (Chapter 1), then present the state of the art in imputation (Chapter 2). Then, we derive some theory on imputation when it is performed with prediction as a goal (Chapters 3 and 4). We then come back to the data to apply what we learn, in order to choose the best imputation method for this problem (Chapter 6) and present our final results (Chapter 6.4).

Chapter 1

Goal and data

1.1 hemorrhagic shock: a lethal but preventable condition

1.1.1 Description

Post-traumatic bleeding is the primary cause of preventable deaths among injured patients around the world [**?**][**?**]. When a person sustains a severe injury (e.g. due to a car accident or violent assault), she may present serious internal or external bleeding. If the injury is serious enough, the natural coagulation process is not sufficient to stop the blood loss. In that case, if too much blood is lost, the patient enters a state called hemorrhagic shock (HS): the body is no longer able to provide vital organs with enough dioxygen to sustain them [**?**]. At this point, it becomes very difficult to save the patient, even by stopping the hemorrhage and transfusing them with enough blood [**?**].

When an individual sustains an injury, she is usually first taken in charge by first responders who evaluate the situation and the patient's state, before transporting her to a trauma center when the patient will be treated. To prevent patients from falling into hemorrhagic shock, procedures have been established in most hospitals to trigger a fast response to suspected haemorhhage [**?**]: massive transfusion (MT) protocols can be activated even before the patient enters the hospital if first response teams deem it necessary. When this happens, the hospital gets ready to transfuse the patient with large amounts of blood as soon as she arrives This way, the time interval between the injury and the transfusion in minimal. Studies [**?**] [**?**] show that early transfusion, followed by surgical bleeding control if necessary, can greatly improve the survival odds of patients that are at risk

of entering hemorrhagic shock.

This means that it is essential to activate the procedure as early as possible if a patient's condition requires it. Unfortunately, it quite hard to evaluate whether a patient is at risk of hemorrhagic shock. While external bleeding (e.g. from a knife wound) is obvious, internal bleeding (usually from blunt trauma) on the other hand is not easily diagnosed visually or using physiological parameters (heart rate, bood pressure, ...) [**?**]. A full-body scan or at least an ultrasound examination [**?**] may be needed. This results in a significant delay in the activation of the procedure — in particular, in those cases the procedure cannot be activated prior to the patient's arrival in the hospital.

1.1.2 Motivation for an assistance tool

As we mentioned, it is both far from obvious and very important to diagnose a risk of HS early, and doctors often fail to do so without advanced tests: a study [**?**] showed doctors to have quite limited performance when trying to predict a patient's risk of HS even after 10 minutes in the hospital. This highlights the difficulty of evaluating the need for the MT procedure before arrival, even when a doctor is present in the ambulance.

This combination of factors means that it makes sense to try to provide tools that would assist a doctor in detecting possible HS. To that end, a number of scoring systems have been developed to evaluate the risk of HS for a patient [**?**] [**?**] [**?**]: the idea is to determine a set of conditions on physiological measurements that determine a numeric score for the patient. The idea is that with well-chosen criteria, the score gives an objective assessment of a patient's condition that can supplement a doctor's expertise, or be used directly as a prediction (e.g. if the score is higher than some threshold, then then patient is at risk). However, the same study that evaluated the performance of doctors' prediction [**?**] showed that numeric criteria perform no better than doctors in predicting HS.

This might mean that it is simply impossible to accurately predict HS before advanced examinations are performed. However, it is also possible, that the relations between HS and physiological measurements are complex enough that a simple hand-made criterion is not enough to capture them. In that case, it would be useful to build a statistical model capable of representing this relationship and of providing hospitals with early estimates of a patient's level of risk.

This is why a team of researchers is trying to develop a tool that would leverage machine learning techniques to predict hemorrhagic shock, using a

database of patient records (cf Section 1.2). Our paper is part of that effort, with a specific focus on missing data imputation.

1.2 The Traumabase data

1.2.1 The Traumabase project

1.2.2 Data overview

General information

The Traumabase contained the records of 7477 patients at the time of this work. On recommendation of the doctors we worked with, we removed all of the patients who sustained penetrating injuries such as knife or gunshot wounds — 826 patients — (because the presence of a hemorrhage is obvious to assess in this case) and those who had a cardiac arrest before their arrival in the hospital — 396 patients — because this level of gravity is always enough to justify an emergency procedure. We also excluded patients who were redirected to the trauma center from another hospital (as opposed to directly by the first responders) —1102 patients— since this does not correspond to our case of study (prehospital evaluation). This leaves us with a total of 5153 patients.

In this population, 500 patients wenth through hemorrhagic shock and 4653 did not.

The traumabase records dozens of variables that trace a patient's history from the moment first responders arrive to the end of the patient's stay in the hospital (i.e. death or recovery). Here we are interested in performing a prehospital evaluation, so when we perform the prediciton we only consider a few measurements that correspond to those performed by the first responders.

Definition of the variables

There are 9 variables in the data that we can use for prediction.

General physical criteria These values are the sex, age and BMI (body-mass index) of the patient. They do not give any direct hint as to whether a patient is in shock, but they are necessary to control for natural differences betwen individuals (for instance, males naturally have a higher level of hemoglobin in their blood than females).

Basic physiological measurements These values are measured by the response team as soon as they arrive on the scene. They are:

1. Heart rate: The heart rate of the patient. Intuitively, if the patient has been losing blood, their heart should be beating faster in order to keep supplying the body with oxygen in spite of the blood loss [**?**]

2. Pulse pressure: The difference between the systolic (maximal) and diastolic (minimal) blood pressure during a heart beat. When the volume of blood is the body is low, this pressure may decrease [**?**]

3. Hemoglobin level: This is the concentration of hemoglobin (Hb) in the blood. The blood is composed, among other things, of red blood cells which contain hemoglobin that is used to carry oxygen. During blood loss, the liquid part of the blood can be regenerated faster than the red cells [**?**] which causes a drop in the Hb concentration. It is easily measured on location using measurement kits.

4. Peripheral oxygen saturation: This value ranges from 0% to 100% and represents the fraction of Hb molecules in the blood carrying dioxygen. During bleeding, if the oxygen carrying capacity is reduced (lower Hb concentration, lower blood flow due to hypotension, ...) then organs will draw more oxygen relative to the total carrying capacity and the saturation will decrease [**?**]. Measurement is easy and standard [**?**].

Glasgow coma scale (GCS) The GCS is a score assessing the conscious state of the patient [**?**]. It is computed from three criteria (eye movement, verbal response, motor functions) and ranges from 3 (deep coma or death) to 15 (fully awake). It gives a standardized way of reporting a patient's consciousness.

Volume expander injection To stabilise the patient and compensate major fluid loss, the emergency responder may decide to inject the patient with volume expanders, that is fluids specifically designed to fill some of the volume of the vascular system in order to rise blood pressure. This is a proxy for the responder's assessment of the patient's gravity, which is useful since many hard-to-quantify factors (paleness, gravity of the incident, general aspect, ...) may impact this assessment and would otherwise be unavailable to us.

This variable gives us the total volume (in mL) of expander that was injected into the patient.

Exploration

Continuous variables: The following table gives a general summary of the continuous covariates:

•	Min	Max	Mean	Median
Age	12	95	37.9	34
BMI	12	100	24.8	24.2
Heart rate	12	222	95.7	93
Pulse pressure	0	169	46.3	45
Hb level	0	19	13.9	14
O2 saturation	0	100	96.5	98
Expander	0	6250	791	500

Their distribution is illustrated in Figure 1.1. We see that all of the variables seem to have a unimodal distribution (some variables such as the expander are artificially rounded by the doctors when reporting, which accounts for the apparent drops in density). Additionally, an all variables but the expander dose, the mean and median are very close together which points towards low skew [**?**].

The age has a rather heavy tail on the right (more older patients). The O2 stauration has a very long lower tail: while almost all patients are above 90% saturation, it is much lower for a few patients.

The differences between the populations of patients with and without shock are in line with our expectations: shocked patients have in general lower Hb levels, a higher heart rate, lower pressure and lower saturation. However, we also see that no single factor gives an easy separation, and that shocked patients can have normal readings for any given measurement.

Categorical variables We have just two variables which can be seen as categorical: the sex and the GCS score. For the GCS, this is a discrete scale from 3 to 15. Its distribution is shown on figure 1.2. As before, shocked patients tend to have a lower score but many of them have a perfect score (15).

As regards the sex, there are 1177 females and 3976 males in the population.

Correlation structure Figure 1.3 shows the correlation bewteen the values of the observations (including the patient outcome). We see that not all variables have obvious correlation, but there are some subgroups of variables that are all correlated (e.g. Glasgow, heart rate, pulse pressure and saturation; or age, BMI and Hb level).

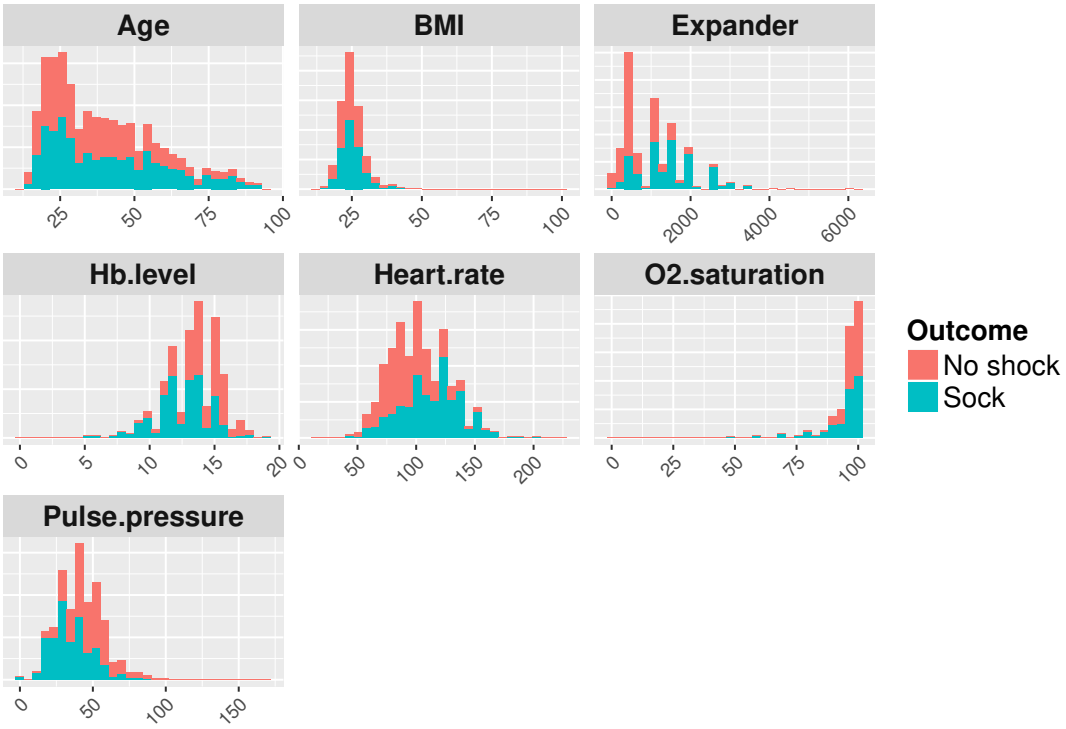


Figure 1.1: Distribution of the continuous variables depending on patient outcome

As expected, the physical measurements (Sex, BMI, age) show no correlation with patient outcome, while all the other variables do.

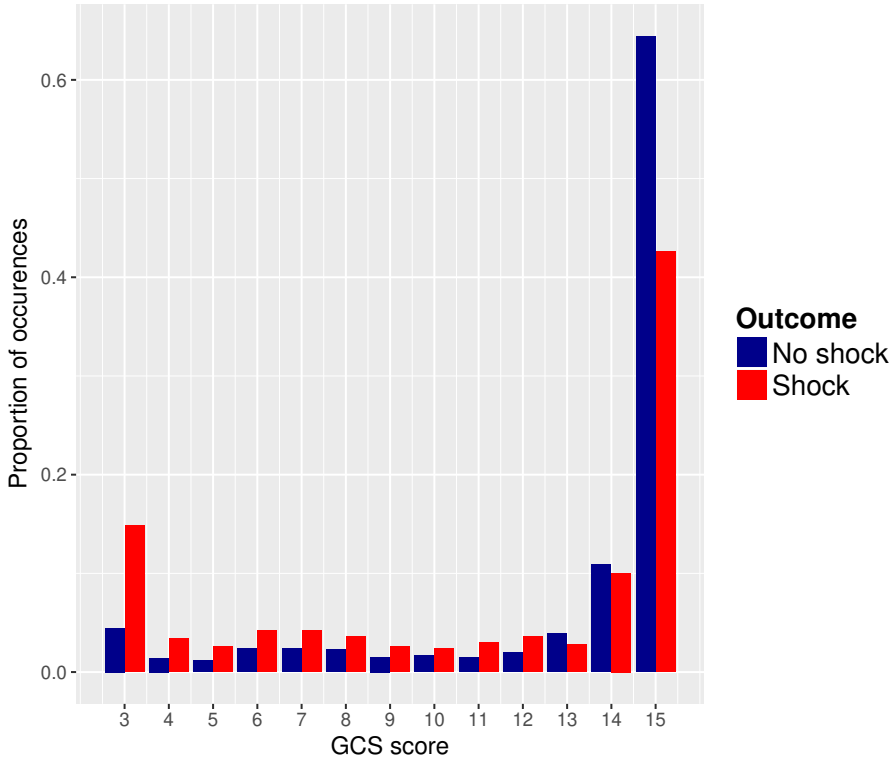


Figure 1.2: Distribution of Glasgow Coma Scale score depending on patient outcome

Sex	1.00	−0.07	0.10	−0.02	−0.04	0.09	0.38	−0.02	−0.03	−0.07
Age	−0.07	1.00	0.23	−0.04	−0.08	0.01	−0.19	−0.10	0.04	0.09
BMI	0.10	0.23	1.00	0.02	0.05	0.02	0.05	−0.05	0.02	0.04
Glasgow	−0.02	−0.04	0.02	1.00	−0.11	0.04	0.10	0.17	−0.22	−0.14
Heart.rate	−0.04	−0.08	0.05	−0.11	1.00	−0.12	−0.03	−0.17	0.22	0.23
Pulse.pressure	0.09	0.01	0.02	0.04	−0.12	1.00	0.11	0.08	−0.20	−0.21
Hb.level	0.38	−0.19	0.05	0.10	−0.03	0.11	1.00	0.04	−0.17	−0.24
O2.saturation	−0.02	−0.10	−0.05	0.17	−0.17	0.08	0.04	1.00	−0.16	−0.12
Expander	−0.03	0.04	0.02	−0.22	0.22	−0.20	−0.17	−0.16	1.00	0.36
Shock	−0.07	0.09	0.04	−0.14	0.23	−0.21	−0.24	−0.12	0.36	1.00

Figure 1.3: Correlation between the measurements (based on complete cases)

Missing data

An important aspect of the Traumabase is that is contains a significant amount of missing data. That is, some measurements or informations about the patients were not collected or not reported into the database, which makes them unavailable to us. In total, 5% of the observations are missing. The table below gives the amount and proportion of missing data for each variable:

•	Amount	Proportion
Sex	0	0%
Age	7	0.1%
BMI	778	15%
GCS	18	0.4%
Heart rate	110	2.1%
Pulse pressure	126	2.5%
Hb level	301	5.8%
O2 saturation	171	3.3%
Expander	795	15.4%

Figure 1.4 shows the repartition of the number of missing observations in the data. 1686 patients (33 %) have at least one missing observation.

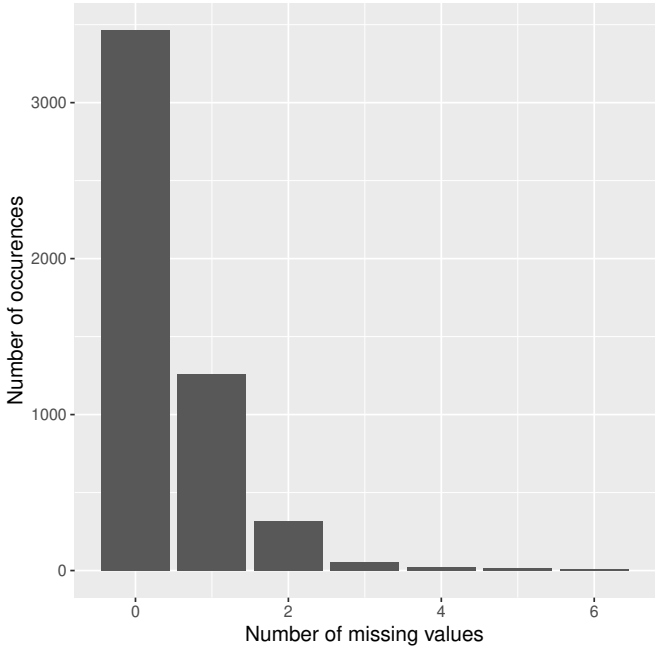


Figure 1.4: Distribution of the number of missing values

Figure **??** shows the correlation between the missingness of the variables. They are mostly uncorrelated, but we see that physiological measurements tend to be missing at the same time.

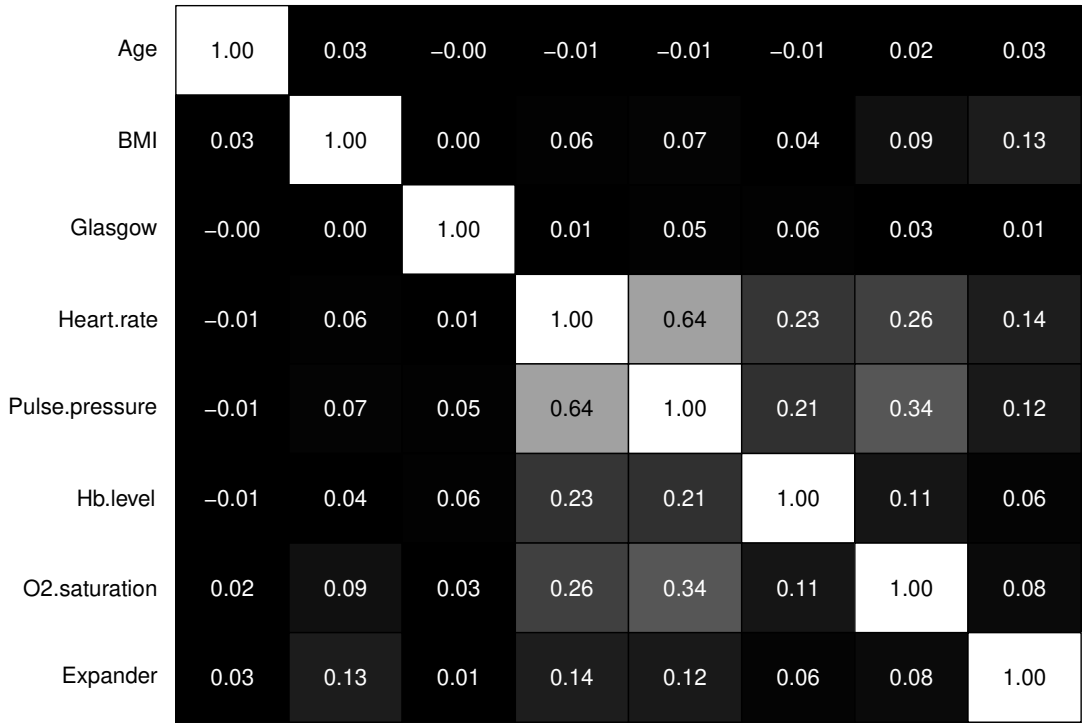


Figure 1.5: Correlation between the missingness of each variable

Lastly, the table below gives the correlation between the missingness of each variable with the patient outcome:

Age	$5.7 \cdot 10^{-3}$
BMI	$5.7 \cdot 10^{-3}$
GCS	$2.8 \cdot 10^{-3}$
Heart rate	$-7.6 \cdot 10^{-3}$
Pulse pressure	$5 \cdot 10^{-2}$
Hb level	$-4.5 \cdot 10^{-2}$
O2 saturation	0.11
Expander	$-2.6 \cdot 10^{-2}$

There is some amount of correlation between the missingness of the O2 saturation and the shock, but it is still quite low so it is hard to tell whether this has any significance.

1.3 Objective and formalization

1.3.1 Objective of this work

Given the importance of treating HS quickly, and the difficulty of detecting it — especially for first responders not specialized in major trauma and do not have access to a hospital's euquipment—, there is a strong case for trying to predict HS automatically during the prehspital phase. This would enable a hospital to have an assessment of a patient's level of risk as soon as first responders reach them, and make preparations in advance to treat them urgently if necessary.

If one is to develop a tool that would predict hemorrhagic shock, an issue that will need to be addressed is that of missing data. Indeed, there are missing observations in the records of 33% of the patients. Although this leaves us with a rather large number of complete cases, using only those would still be a major loss of information. Even more importantly, some of the data may also be missing in the real world when a prediction needs to be made. In that case, there no way around handling the missing observations to output a prediction.

In this work, we will address the particular issue of missing data, and more precisely of imputation: replacing the missing values in the dataset by plausible ones. Indeed, it is possible to create a model that takes missing data into account without trying to fill in the missing values and it has been done successfully in the past [**?**][**?**]. However, existing missing-data implementations are fairly rare, so trying out any model means working almost from the ground up. This greatly limiits our ability to compare several models.

Comparatively, once the dataset is imputed, it can be used with any existing complete-data prediction method. Additionally, it allows a separation of the tasks: the person or team performing the imputation is not necessarily the same as the one performing the subsequent analysis. Of course, this approach has drawbacks as well. In particular, once the dataset is imputed, any method used to perform a prediction with this dataset will use the observed and imputed values indiscriminately, which may lead to errors if the imputation is not correct.

In the following chapters, we investigate the possibility of performing imputation in a predictive context, and how it should be done. Below we present the formal setting of the problem we investigate.

1.3.2 The God/Imputer/Analyst/Practitioner framework

Let us recall the tasks at hand: imputing the data in the Traumabase, then applying a complete-data procedure to learn a model for the outcome. Finally, for new incoming patients, use the new measurements (possibly with missing data) to evaluate whether or not they are at risk of HS.

It is clear that form the point of view of the end user (the hospital or medical practitioner), the performance of this procedure should be judged by its predictive performance on new patients. This separation between historical data and new patients is central to our problem and we investigate it further in chapters 3 and 4.

To formalize this setting, we draw inspiration from the framework proposed by Xie and Meng [**?**] to explore the issue of imputation. In their work, three actors come into play, God, the Imputer and the Analyst. We adapt this framework by adding a fourth actor, the Practitioner, whichrepresents the end user who is only interested in prediction. Their interacton goes as follow:

- "God" (i.e. nature) generates some data $\tilde{X}$ and outcome y based on a process known only to him.
- A dataset X is generated by adding missing values to $\tilde{X}$. X and y are transferred to an Imputer. The imputer is tasked with filling in the missing observations. She chooses an imputation model with parameter α and computes an estimate $\hat{\alpha}$ which she uses to generate a completed dataset X_{imp}.
- The Imputer transfers X_{imp} to an Analyst, along with y. The Analyst does not know which observations were initially missing.
- The Analyst uses X_{imp} and y to learn a predictive model with parameter β. Her estimation for this parameter is $\hat{\beta}$
- God generates a new pair of data and outcome $\tilde{X}_{new}, y_{new}$. Some missing data are added to $\tilde{X}_{new}$ to generate X_{new}.
- The Practitioner receives X_{new}. She has no access to the data used for training. The Analyst and Imputer provide the Practitioner with black-box functions that allow her to perform imputation and prediction on the new data. They are derived from their model and parameter estimate; we call them $f(\cdot, \hat{\alpha}), g(\cdot, \hat{\beta})$.

- The Practitioner uses those functions to compute $X_{new}^{imp} = g(X_{new}, \hat{\alpha})$ and $\hat{y}_{new} = f(X_{new}^{imp}, \hat{\beta})$.
- $\hat{y}_{new}$ is finally compared to y_{new} and the loss $L(y_V, \hat{y}_V)$ is computed. This gives the final performance evaluation of the process.

This process is illustrated in Figure 1.6

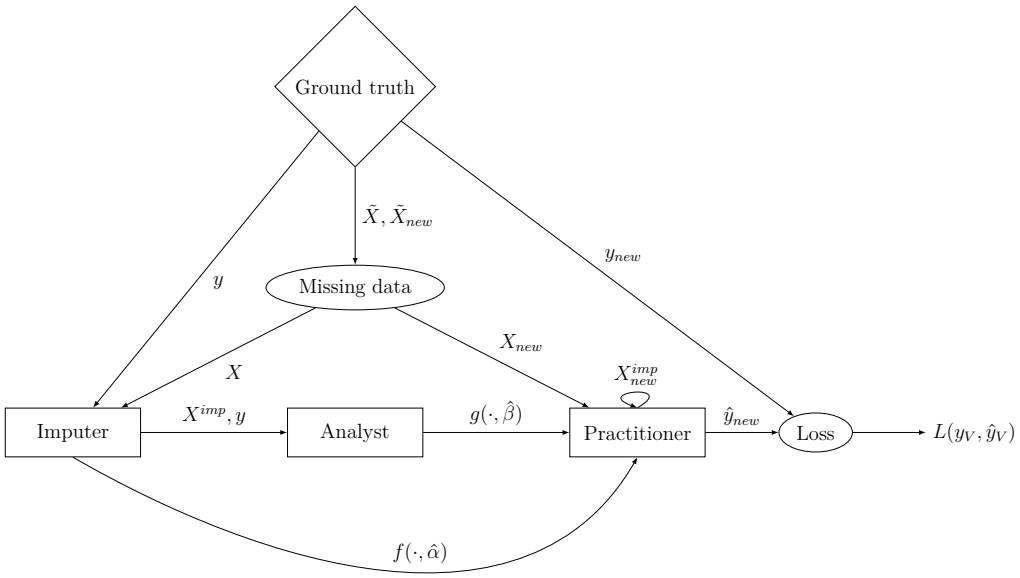


Figure 1.6: Imputation framework

The purpose of this formalization is to clarify the separation between the different phases of the inference, and show how the information is divided. Note that the separation between the Imputer and the Analyst is artificial —we choose it because we are interested in performing imputation separately— while the distinction between the Practitioner and the Analyst is imposed by the problem we are trying to solve: during an intervention, the users need a black-box tool that takes in the available data and outputs a prediction more or less instantly. Even if they had access to the training data, it would be impractical to learn a model all over again every time a new patient comes up. Additionally, the data we are using to learn the model cannot be widely shared since it contains patient records, so the Practitioner will not have access to it.

In a setting where all three roles are regrouped, the sensible way to proceed would be to define a joint model on the full data (including the

response) and use it find the maximum likelihood estimatorfor the unknown outcomes. However, the segmentation of the roles makes it necessary for each agent to work with partial information. In the rest of this work, we investigate the implications of this division, both in terms of theory and of practical implementation.

In Chapter 2, we present an overview of methods used for imputation. Chapter 3 considers the practical difficulties encountered when trying to impute new data with a previously learned model. Chapter 4 explores the asymmetry that exists between missing values present in the training set and in the new data. Then, Chapter 6 goes back to the Traumabase data to explore the best way to perform imputation in this particular case and Chapter 6.4 presents our final results and compares the performance to that of doctors and criteria-based scores.

Chapter 2

Imputation methods

2.1 Main types of imputation

2.1.1 Joint parametric specification

2.1.2 Fully conditional specification (FCS)

In FCS rather than a joint model we define p conditional models $\pi_1, \ldots, \pi_p$ where π_i gives the distribution of variable i conditional on the others. We can then othain an imputed dataset iteratively using the Multiple Imputation by Chained Equations (MICE) algorithm [**?**]

Algorithm 2.1 MICE Algorithm

Input: $X, \pi_1, \ldots, \pi_p$
Output: $\hat{X}$

$\hat{X} \leftarrow$ plausible imputation of the missing data
while not converged **do**
    for $i = 1 \ldots p$ **do**
        $X^{(i)} \leftarrow$ the i^{th} column of X
        $\hat{X}^{(-i)} \leftarrow \hat{X}$ without its i^{th} column
        $\hat{X}^{(i)}_{\text{miss}} \sim P(\hat{X}^{(i)}_{\text{miss}} | X^{(i)}_{\text{obs}}, \hat{X}^{(-i)})$
    end for
end while

The intereset of this approach is that is can be very flexible when the variables have very different distribution profiles.

Chapter 3

Methodology: imputation and the validation split

Let us now go back to the issue of imputing missing data when there is a separation between the Practitioner and the Analyst, that is, when it is necessary to impute new incoming data without having access to the historical data.

As described Chapter 2, a number of methods exist to impute missing data. However, they have been designed with parameter estimation in mind: in that case, there is just one dataset that needs to be inputed. The issue, as we show in this chapter, is that current implementations of these methods may not be suitable when we need to use the same model to impute two separate datasets.

This is not only problematic for our particular problem. As we remind in Section 3.1, it is standard practice when working on prediction to resort to CV, that is to hold out part of the data and use it to validate the model. Section 3.2 shows how this clashes with current implementation of imputation methods. In section 3.3, we investigate possible ways to address this issue and compare them empirically.

3.1 Empirical risk minimization and cross-validation

Let us start by ignoring the issue of missing data and assume that the data is complete. That is, we place ourselves in the same situation as described in 1.3.2 but there is no need for an imputer, and the Analyst directly receives the data X.

As stated in Chapter 1, our end goal is to make good predictions in the real world by learning on historical data. Of course, by definition we do not have access to any future data right now, but we still need to choose a prediction and imputation model, and estimate how it will perform when we use it on the field.

To learn and evaluate the model, we go through two steps: Empirical risk minimization (ERM) and cross-validation (CV). We describe them below

3.1.1 Empirical risk minimization

Let us denote the X a $n \times p$ matrix of covariates and y a response vector of size n, where our goal is to predict the response y_{new} from some future data X_{new}, assuming that $(X_{new}, y_{new}$ follow the same distribution as X, y.

We choose a class of functions $f(\cdot, \beta), \beta \in B$. We want to choose a parameter $\hat{\beta}$ which we can use to predict $\hat{y}_{new} = f(X_{new}, \hat{\beta})$. The quality of a prediction is evaluated by some loss $L(y_{new}, \hat{y}_{new})$. Since we do not have access to X_{new}, our goal is to minimize the risk: $R(\beta) = \mathbb{E}_{X_{new}, y_{new}}(L(y_{new}, f(X_{new}, \beta)))$.

However, we do not know the true distribution of those values. This is why we resort to ERM [7]: we define the empirical risk

$$R_{\text{emp}}(\beta) = \frac{1}{n} \sum_{i=1}^{n} L(y_i, f(X_i, \beta))$$

that is, the average value of the loss when predicting the known y from X with β.

We then select $\hat{\beta} = \arg\min_\beta R_{\text{emp}}(\beta)$ as our ERM estimator for β. However, this is not enough. Once we have chose $\hat{\beta}$, we need to have an estimate of how well this estimate will perform on new data.

This is important because this is what we will take into account if we need to compare multiple choices for f. But the empirical risk gives us no measure of how well our model generalizes, only of how closely it can fit known data. In particular, if the class f is very broad, one may find a β that exactly interpolates the values of y but does not generalize at all (an issue known as overfitting [**?**]

To address the issue of model selection, we need to resort to CV as descried below.

3.1.2 Cross-validation

CV, consists in dividing the available data in two datasets: first we choose $n_A < n$ entries in the dataset that will be used in ERM to learn $\hat{\beta}$: this is

the training dataset X_A and response y_A. We denote $I_A = (i_1, \ldots, i_{n_A})$ the set of indices chosen for the training data.

The rest of the observations are noted X_V and y_V and called the validation dataset. They are used as a substitute for X_{new}, y_{new}

Once this is done, the Analyst performs ERM as before, using only the training data. The obtained parameter $\hat{\beta}$ can then be evaluated with the validation error:

$$R_V(\hat{\beta}) = \frac{1}{n_V} \sum_{i=1, i \notin I_A}^{n} L(y_i, f(X_i, \hat{\beta}))$$

It is this value that we can compare to choose our the model class f. Once the Analyst has decided on a choice of f and $\hat{\beta}$ using ERM and CV, she can send $f(\cdot, \hat{\beta})$ over to the Practitioner so that she can proceed to prediction on new data using $\hat{y_{new}} = f(X_{new}, \hat{\beta})$.

3.2 ERM with missing data: the problem of current methodologies

We now place ourselves in the same context as before, except some values are missing from X, both in the training and the validation data. This means that we are back to a case where there is an Imputer in addition to the Analyst.

3.2.1 Imputation seen as an ERM

Remember that the purpose of this work is to impute the data independently of the model used afterwards for prediction. This means that we cannot perform ERM exactly as before and use any function we like to go from X (which has missig data) to $\hat{y}$. The prediction is the composition of two steps.

Imputation step First we choose an imputation model $X^{\text{complete}} = g(X, \alpha)$ where X^{complete} is the completed dataset and α some parameter. This is similar to the previously described ERM, except we do not know the true data (while we had y to compare to $\hat{y}$, we do not know the true full dataset $\tilde{X}$). Thus we choose $\hat{\alpha}$ to minimize some unsupervised empirical risk

$$R'_{emp}(\alpha) = L'(g(X, \alpha), \alpha)$$

Sometimes the loss L' is related to the likelihood of the completed data according to some distribution[**?**] (though it is not always the case [**?**]). Once this is done, we obtain a completed dataset $\hat{X}$.

Prediction step With imputation done, we can proceed as before to choose a parameter $\hat{\beta}$ that minimizes the empirical risk when using the completed data:

$$R_{\text{emp}}(\beta, \hat{X}) = \frac{1}{n} \sum_{i=1}^{n} L(y_i, f(\hat{X}_i, \beta))$$

Putting it all together, we can define

$$h(X, (\alpha, \beta)) = f(X^{imp}, \beta) = f(g(X, \alpha), \beta)$$

the combined model that takes the observed data as input and outputs a predicted y. Formally, the two successive steps yield:

$$\begin{aligned} \hat{\alpha} &= \arg\min_{\alpha} L'(g(X, \alpha) \\ \hat{\beta} &= \arg\min_{\beta} R_{\text{emp}}(\beta, f(X, \hat{\alpha})) \\ \hat{y} &= h(X, (\hat{\alpha}, \hat{\beta})) \end{aligned}$$

We choose to use this notation to illustrate our point that imputation is an integral part of the ERM, not a separate, preliminary process. In particular, it means that its parameters must be subjected to CV just like those of the prediction. That is, only X_A and y_A are used to estimate $(\hat{\alpha}, \hat{\beta})$ as shown above, while X_V and y_V are held out. Then, we can compute a prediction $\hat{y}_V = h(X_V, (\hat{\beta}, \hat{\alpha}))$ and we compute $L(y_V, \hat{y}_V)$ to evaluate the choice of model.

This implies that just like β, the imputation parameter α should be estimated only on the training data and then used on the validation data. As we will see, this raises an issue with the way current imputation methods are implemented.

3.2.2 Unsuitability of current methods

Implementations of imputation methods have one thing in common: they are used through a single function which takes a dataset with missing values as an input and returns the dataset completed by the method of choice, *without giving the user any access to the imputation model itself.* [**?**] [**?**][**?**]

This is a problem because of how CV is supposed to be performed. Supposedly, one would estimate $(\hat{\alpha}_A, \hat{\beta}_A)$ through ERM, and then make a prediction on the validation set as $h(X_V, (\hat{\beta}_{X_A}, \hat{\alpha}_{X_A}))$. But here, all we have access to is a black-box function $g' : X \mapsto g(X, \hat{\alpha}_X)$ where $\hat{\alpha}_X$ is the optimised parameter for the argument X. This means that one cannot choose what parameters are used to impute the input to function g': a new parameter will be estimated at every call of the function. But in that case, it is impossible to use the same α for the training dataset and for the validation data — or for new data.

This issue has started to arise in the machine learning community in the past few years [10][11][12], but for now no implementation exists that separates the parameter estimation and the imputation itself (except for the very basic imputation by the mean [**?**]). Below, we investigate the alternatives available to perform imputation with held out data.

3.3 Possible solutions

If we were to follow exactly the principles of CV, we would proceed as follows:

Algorithm 3.1 Identical imputation

Input: X, y, $I_A = i, X_i \in X_A$
Output: $\hat{y}_V$

Parameter estimation:
$\hat{\alpha}_A \leftarrow \arg\min_\alpha L'(g(X_A, \alpha)$
$\hat{X}_A \leftarrow g(X_A, \hat{\alpha}_A)$
$\hat{\beta}_A \leftarrow \arg\min_\beta R_{\text{emp}}(\beta, \hat{X}_A)$
Prediction:
$\hat{X}_V \leftarrow g(X_V, \hat{\alpha}_A)$ ▷ *Uses the same α as for the training set*
$\hat{y}_V \leftarrow f(\hat{X}_V, \hat{\beta}_A)$

But this is not possible using a black-box function because we need to recover $\hat{\alpha}_A$ and reuse it with X_V.

3.3.1 Alternatives using current implementations

Methods

Suppose we only have access to the back-box g' described above. Then there are two main ways of performing the imputation.

Grouped imputation Impute all of the data at once before performing the CV split:

Algorithm 3.2 Grouped imputation

Input: X, y, $I_A = i, X_i \in X_A$
Output: $\hat{y}_V$

Imputation:
  $\hat{X} \leftarrow g'(X) = g(X, \hat{\alpha}_X)$
  $(\hat{X}_A, \hat{X}_V) \leftarrow \hat{X}$ ▷ *Split the data after imputation*
Estimation of the prediction parameter
  $\hat{\beta}_A \leftarrow \arg\min_\beta R_{\text{emp}}(\beta, \hat{X}_A)$ ▷ *Note that* $\hat{X}_A = g(X_A, \hat{\alpha}_X)$
Prediction:
  $\hat{y}_V \leftarrow f(\hat{X}_V, \hat{\beta}_A) = f(g(X_V, \hat{\alpha}_X)$

Here, both datasets are indeed imputed with the same parameter α_X but this means that the validation data is used to choose that parameter which then serves to impute the training data. This is contrary to the principles of CV, we are 'cheating' in some way.

Separate imputation Divide the data first, then impute each dataset separately:

Algorithm 3.3 Separate imputation

Input: X, y, $I_A = i, X_i \in X_A$
Output: $\hat{y}_V$

Training parameter estimation:
  $\hat{X}_A \leftarrow g'(X_A) = g(X_A, \hat{\alpha}_A)$
  $\hat{\beta}_A \leftarrow \arg\min_\beta R_{\text{emp}}(\beta, \hat{X}_A)$
Imputation of X_V:
  $\hat{X}_V \leftarrow g'(X_V) = g(X_V, \hat{\alpha}_V)$ ▷ *Imputation made independently from that of* X_A
Prediction:
  $\hat{y}_V \leftarrow f(\hat{X}_V, \hat{\beta}_A)$

Contrarily to grouped imputation, we are not cheating. However, we are using parameter $\hat{\alpha}_V$ to impute X_V, while we learned $\hat{\beta}_A$ on $\hat{X}_A$ which was imputed with $\hat{\alpha}_A$. That is, we are optimizing $h(\cdot, (\hat{\alpha}_A, \hat{\beta}_A$ and predicting with $h(\cdot, (\hat{\alpha}_V, \hat{\beta}_A$.

$\hat{\alpha}_V$, $\hat{\alpha}_A$ and $\hat{\alpha}_X$ are asymptotically the same for large n —since X_A, X_V and X have the same distribution—, so all three methods should be

identical for large n. But for smaller n, harmful effects may be present —overoptimistic validation due to 'cheating' for the grouped imputation, high error due to the difference in parameters for separate imputation.

Need for a new implementation

We want to understand if the alternatives proposed here are good enough to be used if Identical imputation in infeasible (e.g. we want to evaluate a new imputation method that only implements a black box). To do that, we need to be able to compare these with the correct method. That means that for at least one imputation method we need to build an implementation that allows us to separate the estimation and the imputation. That way we will be able to compare its performance with the other alternatives we propose.

Moreover, in addition to this theoretical pursuit, we need this because of what we are trying to achieve with Traumabase: the end goal is to make a recommendation system that can produce a prediction for *a single new patient* arriving to the hospital, without needing to have access to the whole Traumabase data. Without access to the initial training data, this means that only a fully parametric approach can be taken in this particular case (separate imputation is impossible on just one line of data, and grouped imputation requires access to the full data).

Below, we design a very simple imputation method for those purposes.

3.3.2 Multivariate normal conditional expectation

The principle of this imputation is inspired from R package *Amelia* [13], and a large part of the code is from the *norm* package [14]. The idea is to model both X_A and X_V as normally distributed $\mathcal{N}(\mu, \Sigma)$ with unknown parameters. This will allow us to impute the missing data conditionally on the observed data.

Parameter estimation It is possible to approximate maximum-likelihood estimators for μ and σ with an iterative procedure, using the EM algorithm [15] [16]. The algorithm is as follows:

Algorithm 3.4 Normal parameter estimation with EM

Input: X
Output: $\hat{\mu}, \hat{\Sigma}$

$X^{(0)} \leftarrow X$ where missing values are replaced using the observed mean of X (mean imputation)
$t \leftarrow 0$
while not converged **do**
    $(\mu^{(t)}, \Sigma^{(t)}) \leftarrow$ sample mean and covariance matrix of $X^{(t)}$ ▷ *i.e. maximum likelihood estimates on the completed data*
    $X^{(t+1)} \leftarrow \mathbb{E}_{X^{\text{miss}}}(X|X^{\text{obs}}; \mu^{(t)}, \Sigma^{(t)})$ ▷ *missing values are replaced by their expected value under the new parameters*
    $t \leftarrow t + 1$
end while
$\hat{\mu}, \hat{\Sigma} = \mu^{(t)}, \Sigma^{(t)}$

The conditional expectations are easily derived using the Schur complement [17]. For this step, we use a slightly modified version of the code from the *norm* package.

Imputation Once we have the parameters, it is very straightforward to get an imputation of the missing data. Just as during the EM procedure, we impute using the conditional expectation of the dataset conditioned on the observed values:

$$\hat{X} = \mathbb{E}_{X^{\text{miss}}}(X|X^{\text{obs}}; \hat{\mu}, \hat{\Sigma})$$

We implemented this step as well to get a complete imputation procedure divided in two functions that implement the estimation and imputation steps separately. The code is available in package

make package and insert reference

3.3.3 Comparison on simulated data

Now that we have an implementation that separates estimation and imputation, we use it to compare the three imputation procedures on simulated and real data.

insert clean graphs

Chapter 4

Impact of missing data: the case of linear regression

In order to make good decisions for imputation, it is important to understand how it impacts prediction. To gain a better understanding of this issue, we solve a very simple case of cross-validated regression with missing data. Although quite restrictive, this situation provides some insights into the way that missing data impacts prediction performance.

4.1 Problem set-up

We place ourselves in a linerar regression setup with cross-validation (cf Chapter 3). The data is split between a training dataset X_A, y_A and validation dataset X_V, y_V. We use the multi-agent framework described in 1.3.2.

4.1.1 Notations

God's data

The response variable is a noisy linear combination of the covariates in X:

$$\tilde{X_A} = \begin{pmatrix} x_{11} & x_{12} \\ \vdots & \vdots \\ x_{n1} & x_{n2} \end{pmatrix} \quad \text{and} \quad y_A = X_A\beta + \epsilon_A \quad \text{with} \quad \epsilon_A \sim \mathcal{N}(0, \sigma^2)$$

$$\tilde{X_V} = \begin{pmatrix} x_{11}^V & x_{12}^V \\ \vdots & \vdots \\ x_{n_V 1} & x_{n_V 2} \end{pmatrix} \quad \text{and} \quad y_V = X_V\beta + \epsilon_V \quad \text{with} \quad \epsilon_V \sim \mathcal{N}(0, \sigma^2)$$

Observed data

The observed data is God's data with some missing values. Specifically, some observations are missing from the first column of each dataset. We observe the full y^A, but the covariate matrices we actually have access to are:

$$X_A = \begin{pmatrix} ? & x_{12} \\ \vdots & \vdots \\ ? & x_{k_A 2} \\ x_{(k_A+1)1} & x_{(k_A+1)2} \\ \vdots & \vdots \\ x_{n1} & x_{n2} \end{pmatrix}$$

which is sent to the Imputer, and

$$X_V = \begin{pmatrix} ? & x_{12} \\ \vdots & \vdots \\ ? & x_{k_V 2} \\ x_{(k_V+1)1} & x_{(k_V+1)2} \\ \vdots & \vdots \\ x_{n_V 1} & x_{n_V 2} \end{pmatrix}$$

which is sent to the Practitioner. That is, there are k_A and k_v missing values in the datasets.

4.1.2 Imputed data and regression

Principle

The imputer fills in X_A with imputed values, and instructs the Practitioner as to how to fill in X_V. The resulting filled in datasets are:

$$\hat{X}_A = \begin{pmatrix} \phi_1 & x_{12} \\ \vdots & \vdots \\ \phi_{k_A} & x_{k_A 2} \\ x_{(k_A+1)1} & x_{(k_A+1)2} \\ \vdots & \vdots \\ x_{n1} & x_{n2} \end{pmatrix} \quad \text{and} \quad \hat{X}_V = \begin{pmatrix} \psi_1 & x_{12} \\ \vdots & \vdots \\ \psi_{k_V} & x_{k_V 2} \\ x_{(k_V+1)1} & x_{(k_V+1)2} \\ \vdots & \vdots \\ x_{n_V 1} & x_{n_V 2} \end{pmatrix}$$

The ϕ values should depend only on the observed X_A, and the ψ values on both X_V and the imputer's indications. Then, $\hat{X}_A$ is sent by the imputer

to the Analyst. The Analyst only has access to $\hat{X}_A$ and y_A. The end goal is to learn an estimator on the training set that minimizes the expected loss on the validation set:

$$L(y_V, \hat{y}_V) = (y_V - \hat{y_V})^2$$

In line with the principles of ERM and CV (cf 3), the Analyst minimizes the equivalent quantity in the training set. Assuming a linear relationship between the covariantes and response, the least-squares estimate for β is standard [**?**]

$$\hat{\beta}_n = (\hat{X}_A^T \hat{X}_A)^{-1} \hat{X}_A^T y_A$$

$\hat{\beta}$ is then transferred to the Practitioner who can use it to compute a prediction

$$\hat{y}_V (= \hat{X}_V \hat{\beta}_n$$

which will be compared to y_V:

$$L(\hat{y}_V, y_V) = \sum_{i=1}^{n_V} (y_V^{(i)} - \hat{y}_V^{(i)})^2$$

Our end goal is to optimize this metric.

In what we described above, the actions of the Analyst and the Practitioner are completely determined. On the other hand, we have not specified how the Imputer proceeds to the imputation. We want to investigate the effect of the choice of imputation on the expected loss:

$$R(\phi, \psi) = \mathbb{E}_{X_V^{miss}, X_A^{miss}, \epsilon_A, \epsilon_V} [(y_V^{(i)} - \hat{y}_V^{(i)})^2 | X_A, X_V, \phi, \psi]$$

1

Distribution hypotheses

Lastly, for this last expression to have any meaning, fix the distribution of $\tilde{X}$ the true data.

We assume $X \sim \pi$ where the lines of X are independent and identically distributed, and that π is known to the Imputer. This is unlikely in a real setting, but we explore the best-case scenario where we have all the necessary information to perform the imputation in order to isolate the error terms that are specific to the presence of missing data —as opposed to bad imputation.

[1] Even though what we ultimately want is a decision rule for ϕ and ψ, they are only a function of the observed data X_A, X_V, which is fixed here. For simplicity of notation, we write ϕ and ψ as constant values

4.2 Partial resolution

4.2.1 General loss

To be able to estimate the expected loss, we break it up into several components. We first denote

$$\tilde{\beta}_n = (\tilde{X}_A^T \tilde{X}_A)^{-1} \tilde{X}_A^T y_A$$

the estimated parameter we would obtain if the training data were complete. We consider the loss for one given line of validation data $x_V = (x_V^1, x_V^2)$

$$\begin{aligned}
L(y_V, \hat{y}_V) &= (y_V - \hat{y_V})^2 \\
&= (\tilde{x}_V \beta + \epsilon_V - \hat{x}_V \hat{\beta}_n)^2 \\
&= (\tilde{x}_V(\beta - \tilde{\beta}_n) + \tilde{x}_V(\tilde{\beta}_n - \hat{\beta}_n) + (\tilde{x}_V - \hat{x}_V)\hat{\beta}_n + \epsilon_V)^2 \\
&= (\tilde{x}_V(\beta - \tilde{\beta}_n))^2 && (1) \\
&\quad + (\tilde{x}_V(\tilde{\beta}_n - \hat{\beta}_n))^2 && (2) \\
&\quad + ((\tilde{x}_V - \hat{x}_V)\hat{\beta}_n)^2 && (3) \\
&\quad + \tilde{x}_V(\beta - \tilde{\beta}_n)\tilde{x}_V(\tilde{\beta}_n - \hat{\beta}_n) && (4) \\
&\quad + \tilde{x}_V(\beta - \tilde{\beta}_n)(\tilde{x}_V - \hat{x}_V)\hat{\beta}_n && (5) \\
&\quad + \tilde{X}_V(\tilde{\beta}_n - \hat{\beta}_n)(\tilde{x}_V - \hat{x}_V)\hat{\beta}_n && (6) \\
&\quad + \epsilon_V^2 \\
&\quad + \epsilon_V K
\end{aligned}$$

Where C is some term that will not matter (since it has zero expectation —ϵ_V has zero expectation and is independent from the other terms). The risk we want to minimize is the expectation of this loss.

4.2.2 When the training set is fully observed

Imputation

The first thing we can do is to study the situation where the only missing data is in the validation set. In that case, $\tilde{X}_A = \hat{X}_A$ and so $\tilde{\beta}_n = \hat{\beta}_n$, and all we have to choose is ψ. In the previously computed loss, it means that terms (2), (4) and (6) are zero.

Furthermore,

$$\begin{aligned}\beta - \tilde{beta}_n &= \beta - (\tilde{X}_A^T \tilde{X}_A) - 1\tilde{X}_A^T(\tilde{X}_A\beta + \epsilon_A) \\ &= (\tilde{X}_A^T \tilde{X}_A) - 1\tilde{X}_A^T\epsilon_A\end{aligned}$$

And

$$(5) =$$

Furthermore, Cochran's theorem ensures that $(\beta - \tilde{\beta})$ and $\tilde{\beta}$ are independent so term (5) can be factorized and will have zero expectation (since $(\beta - \tilde{\beta})$ has zero expectation).

Term (1) depends only on the true values of X_V, independent of *psi*, so the choice of ψ is not impacted by this term.

This leaves us with only term (3), with expectation:

$$\begin{aligned}\mathbb{E}[((\tilde{X}_V - \hat{X}_V)\tilde{\beta})^2|x_2^V, X_A] &= \mathbb{E}[(x_1^V - \psi)^2\tilde{\beta}_1^2|x_2^V, X_A] \\ &= \mathbb{E}[\tilde{\beta}_1^2|x_2^V, X_A](\mathbb{E}[(x_1^V)^2|x_2^V, X_A] \\ &\quad - 2\psi\mathbb{E}[x_1^V|x_2^V, X_A] + \psi^2)\end{aligned}$$

Once we are there, we can differentiate this expression to easily derive the optimal expression for ψ: $\hat{\psi} = \mathbb{E}[x_1^V|x_2^V]$, the conditional expectation of the missing value.

Incidentally, this does not use any assumption on the distribution of X: this would be true for any joint distribution we choose for the covariates.

Expected loss

First note:

$$\begin{aligned}\eta = \beta - \tilde{\beta} &= \beta - (\tilde{X}_A^T \tilde{X}_A)^{-1}\tilde{X}_A^T y_A \\ &= (\tilde{X}_A^T \tilde{X}_A)^{-1}\tilde{X}_A^T(\tilde{X}_A\beta + \epsilon_A) \\ &= (\tilde{X}_A^T \tilde{X}_A)^{-1}\tilde{X}_A^T\epsilon_A\end{aligned}$$

That is, the difference between the estimated and real parameter is distributed following some centred normal distribution. Let us denote its covariance matrix by $S = \sigma^2(\tilde{X}_A^T \tilde{X}_A)^{-1}$.

Now, term (1) can be expressed as :

$$\begin{aligned}\mathbb{E}[(\tilde{X}_V\eta)^2|x_2^V] &= \mathbb{E}[(x_1^V\eta_1 + x_2^V\eta_2)^2|x_2^V] \\ &= S_{11}\mathbb{E}[(x_1^V)^2|x_2^V] + 2x_2^V S_{12}\mathbb{E}[x_1^V|x_2^V] + S_{22}(x_2^V)^2\end{aligned}$$

Term (3) can be expressed as:

$$\begin{aligned}\mathbb{E}[((\tilde{X}_V - \hat{X}_V)\hat{\beta})^2|x_2^V] &= \mathbb{E}[((x_1^V - \hat{\psi})\tilde{\beta}_1)^2|x_2^V] \\ &= \mathbb{E}[((x_1^V - \hat{\psi})(\beta_1 + \eta_1))^2|x_2^V] \\ &= \beta_1^2 \mathrm{Var}[x_1^V|x_2^V] + S_{11}^2 \mathrm{Var}[x_1^V|x_2^V]\end{aligned}$$

The terms from (1) would be more or less the same if the test data were fully observed. They represent the impact on the prediction of the error made when estimating β.

On the other hand, those from (3) are both positive and unique to the incomplete case. They do not depend on the test data at all. The first term reflects how an error in the imputation of X_V is amplified by the regression coefficient when predicting y. The second one shows how errors in the estimation of β and of X_V combine when predicting y.

Most importantly, the least squares estimator is a strongly consistent one[18], which implies that the only error terms that matter with large n are those that remain when η is set to zero:

$$\beta_1^2 \mathrm{Var}[x_1^V|x_2^V] + \sigma^2$$

The missing data in the validation set adds a term that does not vanish for large n.

When there are more than two covariates An important point is that in a context with more than two covariates, the conditional variance of an observation increases when the number of unobserved variables increases: this means that when the dimension increases, not only do new error terms appear, but the existing ones also increase.

For X_V with $p > 2$ covariates, we denote X_V^{miss} and X_V^{obs} the missing and observed values in X_V. We can now write (3) again:

$$\begin{aligned}\mathbb{E}[((\tilde{X}_V - \hat{X}_V)\beta)^2|X_V^{\text{obs}}] &= \mathbb{E}[(\sum_{\substack{i=1 \\ i\in X_V^{\text{miss}}}}^{p} (x_i^V - \hat{x}_i^V)\beta_i)^2|X_V^{\text{obs}}] \\ &= \mathbb{E}[\sum_{\substack{i=1 \\ i\in X_V^{\text{miss}}}}^{p} \sum_{\substack{j=1 \\ j\in X_V^{\text{miss}}}}^{p} (x_i^V - \hat{x}_i^V)(x_j^V - \hat{x}_j^V)\beta_i\beta_j|X_V^{\text{obs}}]\end{aligned}$$

If, as previously, we choose $\hat{x}_i^V = \mathbb{E}[x_i^V|X_V^{\text{obs}}]$, we end up with:

$$\mathbb{E}[((\tilde{X}_V - \hat{X}_V)\beta)^2|X_V^{\text{obs}}] = \sum_{\substack{i,j=1 \\ i,j\in X_V^{\text{miss}}}}^{p} \beta_i\beta_j Cov(x_i^V, x_j^V|X_V^{\text{obs}})$$

4.2.3 When the data is large and the training data is fully observed

We can suppose that n is large and that we know $\tilde{X_V}$. In that case, we can take the approximation that $\beta = \tilde{\beta}$, that is, the only error in the estimation of β comes from the missing data in the training set. Then, the only error term that is nonzero is (2):

$$(\tilde{X}_V(\hat{\beta} - \tilde{\beta}))^2$$

To complete

4.3 Analysis

4.3.1 Implications

Theoretical consequences If the results above are any indication as to how things go in more complex settings, there are some interesting implications.

First, this confirms our intuition that using the conditional mean is the right thing to do to impute the validation data. This is important, especially as most imputation packages were made for multiple imputation and draw from the conditional distribution instead.

Second, there is a major asymmetry between the training and validation dataset: in the training set, every line works together with the others to help estimate some parameter. In particular, this means that even if all of the imputations are imperfect, with enough observations we can obtain a satisfactory estimate of the parameters (this is similar to the case of statistical inference with missing data, where under some assumptions the estimators are asymptotically consistent [19]). More data adds information, and even if it is incomplete it helps with the estimation.

The validation data is in a completely different situation. Missing data in the validation dataset adds error terms to the data that can be very large and do not vanish asymptotically. Intuitively, even if we have exactly the right β for regression, any error in the estimation of the data will be directly reflected as a prediction error proportional to the regression coefficient, while in the training set this effect is much more indirect. Such errors are inevitable, even if we know the exact distribution of the data, because it is random. The expected loss will be the same for every line on the validation set (with the same missingness pattern), so adding more validation lines will do nothing to reduce the mean error if they also have missing data.

This is important not only from the standpoint of model selection, but also because the validation error gives us an idea of how our model will perform on real-world data: missing data in the data we use for prediction can be a much more severe issue than missing data in our training database.

Implications for our data This could actually be a positive find. Take the example of Traumabase and haemorrhagic shock prediction: part of what our results mean is that the Traumabase can indeed be used to build a prediction tool, even if it has a significant amount of missing data. If the missing data is mostly due to errors of recording, this may mean that it is available in the real world when a doctor uses the tool: if the data used to make important predictions (that is, not a posteriori from the base but in a hospital when a patient needs care) can be kept full, then our estimates will have a chance of being very good, as long as we built our model with a large enough database.

The flip size, of course, is that when data is indeed missing from the validation dataset, there is little we can do to offset the resulting penalty. The missing value has a natural variability, even when controlling for every other observed variable, so even our best guess could have a high error.

4.3.2 Partial multiple imputation

This suggests an idea that could help mitigate this drawback while keeping the computational cost rather low. Using X_A the training data, we can estimate parameters ϕ, ψ, β as usual. Then, keeping these parameters constant, we can make draws from the conditional distribution of X_V and make predictions on the datasets generated this way. Using the quantiles of these predictions, we can build intervals that approximate the possible location of the true value of y and account for uncertainty. The idea behind this is that it is not necessary to multiply impute X_A because with lare n the estimated parameters will not really vary. It is also computationally intensive because it means we have to fit a new model for every imputed dataset. With our method, just one fit is needed and we only perform multiple predictions, which are usually cheaper.

To illustrate these results, we perform an analysis on some very simple simulated data, and check if the properties we derived are visible.

4.3.3 Verification on real and simulated data

We ran an analysis on two sets of data:

1. A simulated dataset designed to respect the assumptions of our model: the X data is generated as a multivariate normal for some random covariance matrix (with $p = 10$ variables), and the response are obtained linearly by scalar product with some parameter β (plus some noise)

2. A very simple real world dataset: the Abalone dataset. It is a regression dataset where the goal is to predict the age of a shell based on 7 measurements. It has no missing data and the covariates have high correlation.There are more that 4000 observations.

On both dataset, we ran the following procedure:

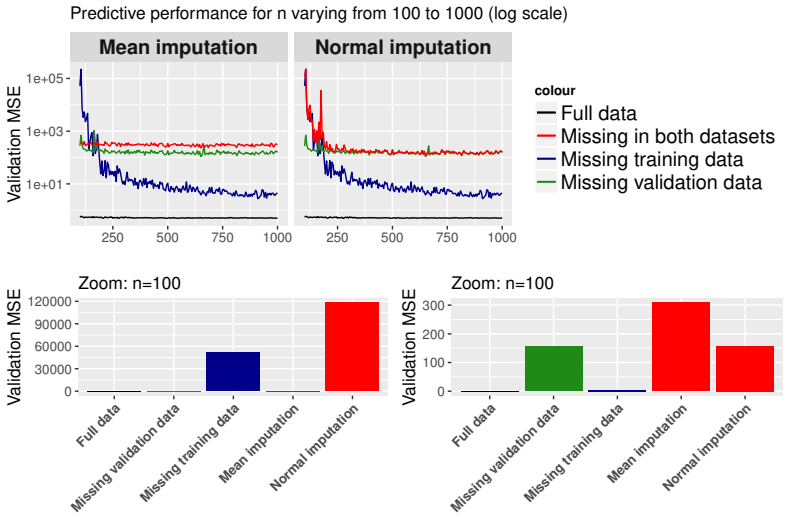
- Split the data in a training and a validation datasets.
- Perform one linear regression on the complete dataset.
- Add some (30%) missing data completely at random.
- Impute the data by the mean, perform a regression.
- Using the complete training data and the validation data with missing values, impute the data as a multivariate normal and perform a regression.
- Using the training data data with missing values and the complete validation, impute the data as a multivariate normal and perform a regression.
- Using both datasets with missing values, impute the data as a multivariate normal and perform a regression.
- Compute the mean squared prediction error for each of the five regressions on the validation data.

This is repeated multiple times to smooth out the variability of the results. We do this for a broad range of n values (for Abalone we select n rows at random for the analysis, for the simulated data we generate n rows each time).

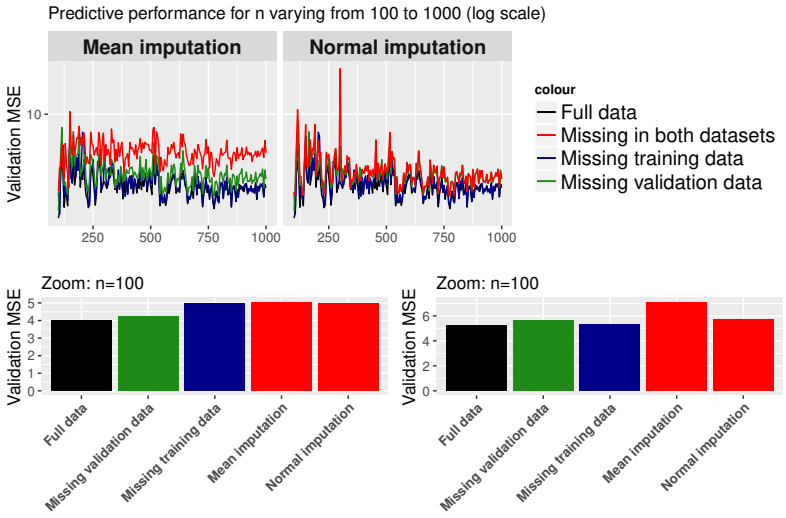
The results are shown in Figure 4.1.

The trends are less visible in the Abalone data, most likely because the base error from model misspecification is higher so the relative variations are smaller. However, we can see some general trends emerge.

For large n, the results match our expectations. The error is much lower when the validation data is known, while there is almost no difference



(a) Simulated data



(b) Abalone data

Figure 4.1: Linear regression with missing data

with and without knowing the real training data: with enough information we make up for the missing data when estimating the parameters. Even on the Abalone data (which is not really normally distributed), the normal imputation performs much better than imputation by the mean.

For small n, everything is very different. The first striking result is that

the main error term comes from the missing data in the training set, this time. This makes sense, as for such a small n it is very difficult to accurately estimate the covariance of the data, so missing values in the training data can seriously impact parameter estimation for the imputation. For the same reason, it is understandable that imputation by the mean works better than normal imputation in this context: with less parameters to estimate, we can at least have a fairly good estimation of the mean rather than a very bad estimation of both the mean and covariance.

Throughout the spectrum of n, we also see that the imputation with missing data everywhere tends to match whichever is worse between the full training data and the full validation data. These values act as lower bounds for the quality of prediction.

Chapter 5

Imputation and prediction: Empirical results

5.1 Is less missing data always better?

When performing an analysis, it is intuitive that we should limit the amount of missing data as much as possible, since missing data pollutes our estimates.

In particular, if the missing data is MCAR —and so the complete cases have exactly the same distribution as those with missing data—, and we have a large enough dataset with many complete cases, it is tempting to use only those complete cases to learn our model. Even in a context where we are training for prediction, and the real-world data will have some mising values we need to hande, it seems that we can use complete cases in the training data to learn both our prediction and imputation parameters as accurately as possible and then use those to predict the new data at best.

However, it may not be so: when imputing missing data, we do not recover the exact initial data. What if these errors change the structure of the dataset enough that a different parameter (different from the one that generated the data) can yield better predictions? In that case, learning our model without any missing data may yield the true parameter bt still not be optimal for prediction.

We investigated this on some simulated data by adding a fixed proportion of missing values to the validation data (mimicking the real-world data) and varying the amount of missing data in the training set:

Algorithm 5.1 Impact of missing data

Input: π_V, m
Input: $L_1, \dots L_m$

$X \sim \mathcal{N}(\mu, \Sigma)$
$y \leftarrow X\beta + \epsilon$ ▷ Where $\epsilon is a normal noise factor$
$X_A, X_V, y_A, y_V \leftarrow$ Random CV split of X, y
for $\pi_A \in [0, \frac{1}{m}, \dots \frac{m-1}{m}]$ **do**
    Add proportion π_A of MCAR missing data to X_A
    Add proportion $\pi_V A$ of MCAR missing data to X_V
    Impute $\hat{X}_A$ and $\hat{X}_V$ using μ_A the observed mean of X_A
    Compute $\hat{\beta}_A$ by linear regression on $\hat{X}_A, y_A$
    Predict $\hat{y}_V = \hat{X}_V \hat{\beta}_A$
    $L_i \leftarrow L(\hat{y}_V, y_V)$
end for

with $\Sigma = (1-\rho)I_p + \rho\mathbb{1}$ (same correlation between all variables). The results are shown in Fig.5.1 for $\rho = 0.9, \pi_V = 0.4$.

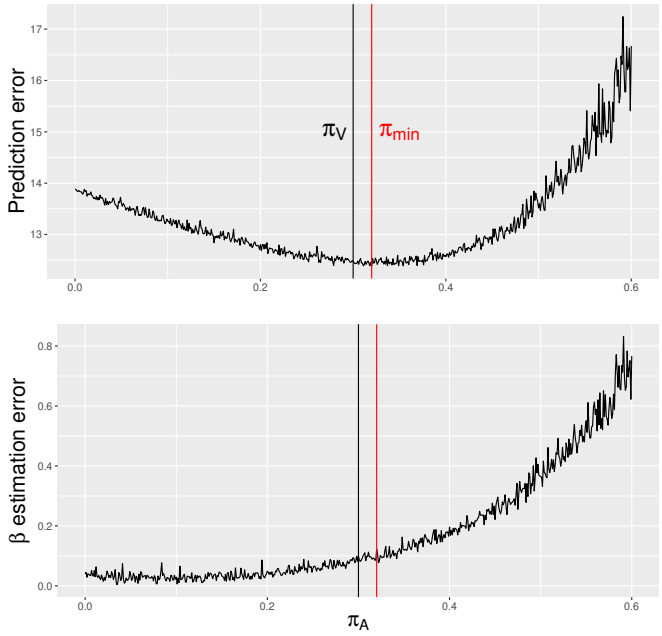


Figure 5.1: Prediction and parameter estimation errors depending on train missing data proportion π_A

bien rappeler que ça ne marche pas avec MVN imputation

5.2 Using y in the imputation

5.3 Multiple imputation

5.3.1 Prediction intervals

5.3.2 Point estimate

Chapter 6

Analysis: imputing the Traumabase data for prediction

6.1 Criteria for evaluation

6.2 Choosing the imputation method

Mention the fact that most methods have the same performance (even mean)
Imputation done before model selection (cf congeniality)

6.3 Methodology

6.4 Results

Conclusion

Bibliography

[1] Donald B Rubin. Inference and missing data. *Biometrika*, 63(3):581–592, 1976.

[2] Roderick JA Little and Donald B Rubin. *Statistical analysis with missing data*, volume 333. John Wiley & Sons, 2014.

[3] Ian H Witten, Eibe Frank, Mark A Hall, and Christopher J Pal. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2016.

[4] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[5] Yvonne Vergouwe, Patrick Royston, Karel GM Moons, and Douglas G Altman. Development and validation of a prediction model with missing predictor data: a practical approach. *Journal of clinical epidemiology*, 63(2):205–214, 2010.

[6] José M Jerez, Ignacio Molina, Pedro J García-Laencina, Emilio Alba, Nuria Ribelles, Miguel Martín, and Leonardo Franco. Missing data imputation using statistical and machine learning methods in a real breast cancer problem. *Artificial intelligence in medicine*, 50(2):105–115, 2010.

[7] Vladimir Naumovich Vapnik. An overview of statistical learning theory. *IEEE transactions on neural networks*, 10(5):988–999, 1999.

[8] Basic version of MICE Imputation github pull request. `https://github.com/scikit-learn/scikit-learn/pull/8478`. Scikit-learn repository.

[9] Joseph L Schafer. *Analysis of incomplete multivariate data*. Chapman and Hall/CRC, 1997.

[10] Get at the final model used in the MICE iterations? `https://github.com/stefvanbuuren/mice/issues/32`, . Github MICE repository.

[11] R MICE impute new observations. `https://stackoverflow.com/questions/40115226/r-mice-impute-new-observations`, . Stackoverflow discussion.

[12] Imputation using MICE: Use the train data to impute the missing test data. `https://stats.stackexchange.com/questions/332342/imputation-using-mice-use-the-train-data-to-impute-the-missing-test-dat` . Stackexchange discussion.

[13] James Honaker, Gary King, Matthew Blackwell, et al. Amelia ii: A program for missing data. *Journal of statistical software*, 45(7):1–47, 2011.

[14] Ported to R by Alvaro A. Novo. Original by Joseph L. Schafer <jls@stat.psu.edu>. *norm: Analysis of multivariate normal datasets with missing values*, 2013. URL `https://CRAN.R-project.org/package=norm`. R package version 1.0-9.5.

[15] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38, 1977.

[16] Joseph L Schafer. *Analysis of incomplete multivariate data*. Chapman and Hall/CRC, 1997.

[17] Fuzhen Zhang. *The Schur complement and its applications*, volume 4. Springer Science & Business Media, 2006.

[18] TW Anderson, John B Taylor, et al. Strong consistency of least squares estimates in normal linear regression. *The Annals of Statistics*, 4(4): 788–790, 1976.

[19] Roderick JA Little and Donald B Rubin. *Statistical analysis with missing data*, volume 333. John Wiley & Sons, 2014.