

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	1
1 Goal and data	3
1.1 hemorrhagic shock: a lethal but preventable condition	3
1.2 The Traumabase data	5
1.3 Objective and formalization	12
2 Imputation methods	15
2.1 Main types of imputation	15
2.2 Multiple imputation	15
2.3 Normality hypothesis: transforming the data	15
3 Methodology: imputation and the validation split	17
3.1 Empirical risk minimization (ERM): classical context	18
3.2 ERM with missing data: the problem of current methodologies	19
3.3 Possible solutions	21

4	Error sources and best imputation: the case of linear regression with missing data	25
4.1	Problem set-up	25
4.2	Partial resolution	28
4.3	Analysis	31
5	Empirical evaluation of some methods	37
5.1	Using y in the imputation	37
5.2	Multiple imputation	37
6	Analysis: imputing the Traumabase data for prediction	39
6.1	Criteria for evaluation	39
6.2	Choosing the imputation method	39
6.3	Methodology	39
7	Results	41
	Conclusion	43
	Bibliography	45

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 7).

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 haemorrhage [?]: 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 is 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 is 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, blood 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 the 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 went 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 prediction 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 between 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 in 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, on 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 saturation 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 between 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).

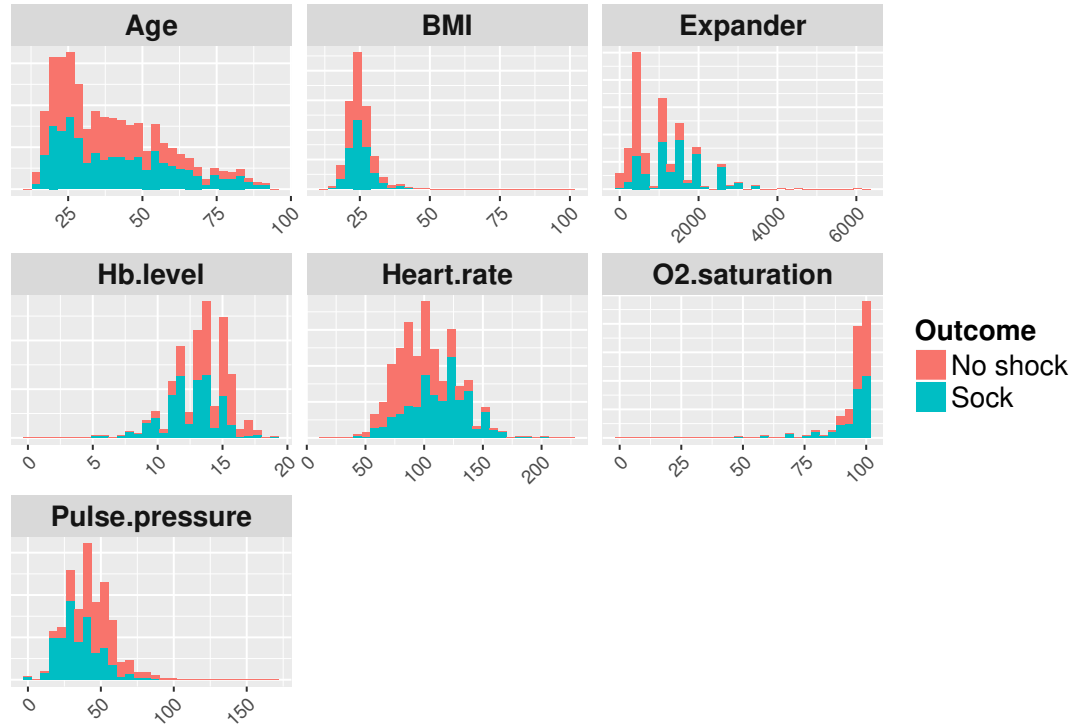


Figure 1.1: Distribution of the continuous variables depending on patient outcome

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

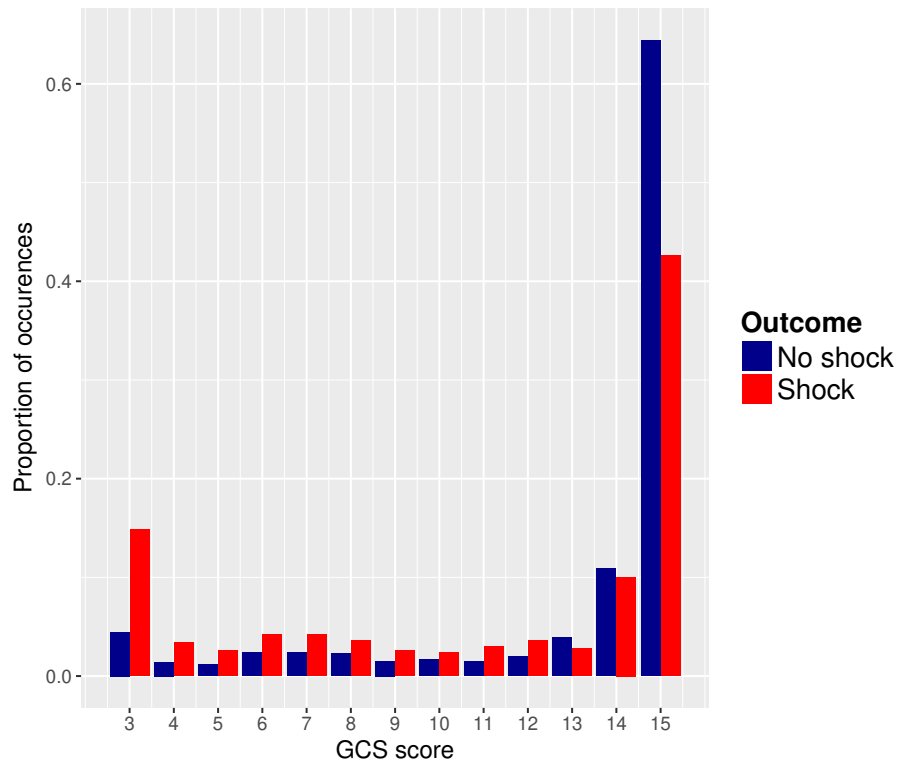


Figure 1.2: Distribution of Glasgow Coma Scale score depending on patient outcome

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 it 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.

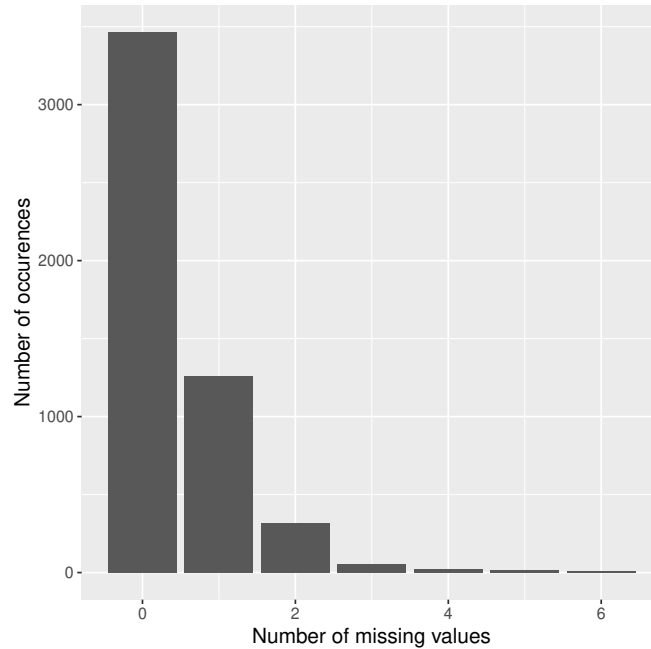


Figure 1.4: Distribution of the number of missing values

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.

Age	1.00	0.03	-0.00	-0.01	-0.01	-0.01	0.02	0.03
BMI	0.03	1.00	0.00	0.06	0.07	0.04	0.09	0.13
Glasgow	-0.00	0.00	1.00	0.01	0.05	0.06	0.03	0.01
Heart.rate	-0.01	0.06	0.01	1.00	0.64	0.23	0.26	0.14
Pulse.pressure	-0.01	0.07	0.05	0.64	1.00	0.21	0.34	0.12
Hb.level	-0.01	0.04	0.06	0.23	0.21	1.00	0.11	0.06
O2.saturation	0.02	0.09	0.03	0.26	0.34	0.11	1.00	0.08
Expander	0.03	0.13	0.01	0.14	0.12	0.06	0.08	1.00

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 prehospital 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 remind the issue we are trying to address: 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 they are at risk of HS.

It is clear that from 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, which represents the end user who is only interested in prediction. Their interaction 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 Imputer and Analyst 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 $X_{new}^{imp} = f(X_{new}, \hat{\alpha})$ and $\hat{y}_{new} = g(X_{new}^{imp}, \hat{\beta})$.
- \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

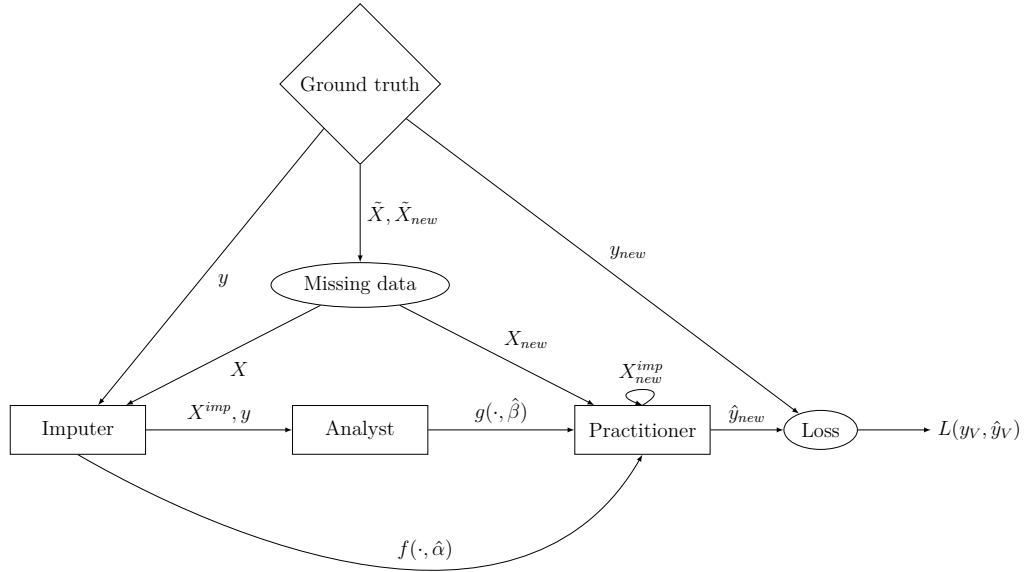


Figure 1.6: Imputation framework

The purpose of this formalization is to clarify the separation between the different phases of the inference, and show how the information is divided. In particular, the Practitioner only knows the

Chapter 2

Imputation methods

2.1 Main types of imputation

2.1.1 Joint parametric specification

2.1.2 Fully conditional specification: the MICE algorithm

2.1.3 Low-rank approximation for imputation

2.1.4 ML-based

2.2 Multiple imputation

2.2.1 Principle

2.2.2 Rubin's rule and prediction aggregation

2.3 Normality hypothesis: transforming the data

Chapter 3

Methodology: imputation and the validation split

The task we are trying to solve is quite peculiar: we are trying to impute the missing values in the data, not to perform a statistical analysis, but to select and train a prediction model. Our final benchmark of performance is not parameter estimation but predictive loss. As the next two chapters will show, although this seems like a small difference, this actually leads to some major changes.

It is interesting to note in that regard how the communities of statistics and machine learning seem to lack any point of convergence on the subject. Missing data have been an active field of research in statistics for a long time [1], and many complex methods have been developed and proven for statistical inference [2] (such as those described in Chapter 2).

On the other hand, there is almost no research on these methods when applied to prediction. Even recent manuals for machine learning [3] generally make only a quick mention of missing data, and in practice it is extremely rare for anything else than imputation by the mean to be used. *Scikit-learn* [4], by far the most-used machine learning package, only proposes imputation by the mean as of now.

A few research papers [5] [6] try to assess the performance of more recent imputation methods when used in a predictive context. However, they do not propose any framework or theory on this endeavour. They take it for granted that they can impute the whole dataset before performing the subsequent analysis. However, when performing prediction there is one significant difference with statistical inference, which we describe further in this chapter: the data is split into one dataset to learn the model, and another one to validate its performance. This raises many questions that we discuss here and in Chapter 4.

After laying out the general framework of Empirical Risk Minimization [7], which is the general paradigm used for prediction, we adapt it to fit the context of missing data. We notice that current implementations of modern imputation methods are incompatible with this framework, and try to devise solutions for this issue.

3.1 Empirical risk minimization (ERM): classical context

We first describe Empirical Risk Minimization (ERM) without missing data, as described in [7].

3.1.1 Context and notations

We are provided with a matrix X of size $n \times p$ and response vector y of size n . Our goal is to learn a model in the form $\hat{y} = f(x, \beta)$, where β is some parameter to choose, \hat{y} is a predicted value for y and f is a fixed parametric predictive function (usually corresponding to a choice of function in a given class).

The quality of a prediction is evaluated with the loss $L(y, \hat{y})$. The end goal in this context is to find the parameter $\hat{\beta}$ which minimizes the risk: $R = \mathbb{E}(L(y, f(X)))$. However, we do not have access to the real expectation of the risk, so we must use a proxy for this value. We define the empirical risk:

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

Empirical Risk Minimization corresponds to choosing β minimizing R_{emp} for our X and y . However, this is not enough. We do not just want the optimal β , we also want a measure of how well the final model would perform on new data. This is important because this is what we will take into account to make our choice of f . But the empirical risk gives us no measure of how well our model generalizes whatsoever, only of how closely it can fit known data.

To address the issue of model selection, the standard practice is to measure the error on some data that were not used to learn the model: it is the principle of cross-validation.

3.1.2 Cross-validation

To perform Cross-validation, we divide the data in two datasets: first we choose $n_A < n$ entries in the dataset to be used to learn β : this is the training dataset X_A and response y_A . We denote $I_A = (i_1, \dots, i_{n_A})$ 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.

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

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

It is this value that we can compare to choose our model class f .

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. The context is almost the same as before: choosing a parametric model that takes as input the observed data and outputs a prediction for y .

3.2.1 Imputation seen as an ERM

Remember that the purpose of this work is to impute the data independently of the predictive model used afterwards. This does not change the framework of ERM but it does mean that we cannot use any function we like to go from X 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 predicting y as we did previously with one caveat: we do not know the true data, even on the training dataset. Thus we choose $\hat{\alpha}$ to minimize some unsupervised loss

$$L'(g(X, \alpha), \alpha)$$

measuring the fit of the model to the data. Generally, this means that we choose a parameter that maximizes the likelihood of the observed data

according to some generative model (though it is not always the case). Once this is done, we obtain a completed dataset \hat{X} .

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

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

Putting it all together, we can define

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

the combined model that takes the observed data as input and outputs a predicted y . This is optimized as

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

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 in theory its parameters *must* be subjected to cross-validation just like those of the prediction. That is X_A the training data are used to estimate $(\hat{\beta}, \hat{\alpha})$ as shown just above. Then, we can compute a prediction $\hat{y}_V = h(X_V, (\hat{\beta}, \hat{\alpha}))$ which can be compared to y_V to evaluate the choice of model.

The bottom line is that just like for the prediction, 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

Over the years, many imputation methods have been proposed, and we describe some in Chapter 2. They have various assumptions and principles but they have one thing in common: they are all implemented (usually in R) via a single function. That function 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 model itself*.

This is a problem because of how cross-validation is supposed to be performed. Normally, one would estimate $(\hat{\beta}_A, \hat{\alpha}_A)$ 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 function $g' : X \mapsto g(X, \hat{\alpha}_X)$ where $\hat{\alpha}_X$ is the optimised parameter for the argument X . It is straightforward to see that using such a function, one can no longer separate the estimation of the parameters and the imputation in itself.

Incidentally, this issue was mentioned recently [8], as part of a larger effort to port the MICE [9] imputation method to Scikit-learn [4]. This package is developed for data scientists, who are used to having separate functions for training and prediction. Likewise, several other mentions of this problem arose in the last two years [10][11][12] with no mention of it earlier. It seems that as more elaborate imputation methods become popular with the machine learning community, the need arises for a change in the way those methods are implemented. However, it is clear that such changes take time, and for now we can do one of two things: make do with what is available, or build our own implementation to adapt one of these methods to our goals.

3.3 Possible solutions

3.3.1 Using current implementations

Methods

There are many ways we can use to approximate the correct procedure, though none of them is theoretically satisfactory. We describe the most natural ones and their caveats.

Grouped imputation Impute all of the data at once before performing the cross-validation split. That is, estimate $\hat{\alpha}_X$ on the whole data then

$$\hat{\beta}_{X_A} = \arg \min_{\beta} R_{emp}(y, h(X, (\beta, \hat{\alpha}_X)))$$

and $\hat{y}_V = h(X_V, (\hat{\beta}_{X_A}, \hat{\alpha}_X))$.

In that case, the parameter α is imputed using the validation data. In particular, imputed values in X_A *depend on those in* X_V , which is contrary to the basic principles of cross-validation where validation data must be held out during parameter estimation. In theory this could falsify the validation error by making it too optimistic (if some hard-to-predict observations are

present in the test set, that would otherwise have high validation error but will not in this case because of our 'cheating').

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

$$\begin{aligned}\hat{\alpha}_{X_A} &= \arg \min_{\alpha} R'_{emp}(X_A, \alpha) \\ \hat{\alpha}_{X_V} &= \arg \min_{\alpha} R'_{emp}(X_V, \alpha) \\ \hat{\beta}_{X_A} &= \arg \min_{\beta} R_{emp}(y, h(X, (\beta, \hat{\alpha}_{X_A}))) \\ \hat{y}_V &= h(X_V, (\hat{\beta}_{X_A}, \hat{\alpha}_{X_V}))\end{aligned}$$

Contrarily to grouped imputation, this does not violate cross-validation at all. However, we are using parameter α_{X_V} to impute the validation data and then apply a predictive model that uses β_{X_A} , optimized for data imputed with α_{X_A} . If sample size is not large enough, it is possible that α_{X_A} and α_{X_V} will be noticeably different. In that case, there is no guarantee that β_{X_A} will still be valid for prediction on the validation dataset that was imputed differently.

Line by line imputation The last option we mention here is to first impute the training data on its own. Then for every line of the validation data we impute it by stacking it with the imputed training data and imputing the whole dataset. Since it is just one line, we can safely assume that the imputation parameters will be those of the training data. The main issue with this method is that is the validation data is rather large, this will be computationally infeasible.

Need for a correct implementation

We want to understand if the alternatives proposed here are good enough to be used if a correct implementation is impossible. 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, ideally without needing to have access to the

whole Traumabase data (which is hard to share because it contains sensitive patient information). 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 the other ones require access to the full data).

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

3.3.2 A new variant: Multivariate Normal conditional expectation with reserved data

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 assume that both X_A and X_V follow a normal distribution $\mathcal{N}(\mu, \Sigma)$ with unknown parameters.

Parameter estimation It is possible to approximate maximum-likelihood estimators for μ and σ iteratively, using the EM algorithm [15] [16]. First, the missing values of X are imputed randomly to give $X^{(1)}$. Then we repeat the following steps:

For $t = 1 \dots$

1. $(\mu^{(t)}, \Sigma^{(t)})$ are computed as the maximum likelihood estimators on the completed data $X^{(t)}$ for the normal distribution (empirical mean and covariance)
2. $X^{(t+1)}$ is computed by replacing the missing values by their expected value under the new parameters:

$$X^{(t+1)} = \mathbb{E}(X|X^{\text{obs}}; \mu^{(t)}, \Sigma^{(t)})$$

The conditional expectation is easily derived using the Schur complement [17].

Once this converges we have an estimated parameter $\hat{\alpha} = (\hat{\mu}, \hat{\Sigma})$.

Imputation Once we have the parameters, it is very straightforward to get an imputation of the missing data. There are two different ways of doing this:

- Replace missing values by their expectation conditional on the observed data:

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

That is, pick the mean of the conditional distribution.

- Draw the missing values from the conditional distribution

In the second case, the *norm* package provides all that is needed. This is what would be used for multiple imputation. However, as we see in Chapter 4, the mean is the optimal choice if we want a single best imputation to use for a prediction model. We implemented this method. We use it below in order to compare the methods mentioned before.

3.3.3 Comparison on simulated data

In order to compare the methods used for imputation

Chapter 4

Error sources and best imputation: the case of linear regression with missing data

In order to make good decisions for imputation, it is important to understand how it impacts prediction. In order to get an idea on this issue, we solve a very simple case of cross-validated regression with missing data. We found no theoretical results whatsoever about the relationship between imputation and prediction.

4.1 Problem set-up

We place ourselves in a simple linear regression context with two covariates, where one observation is missing in the training dataset, and one in the validation dataset.

4.1.1 Notations

True data

The true, unobserved data is a standard regression setup, with the exception that one line of the data is reserved as 'validation data' ¹, with the rest called 'training data' — as is standard in predictive contexts (cf Chapter 3). The response variable y is a noised linear combination of the covariates in

¹For this calculation we only need one line of validation data since we will be using expected values.

X :

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

$$\tilde{X}_V = \begin{pmatrix} x_1^V & x_2^V \end{pmatrix} \quad \text{and} \quad y_V = X_V \beta + \epsilon_V \quad \text{with} \quad \epsilon_V \sim \mathcal{N}(0, \sigma^2)$$

The end goal is to learn an estimator on the training set that minimizes the expected loss on the validation set:

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

This is a well known problem with a simple solution. The regression estimator can be expressed as:

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

with and estimated response

$$\tilde{y}_V = \tilde{X}_V \tilde{\beta}$$

But there is one caveat: the data is actually not fully observed.

Observed data

What we actually have access to is slightly different: one observation is missing in the training set, as is one of the entries in the testing set. We observe the full y^A , but the covariate matrices we actually have access to are:

$$X^A = \begin{pmatrix} ? & x_{12} \\ \vdots & \vdots \\ x_{n1} & x_{n2} \end{pmatrix} \quad \text{and} \quad X^V = \begin{pmatrix} ? & x_2^V \end{pmatrix}$$

4.1.2 Imputed data and regression

Principle

To perform a linear regression similarly to what we did previously, we first have to fill in the blanks: we impute the missing data by replacing them with chosen values ϕ and ψ (which we choose using the value of what we observe).

$$\hat{X}^A = \begin{pmatrix} \phi & x_{12} \\ \vdots & \vdots \\ x_{n1} & x_{n2} \end{pmatrix} \quad \text{and} \quad \hat{X}^V = \begin{pmatrix} \psi & x_2^V \end{pmatrix}$$

This in turn allows us to perform the regression just like before:

$$\hat{\beta} = (\hat{X}_A^T \hat{X}_A)^{-1} \hat{X}_A^T y_A \quad \text{and} \quad \hat{y}_V(\phi, \psi) = \hat{X}_V \hat{\beta}$$

This means that what we really want to minimize is the following risk function:

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

where the only thing we choose is our decision rule for ϕ and ψ .

Distribution hypotheses

Lastly, for this last expression to have any meaning, we need to make some assumption on the distribution of X .

We make a very simple hypotheses of a multivariate normal distribution for the covariates — the parameters are assumed to be known, in real life they would be estimates.

$$\tilde{X}_A \sim \mathcal{N}(\mu, \Sigma) \quad \tilde{X}_V \sim \mathcal{N}(\mu, \Sigma)$$

With these, we know the distribution of the missing observations x_{11} and X_1^V conditional of the observed ones and we can try to find the ϕ and ψ values that give us the best expected loss.

Now that we have these notations and hypotheses, it is possible to solve the problem of the choice of imputation. Here we give a full expression of the loss and partial resolution.

4.2 Partial resolution

4.2.1 General loss

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

$$\begin{aligned} L(y_V, \hat{y}_V) &= (y_V - \hat{y}_V)^2 \\ &= (\tilde{X}_V \beta + \epsilon_V - \hat{X}_V \hat{\beta})^2 \\ &= (\tilde{X}_V(\beta - \tilde{\beta}) + \tilde{X}_V(\tilde{\beta} - \hat{\beta}) + (\tilde{X}_V - \hat{X}_V)\hat{\beta} + \epsilon_V)^2 \\ &= (\tilde{X}_V(\beta - \tilde{\beta}))^2 \end{aligned} \tag{1}$$

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

$$+ ((\tilde{X}_V - \hat{X}_V)\hat{\beta})^2 \tag{3}$$

$$+ \tilde{X}_V(\beta - \tilde{\beta})\tilde{X}_V(\tilde{\beta} - \hat{\beta}) \tag{4}$$

$$+ \tilde{X}_V(\beta - \tilde{\beta})(\tilde{X}_V - \hat{X}_V)\hat{\beta} \tag{5}$$

$$+ \tilde{X}_V(\tilde{\beta} - \hat{\beta})(\tilde{X}_V - \hat{X}_V)\hat{\beta} \tag{6}$$

$$+ \epsilon_V^2$$

$$+ \epsilon_V K$$

Where K is some term that will not matter (because when we take the expectation it will be zero). The risk we want to minimize is the expectation of this loss.

4.2.2 When the validation set is fully observed

Imputation

The first thing we can easily 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 $\tilde{\beta} = \hat{\beta}$, and all we have to choose is ψ . In the previously computed loss, it means that terms (2), (4) and (6) are zero.

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 ψ , 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:

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

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:

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

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:

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

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):

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

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 large 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 than 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.

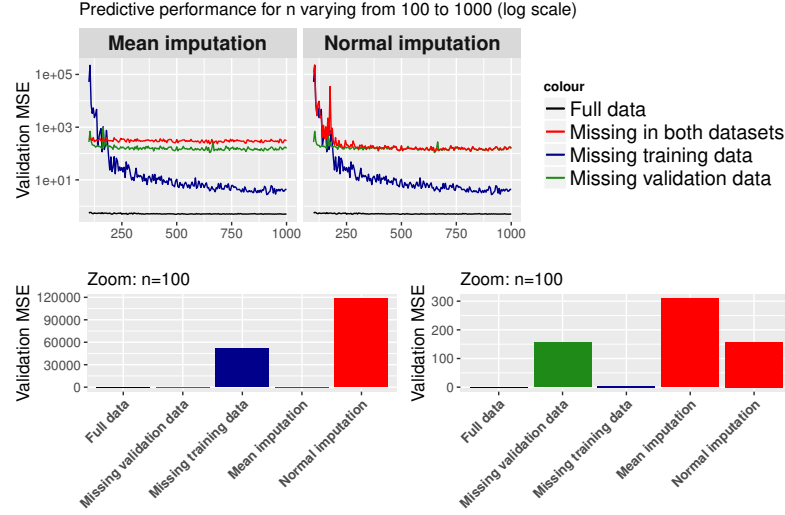
This is repeated multiple times to smooth out the variability of the results. We do this for a broad range of n values (for Abalone we select n rows at random for the analysis, for the simulated data we generate n rows each time).

The results are shown in Figure 4.1.

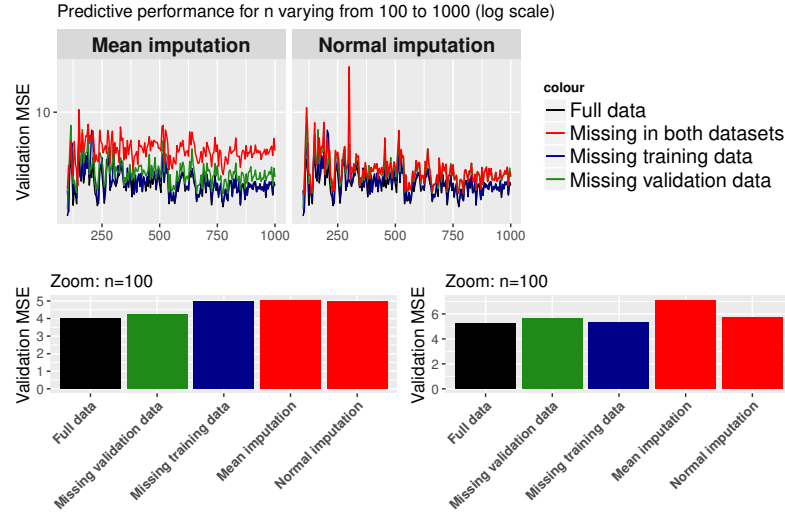
The trends are less visible in the Abalone data, most likely because the base error from model misspecification is higher so the relative variations are smaller. However, we can see some general trends emerge.

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 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



(a) Simulated data



(b) Abalone data

Figure 4.1: Linear regression with missing data

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

Empirical evaluation of some methods

5.1 Using y in the imputation

5.2 Multiple imputation

5.2.1 Prediction intervals

5.2.2 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

Chapter 7

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-data>, . 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.