Handling missing values: model-based approaches

Pierre-Alexandre Mattei

http://pamattei.github.io – @pamattei
pierre-alexandre.mattei@inria.fr

MSc Data Science

What are missing values?

- A few examples
- Mathematical framework

Statistical models of incomplete data

What are missing values?

- A few examples
- Mathematical framework

Statistical models of incomplete data

Example: heath records

	Center	Accident	Age	Sex	Weight	Height	BMI	BP	SBP
1	Beaujon	Fall	54	m	85	NR	NR	180	110
2	Lille	Other	33	m	80	1.8	24.69	130	62
3	Pitie Salpetriere	Gun	26	m	NR	NR	NR	131	62
4	Beaujon	AVP moto	63	m	80	1.8	24.69	145	89
6	Pitie Salpetriere	AVP bicycle	33	m	75	NR	NR	104	86
7	Pitie Salpetriere	AVP pedestrian	30	w	NR	NR	NR	107	66
9	HEGP	White weapon	16	m	98	1.92	26.58	118	54
10	Toulon	White weapon	20	m	NR	NR	NR	124	73

....

	SpO2	Temperature	Lactates	Hb	Glasgow	Transfusion	....
1	97	35.6	<NA>	12.7	12	yes	
2	100	36.5	4.8	11.1	15	no	
3	100	36	3.9	11.4	3	no	
4	100	36.7	1.66	13	15	yes	
6	100	36	NM	14.4	15	no	
7	100	36.6	NM	14.3	15	yes	
9	100	37.5	13	15.9	15	yes	
10	100	36.9	NM	13.7	15	no	

Figure: **Traumabase data set (figure from Julie Josse).**

Example: heath records

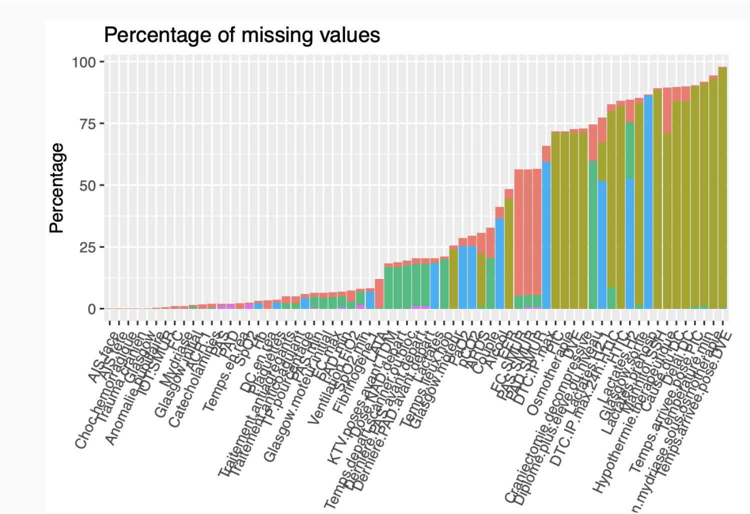

Figure: **Traumabase data set (picture from Julie Josse).**

Example: the Netflix prize

...

Example: semi-supervised image classification

...

Example: semi-supervised medical image segmentation

...

The need for some assumptions

There is a peculiarity when learning with incomplete data. In usual machine learning without missing data, given a very large data set, we usually have guarantees that we will learn the data generating distribution. When there are missing data, things get trickier.

The need for some assumptions

There is a peculiarity when learning with incomplete data. In usual machine learning without missing data, given a very large data set, we usually have guarantees that we will learn the data generating distribution. When there are missing data, things get trickier.

Example 1 (bad thermometer): we have a thermometer that breaks when the temperature is above 39 degrees Celsius. Then, it is impossible to learn the distribution of temperatures above 39 degrees, even given a very large data set!

The need for some assumptions

There is a peculiarity when learning with incomplete data. In usual machine learning without missing data, given a very large data set, we usually have guarantees that we will learn the data generating distribution. When there are missing data, things get trickier.

Example 2 (incomplete questionnaire): in a poll with multiple questions, there is a question about a sensitive issue in a questionnaire (e.g. on a US presidential election). Let's say that the people in favour of the Republican candidate are more likely to choose not to respond to the question. Then, just looking at the observed data, it will be very hard to predict the result of the election, without making additional assumption about the reasons for nonresponse.

The need for some assumptions

Example 2 bis (incomplete questionnaire): in a poll with multiple questions, there is a question about a sensitive issue in a questionnaire (e.g. on a US presidential election). Let's say that the people in favour of the Republican candidate are less likely to answer the phone. Again, just looking at the observed data, it will be very hard to predict the result of the election, even if we do a lot of large polls. Again, we need to make assumptions about the nonresponse process if we want to accurately predict the result of the election.

The need for some assumptions

Example 2 bis (incomplete questionnaire): in a poll with multiple questions, there is a question about a sensitive issue in a questionnaire (e.g. on a US presidential election). Let's say that the people in favour of the Republican candidate are less likely to answer the phone. Again, just looking at the observed data, it will be very hard to predict the result of the election, even if we do a lot of large polls. Again, we need to make assumptions about the nonresponse process if we want to accurately predict the result of the election.

When learning with missing values, **we need to make strong assumptions on the missingness process.** We will see some of these assumptions today, in particular the ones called missing completely at random (MCAR) and missing at random (MAR).

The need for some assumptions

Example 2 bis (incomplete questionnaire): in a poll with multiple questions, there is a question about a sensitive issue in a questionnaire (e.g. on a US presidential election). Let's say that the people in favour of the Republican candidate are less likely to answer the phone. Again, just looking at the observed data, it will be very hard to predict the result of the election, even if we do a lot of large polls. Again, we need to make assumptions about the nonresponse process if we want to accurately predict the result of the election.

When learning with missing values, **we need to make strong assumptions on the missingness process.** We will see some of these assumptions today, in particular the ones called missing completely at random (MCAR) and missing at random (MAR).

This explains why model-based approaches are a natural way to deal with missing values.

A mathematical framework for incomplete data

We assume that there exists a complete data matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$. We do not observe $\mathbf{X}$, and only have access to an incomplete version $\mathbf{Z} \in \tilde{\mathbb{R}}^{n \times p}$ where some values have been replaced by NAs. The values of the observed data matrix $\mathbf{Z}$ belong to $\tilde{\mathbb{R}} = \mathbb{R} \cup \{\mathtt{NA}\}$.
We also define a binary matrix $\mathbf{M} \in \{0,1\}^{n \times d}$ whose nonzero entries correspond to missing values.

$$\mathbf{X} = \begin{pmatrix} 1 & 3 & 9 \\ 2 & 7 & 0 \end{pmatrix}, \mathbf{M} = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}, \mathbf{Z} = \begin{pmatrix} \mathtt{NA} & 3 & \mathtt{NA} \\ 2 & \mathtt{NA} & \mathtt{NA} \end{pmatrix}.$$

Keep in mind that $\mathbf{X}$ is hidden: we only have access to $\mathbf{M}$ and $\mathbf{Z}$, and we will need to build our models and algorithms accordingly.

Supervised *vs* unsupervised

To keep things simple, we are going to consider unsupervised learning of continuous data $\mathbf{X} \in \mathbb{R}^{n \times d}$.

If you want to look at the supervised case, pretty much everything we're going to see today is still valid if you replace $\mathbf{X}$ by $(\mathbf{X}, \mathbf{Y})$. Of course, a few things will need to be adapted.

A statistical framework for incomplete data

Let's try to build a model for our $\mathbf{X}, \mathbf{M}, \mathbf{Z}$ framework

$$\mathbf{X} = \begin{pmatrix} 1 & 3 & 9 \\ 2 & 7 & 0 \end{pmatrix}, \mathbf{M} = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}, \mathbf{Z} = \begin{pmatrix} \mathtt{NA} & 3 & \mathtt{NA} \\ 2 & \mathtt{NA} & \mathtt{NA} \end{pmatrix}.$$

We want to build a statistical model for the data.

A statistical framework for incomplete data

Let's try to build a model for our $\mathbf{X}, \mathbf{M}, \mathbf{Z}$ framework

$$\mathbf{X} = \begin{pmatrix} 1 & 3 & 9 \\ 2 & 7 & 0 \end{pmatrix}, \mathbf{M} = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}, \mathbf{Z} = \begin{pmatrix} \mathtt{NA} & 3 & \mathtt{NA} \\ 2 & \mathtt{NA} & \mathtt{NA} \end{pmatrix}.$$

We want to build a statistical model for the data.

All the data we have access to are in $\mathbf{Z}$, so a first idea would be to directly model $\mathbf{Z}$. The issue with this is that the entries of $\mathbf{Z}$ live in $\tilde{\mathbb{R}} = \mathbb{R} \cup \{\mathtt{NA}\}$, which is not a well-behaved mathematical set.

A statistical framework for incomplete data

Let's try to build a model for our $\mathbf{X}, \mathbf{M}, \mathbf{Z}$ framework

$$\mathbf{X} = \begin{pmatrix} 1 & 3 & 9 \\ 2 & 7 & 0 \end{pmatrix}, \mathbf{M} = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}, \mathbf{Z} = \begin{pmatrix} \mathtt{NA} & 3 & \mathtt{NA} \\ 2 & \mathtt{NA} & \mathtt{NA} \end{pmatrix}.$$

We want to build a statistical model for the data.

All the data we have access to are in $\mathbf{Z}$, so a first idea would be to directly model $\mathbf{Z}$. The issue with this is that the entries of $\mathbf{Z}$ live in $\tilde{\mathbb{R}} = \mathbb{R} \cup \{\mathtt{NA}\}$, which is not a well-behaved mathematical set.

Alternative idea: build a model $p(\mathbf{X}, \mathbf{M})$ rather than directly model $p(\mathbf{Z})$. Why if this easier?

A statistical framework for incomplete data

Let's try to build a model for our $\mathbf{X}, \mathbf{M}, \mathbf{Z}$ framework

$$\mathbf{X} = \begin{pmatrix} 1 & 3 & 9 \\ 2 & 7 & 0 \end{pmatrix}, \mathbf{M} = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}, \mathbf{Z} = \begin{pmatrix} \mathtt{NA} & 3 & \mathtt{NA} \\ 2 & \mathtt{NA} & \mathtt{NA} \end{pmatrix}.$$

We want to build a statistical model for the data.

All the data we have access to are in $\mathbf{Z}$, so a first idea would be to directly model $\mathbf{Z}$. The issue with this is that the entries of $\mathbf{Z}$ live in $\tilde{\mathbb{R}} = \mathbb{R} \cup \{\mathtt{NA}\}$, which is not a well-behaved mathematical set.

Alternative idea: build a model $p(\mathbf{X}, \mathbf{M})$ rather than directly model $p(\mathbf{Z})$. Why if this easier?

- $(\mathbf{X}, \mathbf{M})$ lives in the easier-to-deal-with space $\mathbb{R}^{n \times d} \times \{0, 1\}^{n \times p}$, and in the rest of the course, we learned how to create statistical models for data that lives in this kind of space,

A statistical framework for incomplete data

Let's try to build a model for our $\mathbf{X}, \mathbf{M}, \mathbf{Z}$ framework

$$\mathbf{X} = \begin{pmatrix} 1 & 3 & 9 \\ 2 & 7 & 0 \end{pmatrix}, \mathbf{M} = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}, \mathbf{Z} = \begin{pmatrix} \mathtt{NA} & 3 & \mathtt{NA} \\ 2 & \mathtt{NA} & \mathtt{NA} \end{pmatrix}.$$

We want to build a statistical model for the data.

All the data we have access to are in $\mathbf{Z}$, so a first idea would be to directly model $\mathbf{Z}$. The issue with this is that the entries of $\mathbf{Z}$ live in $\tilde{\mathbb{R}} = \mathbb{R} \cup \{\mathtt{NA}\}$, which is not a well-behaved mathematical set.

Alternative idea: build a model $p(\mathbf{X}, \mathbf{M})$ rather than directly model $p(\mathbf{Z})$. Why if this easier?

- $(\mathbf{X}, \mathbf{M})$ lives in the easier-to-deal-with space $\mathbb{R}^{n \times d} \times \{0, 1\}^{n \times p}$, and in the rest of the course, we learned how to create statistical models for data that lives in this kind of space,
- a joint model $p(\mathbf{X}, \mathbf{M})$ will imply a model $p(\mathbf{Z})$, so we are still indirectly modelling $\mathbf{Z}$.

Building a joint model $p(\mathbf{X}, \mathbf{M})$

We will call $p(\mathbf{X}, \mathbf{M})$ a joint model, because it jointly models the features $\mathbf{X}$ and the missingness pattern $\mathbf{M}$. Part of the features are not observed so this will be a latent-variable model (the latent variables being the missing values).

Building a joint model $p(\mathbf{X}, \mathbf{M})$

We will call $p(\mathbf{X}, \mathbf{M})$ a joint model, because it jointly models the features $\mathbf{X}$ and the missingness pattern $\mathbf{M}$. Part of the features are not observed so this will be a latent-variable model (the latent variables being the missing values).

Today, we will focus on models where the ordering of the observations does not matter, so we will assume that $(\mathbf{x}_1, \mathbf{m}_1), ..., (\mathbf{x}_n, \mathbf{m}_n)$ (the rows of $\mathbf{X}$ and $\mathbf{M}$), are independent and identically distributed samples from a distribution $p(\mathbf{x}, \mathbf{m})$. This means that the model can be written

$$p(\mathbf{X}, \mathbf{M}) = \prod_{i=1}^{n} p(\mathbf{x}_i, \mathbf{m}_i).$$

Building a joint model $p(\mathbf{X}, \mathbf{M})$

We will call $p(\mathbf{X}, \mathbf{M})$ a joint model, because it jointly models the features $\mathbf{X}$ and the missingness pattern $\mathbf{M}$. Part of the features are not observed so this will be a latent-variable model (the latent variables being the missing values).

Today, we will focus on models where the ordering of the observations does not matter, so we will assume that $(\mathbf{x}_1, \mathbf{m}_1), ..., (\mathbf{x}_n, \mathbf{m}_n)$ (the rows of $\mathbf{X}$ and $\mathbf{M}$), are independent and identically distributed samples from a distribution $p(\mathbf{x}, \mathbf{m})$. This means that the model can be written

$$p(\mathbf{X}, \mathbf{M}) = \prod_{i=1}^{n} p(\mathbf{x}_i, \mathbf{m}_i).$$

In that context, what we "just" need to specify is $p(\mathbf{x}, \mathbf{m})$, a distribution over $\mathbb{R}^d \times \{0, 1\}^d$. Now, we will look at the assumptions we mentioned in the beginning of the lecture.

Aparté: Links with logistic regression

Before we start, I just want to remind you that we already studied kinda similar joint models, in the different context of logistic regression, we built models of (x, y)

$$p(x, y) = p(y|x)p(x).$$

Aparté: Links with logistic regression

Before we start, I just want to remind you that we already studied kinda similar joint models, in the different context of logistic regression, we built models of (x, y)

$$p(x, y) = p(y|x)p(x).$$

The idea of logistic regression was to make $p_\theta(y|x)$ dependent on a parameter θ, and assume that $p(x)$ is unknown but does not depend on θ.

Aparté: Links with logistic regression

Before we start, I just want to remind you that we already studied kinda similar joint models, in the different context of logistic regression, we built models of (x, y)

$$p(x, y) = p(y|x)p(x).$$

The idea of logistic regression was to make $p_\theta(y|x)$ dependent on a parameter θ, and assume that $p(x)$ is unknown but does not depend on θ.

With these assumption, we saw that the likelihood can be written

$$\sum_{i=1}^{n} \log p_{\boldsymbol{\theta}}(y_i, \mathbf{x}_i) = \sum_{i=1}^{n} \log p_{\boldsymbol{\theta}}(y_i|\mathbf{x}_i) + \sum_{i=1}^{n} \log p(\mathbf{x}_i),$$

but, since we don't model $p(\mathbf{x})$, $\sum_{i=1}^{n} \log p(\mathbf{x}_i)$ is constant, and maximising the likelihood is equivalent to maximising $\sum_{i=1}^{n} \log p_{\boldsymbol{\theta}}(y_i|\mathbf{x}_i)$.

Aparté: Links with logistic regression

Before we start, I just want to remind you that we already studied kinda similar joint models, in the different context of logistic regression, we built models of (x, y)

$$p(x, y) = p(y|x)p(x).$$

The idea of logistic regression was to make $p_\theta(y|x)$ dependent on a parameter θ, and assume that $p(x)$ is unknown but does not depend on θ.

With these assumption, we saw that the likelihood can be written

$$\sum_{i=1}^{n} \log p_{\boldsymbol{\theta}}(y_i, \mathbf{x}_i) = \sum_{i=1}^{n} \log p_{\boldsymbol{\theta}}(y_i|\mathbf{x}_i) + \sum_{i=1}^{n} \log p(\mathbf{x}_i),$$

but, since we don't model $p(\mathbf{x})$, $\sum_{i=1}^{n} \log p(\mathbf{x}_i)$ is constant, and maximising the likelihood is equivalent to maximising $\sum_{i=1}^{n} \log p_{\boldsymbol{\theta}}(y_i|\mathbf{x}_i)$.

We will use similar tricks to build our joint model $p(\mathbf{x}, \mathbf{m})$.

Building a joint model $p(\mathbf{x}, \mathbf{m})$

Using the product rule from probability theory, any model can be decomposed as

$$p(\mathbf{x}, \mathbf{m}) = p(\mathbf{m}|\mathbf{x})p(\mathbf{x}).$$

Building a joint model $p(\mathbf{x}, \mathbf{m})$

Using the product rule from probability theory, any model can be decomposed as

$$p(\mathbf{x}, \mathbf{m}) = p(\mathbf{m}|\mathbf{x})p(\mathbf{x}).$$

Typically, we can choose for $p(\mathbf{x})$, the model for the complete data, any of the models we've seen in the rest of this course: $p(\mathbf{x})$ could for example be a mixture model, or a single Gaussian, or a variational autoencoder...

Building a joint model $p(\mathbf{x}, \mathbf{m})$

Using the product rule from probability theory, any model can be decomposed as

$$p(\mathbf{x}, \mathbf{m}) = p(\mathbf{m}|\mathbf{x})p(\mathbf{x}).$$

Typically, we can choose for $p(\mathbf{x})$, the model for the complete data, any of the models we've seen in the rest of this course: $p(\mathbf{x})$ could for example be a mixture model, or a single Gaussian, or a variational autoencoder...

Now, what's left is to specify $p(\mathbf{m}|\mathbf{x})$. In general, this is very hard, and we usually need to have strong knowledge about the problem at hand. Indeed, let us go back to the election example: we would need to know which voters are more likely not to respond to model $p(\mathbf{m}|\mathbf{x})$ properly (and knowing this is hard!). Suprisingly, under some suitable assumptions, it is actually possible to not model $p(\mathbf{m}|\mathbf{x})$ at all, similarly to when we did not model $p(x)$ in logistic regression.

Building a joint model $p(\mathbf{x}, \mathbf{m}) = p(\mathbf{m}|\mathbf{x})p(\mathbf{x})$

If we want to ignore modelling $p(\mathbf{m}|\mathbf{x})$ like we ignored $p(x)$ in logistic regression, we need to write down the likelihood. What is the likelihood of incomplete data?

Building a joint model $p(\mathbf{x}, \mathbf{m}) = p(\mathbf{m}|\mathbf{x})p(\mathbf{x})$

If we want to ignore modelling $p(\mathbf{m}|\mathbf{x})$ like we ignored $p(x)$ in logistic regression, we need to write down the likelihood. What is the likelihood of incomplete data?

A convenient thing is to split a complete feature vector $\mathbf{x}$ into observed features $\mathbf{x}^{obs}$ and mission ones $\mathbf{x}^{miss}$, such that $\mathbf{x} = (\mathbf{x}^{obs}, \mathbf{x}^{miss})$. Using this in the equation gives

$$p(\mathbf{x}, \mathbf{m}) = p(\mathbf{m}|\mathbf{x}^{obs}, \mathbf{x}^{miss})p(\mathbf{x}^{obs}, \mathbf{x}^{miss}).$$

Building a joint model $p(\mathbf{x}, \mathbf{m}) = p(\mathbf{m}|\mathbf{x})p(\mathbf{x})$

If we want to ignore modelling $p(\mathbf{m}|\mathbf{x})$ like we ignored $p(x)$ in logistic regression, we need to write down the likelihood. What is the likelihood of incomplete data?

A convenient thing is to split a complete feature vector $\mathbf{x}$ into observed features $\mathbf{x}^{obs}$ and mission ones $\mathbf{x}^{miss}$, such that $\mathbf{x} = (\mathbf{x}^{obs}, \mathbf{x}^{miss})$. Using this in the equation gives

$$p(\mathbf{x}, \mathbf{m}) = p(\mathbf{m}|\mathbf{x}^{obs}, \mathbf{x}^{miss})p(\mathbf{x}^{obs}, \mathbf{x}^{miss}).$$

The likelihood of an incomplete data point is the marginal distribution of what we observe $p(\mathbf{x}^{obs}, \mathbf{m})$. This can be written

Building a joint model $p(\mathbf{x}, \mathbf{m}) = p(\mathbf{m}|\mathbf{x})p(\mathbf{x})$

If we want to ignore modelling $p(\mathbf{m}|\mathbf{x})$ like we ignored $p(x)$ in logistic regression, we need to write down the likelihood. What is the likelihood of incomplete data?

A convenient thing is to split a complete feature vector $\mathbf{x}$ into observed features $\mathbf{x}^{obs}$ and mission ones $\mathbf{x}^{miss}$, such that $\mathbf{x} = (\mathbf{x}^{obs}, \mathbf{x}^{miss})$. Using this in the equation gives

$$p(\mathbf{x}, \mathbf{m}) = p(\mathbf{m}|\mathbf{x}^{obs}, \mathbf{x}^{miss})p(\mathbf{x}^{obs}, \mathbf{x}^{miss}).$$

The likelihood of an incomplete data point is the marginal distribution of what we observe $p(\mathbf{x}^{obs}, \mathbf{m})$. This can be written

$$p(\mathbf{x}^{obs}, \mathbf{m}) = \int p(\mathbf{m}|\mathbf{x}^{obs}, \mathbf{x}^{miss})p(\mathbf{x}^{obs}, \mathbf{x}^{miss})d\mathbf{x}^{miss}.$$

Building a MCAR joint model $p(\mathbf{x}, \mathbf{m}) = p(\mathbf{m}|\mathbf{x})p(\mathbf{x})$

We want to simplify

$$p(\mathbf{x}^{obs}, \mathbf{m}) = \int p(\mathbf{m}|\mathbf{x}^{obs}, \mathbf{x}^{miss})p(\mathbf{x}^{obs}, \mathbf{x}^{miss})d\mathbf{x}^{miss},$$

in a way that avoids modelling explicitly $p(\mathbf{m}|\mathbf{x})$.

Building a MCAR joint model $p(\mathbf{x}, \mathbf{m}) = p(\mathbf{m}|\mathbf{x})p(\mathbf{x})$

We want to simplify

$$p(\mathbf{x}^{obs}, \mathbf{m}) = \int p(\mathbf{m}|\mathbf{x}^{obs}, \mathbf{x}^{miss})p(\mathbf{x}^{obs}, \mathbf{x}^{miss})d\mathbf{x}^{miss},$$

in a way that avoids modelling explicitly $p(\mathbf{m}|\mathbf{x})$.

A first way to do this is to assume that $\mathbf{m}$ and $\mathbf{x}$ are actually independent. This is called the missing completely at random (MCAR) assumption. In that case,

$$p(\mathbf{m}|\mathbf{x}^{obs}, \mathbf{x}^{miss}) = p(\mathbf{m}),$$

and we can write

$$p(\mathbf{x}^{obs}, \mathbf{m}) = p(\mathbf{m}) \int p(\mathbf{x}^{obs}, \mathbf{x}^{miss})d\mathbf{x}^{miss} = p(\mathbf{m})p(\mathbf{x}^{obs}).$$

Building a MCAR joint model $p(\mathbf{x}, \mathbf{m}) = p(\mathbf{m}|\mathbf{x})p(\mathbf{x})$

Let's assume that we have chose a parametric model $p_\theta(\mathbf{x})$ for the features. Under the MCAR assumption, the likelihood of the data is

$$\begin{aligned}\ell(\theta) &= \sum_{i=1}^{n} \log p_\theta(\mathbf{x}_i^{obs}, \mathbf{m}_i) = \sum_{i=1}^{n} \log p(\mathbf{m}_i) p_\theta(\mathbf{x}_i^{obs}) \\ &= \sum_{i=1}^{n} \log p(\mathbf{m}_i) + \sum_{i=1}^{n} \log p_\theta(\mathbf{x}_i^{obs});\end{aligned}$$

Building a MCAR joint model $p(\mathbf{x}, \mathbf{m}) = p(\mathbf{m}|\mathbf{x})p(\mathbf{x})$

Let's assume that we have chose a parametric model $p_\theta(\mathbf{x})$ for the features. Under the MCAR assumption, the likelihood of the data is

$$\ell(\theta) = \sum_{i=1}^{n} \log p_\theta(\mathbf{x}_i^{obs}, \mathbf{m}_i) = \sum_{i=1}^{n} \log p(\mathbf{m}_i) p_\theta(\mathbf{x}_i^{obs})$$
$$= \sum_{i=1}^{n} \log p(\mathbf{m}_i) + \sum_{i=1}^{n} \log p_\theta(\mathbf{x}_i^{obs});$$

and since the red term does not depend on θ, maximising $\ell(\theta)$ is equivalent to maximising

$$\sum_{i=1}^{n} \log p_\theta(\mathbf{x}_i^{obs}),$$

which is the likelihood of the observed features only.