Missing Data Imputation using Optimal Transport

Boris Muzellec [1] **Julie Josse** [2 3 *] **Claire Boyer** [4] **Marco Cuturi** [5 1]

Abstract

Missing data is a crucial issue when applying machine learning algorithms to real-world datasets. Starting from the simple assumption that two batches extracted randomly from the same dataset should share the same distribution, we leverage optimal transport distances to quantify that criterion and turn it into a loss function to impute missing data values. We propose practical methods to minimize these losses using end-to-end learning, that can exploit or not parametric assumptions on the underlying distributions of values. We evaluate our methods on datasets from the UCI repository, in MCAR, MAR and MNAR settings. These experiments show that OT-based methods match or out-perform state-of-the-art imputation methods, even for high percentages of missing values.

1. Introduction

Data collection is usually a messy process, resulting in datasets that have many missing values. This has been an issue for as long as data scientists have prepared, curated and obtained data, and is all the more inevitable given the vast amounts of data currently collected. The literature on the subject is therefore abundant (Little & Rubin, 2019; van Buuren, 2018): a recent survey indicates that there are more than 150 implementations available to handle missing data (Mayer et al., 2019). These methods differ on the objectives of their analysis (estimation of parameters and their variance, matrix completion, prediction), the nature of the variables considered (categorical, mixed, etc.), the assumptions about the data, and the missing data mechanisms. Imputation methods, which consist in filling missing entries with plausible values, are very appealing as they allow to both get a guess for the missing entries as well as to perform (with care) downstream machine learning methods on the completed data. Efficient methods include, among others, methods based on low-rank assumptions (Hastie et al., 2015), iterative random forests (Stekhoven & Buhlmann, 2011) and imputation using variational autoencoders (Mattei & Frellsen, 2019; Ivanov et al., 2019). A desirable property for imputation methods is that they should preserve the joint and marginal distributions of the data. Non-parametric Bayesian strategies (Murray & Reiter, 2016) or recent approaches based on generative adversarial networks (Yoon et al., 2018) are attempts in this direction. However, they can be quite cumbersome to implement in practice.

We argue in this work that the optimal transport (OT) toolbox constitutes a natural, sound and straightforward alternative. Indeed, optimal transport provides geometrically meaningful distances to compare discrete distributions, and therefore data. Furthermore, thanks to recent computational advances grounded on regularization (Cuturi, 2013), OT-based divergences can be computed in a scalable and differentiable way (Peyré et al., 2019). Those advances have allowed to successfully use OT as a loss function in many applications, including multi-label classification (Frogner et al., 2015), inference of pathways (Schiebinger et al., 2019) and generative modeling (Arjovsky et al., 2017; Genevay et al., 2018; Salimans et al., 2018). Considering the similarities between generative modeling and missing data imputation, it is therefore quite natural to use OT as a loss for the latter.

Contributions. This paper presents two main contributions. First, we leverage OT to define a loss function for missing value imputation. This loss function is the mathematical translation of the simple intuition that two random batches from the same dataset should follow the same distribution. Next, we provide algorithms for imputing missing values according to this loss. Two types of algorithms are presented, the first (i) being non-parametric, and the second (ii) defining a class of parametric models. The non-parametric algorithm (i) enjoys the most degrees of freedom, and can therefore output imputations which respect the global shape of the data while taking into account its local features. The parametric algorithm (ii) is trained in a round-robin fashion similar to iterative conditional imputation techniques, as

This research was done while JJ was a visiting researcher at Google Brain Paris. [1]CREST-ENSAE, IP Paris, Palaiseau, France [2]XPOP, INRIA Saclay, France [3]CMAP, UMR7641, École Polytechnique, IP Paris, Palaiseau, France [4]LPSM, Sorbonne Université, ENS Paris, France [5]Google Brain, Paris, France. Correspondence to: Boris Muzellec <boris.muzellec@ensae.fr>.

Proceedings of the 37^{th} International Conference on Machine Learning, Vienna, Austria, PMLR 119, 2020.

implemented for instance in the `mice` package (van Buuren & Groothuis-Oudshoorn, 2011). Compared to the non-parametric method, this algorithm allows to perform out-of-sample imputation. This creates a very flexible framework which can be combined with many imputing strategies, including imputation with Multi-Layer Perceptrons. Finally, these methods are showcased in extensive experiments on a variety of datasets and for different missing values proportions and mechanisms, including the difficult case of informative missing entries. The code to reproduce these experiments is available at `https://github.com/BorisMuzellec/MissingDataOT`.

Notations. Let $\mathbf{\Omega} = (\omega_{ij})_{ij} \in \{0,1\}^{n\times d}$ be a binary mask encoding observed entries, i.e. $\omega_{ij} = 1$ (resp. 0) *iff* the entry (i,j) is observed (resp. missing). We observe the following incomplete data matrix:

$$\mathbf{X} = \mathbf{X}^{(obs)} \odot \mathbf{\Omega} + \texttt{NA} \odot (\mathbb{1}_{n\times d} - \mathbf{\Omega}),$$

where $\mathbf{X}^{(obs)} \in \mathbb{R}^{n\times d}$ contains the observed entries, $\odot$ is the elementwise product and $\mathbb{1}_{n\times d}$ is an $n \times d$ matrix filled with ones. Given the data matrix $\mathbf{X}$, our goal is to construct an estimate $\hat{\mathbf{X}}$ filling the missing entries of $\mathbf{X}$, which can be written as

$$\hat{\mathbf{X}} = \mathbf{X}^{(obs)} \odot \mathbf{\Omega} + \hat{\mathbf{X}}^{(imp)} \odot (\mathbb{1}_{n\times d} - \mathbf{\Omega}),$$

where $\hat{\mathbf{X}}^{(imp)} \in \mathbb{R}^{n\times d}$ contains the imputed values. Let $\mathbf{x}_{i:}$ denote the i-th row of the data set $\mathbf{X}$, such that $\mathbf{X} = (\mathbf{x}_{i:}^T)_{1\leq i\leq n}$. Similarly, $\mathbf{x}_{:j}$ denotes the j-th column (variable) of the data set $\mathbf{X}$, such that $\mathbf{X} = (\mathbf{x}_{:1}|\dots|\mathbf{x}_{:d})$, and $\mathbf{X}_{:-j}$ denotes the dataset $\mathbf{X}$ in which the j-th variable has been removed. For $K \subset \{1,\dots,n\}$ a set of m indices, $\mathbf{X}_K = (\mathbf{x}_{k:})_{k\in K}$ denotes the corresponding batch, and by $\mu_m(\mathbf{X}_K)$ the empirical measure associated to $\mathbf{X}_K$, i.e.

$$\mu_m(\mathbf{X}_K) := \tfrac{1}{m}\sum_{k\in K} \delta_{\mathbf{x}_{k:}}.$$

Finally, $\Delta_n \overset{\text{def}}{=} \{\mathbf{a} \in \mathbb{R}_+^n : \sum_{i=1}^n a_i = 1\}$ is the simplex in dimension n.

2. Background

Missing data. Rubin (1976) defined a widely used - yet controversial (Seaman et al., 2013) - nomenclature for missing values mechanisms. This nomenclature distinguishes between three cases: missing completely at random (MCAR), missing at random (MAR), and missing not at random (MNAR). In MCAR, the missingness is independent of the data, whereas in MAR, the probability of being missing depends only on observed values. A subsequent part of the literature, with notable exceptions (Kim & Ying, 2018; Mohan & Pearl, 2019), only consider these "simple" mechanisms and struggles for the harder yet prevalent MNAR case. MNAR values lead to important biases in the data, as the probability of missingness then depends on the unobserved values. On the other hand, MCAR and MAR are "ignorable" mechanisms in the sense that they do not make it necessary to model explicitly the distribution of missing values when maximizing the observed likelihood.

The naive workaround which consists in deleting observations with missing entries is not an alternative in high dimension. Indeed, let us assume as in Zhu et al. (2019) that $\mathbf{X}$ is a $n \times d$ data matrix in which each entry is missing independently with probability 0.01. When $d = 5$, this would result in around 95% of the individuals (rows) being retained, but for $d = 300$, only around 5% of rows have no missing entries. Hence, providing plausible imputations for missing values quickly becomes necessary. Classical imputation methods impute according to a joint distribution which is either explicit, or implicitly defined through a set of conditional distributions. As an example, explicit joint modeling methods include imputation models that assume a Gaussian distribution for the data, whose parameters are estimated using EM algorithms (Dempster et al., 1977). Missing values are then imputed by drawing from their predictive distribution. A second instance of such joint modeling methods are imputations assuming low-rank structure (Josse et al., 2016). The conditional modeling approach (van Buuren, 2018), also known as "sequential imputation" or "imputation using chained equations" (ice) consists in specifying one model for each variable. It predicts the missing values of each variable using the other variables as explanatory, and cycles through the variables iterating this procedure to update the imputations until predictions stabilize.

Non-parametric methods like k-nearest neighbors imputation (Troyanskaya et al., 2001) or random forest imputation (Stekhoven & Buhlmann, 2011) have also been developed and account for the local geometry of the data. The herein proposed methods lie at the intersection of global and local approaches and are derived in a non-parametric and parametric version.

Wasserstein distances, entropic regularization and Sinkhorn divergences. Let $\alpha = \sum_{i=1}^n a_i\delta_{\mathbf{x}_i}$, $\beta = \sum_{i=1}^{n'} b_i\delta_{\mathbf{y}_i}$ be two discrete distributions, described by their supports $(\mathbf{x}_i)_{i=1}^n \in \mathbb{R}^{n\times p}$ and $(\mathbf{y}_i)_{i=1}^{n'} \in \mathbb{R}^{n'\times p}$ and weight vectors $\mathbf{a} \in \Delta_n$ and $\mathbf{b} \in \Delta_{n'}$. Optimal transport compares α and β by considering the most efficient of transporting the masses $\mathbf{a}$ and $\mathbf{b}$ onto each-other, according to a ground cost between the supports. The (2-)Wasserstein distance corresponds to the case where this ground cost is quadratic:

$$W_2^2(\alpha,\beta) \overset{\text{def}}{=} \min_{\mathbf{P}\in U(\mathbf{a},\mathbf{b})} \langle \mathbf{P}, \mathbf{M}\rangle, \tag{1}$$

where $U(\mathbf{a},\mathbf{b}) \overset{\text{def}}{=} \{\mathbf{P} \in \mathbb{R}^{n\times n'} : \mathbf{P}\mathbb{1}_{n'} = \mathbf{a}, \mathbf{P}^T\mathbb{1}_n = \mathbf{b}\}$ is the set of transportation plans, and $\mathbf{M} = \left(\|x_i - y_j\|^2\right)_{ij} \in$

$\mathbb{R}^{n\times n'}$ is the matrix of pairwise squared distances between the supports. W_2 is not differentiable and requires solving a costly linear program via network simplex methods (Peyré et al., 2019, §3). Entropic regularization alleviates both issues: consider

$$\mathrm{OT}_\varepsilon(\alpha,\beta) \stackrel{\text{def}}{=} \min_{\mathbf{P}\in U(\mathbf{a},\mathbf{b})} \langle \mathbf{P}, \mathbf{M}\rangle + \varepsilon h(\mathbf{P}), \qquad (2)$$

where $\varepsilon > 0$ and $h(\mathbf{P}) \stackrel{\text{def}}{=} \sum_{ij} p_{ij}\log p_{ij}$ is the negative entropy. Then, $\mathrm{OT}_\varepsilon(\alpha,\beta)$ is differentiable and can be solved using Sinkhorn iterations (Cuturi, 2013). However, due to the entropy term, OT_ε is no longer positive. This issue is solved through debiasing, by subtracting auto-correlation terms. Let

$$S_\varepsilon(\alpha,\beta) \stackrel{\text{def}}{=} \mathrm{OT}_\varepsilon(\alpha,\beta) - \frac{1}{2}(\mathrm{OT}_\varepsilon(\alpha,\alpha)+\mathrm{OT}_\varepsilon(\beta,\beta)). \quad (3)$$

Eq. (3) defines the Sinkhorn divergences (Genevay et al., 2018), which are positive, convex, and can be computed with little additional cost compared to entropic OT (Feydy et al., 2019). Sinkhorn divergences hence provide a differentiable and tractable proxy for Wasserstein distances, and will be used in the following.

OT gradient-based methods. Not only are the OT metrics described above good measures of distributional closeness, they are also well-adapted to gradient-based imputation methods. Indeed, let $\mathbf{X}_K, \mathbf{X}_L$ be two batches drawn from $\mathbf{X}$. Then, gradient updates for $\mathrm{OT}_\varepsilon(\mu_m(\mathbf{X}_K), \mu_m(\mathbf{X}_L)), \varepsilon \geq 0$ w.r.t a point $\mathbf{x}_{k:}$ in $\mathbf{X}_K$ correspond to taking steps along the so-called barycentric transport map. Indeed, with (half) quadratic costs, it holds (Cuturi & Doucet, 2014, §4.3) that

$$\nabla_{\mathbf{x}_{k:}} \mathrm{OT}_\varepsilon(\mu_m(\mathbf{X}_K), \mu_m(\mathbf{X}_L)) = \sum_\ell \mathbf{P}^\star_{k\ell}(\mathbf{x}_{k:} - \mathbf{x}_{\ell:}),$$

where $\mathbf{P}^\star$ is the optimal (regularized) transport plan. Therefore, a gradient based-update is of the form

$$\mathbf{x}_{k:} \leftarrow (1-t)\mathbf{x}_{k:} + t\sum_l \mathbf{P}^\star_{kl}\mathbf{x}_{l:}. \qquad (4)$$

In a missing value imputation context, Eq. (4) thus corresponds to updating values to make them closer to the target points given by transportation plans. Building on this fact, OT gradient-based imputation methods are proposed in the next section.

3. Imputing Missing Values using OT

Let $\mathbf{X}_K$ and $\mathbf{X}_L$ be two batches respectively extracted from the complete rows and the incomplete rows in $\mathbf{X}$, such that only the batch $\mathbf{X}_L$ contains missing values. In this one-sided incomplete batch setting, a good imputation should preserve the distribution from the complete batch, meaning that $\mathbf{X}_K$ should be close to $\mathbf{X}_L$ in terms of distributions. The OT-based metrics described in Section 2 provide natural criteria to catch this distributional proximity and derive imputation methods. However, as observed in Section 2, in high dimension or with a high proportion of missing values, it is unlikely or even impossible to obtain batches from $\mathbf{X}$ with no missing values. Nonetheless, a good imputation method should still ensure that the distributions of any two i.i.d. incomplete batches $\mathbf{X}_K$ and $\mathbf{X}_L$, *both* containing missing values, should be close. This implies in particular that OT-metrics between the distributions $\mu_m\mathbf{X}_K$ and $\mu_m\mathbf{X}_L$ should have small values. This criterion, which is weaker than the one above with one-sided missing data but is more amenable, will be considered from now on.

Direct imputation. Algorithm 1 is a direct implementation of this criterion, aiming to impute missing values for quantitative variables by minimizing OT distances between batches. First, missing values of any variable are initialized with the mean of observed entries plus a small amount of noise (to preserve the marginals and to facilitate the optimization). Then, batches are sequentially sampled and the Sinkhorn divergence between batches is minimized with respect to the imputed values, using gradient updates (here using RMSprop (Tieleman & Hinton, 2012)).

Algorithm 1 Batch Sinkhorn Imputation

Input: $\mathbf{X} \in (\mathbb{R}\cup\{\texttt{NA}\})^{\texttt{n}\times\texttt{d}}$, $\mathbf{\Omega} \in \{0,1\}^{n\times d}$, $\alpha,\eta,\varepsilon > 0$, $n \geq m > 0$,

Initialization: for $j = 1,\ldots,d$,

- for i s.t. $\omega_{ij} = 0$, $\hat{x}_{ij} \leftarrow \overline{\mathbf{x}^{obs}_{\cdot j}} + \varepsilon_{ij}$, with $\varepsilon_{ij} \sim \mathcal{N}(0,\eta)$ and $\overline{\mathbf{x}^{obs}_{\cdot j}}$ corresponds to the mean of the observed entries in the j-th variable (missing entries)
- for i s.t. $\omega_{ij} = 1$, $\hat{x}_{ij} \leftarrow x_{ij}$ (observed entries)

for t = 1, 2, ...,t_{max} **do**

Sample two sets K and L of m indices

$\mathcal{L}(\hat{\mathbf{X}}_K, \hat{\mathbf{X}}_L) \leftarrow S_\varepsilon(\mu_m(\hat{\mathbf{X}}_K), \mu_m(\hat{\mathbf{X}}_L))$

$\hat{\mathbf{X}}^{(imp)}_{K\cup L} \leftarrow \hat{\mathbf{X}}^{(imp)}_{K\cup L} - \alpha\text{RMSprop}(\nabla_{\hat{\mathbf{X}}^{(imp)}_{K\cup L}}\mathcal{L})$

end for

Output: $\hat{\mathbf{X}}$

OT as a loss for missing data imputation. Taking a step back, one can see that Algorithm 1 essentially uses Sinkhorn divergences between batches as a loss function to impute values for a model in which "one parameter equals one imputed value". Formally, for a fixed batch size m, this loss is defined as

$$\mathcal{L}_m(\mathbf{X}) \stackrel{\text{def}}{=} \sum_{\substack{K:0\leq k_1<\ldots<k_m\leq n\\ L:0\leq \ell_1<\ldots<\ell_m\leq n}} S_\varepsilon(\mu_m(\mathbf{X}_K), \mu_m(\mathbf{X}_L)). \qquad (5)$$

Eq. (5) corresponds to the "autocorrelation" counterpart of

the minibatch Wasserstein distances described in Fatras et al. (2019); Salimans et al. (2018).

Although Algorithm 1 is straightforward, a downside is that it cannot directly generate imputations for out-of-sample data points with missing values. Hence, a natural extension is to use the loss defined in Eq. (5) to fit parametric imputation models, provided they are differentiable with respect to their parameters. At a high level, this method is described by Algorithm 2. Algorithm 2 takes as an input an imputer

Algorithm 2 Meta Sinkhorn Imputation

Input: $\mathbf{X} \in \mathbb{R}^{n \times d}$, $\mathbf{\Omega} \in \{0,1\}^{n \times d}$, Imputer$(\cdot,\cdot,\cdot)$, Θ_0, $\varepsilon > 0$, $n \geq m > 0$,
$\hat{\mathbf{X}}^0 \leftarrow$ same initialization as in Algorithm 1
$\hat{\Theta} \leftarrow \Theta_0$
for $t = 1, 2, ..., t_{\text{max}}$ **do**
  for $k = 1, 2, ..., K$ **do**
    $\hat{\mathbf{X}} \leftarrow \text{Imputer}(\hat{\mathbf{X}}^t, \Omega, \hat{\Theta})$
    Sample two sets K and L of m indices
    $\mathcal{L}(\hat{\mathbf{X}}_K, \hat{\mathbf{X}}_L) \leftarrow S_\varepsilon(\mu_m(\hat{\mathbf{X}}_K), \mu_m(\hat{\mathbf{X}}_L))$
    $\nabla_\Theta \mathcal{L} \leftarrow \text{AutoDiff}(\mathcal{L}(\hat{\mathbf{X}}_K, \hat{\mathbf{X}}_L))$
    $\hat{\Theta} \leftarrow \hat{\Theta} - \alpha \text{Adam}(\nabla_\Theta \mathcal{L})$
  end for
  $\hat{\mathbf{X}}^{t+1} \leftarrow \text{Imputer}(\hat{\mathbf{X}}^t, \Omega, \hat{\Theta})$
end for
Output: Completed data $\hat{\mathbf{X}} = \hat{\mathbf{X}}^{t_{\text{max}}}$, Imputer$(\cdot,\cdot,\hat{\Theta})$

model with a parameter Θ such that Imputer$(\mathbf{X}, \Omega, \Theta)$ returns imputations for the missing values in $\mathbf{X}$. This imputer has to be differentiable w.r.t. its parameter Θ, so that the batch Sinkhorn loss $\mathcal{L}$ can be back-propagated through $\hat{\mathbf{X}}$ to perform gradient-based updates of Θ. Algorithm 2 does not only return the completed data matrix $\hat{\mathbf{X}}$, but also the trained parameter $\hat{\Theta}$, which can then be re-used to impute missing values in out-of-sample data.

Round-robin imputation. A remaining unaddressed point in Algorithm 2 is how to perform the "$\hat{\mathbf{X}} \leftarrow \text{Imputer}(\hat{\mathbf{X}}^t, \Omega, \Theta)$" step in the presence of missing values. A classical method is to perform imputations over variables in a round-robin fashion, i.e. to iteratively predict missing coordinates using other coordinates as features in a cyclical manner. The main advantage of this method is that it decouples variables being used as inputs and those being imputed. This requires having d sets of parameter $(\theta_j)_{1 \leq j \leq d}$, one for each variable, where each θ_j refers to the parameters used to to predict the j-th variable. The j-th variable is iteratively imputed using the $d-1$ remaining variables, according to the chosen model with parameter θ_j: $\hat{\theta}_j$ is first fitted (using e.g. regression or Bayesian methods), then the j-th variable is imputed. The algorithm then moves to the next variable $j+1$, in a cyclical manner. This round-robin method is implemented for instance in `R`'s `mice` package (van Buuren & Groothuis-Oudshoorn, 2011) or in the `IterativeImputer` method of the `scikit-learn` (Pedregosa et al., 2011) package. When using the Sinkhorn batch loss eq. (5) to fit the imputers, this procedure can be seen as a particular case of Algorithm 2 where the imputer parameter Θ is separable with respect to each variable $(\mathbf{x}_{:j})_{1 \leq j \leq d}$, i.e. Θ consists in d sets of parameter $(\theta_j)_{1 \leq j \leq d}$.

Making this round-robin imputation explicit in the step "$\hat{\mathbf{X}} \leftarrow \text{Imputer}(\hat{\mathbf{X}}^t, \Omega, \Theta)$" of Algorithm 2 leads to Algorithm 3. In Algorithm 3, an imputation $\hat{\mathbf{X}}^t, t = 0, ..., t_{\text{max}}$

Algorithm 3 Round-Robin Sinkhorn Imputation

Input: $\mathbf{X} \in \mathbb{R}^{n \times d}$, $\mathbf{\Omega} \in \{0,1\}^{n \times d}$, Imputer$(\cdot,\cdot,\cdot)$, Θ_0, $\varepsilon > 0$, $n \geq m > 0$,
$\hat{\mathbf{X}}^0 \leftarrow$ same initialization as in Algorithm 1
$(\hat{\theta}_1, ..., \hat{\theta}_d) \leftarrow \Theta_0$
for $t = 1, 2, ..., t_{\text{max}}$ **do**
  for $j = 1, 2, ..., d$ **do**
    for $k = 1, 2, ..., K$ **do**
      $\hat{\mathbf{X}}_{:j} \leftarrow \text{Imputer}(\hat{\mathbf{X}}^t_{:-j}, \Omega_{:j}, \hat{\theta}_j)$
      Sample two sets K and L of m indices
      $\mathcal{L} \leftarrow S_\varepsilon(\mu_m(\hat{\mathbf{X}}_K), \mu_m(\hat{\mathbf{X}}_L))$
      $\nabla_{\theta_j} \mathcal{L} \leftarrow \text{AutoDiff}(\mathcal{L})$
      $\hat{\theta}_j \leftarrow \hat{\theta}_j - \alpha \text{Adam}(\nabla_{\theta_j} \mathcal{L})$
    end for
    $\hat{\mathbf{X}}^t_{:j} \leftarrow \text{Imputer}(\hat{\mathbf{X}}^t_{:-j}, \Omega_{:j}, \hat{\theta}_j)$
  end for
  $\hat{\mathbf{X}}^{t+1} \leftarrow \hat{\mathbf{X}}^t$
end for
Output: Imputations $\hat{\mathbf{X}}^{t_{\text{max}}}$, Imputer$(\cdot,\cdot,\hat{\Theta})$

is updated starting from an initial guess $\hat{\mathbf{X}}^0$. The algorithm then consists in three nested loops. (i) The inner-most loop is dedicated to gradient-based updates of the parameter $\hat{\theta}_j$, as illustrated in Figure 1. Once this inner-most loop is finished, the j-th variable of $\hat{\mathbf{X}}^t$ is updated using the last update of $\hat{\theta}_j$. (ii) This is performed cyclically over all variables of $\hat{\mathbf{X}}^t$, yielding $\hat{\mathbf{X}}^{t+1}$. (iii) This fitting-and-imputation procedure over all variables is repeated until convergence, or until a given number of iterations is reached.

In practice, several improvements on the generic Algorithms 2 and 3 can be implemented:

1. To better estimate Eq. (5), one can sample several pairs of batches (instead of a single one) and define $\mathcal{L}$ as the average of S_ε divergences.

2. For Algorithm 3 in a MCAR setting, instead of sampling in each pair two batches from $\hat{\mathbf{X}}$, one of the two batches can be sampled with no missing value on the j-th variable, and the other with missing values on the

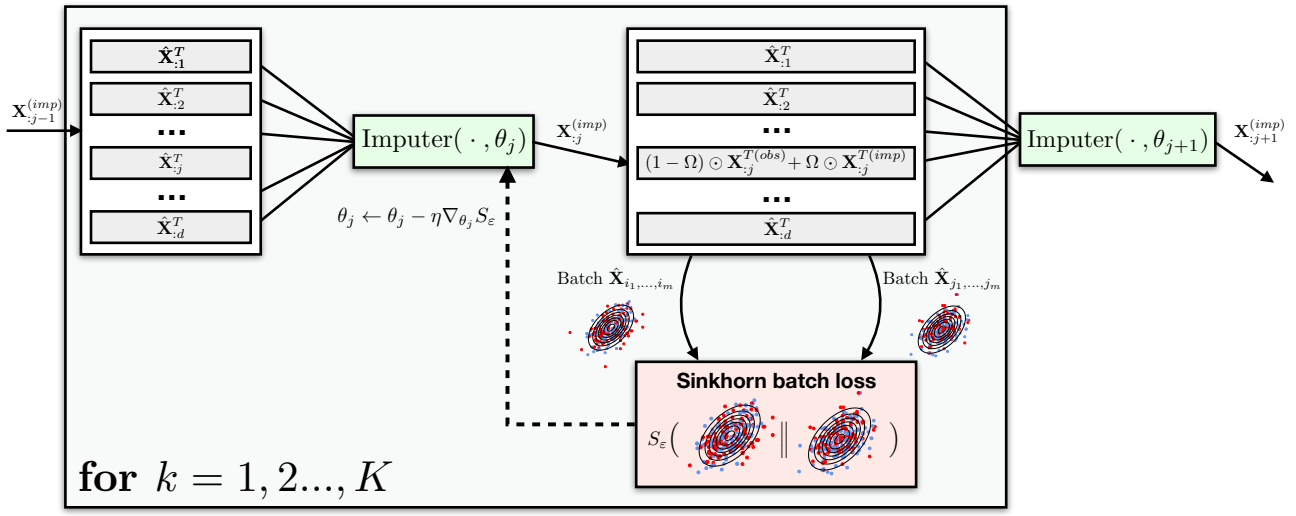

Figure 1: Round-robin imputation: illustration of the imputation of the j-th variable in the inner-most loop of Algorithm 3.

j-th variable. This allows the imputations for the j-th variable to be fitted on actual non-missing values. This helps ensuring that the imputations for the j-th variable will have a marginal distribution close to that of non-missing values.

3. The order in which the variables are imputed can be adapted. A simple heuristic is to impute variables in increasing order of missing values.

4. During training, the loss can be hard to monitor due to the high variance induced by estimating Eq. (5) from a few pairs of batches. Therefore, it can be useful to define a validation set on which fictional additional missing values are sampled to monitor the training of the algorithm, according to the desired accuracy score (e.g. MAE, RMSE or W_2 as in Section 4).

Note that item 2 is *a priori* only legitimate in a MCAR setting. Indeed, under MAR or MNAR assumptions, the distribution of non-missing data is in general not equal to the original (unknown) distribution of missing data.[1] Finally, the use of Adam (Kingma & Ba, 2014) compared to RMSprop in Algorithm 1 is motivated by empirical performance, but does not have a crucial impact on performance. It was observed however that the quality of the imputations given by Algorithm 1 seems to decrease when gradient updates with momentum are used.

4. Experimental Results

Baselines. We compare our methods to three baselines:

[1]Consider as an example census data in which low/high income people are more likely to fail to answer an income survey than medium income people.

(i) **mean** is the coordinate-wise mean imputation;

(ii) **ice** (imputation by chained equations) consists in (iterative) imputation using conditional expectation. Here, we use `scikit-learn`'s (Pedregosa et al., 2011) `iterativeImputer` method, which is based on `mice` (van Buuren & Groothuis-Oudshoorn, 2011). This is one of the most popular methods of imputation as it provides empirically good imputations in many scenario and requires little tuning;

(iii) **softimpute** (Hastie et al., 2015) performs missing values imputation using iterative soft-thresholded SVD's. This method is based on a low-rank assumption for the data and is justified by the fact that many large matrices are well approximated by a low-rank structure (Udell & Townsend, 2019).

Deep learning methods. Additionally, we compare our methods to three DL-based methods:

(iv) **MIWAE** (Mattei & Frellsen, 2019) fits a deep latent variable model (DLVM) (Kingma & Welling, 2014), by optimizing a version of the *importance weighted autoencoder* (IWAE) bound (Burda et al., 2016) adapted to missing data;

(v) **GAIN** (Yoon et al., 2018) is an adaptation of *generative adversarial networks* (GAN) (Goodfellow et al., 2014) to missing data imputation;

(vi) **VAEAC** (Ivanov et al., 2019) are VAEs with easily approximable conditionals that allow to handle missing data.

Transport methods. Three variants of the proposed methods are evaluated:

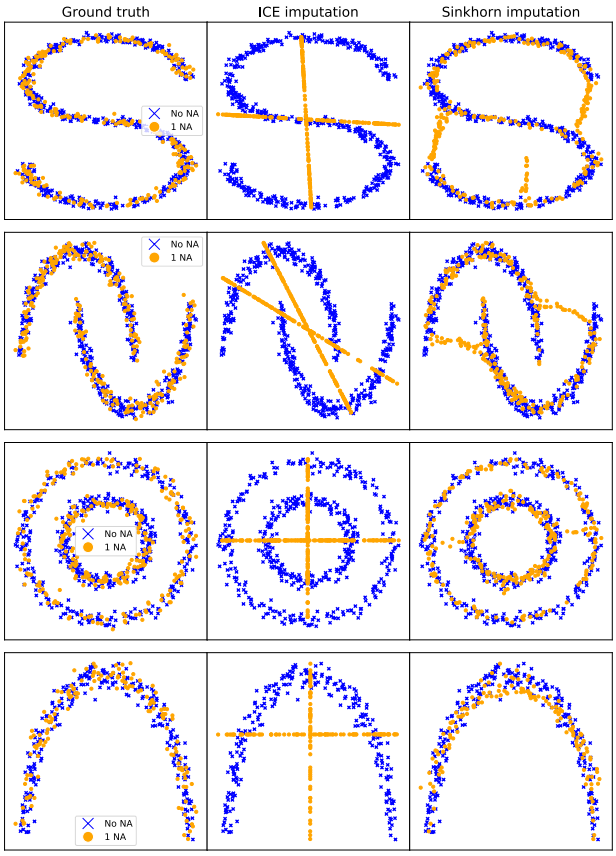

Figure 2: Toy examples: 20 % missing values (MCAR) on toy datasets. Blue points have no missing values, orange points have one missing value on either coordinate. **ice** outputs conditional expectation imputations, which are irrelevant due to the high non-linearity of these examples. Since algorithm 1 does not assume a parametric form for the imputations, it is able to satisfyingly impute missing values.

(vii) **Sinkhorn** designates the direct non-parametric imputation method detailed in Algorithm 1.

For Algorithm 3, two classes of imputers are considered:

(viii) **Linear RR** corresponds to Algorithm 3 where for $1 \leq j \leq d$, Imputer$(\cdot, \theta_j)$ is a linear model w.r.t. the $d-1$ other variables with weights and biases given by θ_j. This is similar to `mice` or `IterativeImputer`, but fitted with the OT loss eq. (5);

(ix) **MLP RR** denotes Algorithm 3 with shallow Multi-Layer Perceptrons (MLP) as imputers. These MLP's have the following architecture: (i) a first $(d-1) \times 2(d-1)$ layer followed by a ReLU layer then (ii) a $2(d-1) \times (d-1)$ layer followed by a ReLU layer and finally (iii) a $(d-1) \times 1$ linear layer. All linear layers have bias terms. Each Imputer$(\cdot, \theta_j), 1 \leq j \leq d$ is one such MLP with a different set of weights θ_j.

Toy experiments. In Figure 2, we generate two-dimensional datasets with strong structures, such as an S-shape, half-moon(s), or concentric circles. A 20% missing rate is introduced (void rows are discarded), and imputations performed using Algorithm 1 or the **ice** method are compared to the ground truth dataset. While the **ice** method is not able to catch the non-linear structure of the distributions at all, **Sinkhorn** performs efficiently by imputing faithfully to the underlying complex data structure (despite the two half-moons and the S-shape being quite challenging). This is remarkable, since Algorithm 1 does not rely on any parametric assumption for the data. This underlines in a low-dimensional setting the flexibility of the proposed method. Finally, note that the trailing points which can be observed for the S shape or the two moons shape come from the fact that Algorithm 1 was used as it is, i.e. with pairs of batches *both* containing missing values, even though these toy examples would have allowed to use batches without missing values. In that case, we obtain imputations that are visually indistinguishable from the ground truth.

Large-scale experimental setup. We evaluate each method on 23 datasets from the UCI machine learning repository[2] (see Table 1) with varying proportions of missing data and different missing data mechanisms. These datasets only contain quantitative features. Prior to running the experiments, the data is whitened (i.e. centered and scaled to variable-wise unit variance). For each dataset, all methods are evaluated on 30 different draws of missing values masks. For all Sinkhorn-based imputation methods, the regularization parameter ϵ is set to 5% of the median distance between initialization values with no further dataset-dependent tuning. If the dataset has more than 256 points, the batch size is fixed to 128, otherwise to $2^{\lfloor \frac{n}{2} \rfloor}$ where n is the size of the dataset. The noise parameter η in Algorithm 1 is fixed to 0.1. For Sinkhorn round-robin models (**Linear RR** and **MLP RR**), the maximum number of cycles is 10, 10 pairs of batches are sampled per gradient update, and an ℓ^2-weight regularization of magnitude 10^{-5} is applied during training. For all 3 Sinkhorn-based methods, we use gradient methods with adaptive step sizes as per algorithms 1 and 3, with an initial step size fixed to 10^{-2}. For **softimpute**, the hyperparameter is selected at each run through cross-validation on a small grid. This CV is performed by sampling additional missing values. For DL-based methods, the implementations provided in open-access by the authors were used[3][4][5], with the hyperparameter settings recommended in the corresponding papers. In particular, for **GAIN** the α parameter is selected using cross-validation. GPUs are used for Sinkhorn and deep learning methods. The code to reproduce

[2] https://archive.ics.uci.edu/ml/index.php
[3] https://github.com/pamattei/miwae
[4] https://github.com/jsyoon0823/GAIN
[5] https://github.com/tigvarts/vaeac

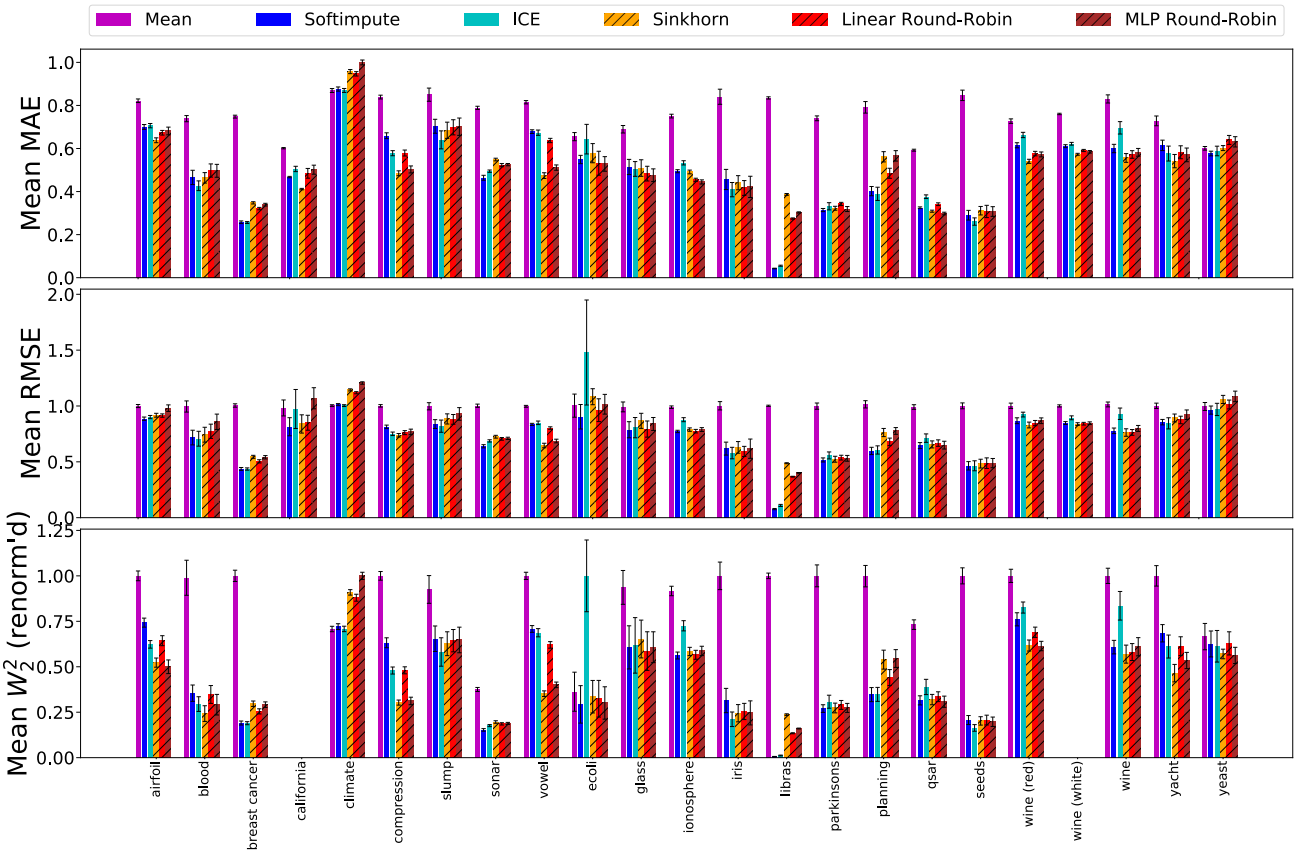

Figure 3: (**30% MCAR**) Imputation methods on 23 datasets from the UCI repository (Table 1). `Sinkhorn` denotes Algorithm 1 and `Linear RR`, `MLP RR` the two instances of Algorithm 3 precedently described. 30% of the values are missing MCAR. All methods are evaluated on 30 random missing values draws. Error bars correspond to $\pm$ 1 std. For readability we display scaled mean W_2^2, i.e. for each dataset we renormalize the results by the maximum W_2^2. For some datasets W_2 results are not displayed due to their large size, which makes evaluating the unregularized W_2 distance costly.

the experiments is available at `https://github.com/BorisMuzellec/MissingDataOT`.

Missing value generation mechanisms. The implementation of a MCAR mechanism is straightfoward. On the contrary, many different mechanisms can lead to a MAR or MNAR setting. We here describe those used in our experiments. In the **MCAR** setting, each value is masked according to the realization of a Bernoulli random variable with a fixed parameter. In the **MAR** setting, for each experiment, a fixed subset of variables that cannot have missing values is sampled. Then, the remaining variables have missing values according to a logistic model with random weights, which takes the non-missing variables as inputs. A bias term is fitted using line search to attain the desired proportion of missing values. Finally, two different mechanisms are implemented in the **MNAR** setting. The first is identical to the previously described MAR mechanism, but the inputs of the logistic model are then masked by a MCAR mechanism. Hence, the logistic model's outcome now depends on potentially missing values. The second mechanism, 'self masked', samples a subset of variables whose values in the lower and upper p-th percentiles are masked according to a Bernoulli random variable, and the values in-between are left not missing. As detailed in the appendix, MCAR experiments were performed with 10%, 30% and 50% missing rates, while MAR and both MNAR settings (quantile and logistic masking) were evaluated with a 30% missing rate.

Metrics. Imputation methods are evaluated according to two "pointwise" metrics: mean absolute error (MAE) and root mean square error (RMSE); and one metric on distributions: the squared Wasserstein distance between empirical distributions on points with missing values. Let $\mathbf{X} \in \mathbb{R}^{n \times d}$ be a dataset with missing values. When (i, j) spots a missing entry, recall that $\hat{x}_{ij}$ denotes the corresponding imputation, and let us note x_{ij}^{true} the ground truth. Let $m_0 \stackrel{\text{def}}{=} \#\{(i,j), \omega_{ij} = 0\}$ and $m_1 \stackrel{\text{def}}{=} \#\{i : \exists j,\ \omega_{ij} = 0\}$ respectively denote the total number of missing values and

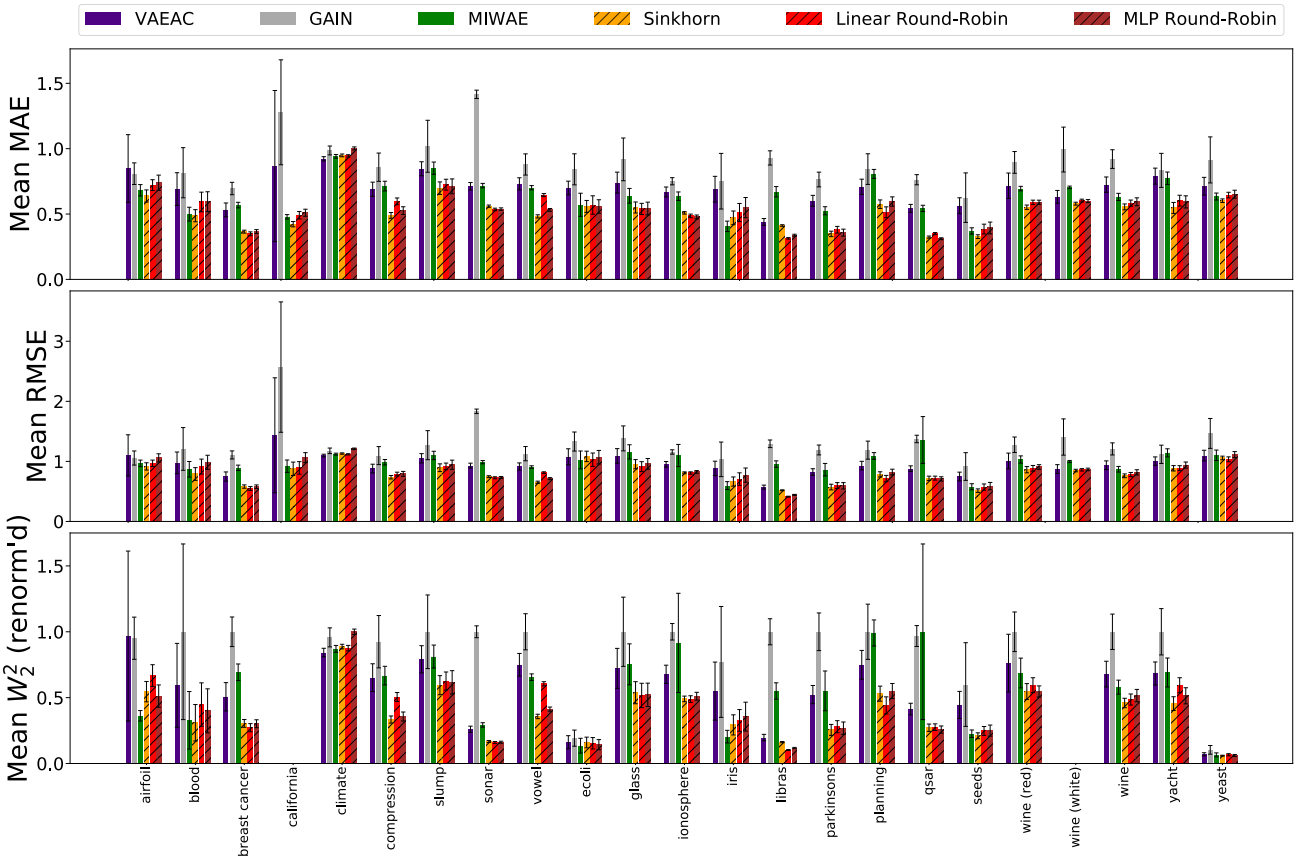

Figure 4: (**30% MNAR**) Imputation methods on 23 datasets from the UCI repository (Table 1). Values are missing MNAR according to the logistic mechanism described in Section 4, with 30% variables used as inputs of a logistic masking model for the 70% remaining variables. 30% of those input variables are then masked at random. Hence, all variables have 30% missing values. All methods are evaluated on the same 30 random missing values draws. Error bars correspond to $\pm$ 1 std. For readability we display scaled mean W_2^2, i.e. for each dataset we renormalize the results by the maximum W_2^2. For some datasets W_2 results are not displayed due to their large size, which makes evaluating the unregularized W_2 distance costly.

the number of data points with at least one missing value. Set $M_1 \stackrel{\text{def}}{=} \{i : \exists j, \omega_{ij} = 0\}$. We define MAE, RMSE and W_2 imputation metrics as

$$\frac{1}{m_0} \sum_{(i,j):\omega_{ij}=0} |x_{i,j}^{\text{true}} - \hat{x}_{ij}|, \qquad \text{(MAE)}$$

$$\sqrt{\frac{1}{m_0} \sum_{(i,j):\omega_{ij}=0} (x_{i,j}^{\text{true}} - \hat{x}_{ij})^2}, \qquad \text{(RMSE)}$$

$$W_2^2 \left(\mu_{m_1}(\hat{\mathbf{X}}_{M_1}), \mu_{m_1}(\mathbf{X}_{M_1}^{(true)}) \right). \qquad (W_2)$$

Results. The complete results of the experiments are reported in the Appendix. In Figure 3 and Figure 4, the proposed methods are respectively compared to baselines and Deep Learning (DL) methods in a MCAR and a logistic masking MNAR setting with 30% missing data. As can be seen from Figure 3, the linear round-robin model matches or out-performs `scikit`'s iterative imputer (**ice**) on MAE and RMSE scores for most datasets. Since both methods are based on the same cyclical linear imputation model but with different loss functions, this shows that the batched Sinkhorn loss in Eq. (5) is well-adapted to imputation with parametric models. Comparison with DL methods (Figure 4) shows that the proposed OT-based methods consistently outperform DL-based methods, and have the additional benefit of having a lower variance in their results overall. Interestingly, while the MAE and RMSE scores of the round-robin MLP model are comparable to that of the linear RR, its W_2 scores are generally better. This suggests that more powerful base imputer models lead to better W_2 scores, from which one can conclude that Eq. (5) is a good proxy for optimizing the unavailable Eq. (1) score, and that Algorithm 3 is efficient at doing so. Furthermore, one can observe that the direct imputation method is very competitive over all data and metrics and is in general the best performing OT-based method, as could be expected from the fact that its imputation model is

Table 1: Summary of datasets

dataset	n	d
airfoil_self_noise	1503	5
blood_transfusion	748	4
breast_cancer_diagnostic	569	30
california	20640	8
climate_model_crashes	540	18
concrete_compression	1030	7
concrete_slump	103	7
connectionist_bench_sonar	208	60
connectionist_bench_vowel	990	10
ecoli	336	7
glass	214	9
ionosphere	351	34
iris	150	4
libras	360	90
parkinsons	195	23
planning_relax	182	12
qsar_biodegradation	1055	41
seeds	210	7
wine	178	13
wine_quality_red	1599	10
wine_quality_white	4898	11
yacht_hydrodynamics	308	6
yeast	1484	8

not restricted by a parametric assumption. This favorable behaviour tends to be exacerbated with a growing proportion of missing data, see Figure 9 in the appendix.

MAR and MNAR. Figure 4 above and Figures 10 to 12 in the appendix display the results of our experiments in MAR and MNAR settings, and show that the proposed methods perform well and are robust to difficult missingness mechanisms. This is remarkable, as the proposed methods do not attempt to model those mechanisms. Finally, note that the few datasets on which the proposed methods do not perform as well as baselines – namely `libras` and to a smaller extent `planning_relax` – remain consistently the same across all missingness mechanisms and missing rates. This suggests that this behavior is due to the particular structure of those datasets, rather than to the missingness mechanisms themselves.

Out-of-sample imputation. As mentioned in Section 3, a key benefit of fitting a parametric imputing model with algorithms 2 and 3 is that the resulting model can then be used to impute missing values in out-of-sample (OOS) data. In Figure 5, we evaluate the Linear RR and MLP RR models in an OOS imputation experiment. We compare the training and OOS MAE, RMSE and OT scores on a collection of datasets selected to have a sufficient number of points. At each run, we randomly sample 70% of the data to be used for training, and the remaining 30% to evaluate OOS imputation. 30% of the values are missing MCAR, uniformly over training and testing sets. Out of the methods presented earlier on, we keep those that allow OOS: for the **ice**, **Linear RR** and **MLP RR** methods, OOS imputation is simply performed using the round-robin scheme without further fitting of the parameters on the new data. For the **mean** baseline, missing values in the testing data are imputed using mean observed values from the training data. Figure 5 confirms the stability at testing time of the good performance of **Linear RR** and **MLP RR**.

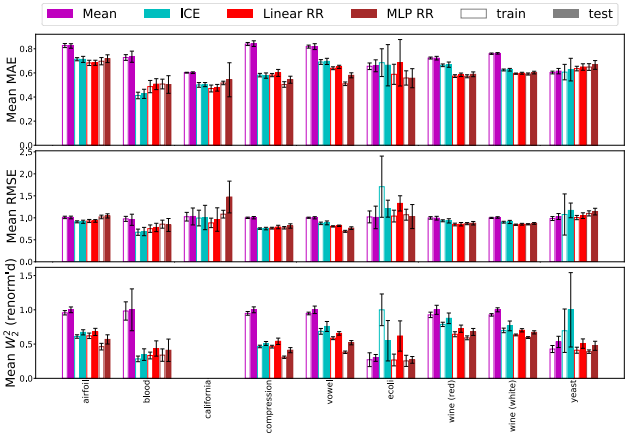

Figure 5: (**OOS**) Out of sample imputation: 70% of the data is used for training (filled bars) and 30 % for testing with fixed parameters (dotted bars). 30% of the values are missing MCAR accross both training and testing sets.

Conclusion

We have shown in this paper how OT metrics could be used to define a relevant loss for missing data imputation. This loss corresponds to the expectation of Sinkhorn divergences between randomly sampled batches. To minimize it, two classes of algorithms were proposed: one that freely estimates one parameter per imputed value, and one that fits a parametric model. The former class does not rely on making parametric assumptions on the underlying data distribution, and can be used in a very wide range of settings. On the other hand, after training, the latter class allows out-of-sample imputation. To make parametric models trainable, the classical round-robin mechanism was used. Experiments on a variety of datasets, and for numerous missing value settings (MCAR, MAR and MNAR with varying missing values proportions) showed that the proposed models are very competitive, even compared to recent methods based on deep learning. These results confirmed that our loss is a good optimizable proxy for imputation metrics. Future work includes further theoretical study of our loss function Eq. (5) within the OT framework.

References

Arjovsky, M., Chintala, S., and Bottou, L. Wasserstein generative adversarial networks. In Precup, D. and Teh, Y. W. (eds.), *Proceedings of the 34th International Conference on Machine Learning*, pp. 214–223, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR.

Burda, Y., Grosse, R., and Salakhutdinov, R. Importance weighted autoencoders. *International Conference on Learning Representations*, 2016.

Cuturi, M. Sinkhorn distances: Lightspeed computation of optimal transport. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, pp. 2292–2300, Red Hook, NY, USA, 2013. Curran Associates Inc.

Cuturi, M. and Doucet, A. Fast computation of Wasserstein barycenters. In Xing, E. P. and Jebara, T. (eds.), *Proceedings of the 31st International Conference on Machine Learning*, number 2, pp. 685–693, Bejing, China, 22–24 Jun 2014. PMLR.

Dempster, A. P., Laird, N. M., and Rubin, D. B. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pp. 1–38, 1977.

Fatras, K., Zine, Y., Flamary, R., Gribonval, R., and Courty, N. Learning with minibatch Wasserstein : asymptotic and gradient properties. *CoRR*, abs/1910.04091, 2019.

Feydy, J., Séjourné, T., Vialard, F., Amari, S., Trouvé, A., and Peyré, G. Interpolating between optimal transport and MMD using Sinkhorn divergences. In *The 22nd International Conference on Artificial Intelligence and Statistics, AISTATS 2019, 16-18 April 2019, Naha, Okinawa, Japan*, pp. 2681–2690, 2019.

Frogner, C., Zhang, C., Mobahi, H., Araya-Polo, M., and Poggio, T. Learning with a Wasserstein loss. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, pp. 2053–2061, Cambridge, MA, USA, 2015. MIT Press.

Genevay, A., Peyre, G., and Cuturi, M. Learning generative models with Sinkhorn divergences. In Storkey, A. and Perez-Cruz, F. (eds.), *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, pp. 1608–1617, Playa Blanca, Lanzarote, Canary Islands, 09–11 Apr 2018. PMLR.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In *Advances in neural information processing systems*, pp. 2672–2680, 2014.

Hastie, T., Mazumder, R., Lee, J. D., and Zadeh, R. Matrix completion and low-rank svd via fast alternating least squares. *J. Mach. Learn. Res.*, 16(1):3367–3402, January 2015. ISSN 1532-4435.

Ivanov, O., Figurnov, M., and Vetrov, D. Variational autoencoder with arbitrary conditioning. *International Conference on Learning Representations*, 2019.

Josse, J., Husson, F., et al. missmda: a package for handling missing values in multivariate data analysis. *Journal of Statistical Software*, 70(1):1–31, 2016.

Kim, J. and Ying, Z. *Data Missing Not at Random: Jae-Kwang Kim, Zhiliang Ying Editors for this Special Issue*. Statistica Sinica. Institute of Statistical Science, Academia Sinica, 2018.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. *International Conference on Learning Representations*, 2014.

Little, R. J. A. and Rubin, D. B. *Statistical analysis with missing data*. John Wiley & Sons, 2019.

Mattei, P.-A. and Frellsen, J. MIWAE: Deep generative modelling and imputation of incomplete data sets. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning*, pp. 4413–4423, Long Beach, California, USA, 09–15 Jun 2019. PMLR.

Mayer, I., Josse, J., Tierney, N., and Vialaneix, N. R-miss-tastic: a unified platform for missing values methods and workflows. *arXiv preprint arXiv:1908.04822*, 2019.

Mohan, K. and Pearl, J. Graphical Models for Processing Missing Data. *Journal of American Statistical Association (JASA)*, 2019.

Murray, J. S. and Reiter, J. P. Multiple imputation of missing categorical and continuous values via bayesian mixture models with local dependence. *Journal of the American Statistical Association*, 111(516):1466–1479, 2016.

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

Peyré, G., Cuturi, M., et al. Computational optimal transport. *Foundations and Trends® in Machine Learning*, 11(5-6): 355–607, 2019.

Rubin, D. B. *Biometrika*, 63(3):581–592, 1976.

Salimans, T., Zhang, H., Radford, A., and Metaxas, D. Improving gans using optimal transport. *arXiv preprint arXiv:1803.05573*, 2018.

Schiebinger, G., Shu, J., Tabaka, M., Cleary, B., Subramanian, V., Solomon, A., Gould, J., Liu, S., Lin, S., Berube, P., et al. Optimal-transport analysis of single-cell gene expression identifies developmental trajectories in reprogramming. *Cell*, 2019.

Seaman, S., Galati, J., Jackson, D., and Carlin, J. What is meant by" missing at random"? *Statistical Science*, pp. 257–268, 2013.

Stekhoven, D. J. and Buhlmann, P. MissForest a nonparametric missing value imputation for mixed-type data. *Bioinformatics*, 28(1):112–118, 10 2011. ISSN 1367-4803. doi: 10.1093/bioinformatics/btr597.

Tieleman, T. and Hinton, G. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 4 (2):26–31, 2012.

Troyanskaya, O., Cantor, M., Sherlock, G., Brown, P., Hastie, T., Tibshirani, R., Botstein, D., and Altman, R. B. Missing value estimation methods for dna microarrays. *Bioinformatics*, 17(6):520–525, 2001.

Udell, M. and Townsend, A. Why are big data matrices approximately low rank? *SIAM Journal on Mathematics of Data Science*, 1(1):144–160, 2019.

van Buuren, S. *Flexible Imputation of Missing Data*. Chapman and Hall/CRC, Boca Raton, FL, 2018.

van Buuren, S. and Groothuis-Oudshoorn, K. mice: Multivariate imputation by chained equations in r. *Journal of Statistical Software, Articles*, 45(3):1–67, 2011. ISSN 1548-7660.

Yoon, J., Jordon, J., and van der Schaar, M. GAIN: Missing data imputation using generative adversarial nets. In Dy, J. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning*, pp. 5689–5698, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR.

Zhu, Z., Wang, T., and Samworth, R. J. High-dimensional principal component analysis with heterogeneous missingness. *arXiv preprint arXiv:1906.12125*, 2019.