


Iterative Imputation of Missing Data using Auto-encoder Dynamics^{*}

Marek Śmieja¹^[0000-0003-2027-4132], Maciej Kołomycki²^[0000-0003-4202-7693],
Łukasz Struski¹^[0000-0003-4006-356X], Mateusz Juda¹^[0000-0003-2023-8430], and
Mário A. T. Figueiredo³^[0000-0002-0970-7745]

¹ Faculty of Mathematics and Computer Science, Jagiellonian University,
Kraków, Poland

`{marek.smieja, lukasz.struski, mateusz.juda}@uj.edu.pl`

² Institute of Applied Informatics, Faculty of Mechanical Engineering, Cracow
University of Technology, Kraków, Poland

`maciej.kolomycki@mech.pk.edu.pl`

³ Instituto de Telecomunicações, Instituto Superior Técnico,
Universidade de Lisboa, Portugal

`mario.figueiredo@tecnico.ulisboa.pt`

Abstract. This paper introduces an approach to missing data imputation based on deep auto-encoder models, adequate to high-dimensional data exhibiting complex dependencies, such as images. The method exploits the properties of the vector field associated to an auto-encoder, which allows to approximate the gradient of the log-density from its reconstruction error, based on which we propose a projected gradient ascent algorithm to obtain the conditionally most probable estimate of the missing values. Our approach does not require any specialized training procedure and can be used together with any auto-encoder model trained on complete data in a classical way. Experiments performed on benchmark datasets show that imputations produced by our model are sharp and realistic.

Keywords: missing data imputation · image inpainting · auto-encoder · dynamical system · auto-encoder’s vector field.

1 Introduction

Missing data imputation is an important problem in machine learning and data analysis, especially when dealing with real-world applications [5, 10, 17]. The typical approach is to directly design a specialized model and train it to fill in absent values. By constructing sophisticated architectures, trained under carefully designed loss functions, state-of-the-art models obtain impressive performance, *e.g.*, in image inpainting [11, 32]. However, a natural question arises: *can*

^{*} The is the extended version of an extended abstract [25] presented at the ICLR Workshop on the Integration of Deep Neural Models and Differential Equations.

we complete missing data at test time using models that were not aware of the imputation task during the training stage?

Our work is motivated by the use of classical parametric (or semi-parametric) density models, such as *Gaussian mixture models* (GMMs) [27], for missing data imputation. In that work, a density is estimated from complete data⁴ in a strictly unsupervised way. To apply the model for missing data imputation, the missing values are replaced either by samples or by maximizers of the estimated conditional density of the missing data, given the observed data. Although the use of a shallow density model, such as a GMM, may allow obtaining the conditional density analytically, such a model may be unable to efficiently capture complex dependencies in high-dimensional data, such as images [24].

While deep generative models, *e.g.*, *generative adversarial networks* (GAN) [8], *variational auto-encoders* (VAE) [12], or *Wasserstein auto-encoders* (WAE) [28], are sufficiently expressive to describe complex dependencies in data, it may be impossible to explicitly obtain or maximize the corresponding conditional density of the missing values due to the nonlinear form of decoder (generator) [20]. The authors of [22] define a pseudo-Gibbs sampling procedure for filling missing values by iterative auto-encoding of incomplete data (see also [9, Section 20.11] for more general formulation). In the case of VAE, this procedure can be modified by adding an option to reject the proposal posterior distribution, which results in Metropolis-within-Gibbs algorithm [18]. Mattei and Frellsen use importance sampling for training VAE on incomplete data as well as for replacing missing values by single or multiple imputation [19]. Similarly, it is challenging to obtain a closed-form expression for such a conditional distribution in GAN, but one can design a procedure to sample from it [15, 31].

We tackle this problem by exploiting the dynamics of auto-encoders' reconstruction function. Based on theoretical results presented in [1], the reconstruction error of a *denoising auto-encoder* (DAE) [29] yields an approximation of the gradient of the log-probability density function, which is (implicitly) estimated from data. We exploit that fact to maximize the conditional density of the missing values, given the observed ones. The conditionally most probable values are found as the attractors of the iterated reconstruction function. We experimentally demonstrate that, in a place of DAE, any auto-encoder model (*e.g.*, AE, VAE, WAE) can be used in the process of replacing missing data at test time without any additional effort at training stage.

Alternatively, our procedure can be interpreted as a type of pseudo-Gibbs sampling. While the pseudo-Gibbs sampling procedure proposed by [22] directly replaces the input by its reconstruction, which may lead to falling out of the true data manifold, our algorithm adds the reconstruction error to the input with a small weight. Similarly to [18], it improves convergence of the algorithm when the posterior approximation is imperfect. Our procedure is also similar to the algorithm proposed in [6, Section 5.2] for NICE flow model, where the gradient

⁴ A GMM can also be learned from incomplete data, but the imputation process does not change.

is given explicitly. Our procedure works for every possible auto-encoder model, even if the gradient is difficult to compute.

We experimentally assess the proposed approach on image datasets, showing that it obtains results comparable to typical deep learning models (with analogous neural network architecture) trained explicitly for missing data imputation. Moreover, by using different initializations in the iterative procedure, we can reach different attractors and, consequently, a diverse set of imputation candidates for the same incomplete input. This makes our model similar to generative models.

The paper is organized as follows. In section 2, we recall known facts concerning AE's vector field and dynamics. Our approach is introduced in Section 3 and, next, experimentally assessed in Section 4.

2 Auto-encoder Dynamics

Because they underlie our approach to missing data imputation, this section reviews relevant facts regarding the vector field associated with an auto-encoder reconstruction function and the associated dynamics.

Auto-encoders (AE) have a long history in the field of artificial neural networks, going back at least to the 1980s [7, 13]. An AE may be viewed as composition of two maps, an *encoder* $f : \mathbb{R}^d \rightarrow Z$ and a *decoder* $g : Z \rightarrow \mathbb{R}^d$, such that $Z \subset \mathbb{R}^l$ is the so-called *latent space*. An AE is trained from data with the goal of making the *reconstruction* function $r := g \circ f$ close to identity, *i.e.*, $r(x) \approx x$, for the training data, by capturing the essential features of that data.

Since an AE does not (and should not) achieve perfect reconstructions (specially for input data far from the training data), we can define an *AE vector field* $v : \mathbb{R}^d \rightarrow \mathbb{R}^d$ associated to the reconstruction function as $v(x) := r(x) - x$. Analogously, we also define an *AE latent vector field* $u : Z \rightarrow Z$, given by $u(z) := f(g(z)) - z$. A natural question arises: what is the structure of the dynamics generated by the vector fields v and u ?

The properties of the vector fields v for a DAE were studied and discussed in [1], where it was shown that the reconstruction error at some point $x \in \mathbb{R}^d$ is approximately equal to the gradient of the log-pdf (*logarithm of the probability density function*) computed at that point, in the low-noise limit, *i.e.*, as $\sigma^2 \rightarrow 0$,

$$\nabla_x \log p_X(x) \approx \frac{r_{\sigma^2}(x) - x}{\sigma^2} = \frac{v_{\sigma^2}(x)}{\sigma^2}, \quad (1)$$

where r_{σ^2} is the reconstruction function of the DAE at denoising level σ^2 and v_{σ^2} is the corresponding vector field. Consequently, the point with the highest log-pdf can be found via gradient ascent, *i.e.*, gradient flow, in the limit of infinitesimal steps, by exploiting this equality. In discrete time, with a step-size of the order of σ^2 , we thus have: $x_{t+1} = r_{\sigma^2}(x_t)$. Notice that, from Equation 1, it is clear that fixed points of this iteration correspond to stationary points of the log-pdf, *i.e.*, zeros of its gradient.

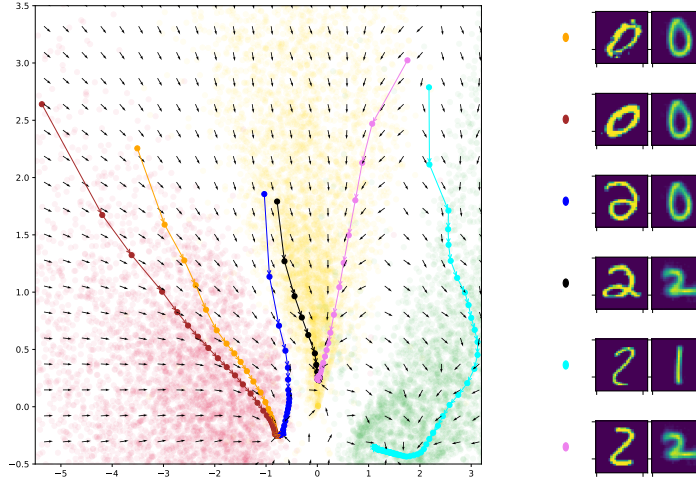


Fig. 1: Example of latent space trajectories (left) for an AE trained on the MNIST dataset (classes 0, 1, 2). Dots represent latent representation of the training examples: 0-red, 1-green, 2-yellow. On the right hand side, for each trajectory, we present its starting point and the attractor reached after 100 iterations.

Analyzing the dynamics resulting from the vector field associated with the reconstruction error may be useful in verifying the quality of an AE. The intuition is that this dynamics (and its counterpart in the latent space) should have stationary points, some of which are locally stable, thus are attractors. The basin of attraction of each attractor should consist of a subset of the input space with points with similar features.

As an example, consider an AE trained on digits 0, 1, and 2 of the MNIST dataset, using latent space dimension $l = 2$. Starting from some latent point $z_0 \in Z$, we draw the latent trajectory generated by the iteration $z_{t+1} := f(g(z_t))$, which is the discrete-time counterpart of the gradient flow explained above. In most cases, we observe the behavior shown in Figure 1: each trajectory travels through the latent space and converges to a fixed point (attractor). For some starting points, a small perturbation may cause the trajectory to converge to different attractor. In Figure 1, we observe such behaviour for the cyan and pink trajectories; their starting points lie close to the boundary between classes 1 and 2. It is also a low density region, so in some sense the AE is not trained enough there. However, in the case of the blue and black trajectories, we see such behaviour also for starting points in the relatively denser area of class 2.

3 Imputation Method

In this section, we show how to use the discrete-time dynamics above described in the context of missing data to obtain imputations with the highest local prob-

ability. A point $x \in \mathbb{R}^d$ with missing components is denoted by a pair (x, J) , where $J \subset \{1, \dots, d\}$ is the set of indices with missing values. For a fully-observed point, $J = \emptyset$. The key question in missing-data imputation is: what is the “best” choice for filling the missing coordinates x_J (restriction of x to unobserved components)? We follow a classical probabilistic approach by choosing the maximizer of the corresponding conditional pdf, given the observed variables $x_{\bar{J}}$, where $\bar{J} = \{1, \dots, d\} \setminus J$ is the set of indices of the observed components of x .

To make the above statement more precise, let p_X be a pdf defined on \mathbb{R}^d . Given a data point with missing components (x, J) , assume that $J \neq \emptyset$, otherwise imputation is unnecessary, and $J \neq \{1, \dots, d\}$, otherwise we do not have an imputation problem. The conditional pdf is given by Bayes law,

$$p(x_J | x_{\bar{J}}) = \frac{p(x_J, x_{\bar{J}})}{p(x_{\bar{J}})} = \frac{p_X(x)}{p(x_{\bar{J}})}, \quad (2)$$

because $x_{J \cup \bar{J}} = x \in \mathbb{R}^d$ (missing and observed). Since we are looking for the maximizer of this conditional pdf, the denominator is irrelevant, thus

$$\hat{x}_J = \arg \max_{x_J \in \mathbb{R}^{|J|}} p(x_J | x_{\bar{J}}) = \arg \max_{y \in \mathbb{R}^d: y_J = x_J} \log p_X(y). \quad (3)$$

To seek the maximizer of the conditional density defined in Equation 3, we propose the following procedure (which we show below corresponds to a projected gradient ascent scheme), based on an AE trained on a dataset with characteristics similar to the data on which imputation will be performed:

1. pick an initial filling \hat{x}_J^0 of the missing part x_J ;
2. iteratively update \hat{x}_J using $\hat{x}_J^{t+1} = \hat{x}_J^t + h [r_\sigma(\hat{x}^t) - \hat{x}^t]_J$.

where h is a step size and $\hat{x}^t = (\hat{x}_J^t, x_{J'}) \in \mathbb{R}^d$ denotes a complete point where the observed components are fixed at the observed values and the missing ones are replaced by the current estimate. This procedure corresponds to moving on an (axes-aligned) affine subspace of dimension $\mathbb{R}^{|J|}$ of the data space \mathbb{R}^d in a direction determined by the gradient of the log-density function (see Equation (1)). Because of the axes-aligned nature of the affine subspace, this coincides with a projected gradient ascent algorithm.

As shown in the Figure 2, the proposed method depends on the initialization \hat{x}_J^0 . We observe that, for the smallest missing window, all initializations lead to the same attractor, thus the same imputations. For mid-sized missing regions, our algorithm with random initialization gives different effect from the one obtained using mean and k-NN filling. For the largest hole, we loose the image features and land in the area of a different class regardless on the initialization.

In practice, we can control the final result by careful selecting the starting point. To make the final imputation the most similar to ground truth, we should pick an initialization using simple imputations, *e.g.*, mean or k-NN. To provide more diverse results, we can use add random noise or samples from some prior distribution for the initialization. Consequently, our method has a generative nature and is capable of creating a wide range of imputations depending on the seed (see next section for more results).

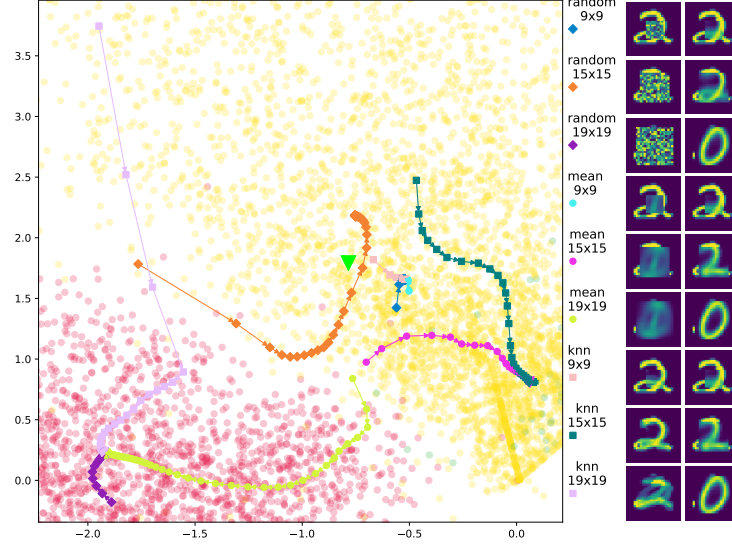


Fig. 2: Illustration of our algorithm (using the same AE as in Figure 1), for missing regions of three different sizes (9×9 , 15×15 , and 19×19) on the same data point from class “2” (green point on the left hand side). Three initialization strategies were used: random noise, mean value over the training set, mean value over 5-NN. We show trajectories in latent space (left) and final imputations (rightmost) for different initializations (second column from the right).

4 Experiments

In this section, we experimentally validate the proposed model. For this purpose, we fit a typical AE on a train set and use it for filling in missing data at test time⁵. To examine the dependence on the initialization, we consider two variants of our model: starting with random noise as initial imputation; initial filling generated using k-NN imputation. We adapted the architecture and the training procedure from [28] (using $\lambda = 0$ to obtain a classical AE).

As a baseline, we apply a pseudo Gibbs sampling (p-Gibbs) [22], where decoded data is directly used as an input to the next iteration. Similarly to our method, we use two variants of initialization: random replacement and k-NN imputation. We also consider a *context encoder* (CE) [21], which is a type of deep AE trained explicitly for filling in missing data. Roughly, a CE takes an

⁵ For a comparison between different auto-encoder models in the proposed procedure the reader is referred to our workshop paper [25].

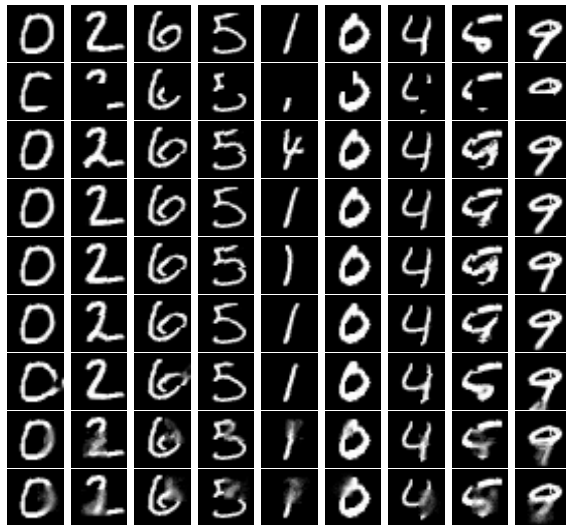


Fig. 3: Reconstructions of incomplete MNIST images. Rows: (1) original image, (2) incomplete image, and imputations using (3) our methods with random initialization, (4) our method with k-NN initialization, (5) p-Gibbs with random initialization, (6) p-Gibbs with k-NN (7) CE (8) k-NN, (9) MICE.

incomplete image (with a mask) and focuses on making the output as similar to the original image as possible by minimizing the MSE on the missing area. To make both approaches (p-Gibbs and CE) fully comparable with our method, we use exactly the same architecture for all models. Additionally, we use two typical imputation methods: (a) k-NN [3], which fills missing values with the corresponding mean values computed from the k nearest training samples (we used $k = 5$); (b) MICE [2, 4], where several imputations are drawn from the conditional pdf using *Markov chain Monte Carlo* sampling. Rather than presenting state-of-the-art performance, which requires more advanced neural network architecture (and modification of training procedure in our case), we focus on showing that our test procedure obtains similar performance to the typical models trained explicitly for the imputation task.

We consider two datasets of gray-scale images: MNIST [14] and Fashion [30]. For each test image of the size 28×28 , we drop a patch of size 13×13 , at a (uniformly) random location. We also use the CelebA dataset [16], which is composed of color face images of size 64×64 , with missing regions of size 25×25 . Analogous missing regions are used for training the CE and MICE.

Figures 3 and 4 present sample results for MNIST and Fashion, respectively. One can observe that the results produced by our method and p-Gibbs with k-NN initialization are visually the most plausible and usually coincide with ground truth. The results obtained by our method with random initialization are also realistic, but differ in some cases from the ground-truth and from p-Gibbs with

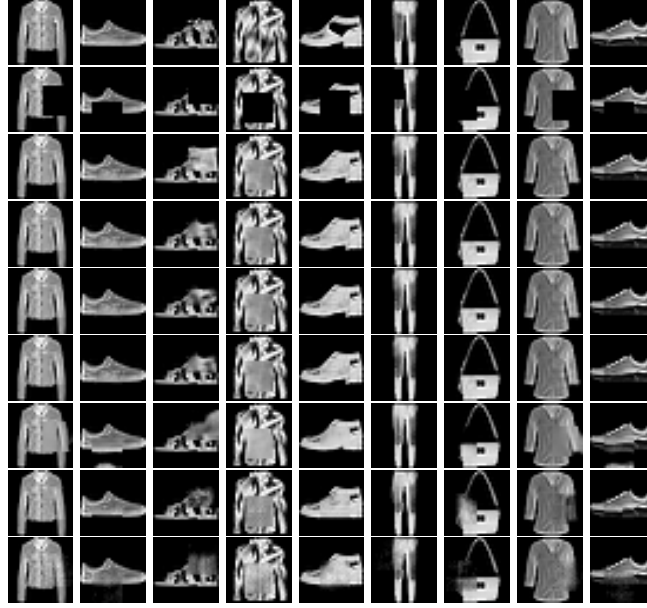


Fig. 4: Reconstructions of incomplete images from Fashion. Each row presents subsequent methods: (1) original image, (2) incomplete image, and imputations using (3) our methods with random initialization, (4) our method with k-NN initialization, (5) p-Gibbs with random initialization, (6) p-Gibbs with k-NN (7) CE (8) k-NN, (9) MICE.

the analogical initialization (2nd and 5th column for MNIST presents positive effect while 3rd column for Fashion illustrates negative results).

Since a CE is trained to fill in missing data by minimizing the MSE, its results usually coincide with the ground-truth average, but many details are missing (see 7th column for Fashion, where this method failed to complete the handbag strap). In contrast, our method aims at finding the most probable replacement, usually yielding sharp images, although maybe different from the ground truth. In the case of the more diverse Fashion dataset, the CE produces a lot of artifacts.

While k-NN presents poor performance on MNIST, its results on Fashion are appealing. Since Fashion contains many similar images, k-NN is able to fill in missing regions with analogous shapes. It is evident that MICE fails to complete missing data with reasonable content.

The results for CelebA dataset are presented in Figure 5. It is clear that using k-NN initialization in our procedure leads to more plausible results than random initialization. As seen in the 1st column, random seed directed a filling trajectory out of true data distribution. On the other hand, the same initialization in the 6th column created forehead bangs, which may be seen as a positive effect. The results produced by p-Gibbs occasionally differ from the ones returned by our

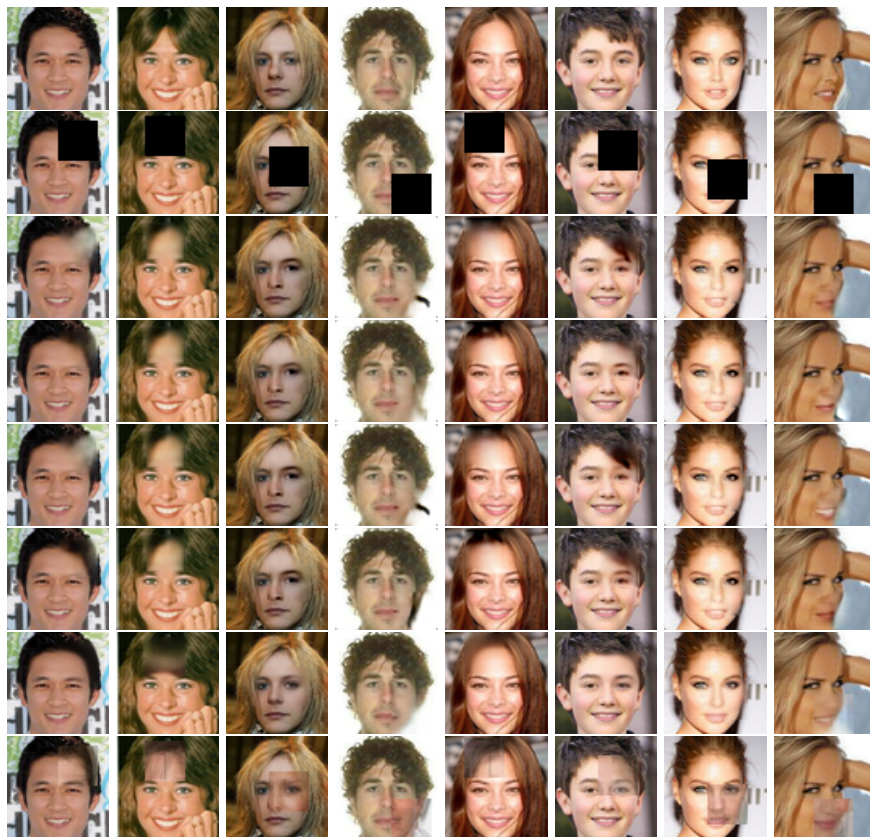


Fig. 5: Reconstructions of partially incomplete images from CelebA. Each row presents subsequent methods: (1) original image, (2) incomplete image, and imputations using (3) our methods with random initialization, (4) our method with k-NN, (5) p-Gibbs with random initialization, (6) p-Gibbs with k-NN (7) CE (8) k-NN.

method (see 9th column for random initialization as well as 4th and 7th column for k-NN). At first glance, the imputations produced by the CE are visually plausible. However, more detailed inspection reveals that the obtained images are often blurry (5th and 8th columns) and sometimes contain artifacts (2nd column). The use of k-NN imputation alone gives bad results. We were unable to run MICE imputation due to high-dimensionality of data.

An interesting aspect is that our method is more “creative” than the others. Varying the initialization, our method can create different styles of the same objects (2nd column for MNIST), examples from different classes (5th column for MNIST) and other data manipulations (longer hairs - 6th column, closed mouths - 8th column for CelebA). This property can be very appealing from a generative perspective, which cannot be easily obtained using typical approaches.

Table 1: SSIM of imputations.

Method	MNIST	Fashion	CelebA
Ours (random init.)	0.830	0.903	0.901
Ours (k-NN init.)	0.875	0.919	0.914
p-Gibbs (random init.)	0.828	0.904	0.899
p-Gibbs (k-NN init.)	0.868	0.918	0.907
CE	0.871	0.879	0.930
k-NN	0.829	0.841	0.857
MICE	0.797	0.809	-

To provide a quantitative assessment of the methods, we measure their *structural similarity* (SSIM) with ground truth. Unlike PSNR or MSE [23], which measure pixel-wise absolute errors, SSIM is based on visible structures in the image. SSIM is calculated for various windows of input images and, for the pixel p , it is defined by [26]:

$$SSIM(p) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} \cdot \frac{2\sigma_{xy} + C_2}{\sigma_x^2 + \sigma_y^2 + C_2},$$

where μ_x, μ_y are the mean values of the patches x and y , respectively, centered at p ; σ_x^2, σ_y^2 denote variances; σ_{xy}^2 denotes covariance; C_1 and C_2 are variables intended to stabilize the division with small denominator; its maximal value 1 is attained for identical images.

The results presented in Table 1 show that our method with k-NN initialization gives the highest resemblance with ground-truth on MNIST and Fashion. While the performance of CE on MNIST is only slightly worse, the disproportion between these methods on Fashion is evident. In the case of CelebA, CE is more accurate than our method, but the difference is not high. It can be observed that p-Gibbs performs slightly worse than our method. The disproportion between shallow (k-NN, MICE) and deep methods (AE, CE) is enormous.

5 Conclusion and future work

In this paper, we proposed a strategy for filling in missing values based on auto-encoder vector field. Our method does not require a training procedure designed for imputation task, but can be used together with any AE trained in a typical way. The idea is to traverse the AE vector field towards an attractor, which is a local maximum of the probability density of function learned by the AE. Experiments showed that this procedure gives comparable results to typical deep models trained explicitly for imputation tasks.

To increase the performance of the proposed procedure, we plan to modify the training procedure of the AE. One option is to simulate the iterative procedure in the training phase, by reconstructing original images from partial imputations. This should stabilize the test stage and prevent from falling out of the true data distribution when initialized from random noise.

Acknowledgements

The work of M. Śmieja was supported by the National Science Centre (Poland) grant no. 2018/31/B/ST6/00993. The work of Ł. Struski was supported by the National Science Centre (Poland) grant no. 2017/25/B/ST6/01271 as well as the Foundation for Polish Science Grant No. POIR.04.04.00-00-14DE/18-00 co-financed by the European Union under the European Regional Development Fund. The work of M. Juda was supported by the National Science Centre (Poland) grant no. 2014/14/A/ST1/00453 and 2015/19/D/ST6/01215.

References

1. Alain, G., Bengio, Y.: What regularized auto-encoders learn from the data-generating distribution. *Journal of Machine Learning Research* **15**, 3563–3593 (2014)
2. Azur, M., Stuart, E., Frangakis, C., Leaf, P.: Multiple imputation by chained equations: what is it and how does it work? *International journal of Methods in Psychiatric Research* **20**, 40–49 (2011)
3. Batista, G., Monard, M.: A study of k-nearest neighbour as an imputation method. *Frontiers in Artificial Intelligence and Applications* **97**, 251–260 (2002)
4. Buuren, S., Groothuis-Oudshoorn, K.: Mice: Multivariate imputation by chained equations in r. *Journal of Statistical Software* pp. 1–68 (2010)
5. Camino, R., Hammerschmidt, C., State, R.: Improving missing data imputation with deep generative models. *arXiv preprint arXiv:1902.10666* (2019)
6. Dinh, L., Krueger, D., Bengio, Y.: Nice: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516* (2014)
7. Gallinari, P., LeCun, Y., Thiria, S., Fogelman-Soulie, F.: *Memoires associatives distribuees*. In: *COGNITIVA 87*. Paris (1987)
8. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: *Advances in Neural Information Processing Systems*. pp. 2672–2680 (2014)
9. Goodfellow, I., Bengio, Y., Courville, A.: *Deep learning*. MIT press (2016)
10. Hwang, U., Jung, D., Yoon, S.: Hexagan: Generative adversarial nets for real world classification. *arXiv preprint arXiv:1902.09913* (2019)
11. Iizuka, S., Simo-Serra, E., Ishikawa, H.: Globally and locally consistent image completion. *ACM Transactions on Graphics (ToG)* **36**(4), 1–14 (2017)
12. Kingma, D., Welling, M.: Auto-encoding variational Bayes. In: *International Conference on Learning Representations* (2014)
13. LeCun, Y.: *Modeles connexionistes de l'apprentissage*. Ph.D. thesis, PhD Thesis, Université de Paris VI (1987)
14. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. *Proceedings of the IEEE* **86**, 2278–2324 (1998)
15. Li, S., Jiang, B., Marlin, B.: MisGAN: Learning from incomplete data with generative adversarial networks. *arXiv preprint arXiv:1902.09599* (2019)
16. Liu, Z., Luo, P., Wang, X., Tang, X.: Deep learning face attributes in the wild. In: *International Conference on Computer Vision* (2015)
17. Luo, Y., Cai, X., Zhang, Y., Xu, J., Xiaojie, Y.: Multivariate time series imputation with generative adversarial networks. In: *Advances in Neural Information Processing Systems*. pp. 1596–1607 (2018)

18. Mattei, P.A., Frellsen, J.: Leveraging the exact likelihood of deep latent variable models. In: *Advances in Neural Information Processing Systems*. pp. 3855–3866 (2018)
19. Mattei, P.A., Frellsen, J.: Miwae: Deep generative modelling and imputation of incomplete data sets. In: *International Conference on Machine Learning*. pp. 4413–4423 (2019)
20. Nazabal, A., Olmos, P.M., Ghahramani, Z., Valera, I.: Handling incomplete heterogeneous data using vaes. *Pattern Recognition* p. 107501 (2020)
21. Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., Efros, A.: Context encoders: Feature learning by inpainting. In: *IEEE Conference on Computer Vision and Pattern Recognition*. pp. 2536–2544 (2016)
22. Rezende, D.J., Mohamed, S., Wierstra, D.: Stochastic backpropagation and approximate inference in deep generative models. *arXiv preprint arXiv:1401.4082* (2014)
23. Sai Hareesh, A., Chandrasekaran, V.: A novel color image inpainting guided by structural similarity index measure and improved color angular radial transform. In: *International Conference on Image Processing, Computer Vision, & Pattern Recognition*. pp. 544–550 (2010)
24. Śmieja, M., Struski, L., Tabor, J., Zieliński, B., Spurek, P.: Processing of missing data by neural networks. In: *Advances in Neural Information Processing Systems*. pp. 2719–2729 (2018)
25. Śmieja, M., Kołomycki, M., Struski, L., Juda, M., Figueiredo, M.A.T.: Can auto-encoders help with filling missing data? In: *ICLR Workshop on Integration of Deep Neural Models and Differential Equations (DeepDiffEq)*. p. 6 (2020)
26. Stagakis, N., Zacharaki, E.I., Moustakas, K.: Hierarchical image inpainting by a deep context encoder exploiting structural similarity and saliency criteria. In: *International Conference on Computer Vision Systems*. pp. 470–479. Springer (2019)
27. Titterton, D., Sedransk, J.: Imputation of missing values using density estimation. *Statistics & Probability Letters* **9**(5), 411–418 (1989)
28. Tolstikhin, I., Bousquet, O., Gelly, S., Schölkopf, B.: Wasserstein auto-encoders (2017), *arXiv:1711.01558*
29. Vincent, P.: A connection between score matching and denoising autoencoders. *Neural Computation* **23**(7) (2011)
30. Xiao, H., Rasul, K., Vollgraf, R.: Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747* (2017)
31. Yoon, J., Jordon, J., Van Der Schaar, M.: Gain: Missing data imputation using generative adversarial nets. *arXiv preprint arXiv:1806.02920* (2018)
32. Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., Huang, T.S.: Generative image inpainting with contextual attention. In: *IEEE Conference on Computer Vision and Pattern Recognition*. pp. 5505–5514 (2018)