Supervised learning with incomplete data via sparse representations

Anonymous authors
Paper under double-blind review

Abstract

This paper addresses the problem of training a classifier on incomplete data and its application to a complete or incomplete test dataset. A supervised learning method is developed to train a general classifier, such as a logistic regression or a deep neural network, using only a limited number of observed entries, assuming sparse representations of data vectors on an unknown dictionary. The proposed method simultaneously learns the classifier, the dictionary and the corresponding sparse representations of each input data sample. A theoretical analysis is also provided comparing this method with the standard imputation approach, which consists on performing data completion followed by training the classifier based on their reconstructions. The limitations of this last "sequential" approach are identified, and a description of how the proposed new "simultaneous" method can overcome the problem of indiscernible observations is provided. Additionally, it is shown that, if it is possible to train a classifier on incomplete observations so that its reconstructions are well separated by a hyperplane, then the same classifier also correctly separates the original (unobserved) data samples. Extensive simulation results are presented on synthetic and well-known reference datasets that demonstrate the effectiveness of the proposed method compared to traditional data imputation methods.

1 Introduction

In many machine learning applications sometimes the measurements are noisy or affected by artifacts, resulting in incomplete data samples. Examples of this situation include: images captured in a self-driving vehicle or robot where objects in the view field can be partially occluded; recommendation systems built from the information gathered by different users where not all the users have fully completed their forms; or medical datasets where typically not all tests can be performed on all patients.

A classical approach of supervised learning with missing entries is to apply entry imputation as a preprocessing step, followed by standard training based on these reconstructions (Little & Rubin, 2014). This "sequential approach" was adopted by several researchers in the literature. For example, some techniques compute missing entries using statistical methods, such as the "mean", "regression" and "multiple" imputation methods (Little & Rubin, 2014). Other completion methods are based on machine learning ideas by estimating missing entries through K-nearest neighbor (Batista & Monard, 2002), Self Organization Maps (SOM) (Fessant & Midenet, 2002), multilayer or recurrent neural networks (Yoon & Lee, 1999; Bengio & Gingras, 1995), tensor completion algorithms (Solé-Casals et al., 2018) and others (García-Laencina et al., 2009). In (Huang et al., 2018), a technique for completing the matrix of features was proposed by imposing low-rankness and incorporating the label information into the completion process. A different approach is to

avoid direct imputation of lost inputs and rely on a probabilistic model of input data, learning model parameters through the Expectation Maximization (EM) algorithm and building a Bayesian classification (Ghahramani & Jordan, 1993; Little & Rubin, 2014). However, the latter "model-based" approach has the disadvantage that it requires a good probabilistic data model, which is usually not available, especially for real-world applications such as those involving natural images.

On the other hand, during the last few years in the signal processing community, there has been a rapid development of the Compressed Sensing (CS) theory and sparse coding algorithms that exploit the redundancy of natural signals (Candes et al., 2006; Donoho, 2006; Eldar & Kutyniok, 2012; Gregor & LeCun, 2010). A particular case of CS is the signal completion problem, i.e. the reconstruction of signals from incomplete measurements. Several algorithms for matrix and tensor (multidimensional signals) completion have been recently proposed relying on different low-dimensional manifold models that are fit to the available entries, allowing for an extrapolation of the unobserved entries. This category of models ranges from sparse representation of vectors (Aharon et al., 2006; Fadili et al., 2008) and tensors (Caiafa & Cichocki, 2012) to low-rank matrix (Candes & Tao, 2010) and tensor decompositions (Liu et al., 2012; Yuan et al., 2018). We would also note that the success of sparse representations in many signal processing applications motivated researchers to use dictionary learning techniques for data classification (Mairal et al., 2008), either using separate sub-dictionaries for each class (Ramirez et al., 2010; Sprechmann & Sapiro, 2010), or using a single dictionary for all classes (Tosic & Frossard, 2011).

In this paper, the goal is to develop a method for training a classifier using incomplete data samples with their labels and to identify the conditions under which such a classifier performs as good as the ideal classifier, i.e. the one that could be obtained from complete observations. We propose to use the sparse representation model for data vectors, and analyze the limitations of the sequential approach, i.e. imputation followed by training (section 2). In section 3 we introduce the simultaneous classification and coding approach, which consists of incorporating a sparse data representation model into a cost function optimized for training the classifier and, at the same time, finding the best sparse representation of the observed data. We provide theoretical conditions under which we can guarantee that the obtained classifier is as good as the ideal classifier. In section 4, a computational method is presented to train a classifier on incomplete data and to test it on complete or incomplete data; in section 5, extensive experimental results are presented using synthetic and well known benchmark datasets illustrating the effectiveness of our proposed algorithms; and finally, in section 6, the main conclusions are outlined.

1.1 Preliminary definitions and problem formulation

We assume a supervised learning scenario with vector samples and their labels $\{\mathbf{x}_i, y_i\}$, $i = 1, 2, \ldots I$, $\mathbf{x}_i \in \mathbf{R}^N$ and $y_i \in \{0, 1, \ldots, C-1\}$ (C classes), so that we only have access to subsets of entries in each data vector along with their labels, which is denoted by $\{\tilde{\mathbf{x}}_i, y_i\}$, $\tilde{\mathbf{x}}_i \in \mathbf{R}^M$ with $M < N$.

The success of CS theory is based on the fact that sparse coding of natural signals is nearly ubiquitous. It is found, for example, in the way that neurons encode sensory information (Olshausen & Field, 1996; 1997; Lee et al., 2007). We define the set of all K-sparse vectors $\Sigma_K^L = \{\mathbf{s} \in \mathbf{R}^L$ such that $\|\mathbf{s}\|_0 \leq K\}$ (containing at most K non-zero entries) and assume that data vectors $\mathbf{x}_i$ admit K-sparse representations over an unknown dictionary $\mathbf{D} \in \mathbf{R}^{N \times L}$ ($L \geq N$), i.e. $\mathbf{x}_i = \mathbf{D}\mathbf{s}_i$, with $\mathbf{s}_i \in \Sigma_K^L$. The columns $\mathbf{d}_i \in \mathbf{R}^N$ of a dictionary are called "atoms" because we can express any data vector as a combination of at most K atoms.

To simplify the mathematical treatment, we shuffle the entries of the full vector $\mathbf{x}_i$ to have the observed values placed in the upper M positions, so that we can write:

$$\mathbf{x}_i = \begin{bmatrix} \tilde{\mathbf{x}}_i \\ \mathbf{z}_i \end{bmatrix}, \text{ with } \mathbf{z}_i \in \mathbf{R}^{(N-M)}. \tag{1}$$

However, our results are valid for any configuration of missing entries for each data vector. Accordingly, we define $\mathbf{D}_i \in \mathbf{R}^{N \times L}$ as the shuffled version of $\mathbf{D}$, such that it stacks in the top M rows those corresponding to the observed entries:

$$\mathbf{D}_i = \begin{bmatrix} \tilde{\mathbf{D}}_i \\ \mathbf{E}_i \end{bmatrix}, \text{ with } \tilde{\mathbf{D}}_i \in \mathbf{R}^{M \times L} \text{ and } \mathbf{E}_i \in \mathbf{R}^{(N-M) \times L}. \tag{2}$$

Some important uniqueness properties of sparse vector reconstructions are characterized by the Restricted Isometry Property (RIP) of a matrix, which is defined as follows: a matrix $\mathbf{A} \in \mathbf{R}^{N \times L}$ satisfies the RIP of order K if there exists $\delta_K \in [0, 1)$ such that $(1-\delta_K)\|\mathbf{s}\|_2^2 \leq \|\mathbf{A}\mathbf{s}\|_2^2 \leq (1+\delta_K)\|\mathbf{s}\|_2^2$, holds for all $\mathbf{s} \in \Sigma_K^L$. RIP was introduced in the context of CS by Candes and Tao (Candès & Tao, 2005) and characterizes matrices which are nearly orthonormal when operating on sparse vectors.

Let us assume that a perfect classifier $p_\Theta(y \mid \mathbf{x})$ in a two classes scenario ($y_i \in \{0, 1\}$), e.g. a logistic regression or deep neural network, can be trained on the complete dataset $\{\mathbf{x}_i, y_i\}$, such that $0.5 < p_\Theta(y_i = 1 \mid \mathbf{x}_i) \leq 1.0$ and $0 \leq p_\Theta(y_i = 0 \mid \mathbf{x}_i) \leq 0.5$, $\forall i = 1, 2, \ldots, I$, where Θ is the set of trained parameters. We want to develop a method to estimate the classifier parameters $\hat{\Theta}$ using only the incomplete data $\{\tilde{\mathbf{x}}_i, y_i\}$ and to identify the conditions under which such classifier is compatible with the ideal classifier. In the following section we will show that, when data imputation based on sparsity followed by training is used, some limitations are clearly identified and, therefore, a more integral approach is needed.

2 The problem of training after data imputation

First, we will provide some intuition about the limitation of the sparsity-based imputation or "sequential" approach through a toy example. We consider the classification of hand-written digit images belonging to two classes: the numbers "3" and "8". We will assume that they admit 2-sparse representations over a dictionary and only the right halves of these digits are observed. Let us consider two example vectors $\mathbf{x}_i$ and $\mathbf{x}_j$ belonging to classes "3" and "8", respectively, as shown in Figure 1 (a-d). Here, we are clearly faced with the *indiscernible observations* problem because our observations of two vectors from different classes are identical ($\tilde{\mathbf{x}}_i = \tilde{\mathbf{x}}_j$). It is obvious that at least two possible 2-sparse representations for the observed data exist, let's say $\tilde{\mathbf{x}}_i = \tilde{\mathbf{D}}_i \mathbf{s}_i = \tilde{\mathbf{D}}_i \mathbf{s}'_i$, with $\mathbf{s}_i \neq \mathbf{s}'_i$. Since the solution is not unique, we are not able to correctly reconstruct the original data vectors based only on the observations and the sparsity assumption.

If the K-sparse representation of the observations $\tilde{\mathbf{x}}_i$ are unique, then $\mathbf{x}_i$ can be perfectly reconstructed from the incomplete observations and the classifier can be successfully trained using these reconstructions, but unfortunately this is not the case most of the times. In the particular case where the dictionary $\mathbf{D}$ is known in advance, there are some conditions on the sampling patterns based on properties of matrices such as coherence, spark or RIP that can guarantee a correct reconstruction (Eldar & Kutyniok, 2012). However, these conditions are difficult to meet in practice. Moreover, in the more general case where the dictionary $\mathbf{D}$ is unknown and needs to be learned from data, it is even more difficult to obtain reconstructions of partially observed data vectors that are well separated. For example, even in the case where complete data vectors are well separated, it could happen that their reconstructions become difficult to separate, leading to suboptimal classifiers.

In this paper, motivated by the limitation of the sequential approach, we propose to incorporate the class information of incomplete samples in order to learn simultaneously their sparsest representations and the classifier, such that the classification error of the reconstructions is minimized. In the next section we will introduce our method and provide a theoretical justification.

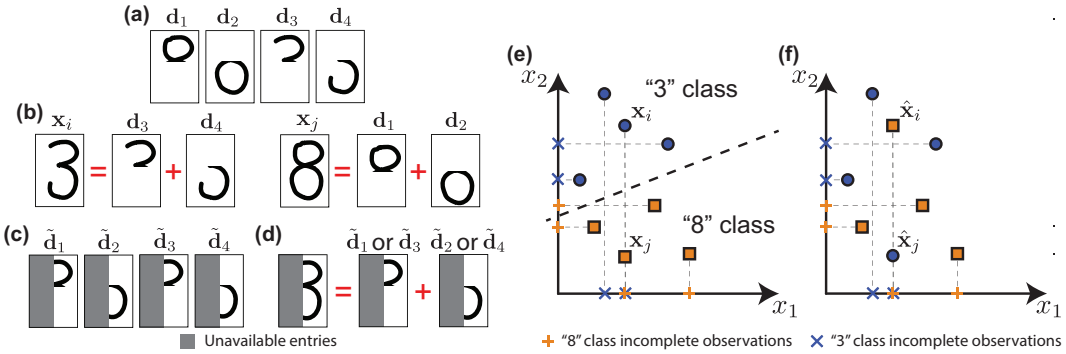

Figure 1: An *indiscernible observation* toy example: (a) 4 out L dictionary elements $\mathbf{d}_i$ (atoms). (b) Digit "3" and "8" can be represented by combining only two atoms in the dictionary (2-sparse representations). (c) Atoms with their left-halves occluded $\tilde{\mathbf{d}}_i$. (d) An occluded digit "3" or "8" admits more than one 2-sparse representation, for example, they can be expressed as the sum of occluded atoms $\tilde{\mathbf{d}}_1$ and $\tilde{\mathbf{d}}_2$, or $\tilde{\mathbf{d}}_3$ and $\tilde{\mathbf{d}}_4$. (e) Simplified two-dimensional representation of samples $\mathbf{x}^T = [x_1, x_2] \in \mathbf{R}^2$ from "3" and "8" classes that are linearly separable where incomplete observations are taken by observing only one entry. Note that $\mathbf{x}_i$ and $\mathbf{x}_j$ belong to different classes but their observations are identical. (f) A wrong reconstruction of data vectors, i.e. $\hat{\mathbf{x}}_i \neq \mathbf{x}_i$ and $\hat{\mathbf{x}}_j \neq \mathbf{x}_j$ can make the set of reconstructed vectors not linearly separable.

3 The simultaneous classification and coding approach

In the previous section, we showed that sparsity-based imputation methods using only the information on available entries are prone to fail because the non-uniqueness of solutions can make the training of a good classifier an impossible task. It is interesting to note that we could solve this problem by incorporating from the beginning the information of the class to which the incomplete data vectors belong to. Let us consider the toy example of previous section, for which we assumed the classes are linearly separable and each data vector admits a K-sparse representation over a dictionary $\mathbf{D} \in \mathbf{R}^{N \times L}$. A two-dimensional simplified visualization for this example is provided in Figure 1(e). When we apply a sparsity-based reconstruction algorithm on observations $\tilde{\mathbf{x}}_i$, and $\tilde{\mathbf{x}}_j$ such that $\tilde{\mathbf{x}}_i = \tilde{\mathbf{x}}_j$ and, assuming for now that dictionary $\mathbf{D}$ is known, we search for the K-sparse representation that is compatible with the measurements, i.e. we find $\mathbf{s} \in \Sigma_K^L$ such that $\tilde{\mathbf{x}}_i = \mathbf{D}_i\mathbf{s}$. In this case, there are at least two equally acceptable solutions $\mathbf{s}_i$ and $\mathbf{s}_j$ corresponding to vectors $\mathbf{x}_i$ and $\mathbf{x}_j$, respectively. If we were to make the wrong decision of assigning $\mathbf{s}_j$ to $\mathbf{x}_i$ and $\mathbf{s}_i$ to $\mathbf{x}_j$, then the resulting set of reconstructed vectors would not be linearly separable, which is illustrated in Figure 1(f), therefore it would be impossible to find a linear classifier. The key observation here is that, in order to assign the proper sparse representation we could also look at the label information y_i and y_j, and make the solution unique by discarding any K-sparse representation resulting in not separable vector reconstructions. This suggests that we would need to train the classifier and find the proper representation not only as sparse as possible but also providing the best separation of classes. In other words, we should simultaneously learn the optimal classifier and find sparse representation of observations. Next, we theoretically analyze the case of using a logistic regression classifier and demonstrate that, if we are able to train a classifier on incomplete

observations such that their sparse representations are well separated by a hyperplane, then the same classifier correctly separates the original (unobserved) data vectors.

3.1 Theoretical guarantees in the logistic regression case

Let us consider the case of a logistic regression classifier (Hastie et al., 2009) where the set of parameters $\Theta = \{\mathbf{w}, b\}$ are a vector $\mathbf{w} \in \mathbf{R}^N$ and a scalar (bias) $b \in \mathbf{R}$, and a perfect classifier exists if there is a hyperplane that separates both classes, i.e., for each data vector $\mathbf{x}_i$: $f(\mathbf{x}_i) = \mathbf{w}^T\mathbf{x}_i + b > 0$ when $y_i = 1$ and $f(\mathbf{x}_i) \leq 0$ if $y_i = 0$. We investigate the conditions under which a perfect classifier can be trained using incomplete data samples and their labels. By using the same shuffling as in equation (1), we can write the partition of the shuffled coefficients $\mathbf{w}_i \in \mathbf{R}^N$ as follows:

$$\mathbf{w}_i = \begin{bmatrix} \tilde{\mathbf{w}}_i \\ \mathbf{u}_i \end{bmatrix}, \text{ where } \mathbf{u}_i \in \mathbf{R}^{(N-M)}. \tag{3}$$

Since the classifier is based on the feature $f(\mathbf{x}_i) = \mathbf{w}_i^T\mathbf{x}_i + b = \tilde{\mathbf{w}}_i^T\tilde{\mathbf{x}}_i + \mathbf{u}_i^T\mathbf{z}_i + b$, the vector $\mathbf{u}_i \in \mathbf{R}^{(N-M)}$ contains the weights associated with unobserved entries $\mathbf{z}_i$. The following definition introduces a way to measure the amount of discriminative information associated with missing entries.

Definition 3.1. Discriminative Missing Information (DMI) μ_i*: Given a classifier* $\{\mathbf{w}, b\}$ *and a partition of observed/missing entries in sample vector* $\mathbf{x}_i$ *such that the shuffled vector of coefficients* $\mathbf{w}_i$ *can be partitioned as in eq. (3), we define the DMI for sample* $\mathbf{x}_i$ *as:* $\mu_i = \|\mathbf{u}_i\|_1 = \sum_{m=1}^{N-M} |\mathbf{u}_i(m)|$.

Intuitively, if the DMI is small enough, the classification will not strongly depend on the unobserved entries $\mathbf{z}_i$ and then a classifier could be trained from incomplete data samples. The following theorem establishes precise conditions under which the existence of a classifier obtained from incomplete data can be guaranteed and thus, a perfect classifier can be successfully trained. The proof can be found in Appendix A

Theorem 3.2. *Given a dataset* $\{\mathbf{x}_i, y_i\}$, $i = 1, 2, \ldots, I$ *with normalized data vectors* ($\|\mathbf{x}_i\| \leq 1$) *admitting a* K*-sparse representation over a dictionary* $\mathbf{D} \in \mathbf{R}^{N\times L}$ *with unit-norm columns and satisfying the RIP of order* K *with constant* δ_K*, and suppose that, we have obtained an alternative dictionary* $\mathbf{D}' \in \mathbf{R}^{N\times L}$ *also satisfying the RIP of order* K *with constant* δ_K *such that, for the incomplete observation* $\tilde{\mathbf{x}}_i \in \mathbf{R}^M$*, the* K*-sparse representation solution is non-unique, i.e.* $\exists \mathbf{s}_i, \mathbf{s}'_i \in \Sigma_K^L$ *such that* $\tilde{\mathbf{x}}_i = \tilde{\mathbf{D}}_i\mathbf{s}_i = \tilde{\mathbf{D}}'_i\mathbf{s}'_i$*, where* $\tilde{\mathbf{D}}_i, \tilde{\mathbf{D}}'_i \in \mathbf{R}^{M\times L}$ *are the matrices containing the observed rows of* $\mathbf{D}$ *and* $\mathbf{D}'$*, respectively;* $\mathbf{s}_i \in \mathbf{R}^L$ *is the vector of coefficients of the true data, i.e.* $\mathbf{x}_i = \mathbf{D}_i\mathbf{s}_i$ *and* $\mathbf{s}'_i$ *provides a plausible reconstruction through* $\hat{\mathbf{x}}_i = \mathbf{D}'_i\mathbf{s}'_i$ *with* $\|\hat{x}_i\| \leq 1$*. If a perfect classifier* $\{\mathbf{w}_i, b\}$ *of the reconstruction* $\hat{\mathbf{x}}_i$ *exists such that*

$$\epsilon > 2K\mu_i/(1 - \delta_K), \tag{4}$$

with $\epsilon > 0$ *being the minimum distance of reconstructed data sample to the separating hyperplane, then the full data vector* $\mathbf{x}_i$ *is also perfectly separated with this classifier, in other words:* $f(\mathbf{x}_i) = \mathbf{w}_i^T\mathbf{x}_i + b > 0$ *(*≤ 0*) if* $y_i = 1$ *(*$y_i = 0$*).*

It is interesting to note that, supposing that we have obtained a classifier from incomplete data, we are able to evaluate condition (4) and figure out whether such a classifier is optimal by testing the condition of the theorem. From condition (4), we also conclude that the following conditions are desirable: (a) very well separated reconstructed vectors (large ϵ); (b) very sparse model (small K); (c) small norm variability of transformed K-sparse vectors through $\mathbf{D}$ (small δ_K), which can be interpreted as a quasi-orthonormal basis behaviour; and (d), small weights assigned to unobserved entries of $\mathbf{x}_i$ (small DMI μ_i).

4 The proposed method

Here, we propose to combine the training of the classifier together with the learning of a dictionary and optimal sparse representations such that reconstructed data vectors are compatible with the observed entries and, at the same time, well separated. To do that we propose to minimize the following global cost function:

$$J(\Theta, \mathbf{D}, \mathbf{s}_i) = \frac{1}{I} \sum_{i=1}^{I} \left\{ J_0(\Theta, \hat{\mathbf{x}}_i, y_i) + \lambda_1 J_1(\mathbf{D}, \mathbf{s}_i) + \lambda_2 J_2(\mathbf{s}_i) \right\}, \tag{5}$$

where Θ contains the classifier parameters, i.e. the vector of coefficients and bias for a logistic regression model, or the vector of weights in a deep neural network classifier architecture; $\mathbf{D} \in \mathbf{R}^{N \times L}$ ($L \geq N$) is a dictionary and $\mathbf{s}_i \in R^L$ are the representation coefficients such that the reconstructed data vectors are $\hat{\mathbf{x}}_i = \mathbf{D}\mathbf{s}_i$. $J_0(\Theta, \hat{\mathbf{x}}_i, y_i)$ is a measure of the classification error for sample i. Typically, we use the crossentropy measure, i.e. $J_0(\Theta, \mathbf{x}, y_i) = -\log(p_\Theta^{y_i}(\mathbf{x}))$, where $p_\Theta^{y_i}(\mathbf{x}_i)$ is the probability assigned by the classifier to sample $\mathbf{x}_i$ as belonging to class y_i. $J_1(\mathbf{D}, \mathbf{s}_i)$ is a measure of the error associated with the sparse representation when it is restricted to the observed entries, and is defined as follows: $J_1(\mathbf{D}, \mathbf{s}_i) = \frac{M}{N} \|\mathbf{m}_i * (\mathbf{x}_i - \mathbf{D}\mathbf{s}_i)\|^2$, where $*$ stands for the entry-wise product, $\mathbf{m}_i \in \mathbf{R}^N$ is the observation mask for sample i, with $m_i(n) \in \{0, 1\}$ such that $m_i(n) = 1$ or 0 if data entry $\mathbf{x}_i(n)$ is available or missing, respectively. $J_2(\mathbf{s}_i) = \frac{1}{N} \|\mathbf{s}_i\|_1$ is the ℓ_1-norm whose minimization promotes the sparsity of the representation (Candès & Tao, 2005). Finally, the hyper-parameters λ_1 and λ_2 allow to give more or less importance to the representation accuracy and its sparsity, with respect to the classification error. Intuitively, minimizing equation (5) favors solutions that not only have sparse representations compatible with observed entries, but also provides reconstructions that are best separated in the given classes.

To minimize the cost function in equation (5) we propose to alternate between the optimization over $\{\Theta, \mathbf{D}\}$ and $\mathbf{s}_i$ ($i = 1, 2, \ldots, I$). As usual, we adopt a first order (gradient based) search of minima, where gradients of functions $J_0(\cdot)$, $J_1(\cdot)$ and $J_2(\cdot)$ are easily derived. Note that in the case of feedforward architectures, the back-propagation technique can be used for the first term $J_0(\cdot)$. It is also noted that function $J_2(\cdot)$ is not differentiable at zero, so we need to avoid zero crossing in every update. In the case where the given test dataset is also incomplete we can use the dictionary learned during the training phase to find the sparsest representation for the given observations, compute the corresponding full vector reconstructions and apply the classifier to them. The algorithms are presented in Appendix B.

5 Experimental results

We synthetically generated $I = 11,000$ ($10,000$ training $+1,000$ test) K-sparse data vectors $\mathbf{x}_i \in \mathbf{R}^{100}$ using a dictionary $\mathbf{D} \in \mathbf{R}^{100 \times 200}$ obtained from a Gaussian distribution with normalized atoms, i.e. $\|\mathbf{D}(:, l)\| = 1, \forall l$. A random hyperplane $\{\mathbf{w}, b\}$ with $\mathbf{w} \in \mathbf{R}^N$, $b \in \mathbf{R}$ was randomly chosen dividing data vectors into two classes according to the sign of the expression $\mathbf{w}^T \mathbf{x}_i + b$, which defines the value of label y_i. We also controlled the degree of separation between the two classes by regenerating all data vectors with distances to the hyperplane lower than a pre-specified threshold d. We applied our simultaneous method, with hyperparameters λ_1 and λ_2 tuned via cross-validation, to train a logistic regression classifier on the training dataset with randomly distributed missing entries, and compared the obtained Test Accuracy against the following standard sequential methods: **Seq. Sparse**: reconstructions are obtained by finding the sparsest representation compatible with the observations solving a LASSO problem; **Zero Fill**: missing entries are filled with zeros, which is equivalent to ignore unknown values; **Mean Unsupervised**: missing entries are filled with the

mean computed on the available values in the same position in the rest of data samples; **Mean Supervised**: as in the previous case but the mean is computed on the samples of the same class vectors only; **K-Nearest Neighbor (KNN)**: as in the previous case but the mean is computed on the K-Nearest Neighbors of the same class only. Since the objective here is to compare the performance of obtained classifiers, we computed the accuracy on the test dataset using all the methods for two cases of degree of separation between classes ($d = 0.0, 0.2$), four levels of sparsity ($K = 4, 8, 32, 128$) and missing values in the training dataset ranging from 25% to 95% as shown in Fig. 2. Our results show that the simultaneous algorithm clearly outperforms all the sequential methods. It is interesting to note that, when classes has some degree of separation ($d = 0.2$), using simple methods as computing the mean or filling with zeros, can give acceptable results but not better than our simultaneous algorithm.

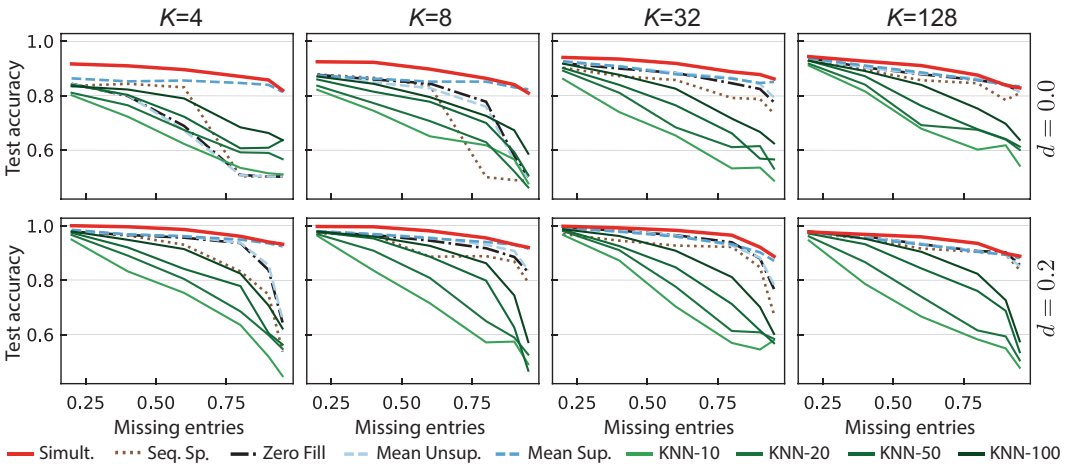

Figure 2: Experimental results on synthetic dataset using our "simultaneous" (red thick solid line) compared to various "sequential" methods. Obtained test accuracy as a function of the percentage of missing entries for two degrees of classes separation $d = 0.0, 0.2$ and four levels of sparsity $K = 4, 8, 32, 128$.

We also applied our algorithm to three popular computer vision datasets: MNIST (LeCun et al., 1989) and Fashion (Xiao et al., 2017) consisting of 70,000 images (60,000 train + 10,000 test) each; and CIFAR10 (Krizhevsky & Hinton, 2009) having 60,000 images (50,000 train + 10,000 test). MNIST/Fashion datasets contains 28×28 gray scale images while CIFAR10 dataset is built upon $32 \times 32 \times 3$ color images of different objects. The corresponding data vector size is $N = 28 \times 28 = 784$ for MNIST/Fashion and $N = 32 \times 32 \times 3 = 3,072$ for CIFAR10. We considered a dictionary of size $N \times N$ and applied our simultaneous algorithm to learn the classifier on incomplete data for these datasets using uniform random missing masks at several levels of missing entries (25%, 50% and 75%) and for 50% random partial occlusions for MNIST/Fashion. We used a logistic regression classifier (single layer neural network) and a 4-layer convolutional neural network (CNN4) for the MNIST/Fashion dataset using batch normalization (BN) in the Fashion dataset, while for CIFAR10 dataset, a residual neural network (Resnet-18, (He et al., 2016)) was implemented. Table 1, columns 3rd through 10th, show the accuracy obtained in the test step, when the model was trained on incomplete data and applied to incomplete as well as to complete test data. The right-most column shows the baseline results obtained by training the model on complete training dataset using the

Dataset	Classifier	Random missing entries %Train / %Test						Occlusion %Train / %Test		Baseline %Train / %Test
		75/75	**50/50**	**25/25**	**75/0**	**50/0**	**25/0**	**50/50**	**50/0**	**0/0**
MNIST	**Log. Reg.**	90.45	93.68	94.14	91.94	93.44	94.43	-	-	91.95
	CNN4	94.62	98.34	98.94	98.14	98.94	98.95	88.55	91.37	98.95
Fashion	**CNN4+BN**	83.71	86.09	86.38	86.39	87.11	87.04	81.73	82.47	90.76
CIFAR10	**Resnet18**	35.50	43.20	45.41	35.40	43.77	45.17	-	-	86.22

Table 1: Test accuracy: columns 3rd through 10th show the results with incomplete training data and incomplete/complete test data. The baseline results were obtained by training the model on complete data.

implementations found in [1] and [2]. We implemented our algorithm in Pytorch 1.0.0 and ran the experiments on a single GPU. Note that for the logistic regression classifier, we obtained better results when training with incomplete data rather than using complete data. Also, it is highlighted that training on incomplete data, with 50% or less random missing entries, provides the same test accuracy as training on complete data for MNIST dataset. The hyper-parameters λ_1 and λ_2 were adjusted by cross-validation through a grid-search, as shown in Appendix C. In Fig. 3 we present some selected examples comparing the original images in the MNIST/Fashion test dataset, their observations using random masks and partial occlusions, and reconstructions computed using their sparse representation over a dictionary learned from the incomplete training data. Additional visual examples are provided in Appendix C.

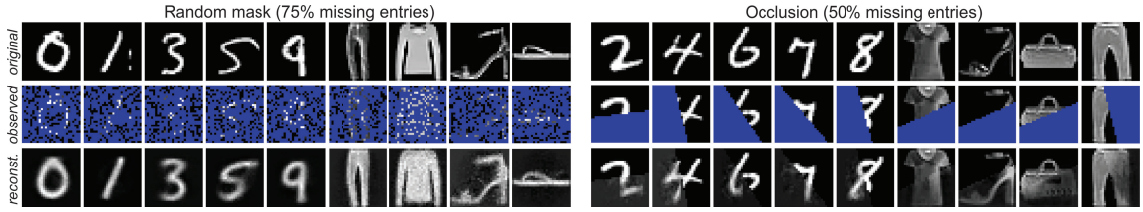

Figure 3: Reconstruction of MNIST/Fashion test data with missing entries: original images (top), incomplete observations (middle) and their reconstructions (bottom).

6 Conclusions

We demonstrated that assuming a sparse representation model for input data vectors allows the successful learning of a general classifier on incomplete data. We analyzed the limitations of the classical imputation approach and demonstrated through experiments that our simultaneous algorithm always outperforms sequential methods for various cases such as LASSO, zero-filling, supervised/unsupervised mean and KNN based methods. Our approach is conceptually similar to the work in Ghahramani & Jordan (1993) but, instead of using a probabilistic model, we use sparse coding, which is known to fit very well for natural signals. It is also noted that in Huang & Aviyente (2006), sparse representations of data vectors were incorporated in the optimization process for maximizing the Fisher discriminant in a context of classification. However, no theoretical guarantee was presented in that paper and our approach can be applied to any deep learning classifier architecture by simply incorporating a sparse data model, and adding two regularization terms to the objective function: one to measure the accuracy of the sparse model to explain the observations, and another to favor sparsity of the data representation, which is based on the ℓ_1 norm.

[1] https://github.com/pytorch/examples/tree/master/mnist
[2] https://github.com/kuangliu/pytorch-cifar

References

M Aharon, M Elad, and A Bruckstein. K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation. *Signal Processing, IEEE Transactions on*, 54(11):4311–4322, 2006.

Gustavo E A P A Batista and Maria Carolina Monard. A Study of K-Nearest Neighbour as an Imputation Method. *HIS*, 2002.

Yoshua Bengio and Francois Gingras. Recurrent Neural Networks for Missing or Asynchronous Data. *NIPS*, 1995.

Cesar F Caiafa and A Cichocki. Computing Sparse representations of multidimensional signals using Kronecker bases. *Neural Computation*, pp. 186–220, December 2012.

E.J. Candès and T Tao. Decoding by Linear Programming. *Institute of Electrical and Electronics Engineers. Transactions on Information Theory*, 51(12):4203–4215, December 2005.

Emmanuel J Candes and Terence Tao. The Power of Convex Relaxation: Near-Optimal Matrix Completion. *Institute of Electrical and Electronics Engineers. Transactions on Information Theory*, 56(5):2053–2080, April 2010.

Emmanuel J Candes, Justin K Romberg, and Terence Tao. Stable signal recovery from incomplete and inaccurate measurements. *Communications on pure and applied mathematics*, 59(8):1207–1223, 2006.

D L Donoho. Compressed sensing. *Institute of Electrical and Electronics Engineers. Transactions on Information Theory*, 52(4):1289–1306, March 2006.

YC Eldar and Gitta Kutyniok. Compressed Sensing: Theory and Applications. *New York: Cambridge Univ. Press*, 20:12, 2012.

M J Fadili, J-L Starck, and F Murtagh. Inpainting and Zooming Using Sparse Representations. *The Computer Journal*, 52(1):64–79, February 2008.

Francoise Fessant and Sophie Midenet. Self-Organising Map for Data Imputation and Correction in Surveys. *Neural Computing and Applications*, 10(4):300–310, 2002.

Pedro J García-Laencina, José-Luis Sancho-Gómez, and Aníbal R Figueiras-Vidal. Pattern classification with missing data: a review. *Neural Computing and Applications*, 19(2):263–282, September 2009.

Zoubin Ghahramani and Michael I Jordan. Supervised learning from incomplete data via an EM approach. *NIPS*, 1993.

Karol Gregor and Yann LeCun. Learning Fast Approximations of Sparse Coding. *ICML*, 2010.

Trevor Hastie, Robert Tibshirani, and Jerome H Friedman. The elements of statistical learning - data mining, inference, and prediction, 2nd Edition. *Springer series in statistics*, 2009.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR*, pp. 770–778. IEEE, 2016.

Ke Huang and Selin Aviyente. Sparse Representation for Signal Classification. *NIPS*, 2006.

Sheng-Jun Huang, Miao Xu, Ming-Kun Xie, Masashi Sugiyama, Gang Niu, and Songcan Chen. Active Feature Acquisition with Supervised Matrix Completion. *arXiv.org*, pp. arXiv:1802.05380, February 2018.

A Krizhevsky and G Hinton. *Learning multiple layers of features from tiny images*. PhD thesis, Toronto University, 2009.

Yann LeCun, Bernhard E Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne E Hubbard, and Lawrence D Jackel. Handwritten Digit Recognition with a Back-Propagation Network. *NIPS*, 1989.

Honglak Lee, Alexis Battle, Rajat Raina, and Andrew Y Ng. Efficient sparse coding algorithms. In *Advances in Neural Information Processing Systems*, pp. 801–808. Stanford University, Palo Alto, United States, December 2007.

Roderick J A Little and Donald B Rubin. *Statistical Analysis with Missing Data*. John Wiley & Sons, August 2014.

Ji Liu, Przemyslaw Musialski, Peter Wonka, and Jieping Ye. Tensor Completion for Estimating Missing Values in Visual Data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):208–220, November 2012.

Julien Mairal, Francis Bach, Jean Ponce, Guillermo Sapiro, and Andrew Zisserman. Supervised Dictionary Learning. *arXiv.org*, pp. arXiv:0809.3083, September 2008.

B A Olshausen and D J Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607–609, June 1996.

BA Olshausen and DJ Field. Sparse coding with an overcomplete basis set: A strategy employed by V1? *Vision research*, 37(23):3311–3325, 1997.

Ignacio Ramirez, Pablo Sprechmann, and Guillermo Sapiro. Classification and clustering via dictionary learning with structured incoherence and shared features. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 3501–3508. University of Minnesota System, United States, IEEE, August 2010.

J Solé-Casals, C F Caiafa, Q Zhao, and A Cichocki. Brain-Computer Interface with Corrupted EEG Data: a Tensor Completion Approach. *Cognitive Computation*, 10(6):1062–1074, July 2018.

Pablo Sprechmann and Guillermo Sapiro. Dictionary learning and sparse coding for unsupervised clustering. In *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, pp. 2042–2045. University of Minnesota System, United States, IEEE, November 2010.

Ivana Tosic and Pascal Frossard. Dictionary Learning. *IEEE Signal Processing Magazine*, 28(2): 27–38, March 2011.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms. *arXiv.org*, pp. arXiv:1708.07747, August 2017.

Song Yee Yoon and Soo Young Lee. Training algorithm with incomplete data for feed-forward neural networks. *Neural Processing Letters*, 10(3):171–179, January 1999.

Longhao Yuan, Chao Li 0013, Danilo P Mandic, Jianting Cao, and Qibin Zhao. Tensor Ring Decomposition with Rank Minimization on Latent Space - An Efficient Approach for Tensor Completion. *arXiv*, cs.LG:arXiv:1809.02288, 2018.

A Proof of Theorem 3.2

Proof. Let us first consider the case with $y_i = 1$. We assume that we are able to find a perfect classifier on the reconstructed full data vector such that

$$f(\hat{\mathbf{x}}_i) = \mathbf{w}_i^T \hat{\mathbf{x}}_i + b > \epsilon > 0, \tag{6}$$

where $\hat{\mathbf{x}}_i = \mathbf{D}_i' \mathbf{s}_i'$. Now, we note that the K-sparse representation solution given the observation is non-unique, i.e. $\exists \mathbf{s}_i, \mathbf{s}_i' \in \Sigma_K^L$, such that $\tilde{\mathbf{x}}_i = \tilde{\mathbf{D}}_i \mathbf{s}_i = \tilde{\mathbf{D}}_i' \mathbf{s}_i'$. We assume that $\mathbf{s}_i$ is the true K-sparse vector such that the unobserved full vector is $\mathbf{x}_i = \mathbf{D}_i \mathbf{s}_i$ and $\mathbf{s}_i'$ provides an alternative reconstruction through $\mathbf{D}_i'$ which explains the observations but $\hat{\mathbf{x}}_i \neq \mathbf{x}_i$.

Analogously to equation (1), we can find a partition of the reconstructed vector $\hat{\mathbf{x}}_i$ as follows:

$$\hat{\mathbf{x}}_i = \begin{bmatrix} \tilde{\mathbf{x}}_i \\ \mathbf{z}_i' \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{D}}_i' \\ \mathbf{E}_i' \end{bmatrix} \mathbf{s}_i', \tag{7}$$

where $\mathbf{z}_i' \in \mathbf{R}^{(N-M)}$, and we want to prove that $f(\hat{\mathbf{x}}_i) > \epsilon$ implies $f(\mathbf{x}_i) > 0$, i.e. the original data vector $\mathbf{x}_i$ (unobserved) is also correctly classified by the logistic regression with parameters $\{\mathbf{w}_i, b\}$. To that end we need to evaluate the following expression:

$$f(\mathbf{x}_i) = \mathbf{w}_i^T \mathbf{x}_i + b = \tilde{\mathbf{w}}_i^T \tilde{\mathbf{x}}_i + \mathbf{u}_i^T \mathbf{z}_i + b, \tag{8}$$

where equation (3) was used. If we add and substract $\mathbf{u}_i^T \mathbf{z}_i'$ and arrange the terms in the previous equation, we get:

$$\begin{aligned} f(\mathbf{x}_i) &= \tilde{\mathbf{w}}_i^T \tilde{\mathbf{x}}_i + \mathbf{u}_i^T \mathbf{z}_i' + b + \mathbf{u}_i^T (\mathbf{z}_i - \mathbf{z}_i'), & (9) \\ &= f(\hat{\mathbf{x}}_i) + \mathbf{u}_i^T \mathbf{E}_i \mathbf{s}_i - \mathbf{u}_i^T \mathbf{E}_i' \mathbf{s}_i'. & (10) \end{aligned}$$

The first term in the right-hand side of last equation is $f(\hat{\mathbf{x}}_i) > \epsilon > 0$ so if we show that $|\mathbf{u}_i^T \mathbf{E}_i \mathbf{s}_i - \mathbf{u}_i^T \mathbf{E}_i' \mathbf{s}_i'| < \epsilon$, then the proof will be complete. To do so, we can write:

$$|\mathbf{u}_i^T \mathbf{E}_i \mathbf{s}_i - \mathbf{u}_i^T \mathbf{E}_i' \mathbf{s}_i'| \leq |\mathbf{u}_i^T \mathbf{E}_i \mathbf{s}_i| + |\mathbf{u}_i^T \mathbf{E}_i' \mathbf{s}_i'|, \tag{11}$$

and

$$|\mathbf{u}_i^T \mathbf{E}_i \mathbf{s}_i| = \Big| \sum_{m=1}^{N-M} \mathbf{u}_i(m) \sum_{n=1}^{N} \mathbf{E}_i(m,n) \mathbf{s}_i(n) \Big| \leq \sum_{m=1}^{N-M} |\mathbf{u}_i(m)| \sum_{n=1}^{N} |\mathbf{E}_i(m,n)| |\mathbf{s}_i(n)|. \tag{12}$$

Since we assumed normalized vectors $\|\mathbf{x}_i\| \leq 1$, by applying the left-hand side of the RIP we obtain: $\|\mathbf{s}_i\| \leq 1/(1-\delta_K)$, which implies that $|s_i(n)| \leq 1/(1-\delta_K)$. Now, taking into account that $|\mathbf{E}_i(m,n)| \leq 1$ (columns of $\mathbf{D}$ are unit-norm) and using the fact that $\mathbf{s}_i \in \Sigma_K^L$ we obtain: $|\mathbf{u}_i^T \mathbf{E}_i \mathbf{s}_i| \leq \frac{K}{(1-\delta_K)} \sum_{m=1}^{N-M} |\mathbf{u}_i(m)|$; and, similarly, we can obtain that $|\mathbf{u}_i^T \mathbf{E}_i \mathbf{s}_i'| \leq \frac{K}{(1-\delta_K)} \sum_{m=1}^{N-M} |\mathbf{u}_i(m)|$. Putting everything together into equation (11) we get:

$$|\mathbf{u}_i^T \mathbf{E}_i \mathbf{s}_i - \mathbf{u}_i^T \mathbf{E}_i' \mathbf{s}_i'| \leq \frac{2K}{(1-\delta_K)} \sum_{m=1}^{N-M} |\mathbf{u}_i(m)| < \epsilon, \tag{13}$$

where we used that $\sum_{m=1}^{N-M} |\mathbf{u}_i(m)| < \frac{\epsilon(1-\delta_K)}{2K}$, which completes the proof for the case $y_i = 1$. Similarly, it is strait-forward to prove that, if $y_i = 0$, having $f(\hat{\mathbf{x}}_i) < -\epsilon < 0$ implies that $f(\mathbf{x}_i) < 0$. □

B Algorithms

Here, the pseudocode of the algorithms discussed in the paper are presented. In Algorithm 1, the simultaneous classification and coding method is described. It consists in the iterative alternation between the update of the classifier's parameters together with the dictionary, and the update of the sparse coefficients for the representation of data observations. Once the classifier is trained, we are able to apply it to incomplete test data by using Algorithm 2, where for fixed Θ and $\mathbf{D}$, we need to find the corresponding sparse coefficients $\mathbf{s}_i$, compute the full data vector estimations and, finally, apply the classifier.

The standard sparsity-based imputation method is presented in Algorithm 3 (sequential approach), which consists of learning first the optimal $\mathbf{D}$ and sparse coefficients $\mathbf{s}_i$ compatible with the incomplete observations (dictionary learning and coding phase), followed by the training phase, where the classifier is tuned in order to minimize the classification error of the reconstructed input data vectors $\hat{\mathbf{x}}_i = \mathbf{D}\mathbf{s}_i$.

Algorithm 1 : Simultaneous classification and coding (training on incomplete data)

Require: Incomplete data vectors and their labels $\{\tilde{\mathbf{x}}_i, y_i\}$, $i = 1, 2, \ldots, I$, hyper-parameters α, λ_1 and λ_2, number of iterations N_{iter} and update rate σ

Ensure: Classifier parameters Θ and sparse representation of full data vectors $\hat{\mathbf{x}}_i = \mathbf{D}\mathbf{s}_i, \forall i$

Initialize $\Theta, \mathbf{D}, \mathbf{s}_i, \forall i$ randomly
for $n \leq N$ **do**
  `Fix` $\mathbf{s}_i$, `update` Θ and $\mathbf{D}$:
  $\Theta = \Theta - \sigma \frac{\partial J}{\partial \Theta}$, $\mathbf{D} = \mathbf{D} - \sigma \frac{\partial J}{\partial \mathbf{D}}$
  `Normalize columns of matrix` $\mathbf{D}$
  `Fix` Θ and $\mathbf{D}$, `update` $\mathbf{s}_i$, $\forall i$:
  $\Delta_i = -\sigma \frac{\partial J}{\partial \mathbf{s}_i}$, $\forall i$
  if $\mathbf{s}_i(j)[\mathbf{s}_i(j) + \Delta_i(j)] < 0$ **then**
    $\Delta_i(j) = -\mathbf{s}_i(j)$; `avoid zero crossing`
  end if
  $\mathbf{s}_i = \mathbf{s}_i + \Delta_i$, $\forall i$
end for
return $\Theta, \mathbf{D}, \mathbf{s}_i, \hat{\mathbf{x}}_i = \mathbf{D}\mathbf{s}_i, \forall i$

Algorithm 2 : Testing on incomplete data

Require: Incomplete data vectors $\{\tilde{\mathbf{x}}_i\}$, $i = 1, 2, \ldots, I$, classifier parameters Θ, dictionary $\mathbf{D}$, hyper-parameters λ_1 and λ_2, number of iterations N_{iter} and update rate σ

Ensure: Class assigned to each vector y_i and sparse representation of full data vectors $\hat{\mathbf{x}}_i = \mathbf{D}\mathbf{s}_i, \forall i$

Sparse coding stage: for fixed dictionary $\mathbf{D}$ find sparse representations of observations $\tilde{\mathbf{x}}_i$
Initialize $\mathbf{s}_i, \forall i$ randomly
for $n \leq N_{iter}$ **do**
  $\Delta_i = -\sigma \left[\lambda_1 \frac{\partial J_1}{\partial \mathbf{s}_i} + \lambda_2 \frac{\partial J_2}{\partial \mathbf{s}_i}\right]$, $\forall i$
  if $\mathbf{s}_i(j)[\mathbf{s}_i(j) + \Delta_i(j)] < 0$ **then**
    $\Delta_i(j) = -\mathbf{s}_i(j)$; `avoid zero crossing`
  end if
  $\mathbf{s}_i = \mathbf{s}_i - \sigma \Delta_i$, $\forall i$
end for
$\hat{\mathbf{x}}_i = \mathbf{D}\mathbf{s}_i, \forall i$; Compute reconstructions of unobserved vector data $\mathbf{x}_i$
Classification stage: apply classifier to reconstructions $\mathbf{x}_i$
$y_i = \arg\max_y (p^y_\Theta(\hat{\mathbf{x}}))$
return $\Theta, y_i, \mathbf{s}_i, \hat{\mathbf{x}}_i, \forall i$

Algorithm 3 : Sequential approach (imputation method)

Require: Incomplete data vectors and their labels $\{\tilde{\mathbf{x}}_i, y_i\}$, $i = 1, 2, \ldots, I$, hyper-parameters α, λ_1 and λ_2, number of iterations N_{iter} and update rate σ

Ensure: Classifier parameters Θ and sparse representation of full data vectors $\hat{\mathbf{x}}_i = \mathbf{D}\mathbf{s}_i, \forall i$

Initialize $\mathbf{D}, \mathbf{s}_i, \forall i$ randomly
`Dictionary learning and coding stage: update` $\mathbf{D}$ and $\mathbf{s}_i$
for $n \leq N$ **do**
  $\mathbf{D} = \mathbf{D} - \sigma \frac{\partial J_1}{\partial \mathbf{D}}$
  `Normalize columns of matrix` $\mathbf{D}$
  $\Delta_i = -\sigma \left[\lambda_1 \frac{\partial J_1}{\partial \mathbf{s}_i} + \lambda_2 \frac{\partial J_2}{\partial \mathbf{s}_i}\right], \forall i$
  if $\mathbf{s}_i(j)[\mathbf{s}_i(j) + \Delta_i(j)] < 0$ **then**
    $\Delta_i(j) = -\mathbf{s}_i(j)$; `avoid zero crossing`
  end if
  $\mathbf{s}_i = \mathbf{s}_i + \Delta_i, \forall i$
end for
$\hat{\mathbf{x}}_i = \mathbf{D}\mathbf{s}_i, \forall i$; Compute reconstructions of unobserved vector data $\mathbf{x}_i$
`Training stage: update` Θ
for $n \leq N$ **do**
  $\Theta = \Theta - \sigma \frac{\partial J_0}{\partial \Theta}$; `Update` Θ:
end for
return $\Theta, \mathbf{D}, \mathbf{s}_i, \hat{\mathbf{x}}_i = \mathbf{D}\mathbf{s}_i, \forall i$

C Additional experimental results

In Table 2 and Figure 4 we present the results of the grid search for hyper-parameter tuning on MNIST and CIFAR10 datasets. We fit our model to the training dataset for a range of values of parameters λ_1 and λ_2 and apply it to the test data set. Figure 4 shows the accuracy obtained for the test dataset with different classifiers and levels of missing entries. We chose the hyper-parameters values such that the test accuracy is maximum, as shown in Table 2.

To visually evaluate our results, additional selected examples of original (complete) images of the test dataset in MNIST, together with their given incomplete observations and obtained reconstructions, are shown in Figure 6

Dataset	**Classifier**	**Random missing entries**						**Occlusion**	
		75%		**50%**		**25%**		**50%**	
		λ_1	λ_2	λ_1	λ_2	λ_1	λ_2	λ_1	λ_2
MNIST	**Log. Reg.**	0.32	1.28	0.64	1.28	0.64	1.28	-	-
	CNN4	1.28	1.28	2.56	1.28	5.12	1.28	10.24	10.24
CIFAR10	**Resnet18**	0.032	0.064	0.013	0.0016	0.013	0.0002	-	-

Table 2: Hyper-parameter tuning: crossvalidated hyperparameters λ_1 and λ_2 obtained for all datasets and classifiers used in our experiments.

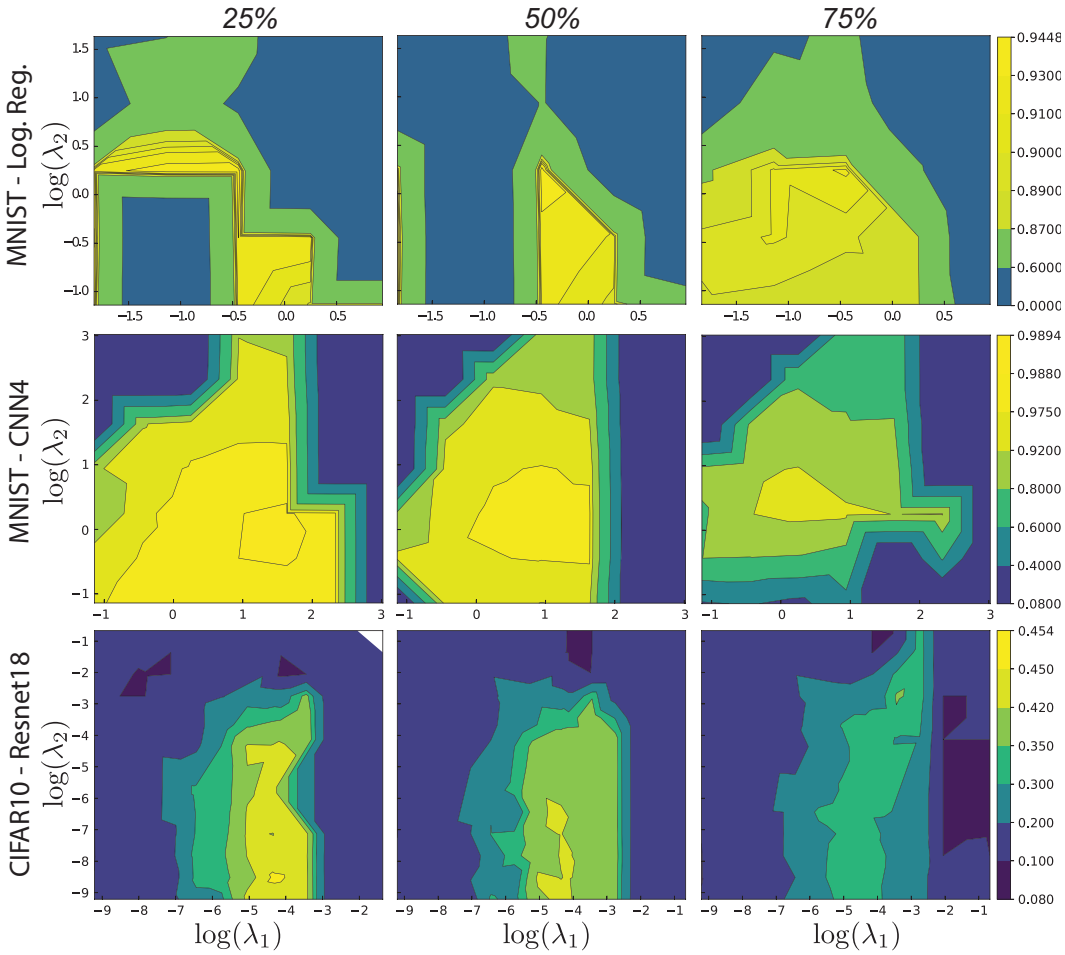

Figure 4: Testing accuracy in the grid search for hyper-parameter tuning: λ_1 and λ_2 were chosen through crossvalidation by maximizing the testing accuracy in all datasets with different levels of missing entries: 25%, 50% and 75%.

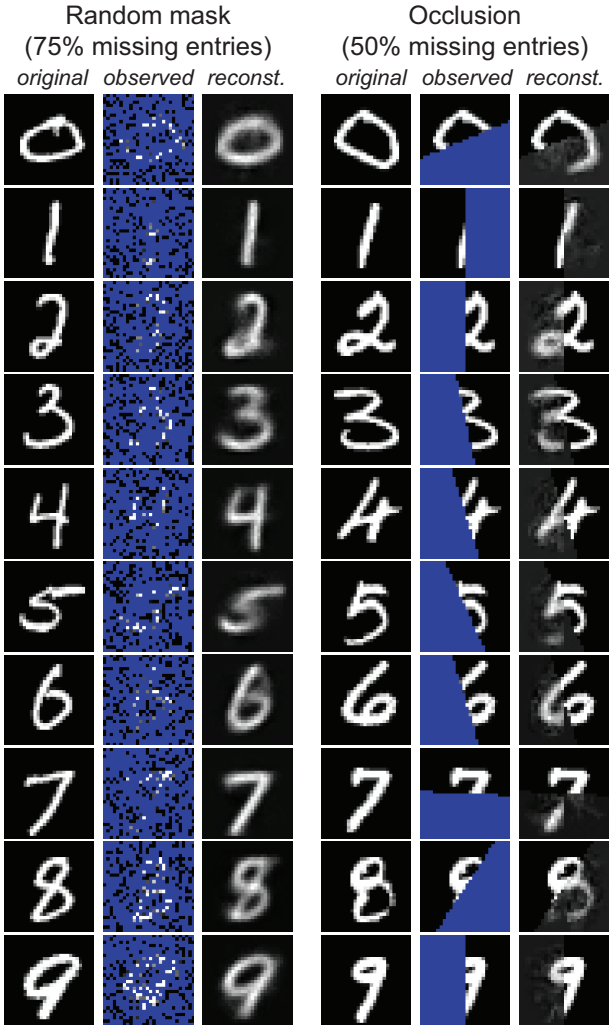

Figure 5: Reconstructions of incomplete test MNIST dataset vectors by applying our simultaneous classification and coding algorithm using the CNN4 architecture.

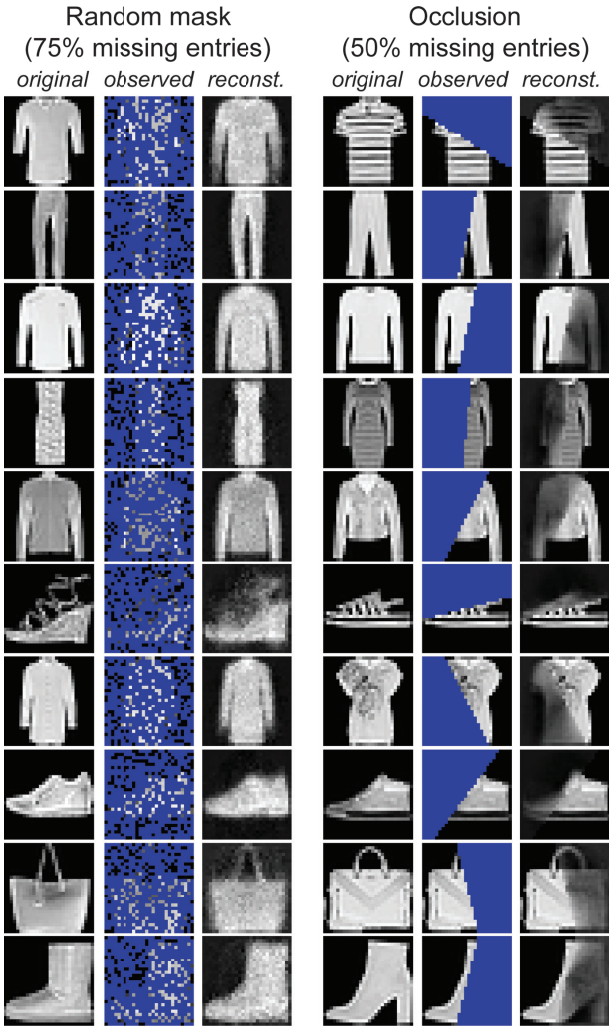

Figure 6: Reconstructions of incomplete test Fashion dataset vectors by applying our simultaneous classification and coding algorithm using the CNN4 architecture with Batch Normalization (BN).