Supplementary material: Learning from Incomplete Features by Simultaneous Training of Neural Networks and Sparse Coding

Cesar F. Caiafa[*]
IAR - CONICET, ARGENTINA
Tensor Learning Team - RIKEN AIP, JAPAN
ccaiafa@gmail.com

Ziyao Wang
School of Automation - SEU, CHINA
Tensor Learning Team - RIKEN AIP, JAPAN
zy_wang@seu.edu.cn

Jordi Solé-Casals
University of Vic (UVic-UCC), SPAIN
jordi.sole@uvic.cat

Qibin Zhao
Tensor Learning Team - RIKEN AIP, JAPAN
qibin.zhao@riken.jp

1. Additional pseudocodes

Here, additional pseudocode of the algorithms discussed in the paper are provided. Once the classifier is trained by using Algorithm 1, 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.

A sparsity-based sequential method is presented in Algorithm 3 (sequential approach), which consists on learning first the optimal dictionary $\mathbf{D}$ and sparse coefficients $\mathbf{s}_i$ compatible with the incomplete observations (dictionary learning and coding phase), followed by the training phase, where the classifier weights are tuned in order to minimize the classification error of the reconstructed input data vectors $\hat{\mathbf{x}}_i = \mathbf{D}\mathbf{s}_i$. It is noted that for the imputation stage (lines 2-12) other and more specialized dictionary learning algorithms with missing data can be applied, such as the ones proposed in [3] for high-dimensional data or [2] for color image data.

2. A condition based on RIP and sparsity

The Restricted Isometry Property (RIP): An overcomplete dictionary $\mathbf{D}$ satisfies the RIP of order K if there exists $\delta_K \in [0,1)$ s.t.

$$(1-\delta_K)\|\mathbf{s}\|_2^2 \leq \|\mathbf{D}\mathbf{s}\|_2^2 \leq (1+\delta_K)\|\mathbf{s}\|_2^2, \qquad \text{(S1)}$$

holds for all $\mathbf{s} \in \Sigma_K^P$. RIP was introduced in [1] and characterizes matrices which are nearly orthonormal when operating on sparse vectors.

In the following theorem, we show that, by imposing conditions on the sparsity level of the representation and the RIP constant of a sub-matrix of the dictionary, we can guarantee to meet the sufficient condition (6).

Algorithm 2 : Testing on incomplete data

```
Require: Incomplete data vectors $\{\mathbf{x}_i^o\}$, $i = 1, 2, \dots, I$, classifier parameters $\Theta$, dictionary $\mathbf{D}$, hyper-parameters $\lambda_1$ and $\lambda_2$, number of iterations $N_{iter}$ and update rate $\sigma_\mathbf{s}$
Ensure: $\hat{y}_i$ and reconstructions $\hat{\mathbf{x}}_i = \mathbf{D}\mathbf{s}_i, \forall i$
Sparse coding stage: for fixed dictionary $\mathbf{D}$ find sparse representations of observations $\mathbf{x}_i^o$
Initialize $\mathbf{s}_i, \forall i$ randomly
for $n \leq N_{iter}$ do
    $\Delta_i = -\sigma_\mathbf{s}[\lambda_1 \frac{\partial J_1}{\partial \mathbf{s}_i} + \lambda_2 \frac{\partial J_2}{\partial \mathbf{s}_i}], \forall i$
    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
$\hat{\mathbf{x}}_i = \mathbf{D}\mathbf{s}_i, \forall i$; Compute reconstructions
Classification stage: apply classifier to reconstructions $\hat{\mathbf{x}}_i$
$\hat{y}_i = \arg\max_y (p_\Theta^y(\hat{\mathbf{x}}_i))$
return $\Theta, \hat{y}_i, \mathbf{s}_i, \hat{\mathbf{x}}_i, \forall i$
```

Theorem 2.1. *Given a dataset $\{\mathbf{x}_i, y_i\}$, $i = 1, 2, \dots, I$ with normalized data vectors ($\|\mathbf{x}_i\| \leq 1$) admitting a K-sparse representation over a dictionary $\mathbf{D} \in \mathbb{R}^{N \times P}$ with unit-norm columns, whose sub-matrices $\mathbf{D}_i^m$ satisfy the RIP of order K with constant δ_K^i, and suppose that, we have obtained an alternative dictionary $\mathbf{D}' \in \mathbb{R}^{N \times P}$, whose sub-matrices $\mathbf{D}_i'^m$ also satisfy the RIP of order K with constant δ_K^i such that, for the incomplete observation $\mathbf{x}_i^o \in \mathbb{R}^{M_i}$, the K-sparse representation solution is non-unique, i.e. $\exists \mathbf{s}_i, \mathbf{s}_i' \in \Sigma_K^P$ such that $\mathbf{x}_i^o = \mathbf{D}_i^o \mathbf{s}_i = \mathbf{D}_i'^o \mathbf{s}_i'$, where $\mathbf{s}_i \in \mathbb{R}^P$ is the vector of coefficients of the true data, i.e. $\mathbf{x}_i = \mathbf{D}\mathbf{s}_i$ and $\mathbf{s}_i'$ provides a plausible reconstruction through $\hat{\mathbf{x}}_i = \mathbf{D}'\mathbf{s}_i'$ with $\|\hat{\mathbf{x}}_i\| \leq 1$. If a perfect classifier $\{\mathbf{w}, b\}$ of the reconstruction*

[*] Corresponding author

Algorithm 3 : Sequential sparsity based approach

```
Require: Incomplete data vectors and their labels $\{\mathbf{x}_i^o, y_i\}$, $i = 1, 2, \ldots, I$, hyper-parameters $\lambda_1$ and $\lambda_2$, number of iterations $N_{iter}$ and update rate $\sigma_\Theta$, $\sigma_\mathbf{D}$ and $\sigma_\mathbf{s}$
Ensure: Classifier weights $\Theta$ and reconstructions $\hat{\mathbf{x}}_i = \mathbf{D}\mathbf{s}_i, \forall i$
Randomly initialize $\mathbf{D}, \mathbf{s}_i, \forall i$
Imputation stage: learning of D and $\mathbf{s}_i$
for $n \leq N_{iter}$ do
    $\mathbf{D} = \mathbf{D} - \sigma_\mathbf{D} \frac{\partial J_1}{\partial \mathbf{D}}$
    Normalize columns of matrix $\mathbf{D}$
    $\Delta_i = -\sigma_\mathbf{s} \big[\lambda_1 \frac{\partial J_1}{\partial \mathbf{s}_i} + \lambda_2 \frac{\partial J_2}{\partial \mathbf{s}_i}\big], \forall i$
    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
Training stage: update $\Theta$
for $n \leq N$ do
    $\Theta = \Theta - \sigma_\Theta \frac{\partial J_0}{\partial \Theta}$;
end for
return $\Theta, \mathbf{D}, \mathbf{s}_i, \hat{\mathbf{x}}_i = \mathbf{D}\mathbf{s}_i, \forall i$
```

$\hat{\mathbf{x}}_i$ exists such that $|f(\hat{\mathbf{x}})| = |\langle \mathbf{w}, \hat{\mathbf{x}} \rangle + b| > \epsilon_i > 0$ and

$$\epsilon_i > 2\|\mathbf{w}_i^m\|_1 \sqrt{\frac{K}{1-\delta_K^i}}, \tag{S2}$$

then the full data vector $\mathbf{x}_i$ is also perfectly separated with this classifier, in other words: $f(\mathbf{x}_i) = \langle \mathbf{w}_i, \mathbf{x}_i \rangle + b > 0$ (≤ 0) if $y_i = 1$ ($y_i = 0$).

Proof. Let us prove that the sufficient condition (6) is met under the hypothesis of Theorem 2.1. Taking into account that $\mathbf{x}_i^m = \mathbf{D}_i^m \mathbf{s}_i$, we can write

$$\begin{aligned} |\langle \mathbf{w}^m, \mathbf{D}_i^m \mathbf{s}_i \rangle| &= \Big| \sum_{j=1}^{N-M_i} \mathbf{w}^m(j) \sum_{n=1}^{N} \mathbf{D}_i^m(j,n) \mathbf{s}_i(n) \Big| \\ &\leq \sum_{j=1}^{N-M_i} |\mathbf{w}^m(j)| \sum_{n=1}^{N} |\mathbf{D}_i^m(j,n)| |\mathbf{s}_i(n)|. \end{aligned} \tag{S3}$$

Since we assumed normalized vectors $\|\mathbf{x}_i\| \leq 1$, by applying the left-hand side of the RIP we obtain: $\|\mathbf{s}_i\| \leq 1/\sqrt{1-\delta_K^i}$, and, taking into account that $\|\mathbf{s}_i\|_1 \leq \sqrt{K}\|\mathbf{s}_i\|$ and $|\mathbf{D}_i^m(j,n)| \leq 1$ (columns of $\mathbf{D}$ are unit-norm), we obtain:

$$|\langle \mathbf{w}^m, \mathbf{D}_i^m \mathbf{s}_i \rangle| \leq \sqrt{\frac{K}{1-\delta_K^i}} \sum_{j=1}^{N-M_i} |\mathbf{w}^m(j)| = \sqrt{\frac{K}{1-\delta_K^i}} \|\mathbf{w}_i^m\|_1. \tag{S4}$$

Similarly, using $\hat{\mathbf{x}_i}^m = \mathbf{D}_i'^m \mathbf{s}_i'$, we can obtain that

$$|\langle \mathbf{w}^m, \mathbf{D}_i'^m \mathbf{s}_i' \rangle| \leq \sqrt{\frac{K}{1-\delta_K^i}} \|\mathbf{w}_i^m\|_1. \tag{S5}$$

Putting equations (S4) and (S5) together with equation (S2) complete the proof of the sufficient condition (6). □

Table S1. Experimental settings for MNIST and CIFAR10 datasets: Number of iterations N_{iter}, batch size bs, learning rate σ_Θ, momentum m, update rate σ (training and test)

Dataset	**Classifier**	N_{iter}	bs	σ_Θ	m	σ (train)	σ (test)
MNIST	**Log. Reg.**	3000	250	0.1	0.5	1.0	5.0
	CNN4	3500	250	05	0.5	0.4	0.5
CIFAR10	**Resnet18**	1000	64	0.01	0.5	1.0	2.5

Table S2. Hyper-parameter tuning: crossvalidated hyperparameters λ_1 and λ_2 obtained for MNIST and CIFAR10 datasets with the classifiers used in our experiments.

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**	0.024	0.008	0.032	0.004	0.032	0.01	-	-

3. Experimental results

3.1. Implementation details

We implemented all the algorithms in Pytorch 1.0.0 on a single GPU. The code is available at[1].

Initializations of dictionary $\mathbf{D}$ and coefficients $\mathbf{s}_i$ were made at random. However, we think some improvements in convergence could be achieved by using some dedicated dictionaries such as the case of Wavelet or Cosine Transform matrices.

To update NN weights (Θ), we used standard Stochastic Gradient Descent (SGD) with learning rate σ_Θ and momentum m, while for updating dictionary $\mathbf{D}$ and vector coefficients $\mathbf{s}_i$, we used fixed update rate $\sigma = \sigma_\mathbf{D} = \sigma_\mathbf{s}$. It is noted that we used different update rates for training and testing stages. In Table S1, we report the settings used for experiments for MNIST and CIFAR10 datasets, which includes Number of iterations N_{iter}, batch size bs, learning rate σ_Θ, momentum m, update rate σ (training and test).

3.2. Hyperparameter tuning

In Table S2 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 a validation data set. Figure S1 shows the validation accuracy obtained with different classifiers and levels of missing entries for MNIST dataset.

3.3. Additional visual results

To visually evaluate our results, additional randomly selected examples of original (complete) images of the test dataset in MNIST and Fashion, together with their given incomplete observations and obtained reconstructions, are shown in Figure S2 and Figure S3

[1] https://github.com/ccaiafa/SimultCodClass.

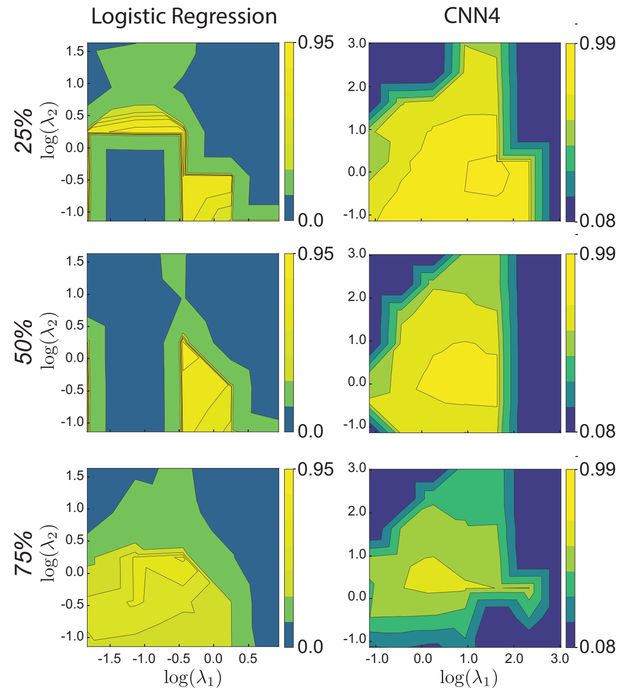

Figure S1. Test accuracy in the grid search for hyper-parameter tuning in MNIST dataset: λ_1 and λ_2 were tuned by cross-validation for various levels of missing entries: 25%, 50% and 75%.

References

[1] Emmanuel J. Candès and Terrence Tao. Decoding by Linear Programming. *IEEE Transactions on Information Theory*, 51(12):4203–4215, 2005.

[2] J Mairal, M Elad, and G Sapiro. Sparse Representation for Color Image Restoration. *IEEE Transactions on Image Processing*, 17(1):53–69, Jan. 2008.

[3] Valeriya Naumova and Karin Schnass. Dictionary learning from incomplete data for efficient image restoration. *EUSIPCO*, 2017.

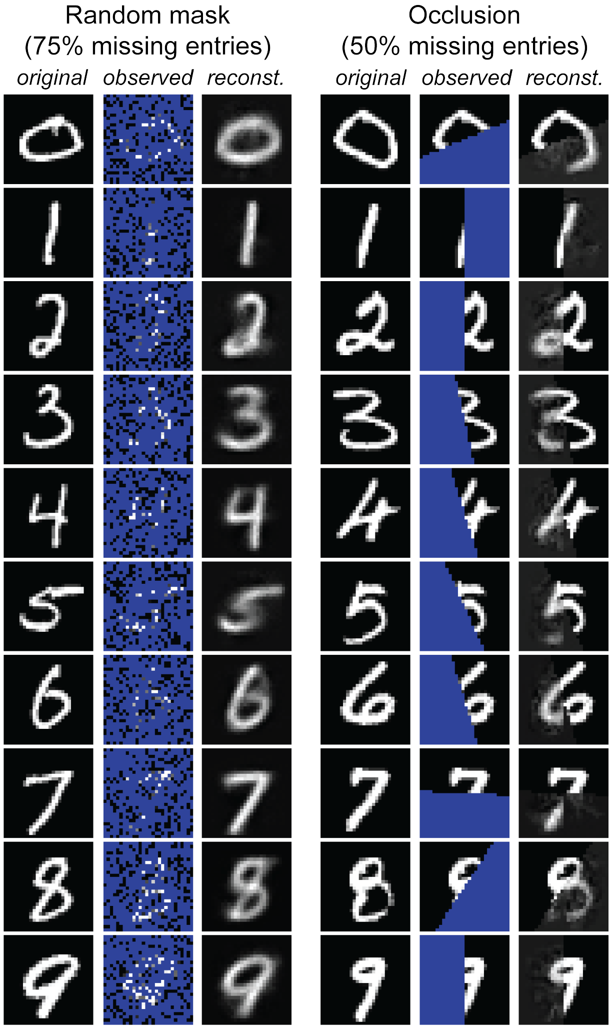

Figure S2. Reconstructions of incomplete test MNIST dataset images by applying our simultaneous classification and coding algorithm with the CNN4 architecture.

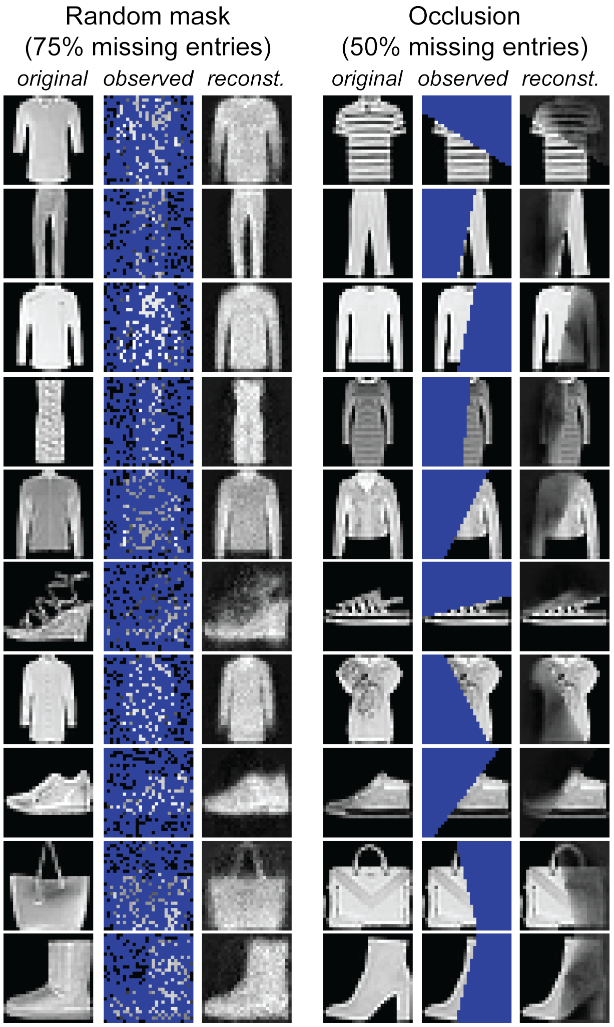

Figure S3. Reconstructions of incomplete test Fashion dataset images by applying our simultaneous classification and coding algorithm with the CNN4 architecture with Batch Normalization (BN).