EE-556 Homework 3

Edoardo Debenedetti

December 12, 2019

1 Multiclass classification

1.1 Theory

1.1.1 Multinomial logistic regression estimator

Proof. Assuming that we can write $\mathbb{P}(b_i = j|\mathbf{a}_i, \mathbf{X})$ as

$$\mathbb{P}(b_i = j|\mathbf{a}_i, \mathbf{X}) = \frac{e^{\mathbf{a}_i^T \mathbf{x}_j}}{\sum_{k=1}^{C} e^{\mathbf{a}_i^T \mathbf{x}_k}} \tag{1}$$

And that all the samples in the matrix $\mathbf{A}$ are i.i.d., we can write $\mathbb{P}(\mathbf{b}|\mathbf{A}, \mathbf{X})$ as

$$\mathbb{P}(\mathbf{b}|\mathbf{A}, \mathbf{X}) = \prod_{i=1}^{n} \frac{e^{\mathbf{a}_i^T \mathbf{x}_{b_i}}}{\sum_{k=1}^{C} e^{\mathbf{a}_i^T \mathbf{x}_k}} \tag{2}$$

Then, we can write $\hat{\mathbf{X}}_{ML}$ as

$$\hat{\mathbf{X}}_{ML} \in \arg\max_{\mathbf{X}} \left\{ \prod_{i=1}^{n} \frac{e^{\mathbf{a}_i^T \mathbf{x}_{b_i}}}{\sum_{k=1}^{C} e^{\mathbf{a}_i^T \mathbf{x}_k}} \right\} \tag{3}$$

However, as we are interested in taking the $\arg\max_{\mathbf{X}}$, we can take the logarithm of the argu-

ment, since log is strictly increasing and does not change the $\arg\max_{\mathbf{X}}$:

$$\arg\max_{\mathbf{X}} \left\{ \prod_{i=1}^{n} \frac{e^{\mathbf{a}_i^T \mathbf{x}_{b_i}}}{\sum_{k=1}^{C} e^{\mathbf{a}_i^T \mathbf{x}_k}} \right\} = \arg\max_{\mathbf{X}} \left\{ \log \prod_{i=1}^{n} \frac{e^{\mathbf{a}_i^T \mathbf{x}_{b_i}}}{\sum_{k=1}^{C} e^{\mathbf{a}_i^T \mathbf{x}_k}} \right\} = \tag{4}$$

$$= \arg\max_{\mathbf{X}} \left\{ \sum_{i=1}^{n} \log \frac{e^{\mathbf{a}_i^T \mathbf{x}_{b_i}}}{\sum_{k=1}^{C} e^{\mathbf{a}_i^T \mathbf{x}_k}} \right\} = \tag{5}$$

$$= \arg\max_{\mathbf{X}} \left\{ \sum_{i=1}^{n} \left(\log e^{\mathbf{a}_i^T \mathbf{x}_{b_i}} - \log \sum_{k=1}^{C} e^{\mathbf{a}_i^T \mathbf{x}_k} \right) \right\} = \tag{6}$$

$$= \arg\max_{\mathbf{X}} \left\{ \sum_{i=1}^{n} \left(\mathbf{a}_i^T \mathbf{x}_{b_i} - \log \sum_{k=1}^{C} e^{\mathbf{a}_i^T \mathbf{x}_k} \right) \right\} \tag{7}$$

Between (4) and (5) we exploited the fact that the log of the products is equal to the sum of the logs, and between (5) and (6) we exploited the fact that the log of a ratio is the difference of the logs. We can now take the negative of the argument of $\arg\max_{\mathbf{X}}$ and transform $\arg\max_{\mathbf{X}}$ in $\arg\min_{\mathbf{X}}$:

$$\arg\max_{\mathbf{X}} \left\{ \sum_{i=1}^{n} \left(\mathbf{a}_i^T \mathbf{x}_{b_i} - \log \sum_{k=1}^{C} e^{\mathbf{a}_i^T \mathbf{x}_k} \right) \right\} = \tag{8}$$

$$= \arg\min_{\mathbf{X}} \left\{ \sum_{i=1}^{n} \left(\log \sum_{k=1}^{C} e^{\mathbf{a}_i^T \mathbf{x}_k} - \mathbf{a}_i^T \mathbf{x}_{b_i} \right) \right\} \tag{9}$$

Hence:

$$\hat{\mathbf{X}}_{ML} \in \arg\min \left\{ f(\mathbf{X}) | f : \mathbb{R}^{d \times C} \to \mathbb{R}, f(\mathbf{X}) = \sum_{i=1}^{n} \left(\log \sum_{k=1}^{C} e^{\mathbf{a}_i^T \mathbf{x}_k} - \mathbf{a}_i^T \mathbf{x}_{b_i} \right), b_i \in \{1, 2, ...C\} \right\} \tag{10}$$

Which is the form for $\hat{\mathbf{X}}_{ML}$ we had to prove. □

1.1.2 Multinomial logistic regression estimator gradient

Proof. Since both sum and subtraction are linear, we can re-write $f(x)$ defined in (10) as

$$f(\mathbf{X}) = \sum_{i=1}^{n} \log \sum_{k=1}^{C} e^{\mathbf{a}_i^T \mathbf{x}_k} - \sum_{i=1}^{n} \mathbf{a}_i^T \mathbf{x}_{b_i} \tag{11}$$

First, let us consider the left part of $f(\mathbf{X})$, that we define as $f_1(\mathbf{X}) = \sum_{i=1}^{n} \log \sum_{k=1}^{C} e^{\mathbf{a}_i^T \mathbf{x}_k}$: we

can then look for the gradient with respect to the j-th column $\mathbf{X}_j$ of $\mathbf{X}$:

$$\nabla_{\mathbf{X}_j} f_1(\mathbf{X}) = \nabla_{\mathbf{X}_j} \sum_{i=1}^{n} \log \sum_{k=1}^{C} e^{\mathbf{a}_i^T \mathbf{x}_k} = \tag{12}$$

$$= \sum_{i=1}^{n} \nabla_{\mathbf{X}_j} \log \sum_{k=1}^{C} e^{\mathbf{a}_i^T \mathbf{x}_k} = \tag{13}$$

$$= \sum_{i=1}^{n} \frac{1}{\sum_{k=1}^{C} e^{\mathbf{a}_i^T \mathbf{x}_k}} \nabla_{\mathbf{X}_j} \sum_{k=1}^{C} e^{\mathbf{a}_i^T \mathbf{x}_k} \tag{14}$$

Then, the gradient of the sum argument is equal to:

$$\nabla_{\mathbf{X}_j} e^{\mathbf{a}_i^T \mathbf{x}_k} = \begin{cases} \mathbf{a}_i^T e^{\mathbf{a}_i^T \mathbf{x}_k} & k = j \\ 0 & k \neq j \end{cases} \tag{15}$$

Hence, the gradient of the sum becomes

$$\nabla_{\mathbf{X}_j} \sum_{k=1}^{C} e^{\mathbf{a}_i^T \mathbf{x}_k} = \mathbf{a}_i^T e^{\mathbf{a}_i^T \mathbf{X}_j} \tag{16}$$

And (14) can be expressed as

$$\nabla_{\mathbf{X}_j} f_1(\mathbf{X}) = \sum_{i=1}^{n} \frac{1}{\sum_{k=1}^{C} e^{\mathbf{a}_i^T \mathbf{x}_k}} \mathbf{a}_i^T e^{\mathbf{a}_i^T \mathbf{X}_j} \tag{17}$$

Moreover, we know that the matrix $e^{\mathbf{AX}}$ is the matrix whose ij-th element is given by $e^{\mathbf{A}_i \mathbf{X}_j} = e^{\mathbf{a}_i^T \mathbf{X}_j}$, that is exactly the leftmost exponent in (17). Plus, if we multiply it by $\mathbf{Z}_i$, as it has been defined in the homework, we get as result:

$$\mathbf{Z}_i e^{\mathbf{A}_i \mathbf{X}_j} = \sum_{i=1}^{n} \frac{1}{\sum_{k=1}^{C} e^{\mathbf{a}_i^T \mathbf{x}_k}} e^{\mathbf{a}_i^T \mathbf{X}_j} \tag{18}$$

If we multiply this result by $\mathbf{A}_j^T$, we then get

$$\mathbf{A}_j^T \mathbf{Z}_i e^{\mathbf{A}_i \mathbf{X}_j} = \sum_{i=1}^{n} \frac{1}{\sum_{k=1}^{C} e^{\mathbf{a}_i^T \mathbf{x}_k}} \mathbf{a}_i^T e^{\mathbf{a}_i^T \mathbf{X}_j} = \nabla_{\mathbf{X}_j} f_1(\mathbf{X}) \tag{19}$$

Expanding the result of the gradient with respect to $\mathbf{X}_j$ to every column, as a result we get that

$$\nabla_{\mathbf{X}} f_1(\mathbf{X}) = \mathbf{A}^T \mathbf{Z} \exp(\mathbf{AX}) \tag{20}$$

Now, let us take in consideration the right part of $f(\mathbf{X})$, that we define as $f_2(\mathbf{X}) = \sum_{i=1}^{n} \mathbf{a}_i^T \mathbf{x}_{b_i}$. We can take the gradient with respect of the j-th column of $\mathbf{X}$:

$$\nabla_{\mathbf{X}_j} f_2(\mathbf{X}) = \nabla_{\mathbf{X}_j} \sum_{i=1}^{n} \mathbf{a}_i^T \mathbf{x}_{b_i} = \tag{21}$$

$$\sum_{i=1}^{n} \nabla_{\mathbf{X}_j} \mathbf{a}_i^T \mathbf{x}_{b_i} \tag{22}$$

We could exchange the sum and the gradient as $\mathbf{X}_j$ does not depend in i. The gradient in (22) is equal to

$$\nabla_{\mathbf{X}_j} \mathbf{a}_i^T \mathbf{x}_{b_i} = \begin{cases} 0 & \mathbf{X}_j \neq x_{b_i} \rightarrow j \neq b_i \\ \mathbf{a}_i^T & \mathbf{X}_j = x_{b_i} \rightarrow j = b_i \end{cases} \tag{23}$$

Summing up along i, we get that the j-th column of $\nabla_{\mathbf{X}_j} f_2(\mathbf{X})$ is given by the sum of all the $\mathbf{a}_i^T$ whose class $b_i = j$. Hence, $\nabla_{\mathbf{X}} f_2(\mathbf{X})$ is the matrix whose $j-th$ column is given by $\nabla_{\mathbf{X}_j} f_2(\mathbf{X})$. It turns out that this matrix can be expressed by the matrix $\mathbf{A}^T\mathbf{Y}$, where $\mathbf{Y}$ is the matrix containing one-hot-encodings. In fact, multiplying $\mathbf{A}^T$ by $\mathbf{Y}$ gives us a matrix whose ij-th element is given by $\mathbf{A}_i^T \mathbf{Y}_j$, where $\mathbf{A}_i^T$ i the vector given by the i-th components of all the each row vector of $\mathbf{A}$, and $\mathbf{Y}_j$ is the vector that is 1 in the indices corresponding to the elements of $\mathbf{A}$ whose class corresponds to j. This inner product, then, sums up all the i-th components of the datapoints belonging to class j. Hence,

$$\nabla_{\mathbf{X}} f_2(\mathbf{X}) = \mathbf{A}^T\mathbf{Y} \tag{24}$$

Combining the results obtained for $\nabla_{\mathbf{X}} f_1(\mathbf{X})$ in (20) and $\nabla_{\mathbf{X}} f_2(\mathbf{X})$ in (24), we get

$$\nabla_{\mathbf{X}} f(\mathbf{X}) = \nabla_{\mathbf{X}} f_1(\mathbf{X}) - \nabla_{\mathbf{X}} f_2(\mathbf{X}) = \tag{25}$$

$$\mathbf{A}^T\mathbf{Z} \exp(\mathbf{A}\mathbf{X}) - \mathbf{A}^T\mathbf{Y} = \tag{26}$$

$$\mathbf{A}^T(\mathbf{Z} \exp(\mathbf{A}\mathbf{X}) - \mathbf{Y}) \tag{27}$$

□

1.1.4 Lipschitz constant

(a) Maximum eigenvalue $\lambda_{max}(\mathbf{a}_i\mathbf{a}_i^T)$

Proof. Given $\mathbf{a}_i \in \mathbb{R}^n$, as $\mathbf{a}_i\mathbf{a}_i^T$ is rank-1, $n-1$ eigenvalues of $\mathbf{a}_i\mathbf{a}_i^T$ are 0, and only 1 is non-zero. This means that, as the sum of the eigenvalues of a matrix is given by its trace,

$$\lambda_{max}(\mathbf{a}_i\mathbf{a}_i^T) = Tr(\mathbf{a}_i\mathbf{a}_i^T) \tag{28}$$

Moreover, since $\mathbf{a}_i\mathbf{a}_i^T$ is the matrix whose k,l-th element is given by $a_{i,k}a_{i,l}$, its k-th diagonal element is given by $a_{i,k}a_{i,k} = a_{i,k}^2$, and then

$$Tr(\mathbf{a}_i\mathbf{a}_i^T) = \sum_{k=1}^{n} a_{i,k}^2 = \|\mathbf{a}_i\|^2 \tag{29}$$

Finally, combining (28) and (29), we get

$$\lambda_{max}(\mathbf{a}_i\mathbf{a}_i^T) = \|\mathbf{a}_i\|^2 \tag{30}$$

□

(b) Lipschitz constant of the gradient

Proof. Given $\nabla^2 f(\mathbf{X}) = \sum_{i=1}^{n} \mathbf{\Sigma}_i \otimes \mathbf{a}_i\mathbf{a}_i^T$ and the fact that we can choose $L = \frac{\|A\|_F^2}{2}$ if

$$\lambda_{max}(\nabla^2 f(\mathbf{X})) \leq \frac{\|A\|_F^2}{2} < \infty \tag{31}$$

We should then look for an upper-bound for $\lambda_{max}(\nabla^2 f(\mathbf{X}))$. We proved that $\lambda_{max}(\mathbf{a}_i\mathbf{a}_i^T) = \|\mathbf{a}_i\|^2$. Now, let us look for the maximum eigenvalue of $\mathbf{\Sigma}_i$. We also know from exercise 1.1.2 that both $\mathbf{\Sigma}_i \succeq 0$ and $\mathbf{a}_i\mathbf{a}_i^T \succeq 0$, $\forall i \in [1, n]$. Then, as $\mathbf{\Sigma}_i \succeq 0$, its $\lambda_{max}(\mathbf{\Sigma}_i) \leq \max_k \sum_l |\mathbf{\Sigma}_{i,kl}|$. If them we take the ℓ-1 norm of each row of $\mathbf{\Sigma}_i$, we get

$$\|\mathbf{\Sigma}_{ij}\|_1 = |\sigma_{ij}(1-\sigma_{ij})| + \sum_{k=1,k\neq j}^{C} |-\sigma_{ij}\sigma_{ik}| \tag{32}$$

Moreover, as each σ is a probability, $0 \leq \sigma_{ij} \leq 1, \forall i, j$. Then:

- The product between two σ is positive
- $1 - \sigma_{ij} \geq 0, \forall i, j$

Then we can remove the absolute values and the minus inside the sum from (32) and rewrite it as

$$\|\mathbf{\Sigma}_{ij}\|_1 = \sigma_{ij}(1-\sigma_{ij}) + \sigma_{ij} \sum_{k=1,k\neq j}^{C} \sigma_{ik} \tag{33}$$

It is worth noting that σ_{ij} has been taken outside the sum as it does not depend on k. Moreover, we can write $\sum_{k=1,k\neq j}^{C} \sigma_{ik}$ as $1 - \sigma_{ij}$. Then (33) becomes

$$\|\mathbf{\Sigma}_{ij}\|_1 = \sigma_{ij}(1-\sigma_{ij}) + \sigma_{ij}(1-\sigma_{ij}) = 2\sigma_{ij}(1-\sigma_{ij}) = 2(\sigma_{ij} - \sigma_{ij}^2) \tag{34}$$

We are now interested in the maximum $\|\mathbf{\Sigma}_{ij}\|_1$ possible. Since σ_{ij} is concave, and so is $-\sigma_{ij}^2$, and since $\sigma_{ij} + (-\sigma_{ij}^2)$ is the sum of two concave functions, it is concave as well. We can then look for a local maximum that will be also a global one. We can than take the derivative of $\|\mathbf{\Sigma}_{ij}\|_1$ as a function of σ_{ij} to 0 and look for the maximizing σ_{ij}:

$$\frac{d}{d\sigma_{ij}}(\sigma_{ij} - \sigma_{ij}^2) = 1 - 2\sigma_{ij} \tag{35}$$

Setting the derivative to 0 gives us

$$\sigma_{ij} = \frac{1}{2} \tag{36}$$

Plugging it into (34) we obtain

$$2(\sigma_{ij} - \sigma_{ij}^2)|_{\sigma_{ij}=\frac{1}{2}} = \frac{1}{2} \tag{37}$$

Then we get that

$$\lambda_{max}(\mathbf{\Sigma}_i) \leq \max_k \sum_l |\mathbf{\Sigma}_{i,kl}| \leq \frac{1}{2} \tag{38}$$

Hence,

$$\lambda_{max}(\mathbf{\Sigma}_i) \leq \frac{1}{2} \tag{39}$$

Recalling that both $\mathbf{\Sigma}_i \succeq 0$ and $\mathbf{a}_i\mathbf{a}_i^T \succeq 0$, $\forall i \in [1, n]$, then

$$\lambda_{max}(\mathbf{\Sigma}_i \otimes \mathbf{a}_i\mathbf{a}_i^T) = \lambda_{max}(\mathbf{\Sigma}_i)\lambda_{max}(\mathbf{a}_i\mathbf{a}_i^T) \tag{40}$$

Using the upper-bounds we have found above, we can then state that

$$\lambda_{max}(\mathbf{\Sigma}_i \otimes \mathbf{a}_i\mathbf{a}_i^T) \leq \frac{1}{2}\|\mathbf{a}_i\|_2^2 \tag{41}$$

Moreover, $\mathbf{\Sigma}_i$ is symmetric by construction, and as it has only real entries, it is Hermitian. Due to the structure described above, also $\mathbf{a}_i\mathbf{a}_i^T$ is symmetric and real, then it is Hermitian. Moreover, since $A^T \otimes B^T = (A \otimes B)^T$, if $A = A^T$ and $B = B^T$, then $(A \otimes B) = (A^T \otimes B^T) = (A \otimes B)^T$. Then, in the specific case where $A = \mathbf{\Sigma}_i$ and $B = \mathbf{a}_i\mathbf{a}_i^T$, $(\mathbf{\Sigma}_i \otimes \mathbf{a}_i\mathbf{a}_i^T) = (\mathbf{\Sigma}_i \otimes \mathbf{a}_i\mathbf{a}_i^T)^T$. As also $\mathbf{\Sigma}_i \otimes \mathbf{a}_i\mathbf{a}_i^T$ has only real entries, it is Hermitian.

Finally, because of Weyl's inequality, we know that $\lambda_{max}(A + B) \leq \lambda_{max}(A) + \lambda_{max}(B)$ if A and B are both Hermitian. Then,

$$\lambda_{max}\left(\sum_{i=1}^n \mathbf{\Sigma}_i \otimes \mathbf{a}_i\mathbf{a}_i^T\right) \leq \sum_{i=1}^n \lambda_{max}(\mathbf{\Sigma}_i \otimes \mathbf{a}_i\mathbf{a}_i^T) \leq \sum_{i=1}^n \frac{1}{2}\|\mathbf{a}_i\|_2^2 = \frac{\|\mathbf{A}\|_F^2}{2} \tag{42}$$

That is what we were looking for in (31). □

1.1.5 ℓ-1 Proximal operator

Proof. Given $g(\mathbf{x}) := \|\mathbf{x}\|_1$,

$$prox_{\lambda g}(\mathbf{z}) = \arg\min_{\mathbf{y}} \{\lambda\|\mathbf{y}\|_1 + \frac{1}{2}\|\mathbf{y}-\mathbf{z}\|_2^2\} \tag{43}$$

As both the functions inside the arg min are convex with respect to $\mathbf{y}$, their sum is convex as well. We can then find the minimizer $\mathbf{y}_{min}$ by taking its gradient with respect to $\mathbf{y}$ to 0.

If we assume $z \in \mathbb{R}$, then also $y \in \mathbb{R}$, and the argument of arg min becomes $\lambda|y| + \frac{1}{2}(y-z)^2$ and we can find the derivative with respect to y:

$$\frac{\mathrm{d}}{\mathrm{dy}}[\lambda|y| + \frac{1}{2}(y-z)^2] = \tag{44}$$

$$= \lambda sign(y) + y - z = \tag{45}$$

$$= \begin{cases} -\lambda + y - z, & y < 0 \\ \lambda + y - z, & y > 0 \end{cases} \tag{46}$$

Equating it to zero, we obtain:

$$y_{min} = \begin{cases} \lambda + z, & y < 0, z < -\lambda \\ z - \lambda, & y > 0, z > \lambda \end{cases} \tag{47}$$

We can see that y_{min} is only defined in the zone where $|z| > \lambda$ (since $\lambda > 0$), as $|y|$ is not differentiable in $y = 0$, and then its derivative is not defined there. However, in order to find the prox operator, we can also use the subdifferential, and we can choose as subgradient 0 and then (47) becomes

$$y_{min} = \begin{cases} \lambda + z, & z < -\lambda \\ z - \lambda, & z > \lambda \\ 0, & |z| < \lambda \end{cases} \tag{48}$$

However, we can rewrite (48) as

$$y_{min} = \begin{cases} \max(z - \lambda, 0), & z \geq 0 \\ \max(-z - \lambda, 0), & z < 0 \end{cases} \tag{49}$$

Which can be compressed into the form

$$y_{min} = \max(|z| - \lambda, 0) \cdot sign(z) \tag{50}$$

Since $\lambda\|\mathbf{y}\|_1 + \frac{1}{2}\|\mathbf{y}-\mathbf{z}\|_2^2 = \sum_{i=1}^{d} |y_i| + \frac{1}{2}(y_i - z_i)^2$, each $y_{i,min}$ is given by $y_{y,min} = \max(|z_i| - \lambda, 0) \cdot sign(z_i)$ and we can then derive the vector $\mathbf{y}$ by applying the operators coordinate-wise, which results in

$$\mathbf{y}_{min} = \max(|\mathbf{z}| - \lambda, 0) \odot sign(\mathbf{z}) \tag{51}$$

Finally,

$$prox_{\lambda g}(\mathbf{z}) = \mathbf{y}_{min} = \max(|\mathbf{z}| - \lambda, 0) \odot sign(\mathbf{z}) \tag{52}$$

□

1.1.6 ℓ-2 Proximal operator

Proof. Given $g(\mathbf{x}) := \frac{1}{2}\|\mathbf{x}\|_2^2$,

$$prox_{\lambda g}(\mathbf{z}) = \arg\min_{\mathbf{y}} \{\frac{\lambda}{2}\|\mathbf{y}\|_2^2 + \frac{1}{2}\|\mathbf{y}-\mathbf{z}\|_2^2\} \tag{53}$$

As, again, both the functions inside the arg min are convex with respect to $\mathbf{y}$, their sum is convex. We can then find the minimizer $\mathbf{y}_{min}$ by taking its gradient with respect to $\mathbf{y}$ to 0:

$$\nabla\{\frac{\lambda}{2}\|\mathbf{y}\|_2^2 + \frac{1}{2}\|\mathbf{y}-\mathbf{z}\|_2^2\} = \lambda\mathbf{y} + \mathbf{y} - \mathbf{z} = \mathbf{y}(1+\lambda) - \mathbf{z} \tag{54}$$

Taking it to zero we get

$$\mathbf{y}_{min}(1+\lambda) = \mathbf{z} \tag{55}$$

$$\mathbf{y}_{min} = \frac{\mathbf{z}}{1+\lambda} \tag{56}$$

Which leads to

$$prox_{\lambda g}(\mathbf{z}) = \frac{\mathbf{z}}{1+\lambda} \tag{57}$$

□

1.2 Handwritten digit classification

1.2.2 Algorithms implementation and convergence

ℓ1 regularization We can see the result of 2000 iterations of ISTA, FISTA and FISTA with gradient scheme restart of ℓ1 regularized logistic regression in figure 1. Their performance is comparable to the theoretical sublinear bound, but worse than the results achieved with the synthetic data shown in Lecture 8. As expected, FISTA restart is the fastest to converge, followed by FISTA. ISTA is the slowest.

Another expected result that has been achieved is the fact that stochastic proximal gradient methods is faster than ISTA, as its complexity per iteration is $\frac{1}{N}$ with respect to ISTA, where

N is the size of the dataset. The final test-set accuracy obtained with the stochastic proximal gradient method is 79.37%, which is lower than the one obtained with FISTA with gradients scheme with restart (89.21%). However, we should keep in mind that the result obtained with PROX-SG is given by 1000 epochs, which have a computational complexity equivalent to 1000 FISTA RESTART iterations.

Visualizing the solution in figure 2, we can see some similarity just for the classes "0" and "3".

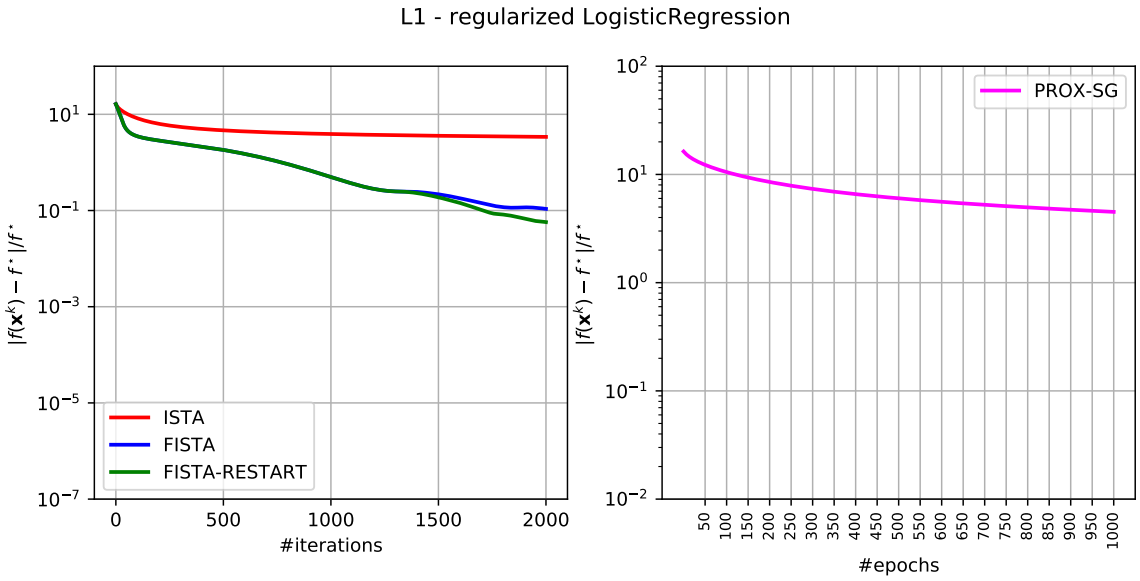

Figure 1: Convergence of ℓ-1 regularized Logistic Regression

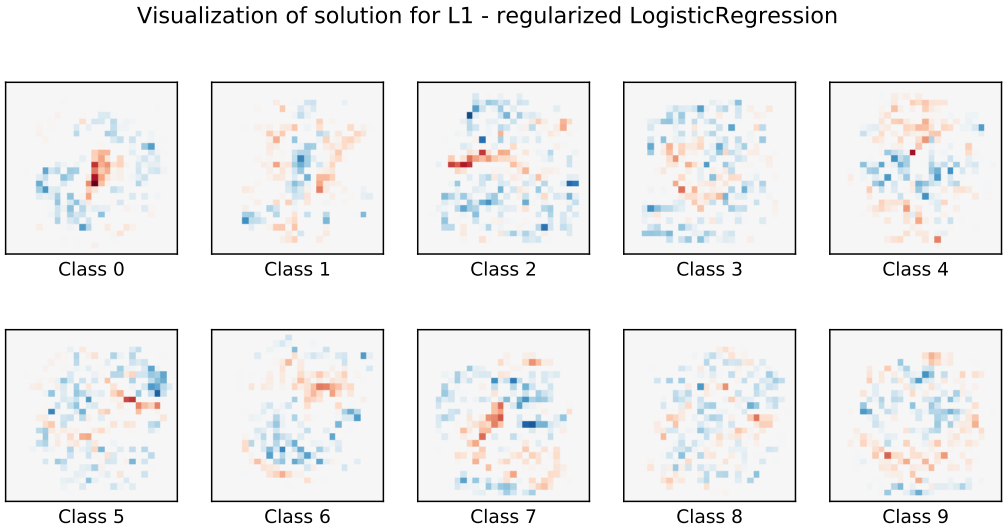

Figure 2: Solution of ℓ-1 regularized Logistic Regression

$\ell 2$ **regularization** With $\ell 2$ regularization we achieve better results both in terms of convergence and solution visualization. As a matter of fact, FISTA with gradient restart scheme condition converges with a linear rate (better than the theoretical bound), and FISTA and ISTA methods converge slightly faster than the corresponding $\ell 1$ regularized methods, even though their rates are still sublinear.

In this case, PROX-SG has the same behavior as above, converging faster than ISTA, and with a lower accuracy than FISTA-RESTART (85.01% vs 89.89% respectively). Also in this case this can be due to the fact PROX-SG result is obtained with a less computationally complex operation.

Moreover, in figure 4 the classes are slightly more identifiable than those in figure 2.

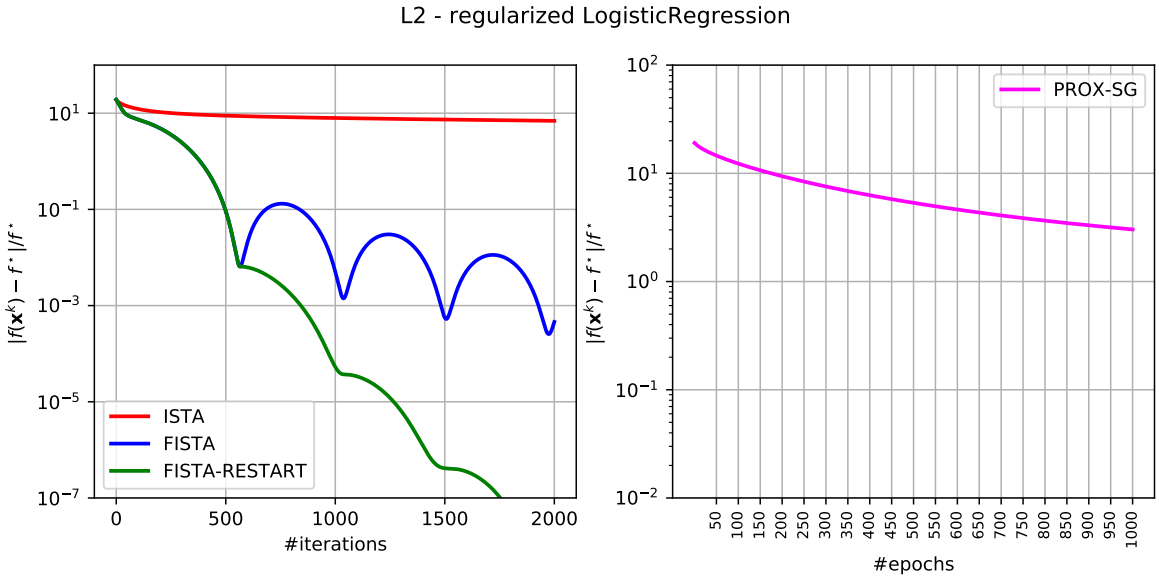

Figure 3: Convergence of ℓ-2 regularized Logistic Regression

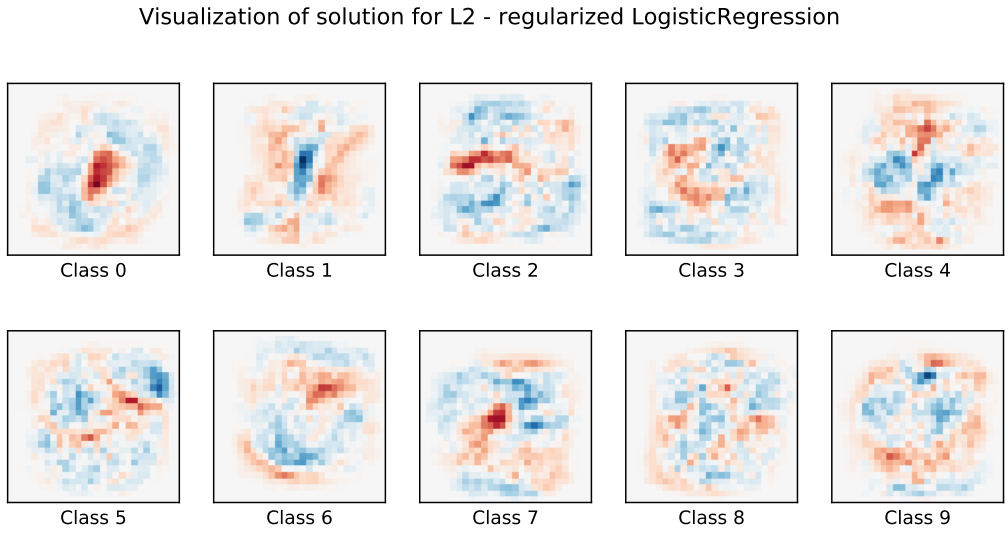

Figure 4: Solution of ℓ-2 regularized Logistic Regression

1.2.3 Logistic Regression and Neural Network comparison

The best performing method is the Neural Network that achieves a 94.7% test-set accuracy. It is followed by $\ell 2$ regularized Logistic Regression, with an 89.89% test-set accuracy, and by $\ell 1$, that achieves an 89.21% test-set accuracy. Even though the difference between the latter two methods is small, the performance achieved by the Neural Network is significantly higher ($\sim 5\%$

more). This result is thanks to the non-linearity of the Neural Network, that makes it more expressive than a generalized linear model, such as Logistic Regression. As a matter of fact, a non-linear function and non-linear feature processing (like the ones used in a Neural Network) allows for better estimation of non-linear bounds between classes, like those encountered in image classification.

2 Image reconstruction

2.1 Properties of TV and ℓ-1 in-painting

2.1.1 Gradients of TV and ℓ-1 in-painting

Given $f(\boldsymbol{\alpha}) = \frac{1}{2}\|\mathbf{b} - \mathbf{P_\Omega W^T}\boldsymbol{\alpha}\|_2^2$,

$$\nabla_{\boldsymbol{\alpha}} f(\boldsymbol{\alpha}) = -\mathbf{W}\mathbf{P}_\Omega^T(\mathbf{b} - \mathbf{P}_\Omega\mathbf{W}^T\boldsymbol{\alpha}) \tag{58}$$

Instead, given $f(\mathbf{x}) = \frac{1}{2}\|\mathbf{b} - \mathbf{P}_\Omega^T\mathbf{x}\|_2^2$,

$$\nabla_{\boldsymbol{x}} f(\boldsymbol{x}) = -\mathbf{P}_\Omega^T(\mathbf{b} - \mathbf{P}_\Omega\mathbf{x}) \tag{59}$$

2.1.2 Lipschitz constants of TV and ℓ-1 in-painting

ℓ-1 in-painting Recalling that $f(\mathbf{x})$ is L-Lipschitz continuous iff $\|f(\mathbf{x}) - f(\mathbf{x})\| \leq L\|\mathbf{x} - \mathbf{y}\|$, we can then check $\|\nabla f(\boldsymbol{\alpha}_1) - \nabla f(\boldsymbol{\alpha}_2)\|$:

$$\|\nabla f(\boldsymbol{\alpha}_1) - \nabla f(\boldsymbol{\alpha}_2)\|_2 = \|-\mathbf{W}\mathbf{P}_\Omega^T(\mathbf{b} - \mathbf{P}_\Omega\mathbf{W}^T\boldsymbol{\alpha}_1) + \mathbf{W}\mathbf{P}_\Omega^T(\mathbf{b} - \mathbf{P}_\Omega\mathbf{W}^T\boldsymbol{\alpha}_2)\|_2 = \tag{60}$$

$$= \|\mathbf{W}\mathbf{P}_\Omega^T\mathbf{P}_\Omega\mathbf{W}^T\boldsymbol{\alpha}_1 - \mathbf{W}\mathbf{P}_\Omega^T\mathbf{P}_\Omega\mathbf{W}^T\boldsymbol{\alpha}_2\|_2 = \tag{61}$$

$$\|\mathbf{W}\mathbf{P}_\Omega^T\mathbf{P}_\Omega\mathbf{W}^T(\boldsymbol{\alpha}_1 - \boldsymbol{\alpha}_2)\|_2 \leq \|\mathbf{W}\mathbf{P}_\Omega^T\mathbf{P}_\Omega\mathbf{W}^T\|_{2\to2}\|\boldsymbol{\alpha}_1 - \boldsymbol{\alpha}_2\|_2 \tag{62}$$

Where we used Cauchy-Schwarz inequality, and $\|\cdot\|_{2\to2}$ is the ℓ-2 to ℓ-2 operator norm.

Then,

$$L = \|\mathbf{W}\mathbf{P}_\Omega^T\mathbf{P}_\Omega\mathbf{W}^T\|_{2\to2} = \tag{63}$$

$$= \|\mathbf{P}_\Omega^T\mathbf{P}_\Omega\|_{2\to2} = \|\mathbf{P}_\Omega^T\mathbf{P}_\Omega\| \tag{64}$$

Where between (63) and (64) we exploited the fact that $\mathbf{W}$ is an orthonormal basis, and where $\|\cdot\|$ is the spectral norm, which we proved, in recitation 3, to be equal to $\|\cdot\|_{2\to2}$. As $\mathbf{P}_\Omega^T\mathbf{P}_\Omega$ is diagonal and its diagonal is made just of 0s (for the vector entries that are not selected) and 1s (for the vector entries which are selected), the spectral norm (i.e. the maximum singular value) is given by 1. Then

$$L = \|\mathbf{P}_\Omega^T\mathbf{P}_\Omega\| = 1 \tag{65}$$

TV in-painting Following the same method as above, we have

$$\|\nabla f(\mathbf{x}_1) - \nabla f(\mathbf{x}_2)\|_2 = \|-\mathbf{P}_\Omega^T(\mathbf{b} - \mathbf{P}_\Omega\mathbf{x}_1) + \mathbf{P}_\Omega^T(\mathbf{b} - \mathbf{P}_\Omega\mathbf{x}_2)\| = \tag{66}$$

$$\|\mathbf{P}_\Omega^T\mathbf{P}_\Omega\mathbf{x}_1 - \mathbf{P}_\Omega^T\mathbf{P}_\Omega\mathbf{x}_2\| = \tag{67}$$

$$\|\mathbf{P}_\Omega^T\mathbf{P}_\Omega(\mathbf{x}_1 - \mathbf{x}_2)\| \leq \|\mathbf{P}_\Omega^T\mathbf{P}_\Omega\|_{2\to 2}\|\mathbf{x}_1 - \mathbf{x}_2\| \tag{68}$$

Again,

$$L = \|\mathbf{P}_\Omega^T\mathbf{P}_\Omega\| = 1 \tag{69}$$

2.2 FISTA implementation and λ sweep

We can see from figure 5 that FISTA algorithm works properly and as expected. In figure 6 we can observe the behavior PSNR (peak signal to noise ratio) as a function of λ. It is better to use a logarithmically-spaced search grid, as it allows to see with which order of magnitude of λ the image is reconstructed in the best way (i.e. PSNR is the highest). In this case I used 15 logarithmically spaced lambdas from 10^{-5} to 10. It is evident from the plot that we get the best performance with $\lambda \in [10^{-4}, 10^{-2}]$. More precisely, using an image of me (the one in figure 7) the best lambdas which have been identified are:

- Best ℓ-1 in-painting lambda = 0.0014
- Best TV in-painting lambda = 0.00052

In figures 7, 8, 9, 10, 11 we can see how the reconstructed images vary with the order of magnitude of the regularization parameter λ. With a smaller λ, the image is more fine-grained, while with a larger one the image is blurrier (and smoother), until it degenerates in a single-color image, such as in figure 11, where the ℓ-1 reconstruction is completely black, which corresponds to a matrix of 0s, the sparsest matrix possible (indeed, what we expect with a large regularization parameter).

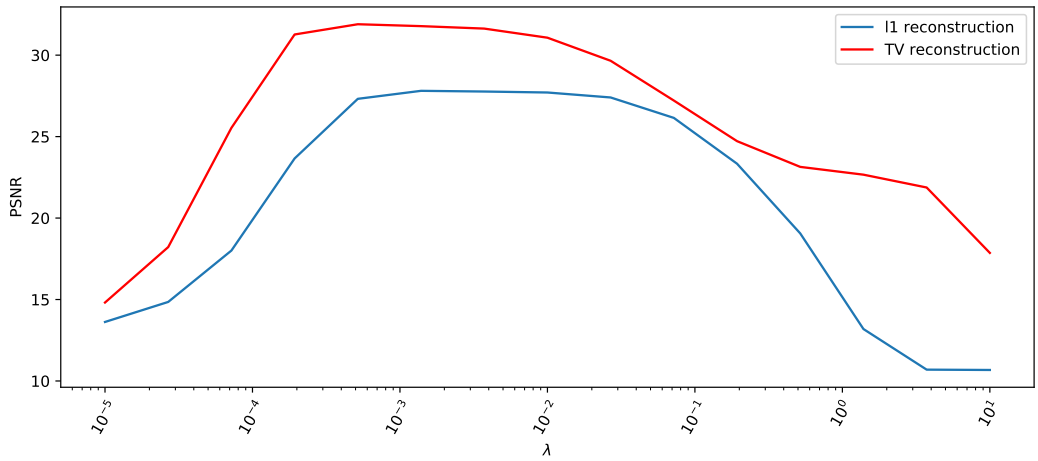

Figure 6: PSNR as a function of the regularization parameter λ

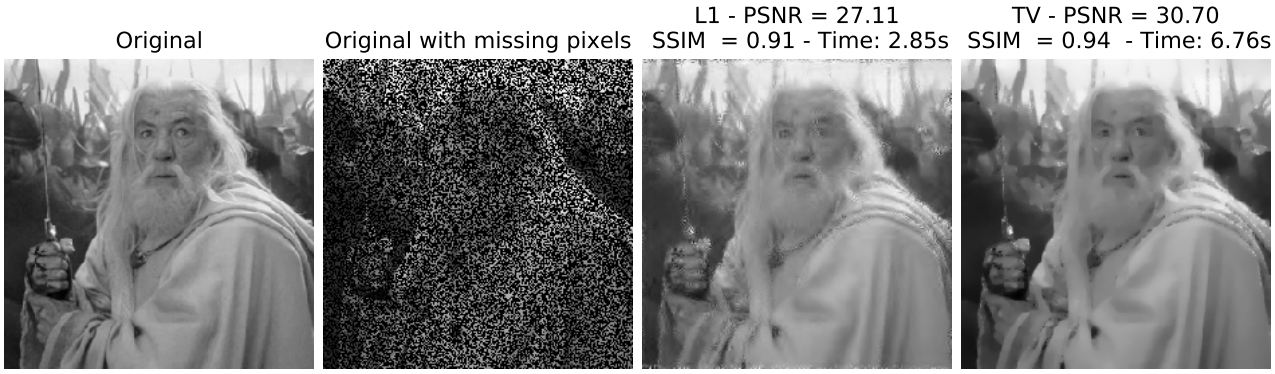

Figure 5: Results obtained with $\lambda = 0.01$ on the reference image

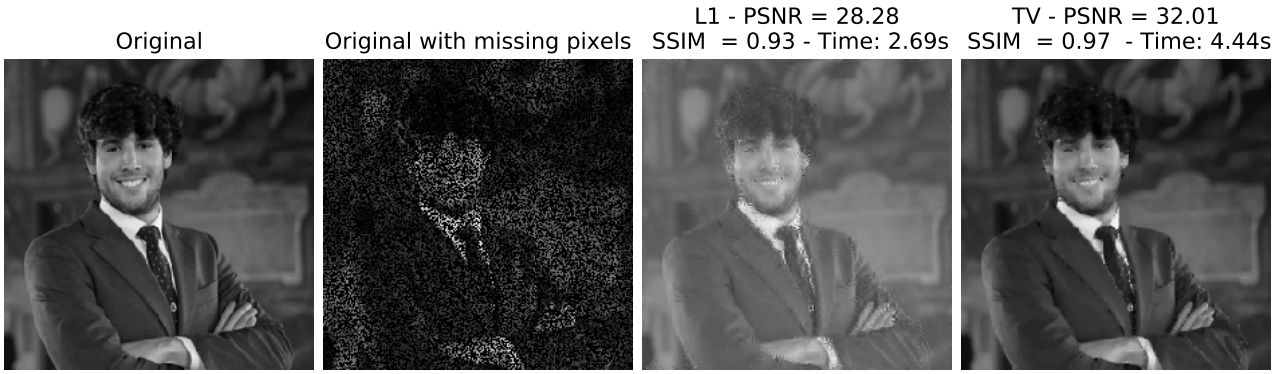

Figure 7: Results obtained with $\lambda = 0.001$

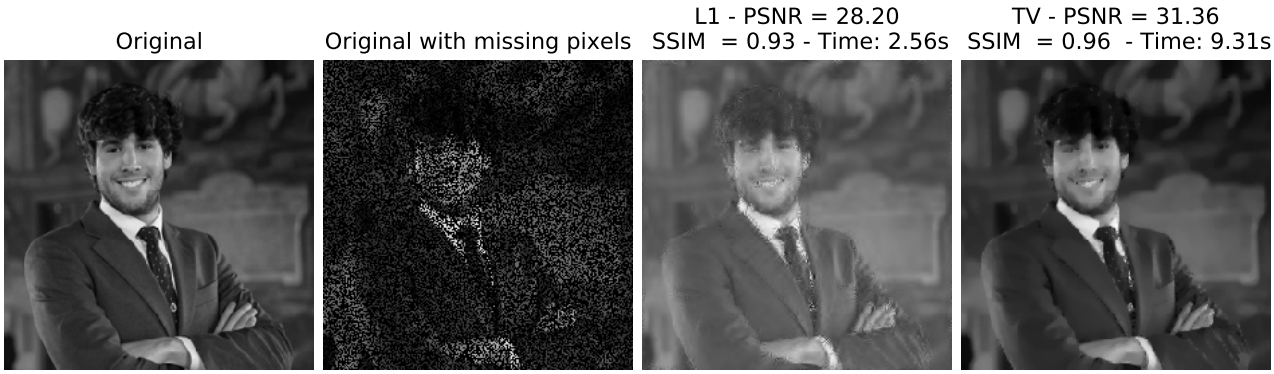

Figure 8: Results obtained with $\lambda = 0.01$

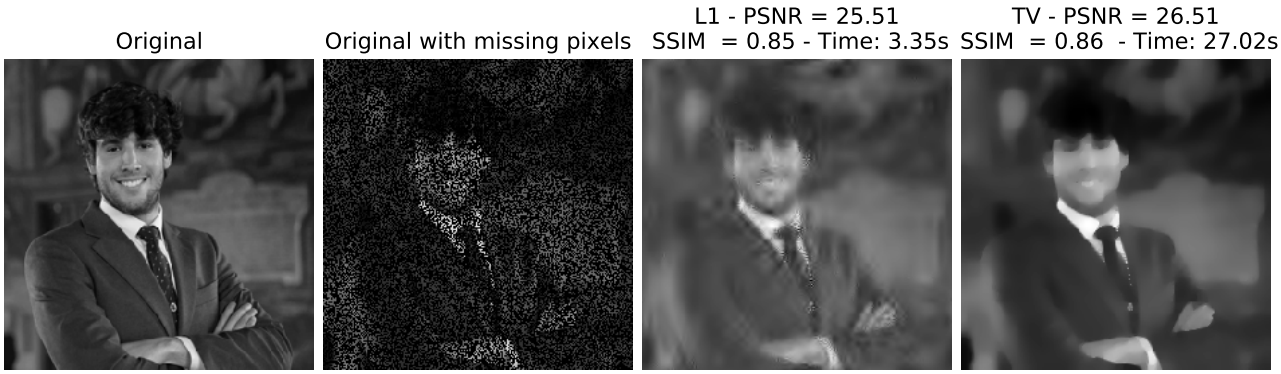

Figure 9: Results obtained with $\lambda = 0.1$

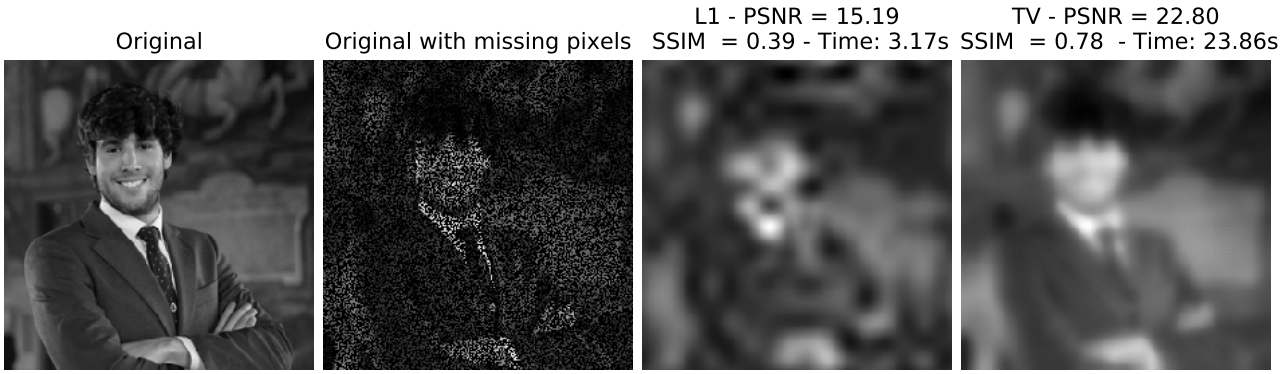

Figure 10: Results obtained with $\lambda = 1$

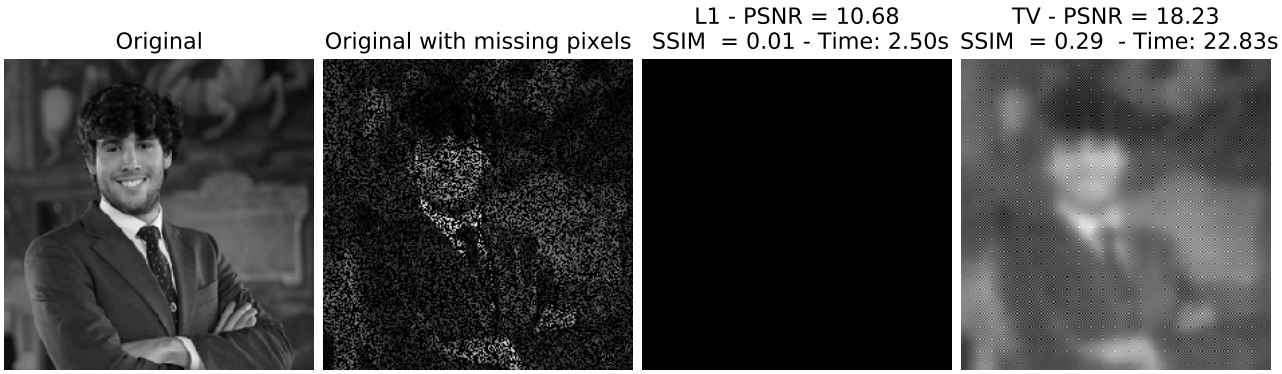

Figure 11: Results obtained with $\lambda = 10$

2.3 Proximal methods convergence

$F^\star$ convergence Using the image in figure 11, the obtained $F^\star = 18.75$. From figure 12 we can observe how the different methods converge with different rates:

- ISTA is the slowest to converge, with a sublinear rate.
- FISTA is a bit faster, but still sublinear, due to its oscillations.
- FISTA with gradient scheme restart is less oscillating than FISTA, and then is able to reach linear convergence rate.

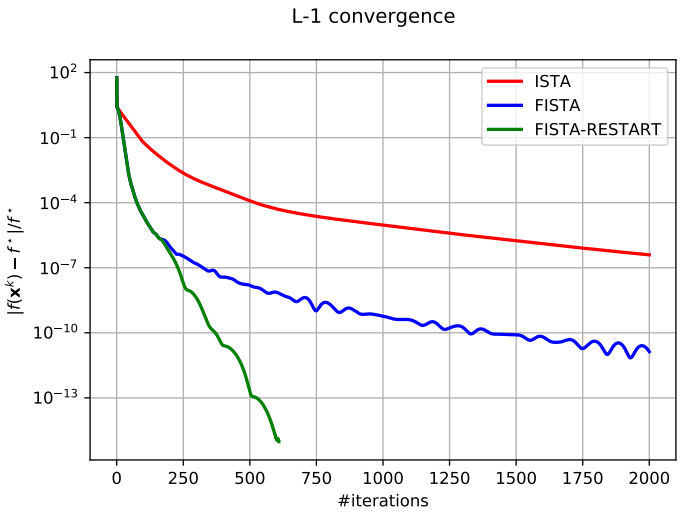

Figure 12: Convergence of ISTA, FISTA and FISTA Restart with respect to $F^\star$

$F^\natural$ convergence Using the same image as above, the obtained $F^\natural = 24.42$. It is worth noting that, for problem (11) in exercise 2 (ℓ-1 in-painting), we have $F^\natural(\boldsymbol{\alpha}^\natural) = \frac{1}{2}\|\mathbf{b}^\natural - \mathbf{P}_{\boldsymbol{\Omega}}\mathbf{W}^{\mathbf{T}}\boldsymbol{\alpha}^\natural\|_2^2 + \lambda_{\ell 1}\|\boldsymbol{\alpha}\|_1$, where $\boldsymbol{\alpha}^\natural = \mathbf{W}\mathbf{x}^\natural$, and $\mathbf{b} = \mathbf{P}_{\boldsymbol{\Omega}}\mathbf{x}^\natural$. Hence, $\mathbf{W}^T\boldsymbol{\alpha}^\natural = \mathbf{W}^T\mathbf{W}\mathbf{x}^\natural = \mathbf{x}^\natural$, and then $F^\natural(\boldsymbol{\alpha}^\natural) = \frac{1}{2}\|\mathbf{b}^\natural - \mathbf{P}_{\boldsymbol{\Omega}}\mathbf{x}^\natural\|_2^2 + \lambda_{\ell 1}\|\boldsymbol{\alpha}^\natural\|_1 = \frac{1}{2}\|\mathbf{b}^\natural - \mathbf{b}^\natural\|_2^2 + \lambda_{\ell 1}\|\boldsymbol{\alpha}^\natural\|_1 = \lambda_{\ell 1}\|\boldsymbol{\alpha}^\natural\|_1$, hence:

$$F^\natural = \lambda_{\ell 1}\|\mathbf{W}\mathbf{x}^\natural\|_1 \tag{70}$$

From figure 13 we can observe that no method converges to $F^\natural$, but this is due to the fact that $F^\star \neq F^\natural$. More specifically, $F^\star < F^\natural$, and this is the reason behind the lower peak we can observe in figure 13 between the first and the 100-th iteration. As a matter of fact, since $F(\boldsymbol{\alpha})$ starts with a large value, while decreasing has values are similar to $F^\natural$ and continues straight towards $F^\star$. Next, as all the methods converge towards $F^\star$, they reach a fixed distance with respect to $F^\natural$, and this is the reason behind the fact that from the 200-th iteration on we have a horizontal line.

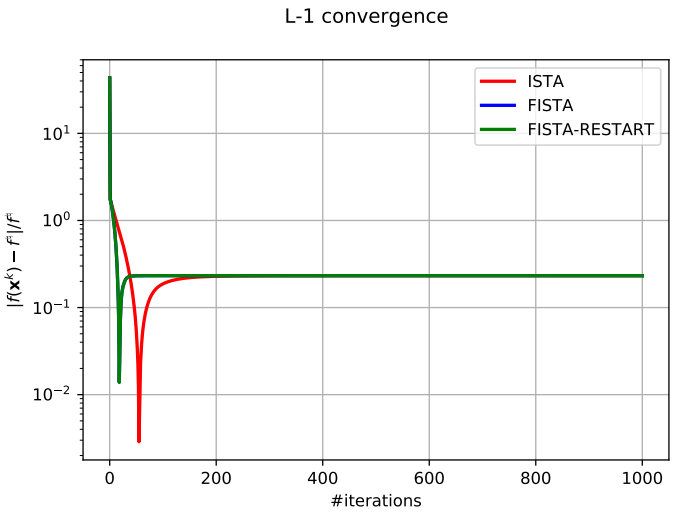

Figure 13: Convergence of ISTA, FISTA and FISTA Restart with respect to $F^\natural$

2.4 Wavelet in-painting and NN unrolling comparison

2.4.1 500 iterations

From figures 17, 18 and 19, respectively relative to in-painting using ℓ-1 and TV-norm regularizers, and unrolling NN we can see how the different methods perform with 500 iterations. It is clear that TV-norm regularizer outperforms the other 2 methods, both for the image quality and PSNR. Moreover, the error mask $|\mathbf{x}^\natural - \mathbf{x}|$ is darker, that means that there is a smaller difference with the real image. The cause of the poor result obtained with unrolling could be due to the fact too many unrolling steps lead to a too much smooth and sparse image.

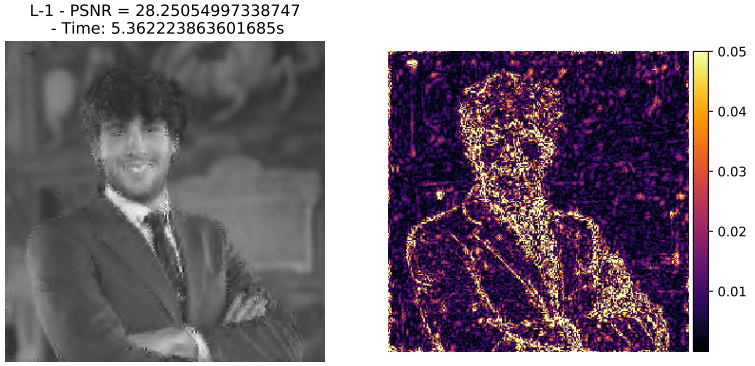

Figure 14: Results with 500 iterations and ℓ-1 norm regularizer, with error mask on the left

Figure 15: Results with 500 iterations and TV-norm regularizer, with error mask on the left

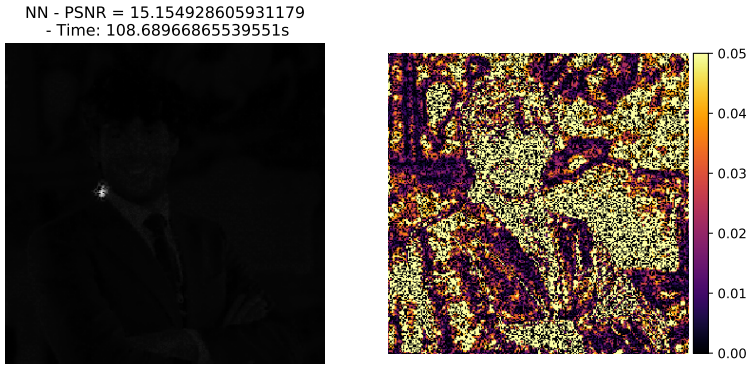

Figure 16: Results with 500 iterations and unrolled NN, with error mask on the left

2.4.2 5 iterations

In this case, the neural network outperforms the other two methods, as just five iterations are not enough to reconstruct the image. However, even though the reconstruction time for the NN is still low, we should keep in mind that the neural network has been already trained, while problems (11) and (12) start from scratch.

A possible tradeoff A good tradeoff between the unrolled method and the classical iterative approaches could be given by a non-linear feature augmentation (e.g. polynomial expansion) of the kept pixels $\mathbf{b}$ to give the iterative approaches more expressiveness.

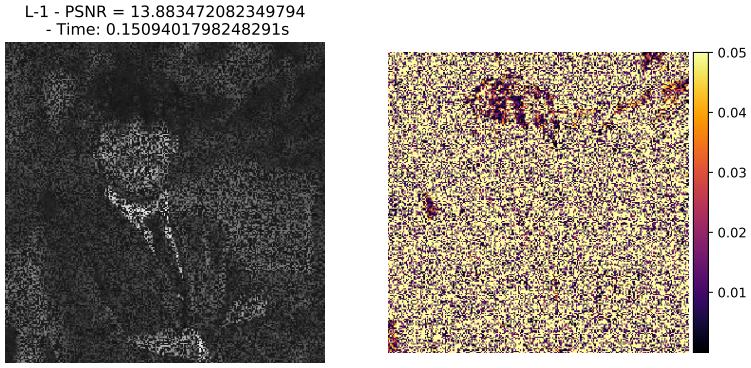

Figure 17: Results with 5 iterations and ℓ-1 norm regularizer, with error mask on the left

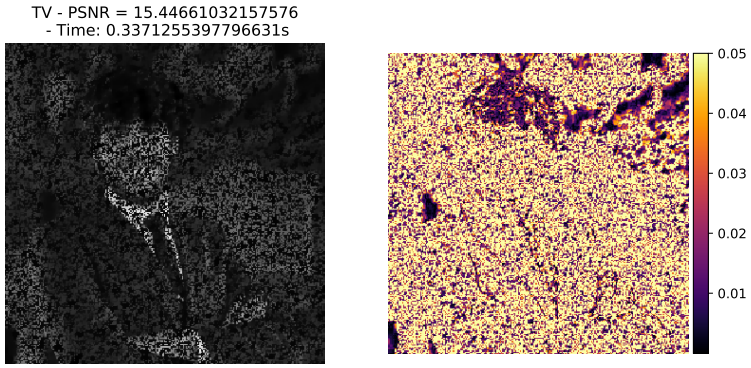

Figure 18: Results with 5 iterations and TV-norm regularizer, with error mask on the left

Figure 19: Results with 5 iterations and unrolled NN, with error mask on the left