Homework 11

Due May 10 at 11 pm

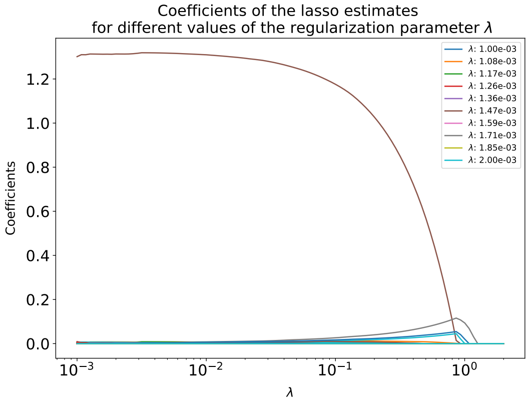
Yves Greatti - yg390

1. (Lasso and ℓ_0) The file `X.txt` contains a 50×300 matrix X, and the file `y.txt` contains the 50×1 vector y. Each line of each file represents a row of the corresponding matrix, and the values on each line are space-delimited.

   (a) Consider the lasso problem

   $$\min_{\beta} \frac{1}{2n} \|X\beta - y\|^2 + \lambda \|\beta\|_1$$

   where $\lambda > 0$ is a parameter and $n = 50$. Construct a (semilogx) plot that draws a separate path for each coefficient value as a function of λ. Include values of λ between 0.01 and 2 (you can include more if you want), and make your values spaced evenly on the log axis (e.g., `np.geomspace`). You can solve the lasso problem using whatever code/library you want.



   (b) Determine the minimizer of

   $$\begin{array}{ll} \text{minimize} & \|\beta\|_0 \\ \text{subject to} & X\beta = y. \end{array}$$

   Assume that the minimizer has small ℓ_0 norm, i.e $\ell_0 \leq 2$. Explain your strategy and justify that it finds the minimizer. Report the nonzero coefficients of the minimizer, and their values. Remember that two floating point values may be different for numerical reasons even if they represent the same value.

   We want to minimize the reconstruction loss $\arg\min ||X\beta - y||_2^2$ with coefficient estimates β having the minimum non-zeros elements. If we have to select only one column of X, we would choose the column which results in the largest projection of y onto that column. After selecting this column, we then solve the linear regression problem using that column with y which gives us one coefficient estimate. We now compute the residual between y and that projection which we use as our new y. Reiterating the previous test, we find the second biggest ”contributor” column by computing the inner product between the $residual$ and each of the columns of X. After identifying the new ”best” column we now solve a linear regression problem, using the two selected columns so far, and y, which results in two new coefficient estimates, which is the sparse representation of y using two columns of X. We keep iterating until we have the minimum number of nonzero coefficients β, the residual is less than a threshold, or we reach a maximum number of iterations.

```
def orthogonal_mp(X:np.ndarray,
        y:np.ndarray,
        n_nonzero_coefs,
        eps_min: np.float64 = 1e-3,
        iter_max: int = 1000):

    def stopping_condition(coef, n_nonzero_coefs):

        for i in range(len(coef)):
            if coef[i].shape[0] == n_nonzero_coefs:
                return True
        return False

    col_idx = list()
    coefs = list()
    residual = y

    for _ in range(iter_max):
        i = np.abs(np.dot(X.T, residual)).argmax()
        if i not in col_idx:
            col_idx.append(i)

        coefi, _, _, _ = np.linalg.lstsq(X[:, col_idx], y)
        coefs.append(coefi)
        residual = y - np.dot(X[:,col_idx], coefi)

        if stopping_condition(coefs, n_nonzero_coefs):
            print(f"Found required number
            of non-zero coefficients:{n_nonzero_coefs}")
            break

        #print(np.inner(residual, residual))
        if np.linalg.norm(residual) <= eps_min:
            print(f"Residual too small, less than {eps_min}")
            break

    return coefs, col_idx
```

Applying this algorithm for $\ell_0 \leq 2$, we found

column 117	column 239
0.33641899	0.51962402

(c) Will your strategy in (b) always find the optimal minimizer of any least-squares problem with ℓ_0 regularization? No, the strategy in (b) will return a minimizer but not necessarily the optimal one. For example, due to numerical reasons if columns of the matrix X are very coherent (inner product between columns, very close to each other), then the algorithm might not select the right column (it has been shown that if mutual incoherence property condition is satisfied, defined as $\max_{(i,j),i\neq j} \langle X[:,i,X[:,j]\rangle < \frac{1}{2K-1}$ then K-sparse signal can be recovered).

2. (Proximal operator) The proximal operator of a function $f : \mathbb{R}^n \to \mathbb{R}$ is defined as

$$\operatorname{prox}_f(y) := \arg\min_x f(x) + \frac{1}{2}||x - y||_2^2. \tag{1}$$

(a) Derive the proximal operator of the squared ℓ_2 norm weighted by a constant $\alpha > 0$, i.e. $f(x) = \alpha ||x||_2^2$.

$$\operatorname{prox}_f(y) := \arg\min_x \alpha ||x||_2^2 + \frac{1}{2}||x - y||_2^2$$

The two terms are quadratic, therefore differentiable, the gradient is

$$\nabla_x \operatorname{prox}_f(y) = 2\alpha x + (x - y)$$

Setting the gradient to zero, yields:

$$\begin{aligned} 2\alpha x + (x - y) &= 0 \\ x &= \frac{1}{1 + 2\alpha} y \\ \operatorname{prox}_f(y) &= \frac{1}{1 + 2\alpha} y, \alpha > 0 \end{aligned}$$

(b) Prove that the proximal operator of the ℓ_1 norm weighted by a constant $\alpha > 0$ is a soft-thresholding operator,

$$\operatorname{prox}_{\alpha ||\cdot||_1}(y) = \mathcal{S}_\alpha(y), \tag{2}$$

where

$$\mathcal{S}_\alpha(y)[i] := \begin{cases} y[i] - \operatorname{sign}(y[i])\alpha & \text{if } |y[i]| \geq \alpha, \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

$$\operatorname{prox}_{\alpha ||\cdot||_1}(y) = \alpha \|x\|_1 + \frac{1}{2}||x - y||_2^2, \alpha > 0$$

And we are looking for:

$$\begin{aligned} 0 &\in \partial(\alpha \|x\|_1) + \nabla_x(\frac{1}{2}||x - y||_2^2) \\ 0 &\in \alpha\, \partial(\|x\|_1) + (x - y) \end{aligned}$$

We examine each component of x and y separately. Assume first that $x[i] \neq 0$ then $\partial(\|x\|_1) = \text{sign}\,(x[i])$, setting the subgradient to 0, we have:

$$\begin{aligned} x[i] - y[i] + \alpha\,\text{sign}\,(x[i]) = 0 \\ x[i] = y[i] - \alpha\,\text{sign}\,(x[i]) \end{aligned}$$

Note that

$$\begin{aligned} x[i] < 0, \text{sign}\,(x[i]) = -1 \to y[i] + \alpha < 0 \quad &\text{or } y[i] < -\alpha < 0 \\ x[i] > 0, \text{sign}\,(x[i]) = 1 \to y[i] - \alpha > 0 \quad &\text{or } y[i] > \alpha > 0 \end{aligned}$$

thus in this case $\text{sign}\,(x[i]) = \text{sign}\,(y[i])$ and the optimal point is $y[i] - \alpha\,\text{sign}\,(y[i])$. In the case where $x[i] = 0$, let $\gamma = \partial(\|x\|_1), |\gamma| \leq 1$ then it holds

$$\begin{aligned} x[i] - y[i] + \alpha\gamma = 0 \to y[i] - \alpha\gamma = 0 \\ y[i] = \gamma\alpha \\ |y[i]| \leq \alpha \end{aligned}$$

Putting all together, we get

$$\text{prox}_{\alpha\,||\cdot||_1}(y) = \begin{cases} y\,[i] - \text{sign}\,(y\,[i])\,\alpha & \text{if } |y\,[i]| \geq \alpha, \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

(c) Prove that if $X \in \mathbb{R}^{p\times n}$ has orthonormal rows ($p \leq n$) and $y \in \mathbb{R}^n$, then for any function f

$$\arg\min_\beta \frac{1}{2}\left|\left|y - X^T\beta\right|\right|_2^2 + f(\beta) = \arg\min_\beta \frac{1}{2}\,||Xy - \beta||_2^2 + f(\beta). \tag{5}$$

The two expressions for the same function f differs on the first term, so we want to show that

$$\arg\min_\beta \left|\left|y - X^T\beta\right|\right|_2^2 = \arg\min_\beta ||Xy - \beta||_2^2$$

X having orthonormal rows: $XX^T = I$, and

$$\begin{aligned} \left|\left|y - X^T\beta\right|\right|_2^2 &= (y^T - \beta^T X)(y - X^T\beta) \\ &= y^Ty - y^TX^T\beta - \beta^TXy + \beta^TXX^T\beta \\ &= y^Ty - y^TX^T\beta - \beta^TXy + \beta^T\beta \\ ||Xy - \beta||_2^2 &= (y^TX^T - \beta^T)(Xy - \beta) \\ &= y^TX^TXy - y^TX^T\beta - \beta^TXy + \beta^T\beta \\ \arg\min_\beta \left|\left|y - X^T\beta\right|\right|_2^2 &= \arg\min_\beta ||Xy - \beta||_2^2 \text{ eliminating terms independant of } \beta \end{aligned}$$

(d) Use the answers to the previous questions to compare the ridge-regression and lasso estimators for a regression problem where the features are orthonormal.

The use of l_1, l_2 norms gives rise to the problems, for $\lambda > 0$:

$$\frac{1}{2}\arg\min_{\beta} \|y - X^T\beta\|_2^2 + \lambda\|\beta\|_2^2 \text{ Ridge regression}$$
$$\frac{1}{2}\arg\min_{\beta} \|y - X^T\beta\|_2^2 + \lambda\|\beta\|_1 \text{ Lasso regression}$$

which is equivalent from part c) to

$$\arg\min_{\beta} \lambda\|\beta\|_2^2 + \frac{1}{2}\|\beta - Xy\|_2^2 \text{ Ridge regression}$$
$$\arg\min_{\beta} \lambda\|\beta\|_1 + \frac{1}{2}\|\beta - Xy\|_2^2 \text{ Lasso regression}$$

From part a) and b), the solutions of these two problems are the proximal operators:

$$\beta_{\text{ridge}} = \frac{1}{1+2\lambda}Xy$$
$$\beta_{\text{lasso}} = \mathcal{S}_\lambda\left(Xy\right)$$

We see that the lasso solution shows sparsity: when the component wise least-square coefficients, Xy, are not small they are shrunken towards 0 by λ, and set to 0 when they are small. In contrast the ridge regression estimates are never sparse, all scaled with a single factor inversely proportional to λ. When λ is 0, ridge and lasso estimates are the OLS estimates.

3. (Proximal gradient method)

(a) The first-order approximation to a function $f : \mathbb{R}^p \to \mathbb{R}$ at $x \in \mathbb{R}^p$ equals

$$f\left(x\right) + \nabla f\left(x\right)^T \left(y - x\right). \tag{6}$$

We want to minimize this first-order approximation locally. To this end we fix a real constant $\alpha > 0$ and augment the approximation with an ℓ_2-norm term that keeps us close to x,

$$f_x(y) := f\left(x\right) + \nabla f\left(x\right)^T \left(y - x\right) + \frac{1}{2\,\alpha}\left|\left|y - x\right|\right|_2^2. \tag{7}$$

Prove that the minimum of f_x is the gradient descent update $x - \alpha \nabla f\left(x\right)$.
For $\alpha > 0$,

$$\nabla f_x(y) = \nabla f\left(x\right) + \frac{1}{\alpha}(y - x)$$

Setting the gradient to 0 gives

$$\begin{aligned} \alpha \nabla f\left(x\right) + (y - x) = 0 \\ y = x - \alpha \nabla f\left(x\right) \end{aligned}$$

(b) Inspired by the previous question, how would you modify gradient descent to minimize a function of the form

$$h(x) = f_1(x) + f_2(x), \tag{8}$$

where f_1 is differentiable, and f_2 is nondifferentiable but has a proximal operator that is easy to compute?
At iteration $x^{(k)}$, we want to stay close to the gradient update for $f_1(x^{(k)})$ and minimize $f_2(x^{(k)})$

$$\begin{aligned}
x^{(k+1)} &= \arg\min_x f_2\left(x^{(k)}\right) + f_1\left(x^{(k)}\right) + \nabla f_1\left(x^{(k)}\right)^T\left(x - x^{(k)}\right) + \frac{1}{2\,\alpha}\left|\left|x - x^{(k)}\right|\right|_2^2 \\
&= \arg\min_x f_2\left(x^{(k)}\right) + 2\alpha f_1\left(x^{(k)}\right) + \Bigg(2\alpha\nabla f_1\left(x^{(k)}\right)^T\left(x - x^{(k)}\right) + \left|\left|x - x^{(k)}\right|\right|_2^2 \\
&+ \left|\left|\alpha\nabla f_1\left(x^{(k)}\right)^T\right|\right|_2^2\Bigg) - \left|\left|\alpha\nabla f_1\left(x^{(k)}\right)^T\right|\right|_2^2 \\
&= \arg\min_x f_2\left(x^{(k)}\right) + 2\alpha f_1\left(x^{(k)}\right) + \left|\left|\alpha\nabla f_1\left(x^{(k)}\right)^T + (x - x^{(k)})\right|\right|_2^2 - \left|\left|\alpha\nabla f_1\left(x^{(k)}\right)^T\right|\right|_2^2 \\
&= \arg\min_x f_2\left(x^{(k)}\right) + f_1\left(x^{(k)}\right) + \frac{1}{2\,\alpha}\left|\left|\alpha\nabla f_1\left(x^{(k)}\right)^T + (x - x^{(k)})\right|\right|_2^2 - \frac{1}{2\,\alpha}\left|\left|\alpha\nabla f_1\left(x^{(k)}\right)^T\right|\right|_2^2 \\
&= \arg\min_x f_2\left(x^{(k)}\right) + \frac{1}{2\,\alpha}\left|\left|\alpha\nabla f_1\left(x^{(k)}\right)^T + (x - x^{(k)})\right|\right|_2^2 \quad \text{removing terms independant of } x \\
&= \arg\min_x f_2\left(x^{(k)}\right) + \frac{1}{2\,\alpha}\left|\left|x - (x^{(k)} - \alpha\nabla f_1\left(x^{(k)}\right)^T)\right|\right|_2 \\
&= \operatorname{prox}_{f_2}\left(x^{(k)} - \alpha\nabla f_1\left(x^{(k)}\right)\right)
\end{aligned}$$

(c) Show that a vector x^* is a solution to

$$\text{minimize} \quad f_1(x) + f_2(x)\,, \tag{9}$$

where f_1 is differentiable, and f_2 is nondifferentiable, and both functions are convex, if and only if it is a fixed point of the iteration you proposed in the previous question for any $\alpha > 0$.

if x^* minimizes $f_1(x) + f_2(x)$ then

$$x^* = \operatorname{prox}_{f_2}(x^{(k)} - \alpha\nabla f_1(x^{(k)})$$

since both functions are convex

$$\begin{aligned}
&\Leftrightarrow 0 \in \partial_x\Bigg(f_2\left(x^{(k)}\right) + f_1\left(x^{(k)}\right) + \nabla f_1\left(x^{(k)}\right)^T\left(x - x^{(k)}\right) + \frac{1}{2\,\alpha}\left|\left|x - x^{(k)}\right|\right|_2^2\Bigg) \\
&\Leftrightarrow \alpha\nabla_x f_1\left(x^{(k)}\right)^T + x - x^{(k)} = 0 \\
&\Leftrightarrow x^* = x^{(k)} - \alpha\nabla_x f_1\left(x^{(k)}\right)^T
\end{aligned}$$

4. (Iterative shrinkage-thresholding algorithm)

   (a) What is the proximal gradient update corresponding to the lasso problem defined below? Your answer will involve a hyperparameter which we will call as α.

$$\frac{1}{2}\,||y - X\beta||_2^2 + \lambda|\beta|_1$$

   From 2.(b), we have established:

$$\begin{aligned} \text{prox}_{\alpha\,||\cdot||_1}\left(x\right) &= \mathcal{S}_\alpha\left(x\right) \\ \mathcal{S}_\alpha\left(x\right)\left[i\right] &= \text{sign}\left(x\left[i\right]\right)\max\left(\left|\left(x\left[i\right]\right)\right| - \alpha, 0\right) \end{aligned}$$

   Combining it with 3.(b) gives for

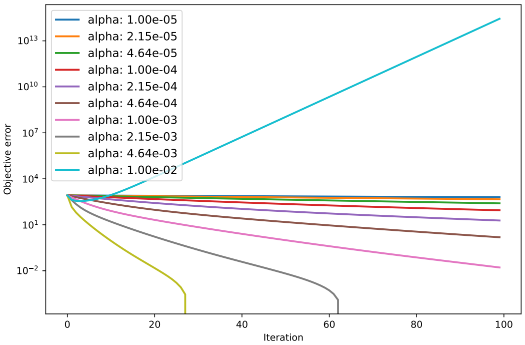
$$\begin{aligned} g(x) &= f(x) + \lambda|x|_1 \; f \;\text{ being differentiable} \\ x^{(k+1)} &= \text{prox}_{\alpha\lambda\,||\cdot||_1}\left(x^{(k)} - \alpha\nabla_x f\left(x^{(k)}\right)^T\right) = \mathcal{S}_{\alpha\lambda}\left(x^{(k)} - \alpha\nabla_x f\left(x^{(k)}\right)^T\right) \end{aligned}$$

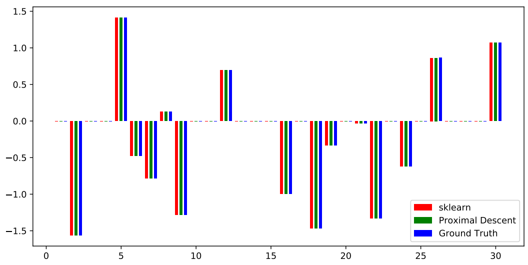
   Plugging it back leads to the proximal gradient update corresponding to the lasso problem

$$\beta^{(k+1)} = \mathcal{S}_{\alpha\lambda}(\beta^{(k)} + \alpha X^T(y - X\beta^{(k)}))$$

   (b) How would you check whether you have reached an optimum? How would you modify this to take into account possible numerical inaccuracies? For each iteration we can compare the value of the objective function for the new estimates β with the value obtained for the previous estimated coefficients and if the absolute difference is less than a tolerance (very small value) then we know we have reached an optimum (we can also improve that logic and check that the difference did not decrease for the last n iterations).

   (c) Implement the method and apply it to the problem in `pgd_lasso-question.ipynb`. You have to fill in blocks of code corresponds to the proximal gradient update step and termination condition. Report all the generated plots.




1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
0
5
10
15
20
25
30
sklearn
Proximal Descent
Ground Truth