Exercises

For each of the following functions, find an expression for $\mathrm{prox}_{\lambda f}(\mathbf{x})$ for any $\lambda > 0$ and $\mathbf{x}$.

1. elastic net: $f(\mathbf{x}) = \lambda\|\mathbf{x}\|_2^2 + \mu\|\mathbf{x}\|_1$ $(\lambda, \mu > 0)$ (just $\mathrm{prox}_f(\mathbf{x})$)
2. $f(t) = \max\{0, t\}$
3. $f(t) = \max\{t, 1 - 3t\}$
4. $f(t) = [\max\{0, t\}]^2$
5. $f(\mathbf{x}) = \mathbf{x}_{[1]} \equiv \max\{x_1, x_2, \ldots, x_n\}$
6. $f(\mathbf{x}) = 2x_{[1]} + x_{[2]}$. Write a code implementing prox_f . Use this code to find $\mathrm{prox}_f((2, 1, 4, 1, 2, 1))$ Final answer: $(1.5, 1, 2, 1, 1.5, 1)$
7. $f(\mathbf{x}) = |\mathbf{a}^T\mathbf{x}|$, $\mathbf{a} \in \mathbb{R}^n \setminus \{\mathbf{0}\}$
8. $f(t) = \begin{cases} \frac{1}{t} & t > 0, \\ \infty & \text{else.} \end{cases}$
9. $f(\mathbf{X}) = \begin{cases} \mathrm{tr}(\mathbf{X}^{-1}) & \mathbf{X} \succ \mathbf{0} \\ \infty & \text{else} \end{cases}$ (over $\mathbb{S}^n$). Write a code implementing prox_f .

   Use this code to find $\mathrm{prox}_f\left[\begin{pmatrix} 3 & 1 \\ 1 & 4 \end{pmatrix}\right]$ Final answer: $\begin{pmatrix} 3.1251 & 0.9511 \\ 0.9511 & 4.0762 \end{pmatrix}$
10. $f(\mathbf{x}) = (\|\mathbf{x}\|_2 - 1)^2$

Exercise 0 Contd.

Suppose that $\mathbf{A}, \mathbf{b}, \lambda_1, \lambda_2$ are generated via

MATLAB

```
m=100;
n=120;
a = [0:m-1]+1;
b = [0:n-1]+0.5;
A = sin(10*(a'*b).^3);
xi = sin(31*[1:n].^3)';
b = A*xi;
lambda1=2;
lambda2=0.5;
```

Python

```
import numpy as np
m = 100
n = 120
a = np.arange(0,m)+1
b = np.arange(0,n)+0.5
A = np.sin(10 * np.outer(a,b)**3)
xi = np.sin(31 * np.arange(1,n+1)**3)
b = A @ xi
lambda1 = 2
lambda2 = 0.5
```

Implement the following methods in Python/MATLAB:

- proximal gradient with $f(\mathbf{x}) = \frac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 + \frac{\lambda_1}{2}\|\mathbf{x}\|_2^2, g(\mathbf{x}) = \lambda_2\|\mathbf{x}\|_1$.
- FISTA with $f(\mathbf{x}) = \frac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 + \frac{\lambda_1}{2}\|\mathbf{x}\|_2^2, g(\mathbf{x}) = \lambda_2\|\mathbf{x}\|_1$.
- V-FISTA with $f(\mathbf{x}) = \frac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 + \frac{\lambda_1}{2}\|\mathbf{x}\|_2^2, g(\mathbf{x}) = \lambda_2\|\mathbf{x}\|_1$.
- V-FISTA with $f(\mathbf{x}) = \frac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2, g(\mathbf{x}) = \frac{\lambda_1}{2}\|\mathbf{x}\|_2^2 + \lambda_2\|\mathbf{x}\|_1$. (yes - not a "legal" method. Use λ_1 as the strong-convexity parameter)

Run 100 iterations of each of the methods. Start each of the methods with the all-zeros vectors and use a constant stepsize. Plot the values of $F(\mathbf{x}^k) - F_{\text{opt}}$ of the four methods (in the same plot). Use log-scale in the y-axis. Which of the methods performed best? write the first four components of the solution generated by the methods.

Exercise 1

Consider the problem

$$\min \sqrt{\mathbf{x}^T\mathbf{Q}\mathbf{x} + 2\mathbf{b}^T\mathbf{x} + c} + 0.2\|\mathbf{D}\mathbf{x} + \mathbf{e}\|_1,$$

where $\mathbf{Q} \in \mathbb{R}^{30\times 30}$, $\mathbf{b} \in \mathbb{R}^{30}$, $c \in \mathbb{R}$, $\mathbf{D} \in \mathbb{R}^{10\times 30}$. The matrix $\mathbf{Q}$ is positive definite.

(a) Show that under the condition $c > \mathbf{b}^T\mathbf{Q}^{-1}\mathbf{b}$, the problem is well-defined (the expression inside the square root is always nonnegative)

(b) Show that under the above condition, the problem is convex.

(c) Assume that $\mathbf{D}\mathbf{D}^T = \mathbf{I}$. Write explicitly the proximal gradient method and FISTA with constant stepsize $1/L_f$ for solving the problem (taking $\|\mathbf{D}\mathbf{x}\|_1$ as the nonsmooth part). Explain how to compute a Lipschitz constant of the differentiable part.

Exercise 3: Soft Margin SVM

Given a set of data points $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n \in \mathbb{R}^d$ and corresponding labels $y_1, y_2, \ldots, y_n$. The soft margin SVM problem is given by

$$\min \left\{ \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{i=1}^{n} \max\{0, 1 - y_i \mathbf{w}^T \mathbf{x}_i\} \right\}.$$

(a) Write explicitly (including e.g. prox computations, Lipschitz constants) the DPG and FDPG for solving the problem.
(b) Suppose that the data is generated as follows:

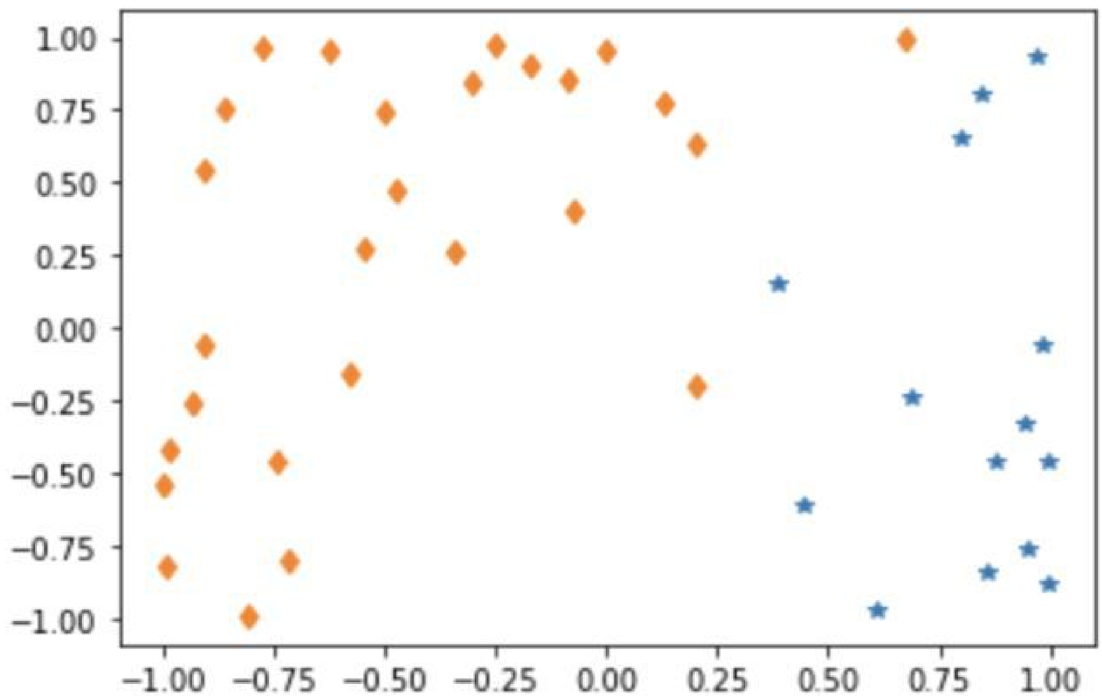
MATLAB

```
x=sin(10*[1:40].^3)';
y=sin(28*[1:40].^3)';
cl=[2*x<y+0.5]+1;
x = [x;0.2];
y = [y;-0.2];
cl = [cl;2];
A1=[x(cl==1),y(cl==1)];
A2=[x(cl==2),y(cl==2)];
figure(1)
plot(A1(:,1),A1(:,2),'*','MarkerSize',6)
hold on
plot(A2(:,1),A2(:,2),'d','MarkerSize',6)
```

Python

```
import numpy as np
import matplotlib.pyplot as plt
x = np.sin(10*np.arange(1,41)**3)
y = np.sin(28*np.arange(1,41)**3)
cl = (2*x<y+0.5)+1
x = np.hstack((x,0.2))
y = np.hstack((y,-0.2))
cl = np.hstack((cl,2))
A1=np.column_stack((x[cl==1], y[cl==1]))
A2=np.column_stack((x[cl==2], y[cl==2]))
plt.plot(A1[:,0], A1[:,1],'*')
plt.plot(A2[:,0], A2[:,1],'d')
plt.show()
```

Exercise 3 Contd.



Write a MATLAB/Python code that implements 40 iterations of the DPG and FDPG methods. Write the solutions produced by each of the methods. For each of the solutions, plot the two classes of points along with the corresponding hyperplane.

Exercise

Exercise 2: Consider the problem

$$\min\left\{-\sum_{i=1}^{m}\log(\mathbf{a}_i^T\mathbf{x}-b_i)+\sum_{i=1}^{n-2}\sqrt{(x_i-x_{i+1})^2+(x_{i+1}-x_{i+2})^2}:\mathbf{a}_i^T\mathbf{x}>b_i, i\in[m]\right\}$$

where $\mathbf{a}_i \in \mathbb{R}^n$ and $b_i \in \mathbb{R}$ for all $i \in [m]$.

(a) Write an ADLPMM method for solving the problem.

(b) Write the CP method for solving the problem.

(c) Assume that $\mathbf{A}$ and $\mathbf{b}$ are generated via

MATLAB

```
m=30;
n=25;
a = [0:m-1]+1;
b = [0:n-1]+0.5;
A = sin(10*(a'*b).^3);
xi = sin(31*[1:n].^3)';
b = A*xi+sin(23*[1:m].^3)'+1.5
```

Python

```
import numpy as np
import numpy.linalg as la
m = 30
n = 25
a = np.arange(m)+1
b = np.arange(n)+0.5
A = np.sin(10*np.outer(a,b)**3)
xi = np.sin(31*np.arange(1,n+1)**3)
b = A@xi+np.sin(23*np.arange(1,m+1)**3)+1.5
```

Exercise 2 Contd.

Implement the CP and ADLPMM in MATLAB/Python. Run the ADLPPMM method with $\rho = 1$ and $\rho = \frac{1}{\|\mathbf{A}\|_2}$. Run the CP method with $\tau = \sigma = \frac{1}{\|\mathbf{A}\|_2}$ and also $\tau = \frac{1}{\|\mathbf{A}\|_2^2}, \sigma = 1$. Run 60 iterations of each of the methods ($4 = 2 \times 2$ runs). Initialize all the vectors with zeros. Print the first three components (x_1, x_2, x_3) generated by each of the methods. Add a plot of $f(\mathbf{x}^k)$ of the four methods (in the same plot). Plot only the iterations in which the vector is feasible!

(d) Consider the problem

$$\min \left\{ \frac{\|\mathbf{x}\|_2^2}{2} - \sum_{i=1}^{m} \log(\mathbf{a}_i^T \mathbf{x} - b_i) + \sum_{i=1}^{n-2} \sqrt{(x_i - x_{i+1})^2 + (x_{i+1} - x_{i+2})^2} : \mathbf{a}_i^T \mathbf{x} > b_i \right\}$$

Write an accelerated CP (ACP) method as well as the FDPG method for solving the problem.

(e) Implement the two methods from part (d) in MATLAB/Python on the data generated in part (c). For the ACP method use $\tau_0 = \sigma_0 = \frac{1}{\|\mathbf{A}\|_2}$. Run 100 iterations of the two methods. Write the the first three components produced by each of the methods and plot the function values generated by each of the two methods (in the same plot). Plot only the iterations in which the vector is feasible.