

## CSE 203B W22 Homework 3

Due Time : 11:50pm, Wednesday Feb. 9, 2022 Submit to Gradescope  
Gradescope: <https://gradescope.com/>

In this homework, we work on exercises from text book including level sets of convex, concave, quasi-convex, quasi-concave functions (3.1, 3.2), second-order conditions for convexity on affine sets (3.9), Kullback-Leibler divergence (3.13), saddle points of convex-concave functions (3.14) determination of convex, concave, quasi-convex, quasi-concave functions (3.16), conjugate functions (3.36), and gradient and Hessian of conjugate functions (3.40). Extra assignments are given on conjugate function and compressed sensing using different norms.

Total points: 30. Exercises are graded by completion, assignments are graded by content.

### I. Exercises from textbook chapter 2 (8 pts, 1pt for each problem)

3.1, 3.2, 3.9, 3.13, 3.14, 3.16, 3.36, 3.40.

### II. Assignments

II. 1. Conjugate Functions. (4 pts)

Find the conjugate functions of the following functions.

$$(1) f(x) = x^T A x + b^T x, \text{ where } x \in R^n, A \in R^{n \times n}, b \in R^n.$$

The conjugate is defined as:  $f^*(y) = \sup_x (y^T x - x^T A x - b^T x) = \sup_x ((y - b)^T x - x^T A x)$ .  
Let  $g(x, y) = (y - b)^T x - x^T A x$ .

Discuss your solution for the following three cases:

i. matrix A is symmetric and positive definite,

Since  $\forall x \neq 0, x^T A x > 0$ , we can conclude that this quadratic term overshadows the linear terms as  $x \rightarrow \pm\infty$ , and because of the minus sign,  $g(x, y) \rightarrow -\infty$ . Since this is inconclusive, we can check the derivative:

$$\frac{\partial g(x, y)}{\partial x} = y - 2Ax - b = 0 \implies \hat{x} = A^{-1} \left( \frac{y - b}{2} \right)$$

$$\begin{aligned} f^*(y) &= (y - b)^T A^{-1} \left( \frac{y - b}{2} \right) - \left( \frac{y - b}{2} \right)^T A^{-1} A A^{-1} \left( \frac{y - b}{2} \right) \\ &= \left( \frac{y - b}{2} \right)^T A^{-1} \left( \frac{y - b}{2} \right) \end{aligned}$$

ii.  $A$  is positive definite but not symmetric,

The same logic as the first case applies, so we can check the derivative again:

$$\frac{\partial g(x, y)}{\partial x} = y - (A + A^T)x - b = 0 \implies \hat{x} = (A + A^T)^{-1}(y - b)$$

$$\begin{aligned} f^*(y) &= (y - b)^T (A + A^T)^{-1} (y - b) - (y - b)^T (A + A^T)^{-1} A (A + A^T)^{-1} (y - b) \\ &= (y - b)^T (A + A^T)^{-1} (y - b) - (y - b)^T (A + A^T)^{-1} \left( \frac{A + A^T}{2} \right) (A + A^T)^{-1} (y - b) \\ &= \frac{1}{2} (y - b)^T (A + A^T)^{-1} (y - b) \end{aligned}$$

Note: when we have a form  $v^T A v$ , notice that this is a scalar, so  $v^T A v = (v^T A v)^T = v^T A^T v$ . Add  $v^T A v$  to both sides:  $2v^T A v = v^T A v + v^T A^T v = v^T (A + A^T) v$ . Finally simplify:  $v^T A v = v^T \left( \frac{A + A^T}{2} \right) v$ .

iii.  $A$  is not positive definite.

There are two scenarios to consider here:

- Positive semidefinite case:  $\forall x, x^T A x \geq 0$ , with  $\exists z \neq 0, z^T A z = 0$ . Set  $x = tz$ . If we check the limit of  $g(x, y)$  as  $t \rightarrow \pm\infty$  (depending on the sign of  $(y - b)^T z$ ):

$$\begin{aligned} \lim_{t \rightarrow \pm\infty} ((y - b)^T x - x^T A x) &= \lim_{t \rightarrow \pm\infty} (t(y - b)^T z - t^2 z^T A z) \\ &= \lim_{t \rightarrow \pm\infty} (t(y - b)^T z) \rightarrow \infty, \text{ if } (y - b)^T z \neq 0 \end{aligned}$$

If  $\forall z \neq 0 \wedge z^T A z = 0, (y - b)^T z = 0$ , then the conjugate is finite. Note that it isn't directly 0 however, since we have to check all  $z$  such that  $z^T A z \geq 0$  to conclude that. When we do, we find that a  $z$  that satisfies  $z^T A z > 0$  can achieve values  $\geq 0$ , and that's just the answer from part (ii), except a pseudoinverse must be used instead of the inverse. Therefore, for the semidefinite case, we can conclude the following:

$$f^*(y) = \begin{cases} \infty & \exists z \neq 0, z^T A z = 0 \wedge (y - b)^T z \neq 0 \\ \frac{1}{2} (y - b)^T (A + A^T)^\dagger (y - b) & \text{otherwise} \end{cases}$$

- Non-positive semidefinite case:  $\exists z, z^T A z < 0$ . Set  $x = tz$ . If we check the limit of  $g(x, y)$  as  $t \rightarrow \pm\infty$ :

$$\lim_{t \rightarrow \pm\infty} ((y - b)^T x - x^T A x) = \lim_{t \rightarrow \pm\infty} (t(y - b)^T z - t^2 z^T A z) \rightarrow \infty$$

Putting the semidefinite and non-semidefinite cases together, we have the following conjugate:

$$f^*(y) = \begin{cases} \infty & \exists z, z^T A z < 0 \vee (z \neq 0 \wedge z^T A z = 0 \wedge (y - b)^T z \neq 0) \\ \frac{1}{2}(y - b)^T (A + A^T)^\dagger (y - b) & \text{otherwise} \end{cases}$$

An alternative solution involves the realization that for the positive semidefinite case, we also have:

$$\frac{\partial g(x, y)}{\partial x} = y - (A + A^T)x - b = 0 \implies (A + A^T)x = y - b$$

A consideration to make here is whether we have  $y - b \in \text{range}(A + A^T)$ , or equivalently  $y - b \perp \text{nullspace}(A + A^T)$ . When  $(A + A^T)x = y - b$  is solvable, we can define the solution using (at the very least) the pseudoinverse. When it is not solvable, then we do not have a valid optimum point, and since  $\frac{\partial^2 g(x, y)}{\partial x^2}$  is negative semidefinite when  $A$  is positive semidefinite, we can conclude that  $g$  can grow boundlessly. In the non-positive semidefinite case, since  $\frac{\partial^2 g(x, y)}{\partial x^2}$  is non-negative semidefinite when  $A$  is non-positive semidefinite,  $g$  is non-concave, and nonconcave means there's a direction along which the function grows without bound (and this is what the limit argument also utilizes.) Therefore, the following is also a valid answer:

$$f^*(y) = \begin{cases} \frac{1}{2}(y - b)^T (A + A^T)^\dagger (y - b) & (\forall z, z^T A z \geq 0) \wedge y - b \in \text{range}(A + A^T) \\ \infty & \text{otherwise} \end{cases}$$

(2)

$$f(x) = \begin{cases} \frac{1}{2}x^T x & \|x\|_2 \leq 1 \\ (x^T x)^{\frac{1}{2}} - \frac{1}{2} & \|x\|_2 > 1 \end{cases}$$

$$f^*(y) = \max \left\{ \sup_{\|x\|_2 \leq 1} \left( y^T x - \frac{1}{2}x^T x \right), \sup_{\|x\|_2 > 1} \left( y^T x - (x^T x)^{\frac{1}{2}} + \frac{1}{2} \right) \right\}$$

For the  $\|x\|_2 \leq 1$  case, since  $x$  is bounded by its norm, and we don't have any norm terms as a divisor etc., we can take the derivative to figure out the supremum. Let  $g(x, y) = y^T x - \frac{1}{2}x^T x$ . Then:

$$\frac{\partial g(x, y)}{\partial x} = y - x = 0 \implies \hat{x} = y \implies \sup_{\|x\|_2 \leq 1} \left( y^T x - \frac{1}{2}x^T x \right) = \frac{1}{2}y^T y, \quad \|y\|_2 \leq 1$$

For the  $\|x\|_2 > 1$  case, consider the following:

- In the range of  $y$  that satisfies  $\|y\|_2 \leq 1$ , we have that:

$$\begin{aligned} y^T x &\leq \|y\|_2 \|x\|_2 \\ y^T x - \|x\|_2 &\leq (\|y\|_2 - 1) \|x\|_2 \\ y^T x - (x^T x)^{\frac{1}{2}} &\leq \left( (y^T y)^{\frac{1}{2}} - 1 \right) \|x\|_2 \\ y^T x - (x^T x)^{\frac{1}{2}} + \frac{1}{2} &\leq \left( (y^T y)^{\frac{1}{2}} - 1 \right) \|x\|_2 + \frac{1}{2} \end{aligned}$$

Notice that since  $\|y\|_2 = (y^T y)^{\frac{1}{2}} \leq 1$ , the  $\left((y^T y)^{\frac{1}{2}} - 1\right) \|x\|_2$  term is negative. To maximize this bound, we can only minimize  $\|x\|_2$ , which may only approach 1 as the smallest value here. Since  $y^T x \leq \|y\|_2 \|x\|_2$  is a bound where equality is achievable for  $\|x\|_2 > 1$  (i.e. the bound is tight), we have the following supremum:

$$\sup_{\|x\|_2 > 1} \left( y^T x - (x^T x)^{\frac{1}{2}} + \frac{1}{2} \right) = (y^T y)^{\frac{1}{2}} - \frac{1}{2}, \quad \|y\|_2 \leq 1$$

- In the range of  $y$  that satisfies  $\|y\|_2 > 1$ , consider the dual norm formula for the 2-norm (which is self-dual):

$$\|y\|_2 = \sup_z \frac{y^T z}{\|z\|_2}$$

Let  $z^*$  be the argument that gives the supremum in this case. Then:

$$\begin{aligned} \|y\|_2 &> 1 \\ \frac{y^T z^*}{\|z^*\|_2} &> 1 \\ y^T z^* &> \|z^*\|_2 \\ y^T z^* - \|z^*\|_2 &> 0 \\ y^T z^* - ((z^*)^T (z^*))^{\frac{1}{2}} &> 0 \end{aligned}$$

Let  $x = tz^*$ ,  $t \in \mathbb{R}_{++}$ . We can look at what happens to our supremum formula as  $t \rightarrow \infty$ :

$$\begin{aligned} \lim_{t \rightarrow \infty} \left( y^T x - (x^T x)^{\frac{1}{2}} + \frac{1}{2} \right) &= \lim_{t \rightarrow \infty} \left( ty^T z^* - ((tz^*)^T (tz^*))^{\frac{1}{2}} + \frac{1}{2} \right) \\ &= \lim_{t \rightarrow \infty} \left( ty^T z^* - (t^2 (z^*)^T (z^*))^{\frac{1}{2}} + \frac{1}{2} \right) \\ &= \lim_{t \rightarrow \infty} \left( ty^T z^* - t((z^*)^T (z^*))^{\frac{1}{2}} + \frac{1}{2} \right) \\ &= \lim_{t \rightarrow \infty} \left( t \left( y^T z^* - ((z^*)^T (z^*))^{\frac{1}{2}} \right) + \frac{1}{2} \right) \rightarrow \infty \\ \implies \sup_{\|x\|_2 > 1} \left( y^T x - (x^T x)^{\frac{1}{2}} + \frac{1}{2} \right) &= \infty, \quad \|y\|_2 > 1 \end{aligned}$$

Putting it all together, we have:

$$f^*(y) = \begin{cases} \max \left\{ \frac{1}{2} y^T y, (y^T y)^{\frac{1}{2}} - \frac{1}{2} \right\} & \|y\|_2 \leq 1 \\ \infty & \|y\|_2 > 1 \end{cases} = \begin{cases} \frac{1}{2} y^T y & \|y\|_2 \leq 1 \\ \infty & \|y\|_2 > 1 \end{cases}$$

## II. 2. Compressed Sensing. (18 pts)

For compressed sensing, we formulate the sensed signal  $y_t$  with an equation  $y = Ax + b$ , where  $y \in R^m, A \in R^{mn}, x \in R^n, b \in R^m$ . We have vector  $y$  contains the input (hw3\_signal.txt), which is generated by a sparse linear combination of sinusoidal waves,  $y_t \approx \sum_k a_{2k} \sin(2\pi f_k t) + a_{2k+1} \cos(2\pi f_k t)$ , matrix  $A$  is a dictionary of *sine* and *cosine* patterns and your job is to figure out the coefficients  $a_k$  using vector  $x$  with noise  $b$ . It is known that the data is a 5 second data sampled at 1k Hz. Also, the frequencies (in the unit of Hertz) are in the set of  $\{i | i \in \mathbb{N}, 1 \leq i \leq 100\}$ .

Please formulate the problem into an optimization problem that minimizes the sum of the squared  $L_2$  norm of the noise  $b$  and the  $L_\alpha$  norm of the frequency components  $x$ . Once you derive the formulation, use the convex optimization programming tool to compute the numerical values of the amplitudes and figure out the frequencies of the signal.

II.2.1. Work on the following with  $\alpha = 2$ , i.e.  $\min_{x,b} \|x\|_{\alpha=2} + \lambda \|b\|_2$ , such that  $y = Ax + b$ . (6 pts)

- (i) Write the formulation. Note that we need a weight  $\lambda$  to balance between the squared  $L_2$  norm of the noise and the  $L_\alpha$  norm of the frequency components in the objective function.
- (ii) Show your results with three or more samples of weight  $\lambda$ .
- (iii) Show and explain your best choice of the weight.

II.2.2. Repeat the items of 1 with  $\alpha = 1$ . (6 pts)

II.2.3. Try to repeat the items of 1 with  $\alpha = 0.5$ . Explain your solution. (6 pts)

Hint: You should get the major three frequency components in your solution. ps: CVX is a Matlab-based modeling system for convex optimization created by Professor Stephen Boyd. For more details, please refer to the tutorial for CVX: [https://web.stanford.edu/class/ee364a/lectures/cvx\\_tutorial.pdf](https://web.stanford.edu/class/ee364a/lectures/cvx_tutorial.pdf)

```

# download the data
!wget https://cseweb.ucsd.edu//classes/wi22/cse203B-a/hws/hw3_signal.txt

--2022-02-03 18:08:01-- https://cseweb.ucsd.edu//classes/wi22/cse203B-a/hws/hw3
Resolving cseweb.ucsd.edu (cseweb.ucsd.edu)... 132.239.8.30
Connecting to cseweb.ucsd.edu (cseweb.ucsd.edu)|132.239.8.30|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 127528 (125K) [text/plain]
Saving to: 'hw3_signal.txt'

hw3_signal.txt      100%[=====>] 124.54K   590KB/s   in 0.2s

2022-02-03 18:08:02 (590 KB/s) - 'hw3_signal.txt' saved [127528/127528]

import cvxpy as cp # cvxpy for convex opt.
import numpy as np # numpy for linear algebra
import matplotlib.pyplot as plt # matplotlib for plotting

# solution
#sln = np.array([0.000000000000000000e+00, 1.999100619435016934e+00, 0.000000000000000000e+00])
sln = np.array([0.,0.,1.99910062,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,1.010533])

# Load and plot the observed signal

```

```

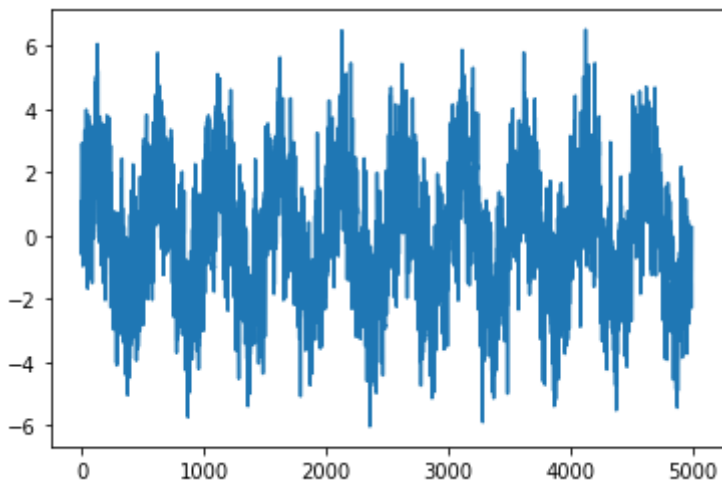
signal = np.loadtxt('hw3_signal.txt')
signal = np.reshape(signal, (-1,1))
print('length of signal (5s, 1k Hz): {:.2f} mean of signal: {:.2f} std of signal: {:.2f}')
plt.plot(signal)

```

```

length of signal (5s, 1k Hz): 5001.00 mean of signal: 0.00 std of signal: 2.02
[<matplotlib.lines.Line2D at 0x7f6caf69b850>]

```



```

# Known parameters

```

```

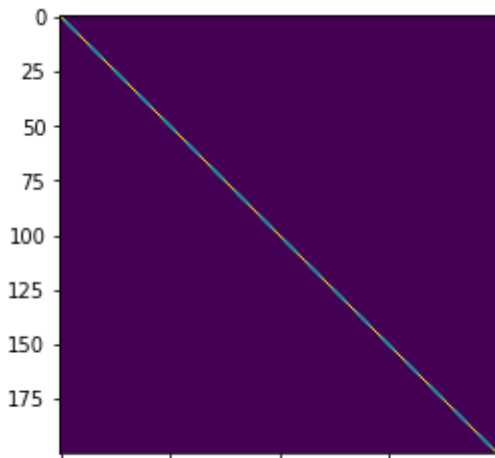
t = np.linspace(0,5,num=5001) # range in seconds
f = np.linspace(1,100, num=100) # frequency range
Fsin = lambda fs, ts: np.sin(2*np.pi*fs*ts)
Fcos = lambda fc, tc: np.cos(2*np.pi*fc*tc)
Phi = np.array([[summand for fk in f
                 for summand in (Fsin(fk, tk), Fcos(fk,tk))]
                 for tk in t]) # Fourier basis for the vector space of t-periodic func

# are the columns of Phi orthogonal?
print(Phi.shape)
plt.imshow(Phi.T@Phi)

```

(5001, 200)

<matplotlib.image.AxesImage at 0x7f6caf1fe5d0>



```
# Problem parameters / variables
```

```
x = cp.Variable((200,1)) # sparse set of coefficients on the basis functions (variable)
```

```
lamb = cp.Parameter(nonneg=True) # tradeoff reconstruction / sparsity
lamb.value = 1e-1
```

```
A = cp.Parameter((5001,200)) # dictionary param
A.value = Phi
```

```
y = cp.Parameter((5001,1)) # observation param
y.value = signal
```

```
alpha = 1 # p-norm regularization- want to use one that encourages sparsity (e.g. alpha=1)
```

```
# Solve problem & print loss
```

```
# \min_x ||x||_{\alpha} + \lambda*(||Ax - y||_2)^2
```

```
obj = cp.Minimize(cp.norm(x, alpha) + lamb*cp.norm(y-A*x, 2)**2)
```

```
prob = cp.Problem(obj)
```

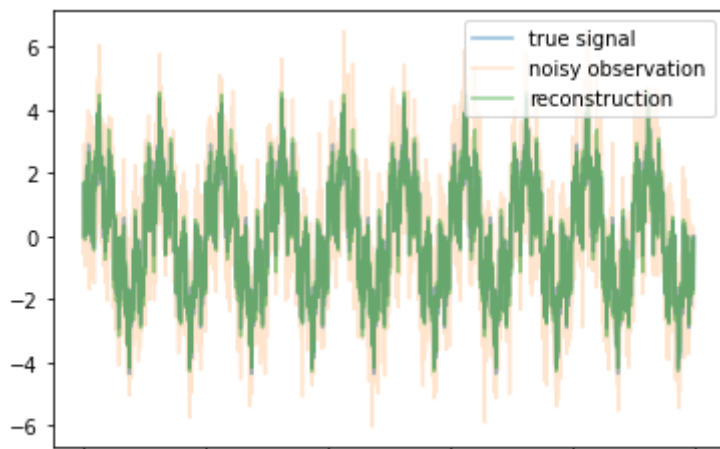
```
print('final objective')
prob.solve()
```

```
final objective
476.2807125774311
```

```
# Plot signal reconstruction
```

```
plt.plot(Phi@sln,label='true signal',alpha=0.5)
plt.plot(signal,label='noisy observation',alpha=0.2)
plt.plot(Phi@x.value,label='reconstruction',alpha=0.5)
plt.legend()
```

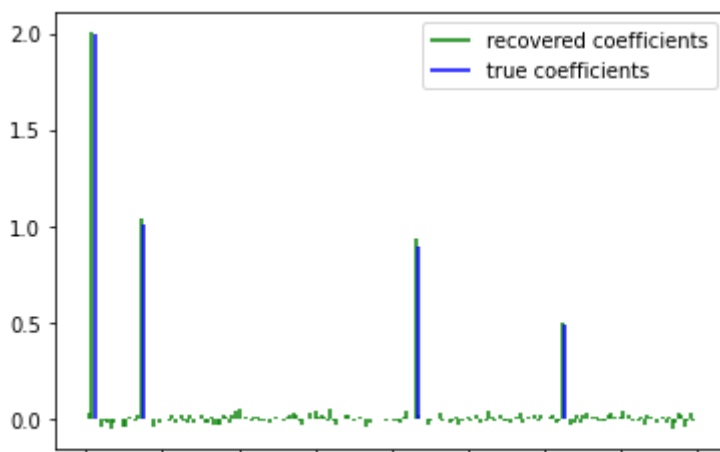
<matplotlib.legend.Legend at 0x7f6caeb18710>



```
# Plot recovered coefficient values (solution is offset by 1)
```

```
plt.vlines(np.arange(200),0,x.value,label='recovered coefficients',color='green')
plt.vlines(np.arange(201),0,[0]+list(sln),label='true coefficients',color='blue')
plt.legend()
```

<matplotlib.legend.Legend at 0x7f6cad416d10>





```

# How to choose \lambda?
# depends on your tradeoff. Usually want to choose one
# corresponding to a "knee point" (e.g. sparsest solution for a given reconstruction t

from tqdm import tqdm

loss = []
recons = []
sparsity = []
lambdas = [1,5e-1,1e-1,1e-2,1e-3]
for l in tqdm(lambdas):
    lamb.value = l
    obj = cp.Minimize(cp.norm(x, alpha) + lamb*cp.norm(y-A*x, 2)**2)
    prob = cp.Problem(obj)
    lval = prob.solve()
    loss.append(lval)
    recons.append(np.linalg.norm(signal - Phi*x.value)**2)
    sparsity.append(np.linalg.norm(x.value,alpha))

fig, axs = plt.subplots(1,3,sharex=True,figsize=(12,4))
axs[0].set_title('loss')
axs[0].plot(np.log(lambdas), loss)
axs[1].set_title('reconstruction')
axs[1].plot(np.log(lambdas), sparsity)
axs[2].set_title('sparsity')
axs[2].plot(np.log(lambdas), recons)

axs[1].set_xlabel('log(\lambda)')

```

```

100% |██████████| 5/5 [01:15<00:00, 15.12s/it]
Text(0.5, 0, 'log(\lambda)')

```

