ECE 6270, Spring 2026

Homework #6

Due Sunday, April 26, at 11:59pm

1. Prepare a one paragraph summary of what we talked about since the last assignment. I do not want just a bulleted list of topics, I want you to use complete sentences and establish context (Why is what we have learned relevant? How does it connect with other classes?). The more insight you give, the better.

2. You have a large amount of money M that you are going to gamble on a horse race. You want to be smart about it, though.

   There are N horses running in the race. You will divide up your money to place a bet of x_n on each of them. Clearly,

$$\sum_{n=1}^{N} x_n = M.$$

   As with any parimutuel betting scenario, if horse n wins, the payout to you is proportional to the amount you bet on horse n versus what everybody else (the "public") bet on horse n. If you wager x_n on horse n and the public wagers s_n then

$$\begin{aligned}\text{payout if horse } n \text{ wins} &= C \cdot (\text{total amount of money bet on all horses}) \cdot \frac{x_n}{x_n + s_n} \\ &= C \cdot \left(M + \sum_{n=1}^{N} s_n \right) \frac{x_n}{x_n + s_n}.\end{aligned}$$

   The constant C above is less than 1, and represents the fact that the track takes a cut of all the bets (the "vigorish" or "vig" is $1 - C$). A typical value of C might be 0.8 or 0.9.

   The reason you are betting is that you have two pieces of key knowledge about this race. First, you know the *actual* probability p_n that horse n will win. Second, you know s_n, the amount that the public will end up placing on horse n.

   (a) With your knowledge of the probabilities $\{p_n\}_{n=1}^{N}$ and public money $\{s_n\}_{n=1}^{N}$, write down a convex optimization program answer will tell you how much to bet on each horse to maximize your expected return. (In the end, you should be maximizing a concave function over a convex set.)

   (b) Using Fenchel duality, show how this expected payout can be computed by solving an optimization program in one variable. (Hint: look at the resource allocation example in the notes.) All of the relevant functions are given to you here, so you can (and should) compute their conjugates explicitly.

   (c) Show how the primal optimal solution (the best x_n) can be recovered from the (single variable) dual solution.

   (d) Here are the track odds right before closing as you are making your bet:[1]:

[1] These are close to the odds from the morning of the 2022 Kentucky Derby, but I have just included the 8 most interesting horses (out of 20) and adjusted the odds accordingly to give the implied advantage back to the track.

1.	Zandon	3-1
2.	Epicenter	7-2
3.	Messier	5-1
4.	Mo Donegal	6-1
5.	White Abarrio	6-1
6.	Taiba	10-1
7.	Zozos	20-1
8.	Rich Strike	80-1

Saying horse n has track odds A-B simply means that a bet of B will yield A in profit (and a total return of $A+B$) should horse n win. The expected return of betting on horse n is zero (meaning it is a "fair bet") if the probability that horse n wins is $B/(A+B)$. These odds are set based on how the public has wagered so far[2] and are given by the relation

$$\frac{B}{B+A} = \frac{1}{C} \cdot \frac{s_n}{\sum_{n=1}^{N} s_n}.$$

That is, the track-odds assign probabilities based on the proportion of betting on each of the horses after the vig has been incorporated.

Note that by examining these odds, you can infer what the vig is $(1-C)$. If you know the total amount the public has wagered you can also determine the s_n's. Suppose that the public has wagered a total of \$20 million on this race.

You happen to know the trainer for Rich Strike, and he thinks that the horse has a much better chance of winning the race than public believes. Based on this information, you judge the true win probabilities as

1.	Zandon	25%
2.	Epicenter	25%
3.	Messier	10%
4.	Mo Donegal	8%
5.	White Abarrio	8%
6.	Taiba	8%
7.	Zozos	6%
8.	Rich Strike	10%

You have \$500,000. How much do you bet on each horse? What is your expected return? (You should calculate your return using the value of C and the s_n's determined by your optimization problem.) What is the standard deviation (a simple measure of risk) of your expected return?

(e) Rich Strike won the race.[3] How much profit did you make?

(f) Explore how your answer changes if you bet larger amounts (in terms of the strategy, expected return, and risk). Can you explain why your betting strategy changes as the amount wagered goes up?

[2]These odds are based on everyone's money except yours that has been bet so far.

[3]This actually happened, and is considered the one of the most stunning results in the history of the Kentucky Derby. Watch it here: https://www.youtube.com/watch?v=wIYD42DV3Ro.

3. Let f be a convex function on $\mathbb{R}^N$ with closed epigraph, and let f^* be its Fechel conjugate

$$f^*(\boldsymbol{\nu}) = \sup_{\boldsymbol{x}\in\mathbb{R}^N} \left(\boldsymbol{\nu}^{\mathrm{T}}\boldsymbol{x} - f(\boldsymbol{x})\right).$$

   (a) Use the properties of the Fenchel conjugate in the notes to show that for any $\boldsymbol{x}$

$$\mathrm{prox}_f(\boldsymbol{x}) + \mathrm{prox}_{f^*}(\boldsymbol{x}) = \boldsymbol{x}.$$

   (b) Generalize your argument above to show that for any $\lambda > 0$

$$\mathrm{prox}_{\lambda f}(\boldsymbol{x}) + \lambda\,\mathrm{prox}_{\lambda^{-1} f^*}(\boldsymbol{x}/\lambda) = \boldsymbol{x}.$$

   (c) Suppose that you have code that computes the proximal operator for $f(\boldsymbol{x}) = \|\boldsymbol{x}\|_\infty$. (You have this code from an earlier homework!). Describe how you could use this code to compute projections onto the ℓ_1 ball; that is, given $\boldsymbol{z} \in \mathbb{R}^N$, solve

$$P_{\tau\ell_1}(\boldsymbol{z}) = \arg\min_{\|\boldsymbol{y}\|_1\le\tau} \|\boldsymbol{y} - \boldsymbol{z}\|_2^2$$

4. There is a close relationship between the subdifferential of a function f and its Fenchel conjugate f^*. In this problem you will explore some of these connections. In the problems below, we will assume that f is convex.

   (a) Show that if $\boldsymbol{\nu} \in \partial f(\boldsymbol{x})$, then $\boldsymbol{x} \in \partial f^*(\boldsymbol{\nu})$.

   (b) Now suppose that f is closed (and hence $f^{**} = f$). Show that if $\boldsymbol{x} \in \partial f^*(\boldsymbol{\nu})$ then $\boldsymbol{\nu} \in \partial f(\boldsymbol{x})$.

5. Consider the optimization program

$$\underset{\boldsymbol{x}\in\mathbb{R}^N}{\text{minimize}}\ \|\boldsymbol{x}\|_1 \quad \text{subject to} \quad \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}, \tag{1}$$

   where $\boldsymbol{A}$ is a $M \times N$ matrix with $M < N$ and has full row rank ($\mathrm{rank}(\boldsymbol{A}) = M$).

   (a) Find the Lagrange dual program. (The answer to this is basically in the notes.)

   (b) Using strong duality, show that any primal solution $\boldsymbol{x}^\star$ and any dual solution $\boldsymbol{\nu}^\star$ obey

$$\text{for } \boldsymbol{u}^\star = \boldsymbol{A}^{\mathrm{T}}\boldsymbol{\nu}^\star, \quad \begin{cases} u_n^\star = 1, & x_n^\star > 0 \\ u_n^\star = -1, & x_n^\star < 0 \\ |u_n^\star| \le 1, & x_n^\star = 0 \end{cases}.$$

   (c) Argue that there is a solution to (1) that has at most M non-zero components. Do this by showing that if there is a solution $\boldsymbol{x}^\star$ to (1) with $M' > M$ nonzero terms, then there is a $\boldsymbol{x}^{\star\star}$ with $\|\boldsymbol{x}^{\star\star}\|_1 = \|\boldsymbol{x}^\star\|_1$ that has $M' - 1$ nonzero terms. (Hint: Since $\mathrm{rank}(\boldsymbol{A}) = M$, any $M + 1$ columns of $\boldsymbol{A}$ are linearly dependent. That is, given any $M + 1$ locations, we can find a vector in the null space of $\boldsymbol{A}$ that is nonzero only on these locations.)

6. Training a soft-margin support vector machine amounts to solving an optimization program of the form

$$\underset{\boldsymbol{x}\in\mathbb{R}^N}{\text{minimize}} \sum_{m=1}^{M} \max(\boldsymbol{a}_m^{\mathrm{T}}\boldsymbol{x}, b_m) + \tau\|\boldsymbol{x}\|_2^2. \tag{2}$$

(a) Derive the prox operator for

$$g(\boldsymbol{z}) = \sum_{m=1}^{M} \max(z_m, b_m).$$

That is, give the solution to

$$\text{prox}_{\alpha g}(\boldsymbol{u}) = \arg\min_{\boldsymbol{z}} \left(g(\boldsymbol{z}) + \frac{1}{2\alpha}\|\boldsymbol{z}-\boldsymbol{u}\|_2^2 \right).$$

The key to this is of course computing the subdifferential of g.

(b) Using your answer to (a), write down an ADMM algorithm for solving (2). (Hint: use $\boldsymbol{z} = \boldsymbol{A}\boldsymbol{x}$, where the $\boldsymbol{a}_m^{\mathrm{T}}$ are the rows of $\boldsymbol{A}$.) You answer will be iterations over a sequence of least-squares problems (for which you should provide the closed-form solution) and prox problems of the type you just solved.

7. In this problem we will explore two alternative approaches to solving a simple variant of the least squares problem where we add the constraint that the solution is non-negative, i.e., we wish to solve

$$\underset{\boldsymbol{x}\in\mathbb{R}^N}{\text{minimize}} \ \frac{1}{2}\|\boldsymbol{y}-\boldsymbol{A}\boldsymbol{x}\|_2^2 \quad \text{subject to} \quad \boldsymbol{x} \geq \boldsymbol{0}.$$

This is natural in many practical applications where the entries of $\boldsymbol{x}$ have physical interpretations (e.g., light intensity, power, concentration of some physical material, etc.) that don't really make sense as negative quantities. We will assume that $\boldsymbol{A}$ is an $M \times N$ matrix with $M > N$ and that $\boldsymbol{A}$ has full column rank ($\text{rank}(\boldsymbol{A}) = N$).

The following python code sets up a random instance of this problem:

```
import numpy as np
m = 40
n = 15
np.random.seed(6270)
A = np.random.randn(m, n)
y = np.random.randn(m)
```

In the problems below, use this instantiation and submit your code as a single file.

(a) Implement the projected gradient descent approach described in the notes. Note that even though you are solving a least-squares problem, it is not clear how to solve for the optimal step size with the positivity constraint. You should either use a line search to choose α, or just take a fixed step size. For guidance, the theory guarantees convergence if $\alpha \leq 1/\|\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\|_2$.

You also need to be careful defining a stopping criterion. You cannot expect that the norm of the gradient will be zero at the solution. Instead you could either define a stopping criterion involving $\|\boldsymbol{x}^{(k)} - \boldsymbol{x}^{(k-1)}\|_2$, or alternatively you could do something inspired by part (c) above.

(b) Derive the Lagrangian function for this optimization problem (pay careful attention to the sign of each term).

(c) Show that the dual optimization problem is itself another nonnegative least squares problem. Do this by deriving the dual function $d(\boldsymbol{\lambda})$ by first finding the $\boldsymbol{x}$ that minimizes the Lagrangian $L(\boldsymbol{x}, \boldsymbol{\lambda})$, and then plugging this into $L(\boldsymbol{x}, \boldsymbol{\lambda})$. Simplify the dual optimization problem as much as possible.

(d) Given the dual solution $\boldsymbol{\lambda}^\star$, give a closed-form expression for $\boldsymbol{x}^\star$. This is of course the minimizer of $L(\boldsymbol{x}, \boldsymbol{\lambda}^\star)$ ... you probably derived the expression for this already when you were solving part (c).

(e) Compute the gradient of the dual $\nabla d(\boldsymbol{\lambda})$. Use this to implement a "dual projected gradient ascent" algorithm on the dual. Run it on the same problem instance as in part (a), and verify that the solutions are the same. (You will of course have to use your dual solution to get the primal solution.)

(f) For your solutions to this problem instance, verify that strong duality holds, $d(\boldsymbol{\lambda}^\star) = \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}^\star\|_2^2$.

(g) Write down the KKT conditions. Verify that they hold for your $\boldsymbol{x}^\star, \boldsymbol{\lambda}^\star$ for the given problem instance.