COMPSCI 514: Optional Problem Set 5

Released: Wednesday 11/18.

Due: Monday 11/30 by 8:00pm in Gradescope.

This problem set is optional. If you complete it, we will use your 4 out of 5 highest problem set grades in computing your final course grade.

Instructions:

- Each group should work together to produce a single solution set. One member should submit a solution pdf to Gradescope, marking the other members as part of their group.
- You may talk to members of other groups at a high level about the problems but not work through the solutions in detail together.
- You must show your work/derive any answers as part of the solutions to receive full credit.

1. Convex Functions and Sets Warm Up (13 points)

Recall that for a convex set $\mathcal{S}$, the projection function $P_{\mathcal{S}}(\vec{z})$ returns $\vec{y} \in \arg\min_{\vec{y}\in\mathcal{S}} \|\vec{z}-\vec{y}\|_2$.

1. (3 points) Show that for any $\vec{b} \in \mathbb{R}^d$ and $W \in \mathbb{R}$, the set $\mathcal{S}_{\vec{b},W} = \{\vec{v} \in \mathbb{R}^d : \langle \vec{b}, \vec{v}\rangle \geq W\}$ is convex. What is the projection function $P_{\mathcal{S}_{\vec{b},W}}(\vec{z})$? Prove that this is the projection function.
2. (4 points) Show that for any convex set $\mathcal{S}$, the projection function is unique. I.e., for any $\vec{z}$, there is a unique $\vec{y} \in \mathcal{S}$ with $\vec{y} = \arg\min_{\vec{y}\in\mathcal{S}} \|\vec{z}-\vec{y}\|_2$. **Hint:** Try a proof by contradiction that uses the definition of a convex set. It might help to draw a picture first to get the intuition before trying to write a formal proof.
3. Consider the minimum cut problem on a graph G with n nodes and Laplacian $\mathbf{L}$.
   (a) (2 points) Argue that this problem is equivalent to solving:

$$\min_{\substack{\vec{x}\in\{-1,1\}^n \\ \vec{x}\neq[1,1,...,1],\ \vec{x}\neq[-1,-1,...,-1]}} c(\vec{x}) \text{ where } c(\vec{x}) = \vec{x}^T\mathbf{L}\vec{x}.$$

   (b) (2 points) Prove that the objective function $c(\vec{x}) = \vec{x}^T\mathbf{L}\vec{x}$ is convex. **Hint:** It may be helpful to use that the sum of convex functions is convex.

   (c) (2 points) Is min-cut a convex optimization problem over a convex constraint set?

2. Gradient Descent, Linear Systems, and the Power Method (13 points)

Consider any data matrix $\mathbf{A} \in \mathbb{R}^{n \times d}$ and label vector $\vec{b} \in \mathbb{R}^n$. Define the least squares regression function:

$$f(\vec{x}) = \|\mathbf{A}\vec{x} - \vec{b}\|_2^2.$$

1. (2 points) Is $f(\vec{x})$ G-Lipschitz for some finite G? Why or why not? Can the gradient descent analysis shown in class be applied? **Hint:** Compute the gradient as a function of $\mathbf{A}, \vec{b}$, and $\vec{x}$. This will be useful in part (2) as well.

2. (3 points) Let $\mathbf{B} = \mathbf{I} - 2\eta\mathbf{A}^T\mathbf{A}$ and $\vec{x}_\star = \arg\min_{\vec{x} \in \mathbb{R}^d} f(\vec{x})$. Prove that the gradient descent update for optimizing $f(\vec{x})$ is equivalent to:

$$\vec{x}^{(i+1)} = \mathbf{B}(\vec{x}^{(i)} - \vec{x}_\star) + \vec{x}_\star.$$

   Hint: Write down the gradient update equation and an expansion of the above equation. Then identify what you need to prove for these two equations to be equal.

3. (2 points) Prove that for $\eta = \frac{1}{2\lambda_1(\mathbf{A}^T\mathbf{A})}$, the eigenvalues of $\mathbf{B}$ all fall in the range $\left[0, 1 - \frac{\lambda_d(\mathbf{A}^T\mathbf{A})}{\lambda_1(\mathbf{A}^T\mathbf{A})}\right]$.

4. (6 points) Use the above to show for any $\epsilon \geq 0$, after $t = O\left(\frac{\lambda_1(\mathbf{A}^T\mathbf{A})}{\lambda_d(\mathbf{A}^T\mathbf{A})} \cdot \log(1/\epsilon)\right)$ iterations, the t^{th} iterate of gradient descent satisfies $\|\vec{x}^{(t)} - \vec{x}_\star\|_2 \leq \epsilon\|\vec{x}_\star\|_2$, assuming that we initialize with $\vec{x}^{(0)} = \vec{0}$ and use the η found in part (3).

Note: The ratio $\frac{\lambda_1(\mathbf{A}^T\mathbf{A})}{\lambda_d(\mathbf{A}^T\mathbf{A})}$ is known as the *condition number* of $\mathbf{A}^T\mathbf{A}$. It is important since it governs the convergence rate of gradient descent and many other iterative methods in solving least squares regression and linear systems.

3. Online Optimization and Portfolio Management (10 points + 8 bonus)

Adapted from Homework 3 of https://www.cs.princeton.edu/courses/archive/fall16/cos521/.

Assume that you have a fixed sum of money that you would like to invest in n stocks to maximize your total return over T days. For day $t = 1, 2, \ldots, T$, let $r_i^{(t)}$ be the relative price change of stock i on day t. I.e.,

$$r_i^{(t)} = \frac{\text{Price of stock } i \text{ on day } t}{\text{Price of stock } i \text{ on day } t-1}.$$

Let $\vec{r}^{(t)} \in \mathbb{R}^n$ be the vector with i^{th} entry equal to $r_i^{(t)}$.

One common portfolio management strategy is *Constant Rebalanced Portfolio* (CRB): decide on a fixed proportion of money to put into each stock and buy/sell individual stocks each day to maintain this proportion. Intuitively, when the price of a stock drops, you will buy more shares to maintain its portion of the portfolio, and when the price goes up, you will sell, causing the strategy to generally 'buy low and sell high'.

Let Δ^n be the n-dimensional simplex: the set of all vectors $\vec{x} \in \mathbb{R}^n$ with $\vec{x}(i) \geq 0$ for all i and $\sum_{i=1}^n \vec{x}(i) = 1$. That is, the set of all valid allocations of a portfolio's value to n stocks. For a portfolio allocation $\vec{x} \in \Delta^n$, we can see that the return on day t is given by $\langle \vec{r}^{(t)}, \vec{x} \rangle$ and the total return after T days (assuming 0 transaction costs) is:

$$\prod_{t=1}^{T} \langle \vec{r}^{(t)}, \vec{x} \rangle.$$

1. (6 points) Show that finding a portfolio allocation maximizing the return over T days is equivalent to finding:

$$\vec{x} \in \underset{\vec{x} \in \Delta^n}{\arg\min} f(\vec{x})$$

where $f(\vec{x}) \overset{\text{def}}{=} \sum_{t=1}^{T} -\log(\langle \vec{r}^{(t)}, \vec{x} \rangle)$. Show that this optimization problem is a convex optimization problem over a convex constraint set, and hence can be approximately minimized with projected gradient descent.

   Hint: You may use that for a scalar function $f : \mathbb{R} \to \mathbb{R}$, if the second derivative $f''(y)$ is positive for all y, then f is convex.

2. (4 points) What is $\vec{\nabla} f(\vec{x})$?

3. **Bonus:** (8 points) Download the S&P 500 stock data on the course page, which contains prices for 490 stocks over 1000 trading days from 2001-2005.

   (a) (2 points) If an investor places all their money in the single best stock, what is the return obtained over 1000 days (as a percentage of the initial quantity invested)? What if they place an equal amount of money initially in each stock? Finally, what if they use a CRP strategy that allocates an equal percentage of the portfolio to each stock (i.e., uses allocation $\vec{x}$ with $\vec{x}(i) = 1/n$ for all i)? **Hint:** Start by using the price data to compute $\vec{r}^{(t)}$ for $t = 1, \ldots, 999$ (let the first price be the price at day 0).

   (b) (3 points) Implement projected gradient descent to find a CRP allocation $\vec{x}$ that performs better than the single best stock over 1000 days. Report the total return when using this allocation and list the 10 stocks that are allocated the largest percentage of the portfolio (along with their respective allocations). If fewer than 10 stocks have nonzero allocations, just list those with nonzero allocations. The following may be helpful to implement the projection step onto Δ^n: https://eng.ucmerced.edu/people/wwang5/papers/SimplexProj.pdf. You may try a number of different step sizes η, reporting the best result obtained. Write a couple sentences discussing how you picked η and the iteration count.

   (c) (3 points) The above optimization of course requires knowing the stock prices ahead of time, so isn't of much use in practice. Implement *online gradient descent* to find a portfolio re-balancing strategy that an investor could actually use. Note that in this strategy, a different allocation $\vec{x}^{(t)}$ may be used at each day t. How does the return of this allocation compare to the static allocation of part (b) and to the baselines of part (a)? Again, you may try a number of different step sizes η, reporting the best result obtained. In reality, η itself would be learned based on prior data or in an online way. Write a couple sentences discussing how you picked η and how your implementation differs from the gradient descent implementation of part (b).