COMPSCI 514: Optional Problem Set 5

Due: December 13th by 11:59pm in Gradescope.

This problem set is optional. If you complete it, the grade can be used to replace your lowest problem set 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 (14 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. (4 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 (2 points). What is the projection function $P_{\mathcal{S}_{\vec{b},W}}(\vec{z})$? Prove that this is the projection function (2 points).
2. (2 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:

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

   (b) (2 points) Prove that a sum of two convex functions is always convex.
   (c) (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 part (2) here.
   (d) (2 points) Is min-cut a convex optimization problem over a convex constraint set?

2. Understanding Gradient Descent (8 points)

1. (2 points) In our gradient descent analysis we showed that for large enough t, $\frac{1}{t}\sum_{i=1}^{t} f(\vec{\theta}^{(i)}) \leq f(\vec{\theta}^*) + \epsilon$ which implies that $f(\hat{\theta}) \leq f(\vec{\theta}^*) + \epsilon$ for the best iterate $\hat{\theta} = \arg\min_{\vec{\theta}^{(1)},\ldots,\vec{\theta}^{(t)}} f(\vec{\theta}^{(i)})$. Prove that if we instead set $\bar{\theta} = \frac{1}{t}\sum_{i=1}^{t} \vec{\theta}^{(i)}$ (i.e., $\bar{\theta}$ is the average iterate) then we also have $f(\bar{\theta}) \leq f(\vec{\theta}^*) + \epsilon$. This strategy is often used, e.g., when using stochastic gradient descent for large datasets, since determining the best iterate can be much more expensive than just storing a running average. **Hint:** Use that f is convex.

2. Let $f : \mathbb{R}^d \to \mathbb{R}$ be a G-Lipschitz function, i.e., $\|\nabla f(\theta)\|_2 \leq G$ for all θ.

   (a) (2 points) If $\theta^{(i+1)} = \theta^{(i)} - \eta \nabla f(\theta^{(i)})$, give an upper bound on $\|\theta^{(i+1)} - \theta^{(i)}\|_2$ in terms of η and G.

   (b) (2 points) In our fixed step size gradient algorithm we set $t = \frac{R^2G^2}{\epsilon^2}$ and $\eta = \frac{R}{G\sqrt{t}}$. Under these settings, what is the worst case increase in function value from step i to step $i+1$? I.e., give an upper bound on $f(\theta^{(i+1)}) - f(\theta^{(i)})$. Does this make intuitive sense? **Hint:** Use part (1).

   (c) (2 points) Consider the case of projected gradient descent over a convex set $\mathcal{S}$. So $\theta^{(i+1)} = P_{\mathcal{S}}(\theta^{out})$ for $\theta^{out} = \theta^{(i)} - \eta \nabla f(\theta^{(i)})$. Show that the bounds of (a), (b) still hold.

3. Gradient Descent, Linear Systems, and the Power Method (14 points)

Consider any $\mathbf{A} \in \mathbb{R}^{n \times d}$ with SVD $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$. One of the most classic data fitting methods, least squares regression is: given a vector $\vec{b} \in \mathbb{R}^n$, find:

$$\vec{x}_* \in \arg\min_{\vec{x} \in \mathbb{R}^d} \|\mathbf{A}\vec{x} - \vec{b}\|_2^2. \tag{1}$$

The rows of $\mathbf{A}$ represent d-dimensional data points, the entries of $\vec{b}$ represent observations at these points, and $\mathbf{A}\vec{x}_*$ is the 'line of best fit', which attempts to fit these observations as closely as possible with a linear function of the rows.

1. (2 points) Prove that $\vec{x}_* = \mathbf{V}\mathbf{\Sigma}^{-1}\mathbf{U}^T\vec{b}$ satisfies equation (1) above. **Hint:** The solution will involve a projection matrix and does not require using any calculus.

2. (2 points) Define the least squares regression function as $f(\vec{x}) = \|\mathbf{A}\vec{x} - \vec{b}\|_2^2$. 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.

3. (2 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 and use part (1).

4. (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 $\left[0, 1 - \frac{\lambda_d(\mathbf{A}^T\mathbf{A})}{\lambda_1(\mathbf{A}^T\mathbf{A})}\right]$.

5. (2 points) Use the above to show for after t iterations, the t^{th} iterate of gradient descent satisfies $\|\vec{x}^{(t)} - \vec{x}_\star\|_2 \leq \left(1 - \frac{\lambda_d(\mathbf{X}^T\mathbf{X})}{\lambda_1(\mathbf{X}^T\mathbf{X})}\right)^t \cdot \|\vec{x}^{(0)} - \vec{x}_\star\|_2$, assuming that we use the η found in part (3). **Hint:** This question and the one below it will use some ideas from the power method proof given in class.

6. (2 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).

7. (2 points) 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. Let ϵ be a small constant (e.g., $\epsilon = .01$). Ignoring constants, compare the asymptotic runtime of computing $\vec{x}^{(t)}$ in part (6) with that of computing an exact solution via SVD in part (1). When is one faster than the other? **Note:** Assume that a full SVD requires $O(nd^2)$ time to compute and that multiplying a vector by $\mathbf{A}$ or $\mathbf{A}^T$ requires $O(nd)$ time.