10725/36725 Optimization
Homework 1

Due September 20, 2012 at beginning of class

Instructions: There are four questions in this assignment. Please submit your homework as 4 separate sets of pages with your name and userid on each set. For the last question which involves coding, please print out your code and graphs and attach them to the written part of your homework. Refer to the course webpage for policies regarding collaboration, due dates, and extensions.

1 Convexity (Kevin)

1.1 Sets

Let $A \subseteq \mathbb{R}^n$ be a closed set with non-empty interior that has a supporting hyperplane at every point on its boundary.

(a) [4 pts] Show that A is convex.

Let $X, Y \subseteq \mathbb{R}^n$ be disjoint convex sets, let $\{x \mid a^T x + b = 0\}$ be a separating hyperplane and let $f(x) = Cx + d$ be a function, where $C \in \mathbb{R}^{m \times n}, d \in \mathbb{R}^m$.

(b) [3 pts] Given that the sets $f(X)$ and $f(Y)$ are disjoint, find a hyperplane, $\{y \mid \alpha^T y + \beta = 0\}$, that separates $f(X)$ and $f(Y)$.

Here, $f(A) = \{y \mid y = f(x), x \in A\}$.

1.2 Voronoi Decomposition

Let $a, b \in \mathbb{R}^n$ such that $a \neq b$.

(a) [2 pts] Show that the set $\{x \mid \|x - a\|_2 \leq \|x - b\|_2\}$ is a halfspace.

Let $x_1, \dots, x_k \in \mathbb{R}^n$ and let $V_i = \{x \mid \|x_i - x\|_2 \le \|x_j - x\|_2, j \neq i\}$.

(b) [2 pts] Show that V_1 is a polyhedron. That is, $V_1 = \{x \mid A_1 x \le b_1\}$.

Let $P_i = \{x \mid A_i x \le b_i\}$ be disjoint polyhedra that cover $\mathbb{R}^n$.

(c) [3 pts] Does there exist points $x_1, \dots, x_k \in \mathbb{R}^n$ such that $V_i = P_i$ for $i = 1, \dots, k$? If so, provide the points, otherwise construct a counterexample.

1.3 Farkas' Lemma

Let $A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m$ such that there is some x where $Ax = b$.

(a) [6 pts] Show that **either** there exists $x > 0$ where $Ax = b$, **or** there exists λ such that $A^T\lambda \ge 0$, $A^T\lambda \neq 0$ and $b^T\lambda \le 0$.

Hint: First show that $c^T x = d$ for all x such that $Ax = b$ if and only if there exists λ such that $c = A^T\lambda, d = b^T\lambda$.

1.4 Functions

(a) [3 pts] Show that the function $f(x) = (\sum_{i=1}^n x_i^p)^{1/p}$ is concave on $\mathbb{R}^n_{++}$ for all $p \in (0, 1)$.

Hint: consider the log-sum-exp and geometric mean functions in Boyd.

Let $G = (V, E, c, s, t)$ be an undirected graph with edge capacities $c(u, v) \ge 0$, source vertex s and sink vertex t. We say a vector $y \in \mathbb{R}^E$ is a valid *s-t-flow* if:

- $y(u, v) \le c(u, v), \forall (u, v) \in E$, (*Capacity*)
- $y(u, v) = -y(v, u), \forall (u, v) \in E$, and (*Skew Symmetry*)
- $\sum_{v \in V} y(u, v) = 0, \forall u \in V \setminus \{s, t\}$. (*Flow Conservation*)

Let $f(x) = \min_y \|x - y\|$, subject to: y is a valid s-t-flow.

(b) [2 pts] Show that $f(x)$ is convex.

2 The meanest nice functions (Shiva)

In this question, as in class, we consider optimizing 'nice' functions:

- convex on $\mathbb{R}^n$,

- Lipschitz-continuous derivative: for some $L > 0$, $||\nabla f(x) - \nabla f(y)|| \leq L||x - y||$ for all x, y.

We also restrict attention to optimization algorithms that are:

- first-order: the only information the algorithm gets about the function are gradients $\nabla f(x^{(k)})$. (As in class, $x_i^{(k)}$ refers to the i'th component of $x^{(k)}$, the n-dimensional vector chosen by the algorithm on the k'th iteration.)
- gradient-summing: $x^{(k)}$ is a linear combination of $x^{(0)}$ and $\nabla f(x^{(0)})$ through $\nabla f(x^{(k-1)})$.

In this question, we will prove the lower bound that was (or soon will be) stated in class.

Run any first-order, gradient summing optimization algorithm starting from any point $x^{(0)} \in \mathbb{R}^n$ for K iterations, where $1 \leq K \leq \frac{1}{2}(n-1)$, yielding the last iterate $x^{(K)}$. There is a nice function f such that

$$||x^{(K)} - \bar{x}||^2 \geq \frac{1}{32}||x^{(0)} - \bar{x}||^2,$$

$$f(x^{(K)}) - \bar{f} \geq \frac{3L||x^{(0)} - \bar{x}||^2}{32(K+1)^2},$$

where the minimizer and minimal cost are $\bar{x} = argmin_x f(x)$ and $\bar{f} = f(\bar{x})$, respectively. Thus, even upon nice functions, the convergence of rate of any such algorithm is $\Omega(1/K^2)$.

Let's prove this lower bound together. First, assume that $x^{(0)} = 0$.

(a) [3 pts] Justify this assumption. That is, if the theorem holds for $x^{(0)} = 0$, prove it still holds when $x^{(0)} \neq 0$.

This makes $||x^{(0)} - \bar{x}|| = ||\bar{x}||$ which will come in handy later. It also makes the initial solution 'uninitialized' in every dimension. Since we required $K \leq \frac{1}{2}(n-1)$, you might have guessed there are too many dimensions for any algorithm to handle within K iterations. Indeed, we will set up f so that, after k iterations, $x^{(k)}$ is non-zero only in its first k coordinates. Since the algorithm is gradient-summing, we just need the gradients to satisfy the same condition:

$$\forall 1 \leq k \leq K, \nabla f(x^{(k-1)}) = (\underbrace{\ldots}_{k \text{ numbers}}, \underbrace{\ldots}_{n-k \text{ zeroes}}) \tag{1}$$

The lower bound on $||x^{(K)} - \bar{x}||^2$ will follow from a lower bound on the last $n - K$ coordinates of $\bar{x}$. (It would be convenient to have a formula for each coordinate of $\bar{x}$.) The lower bound on $f(x^{(K)}) - \bar{f}$ involves a bit more thought. Imagine we have a function g which agrees with f at the algorithm's queries:

$$\forall 1 \leq k \leq K, g(x^{(k)}) = f(x^{(k)}) \tag{2}$$

By definition, $\bar{g} = \text{argmin}_x g(x) \leq g(x^{(k)})$. By the previous two facts, $f(x^{(K)}) - \bar{f} \geq \bar{g} - \bar{f}$. (Now it would be convenient to have formulae for $\bar{g}$ and $\bar{f}$.)

Rather than appealing to information theory, communication complexity, the small-set expansion conjecture, or the like, we will take a direct approach: we will explicitly construct f and g which satisfy (1) and (2), and yield the aforementioned convenient formulae. Both functions come from the same mean family: let $f = \zeta^{(2K+1)}$ and $g = \zeta^{(K)}$, where

$$\zeta^{(k)}(x) = \frac{L}{4}\left[\frac{1}{2}\left(x_1^2 + \sum_{i=1}^{k-1}(x_i - x_{i+1})^2 + x_k^2\right) - x_1\right]$$

(The ugliness of the RHS is balanced by the typographic dubiousness of the LHS.) Convince yourself, or just take my word, that these functions are convex and continuously differentiable. Now for some math; include all your work!

(b) [3 pts] Show that $\nabla^2\zeta^{(k)}(x) = \frac{L}{4}A^{(k)}$ for some $n \times n$ matrix $A^{(k)}$. (Write it out for a few k and figure out what the block structure is.)

(c) [3 pts] Write $\nabla\zeta^{(k)}(x)$ in terms of $A^{(k)}$. Determine $\bar{x}_j^{(k)} = \text{argmin}_x \zeta^{(k)}(x)$ (for all coordinates $1 \leq j \leq n$) and $\bar{\zeta}^{(k)} = \zeta^{(k)}(\bar{x}^{(k)})$ in terms of k, j, and L. (Just basic calculus.)

(d) [3 pts] Check that $\nabla\zeta^{(k)}$ is L-Lipschitz. (The definition of L-Lipschitz immediately suggests a way to check.)

(e) [3 pts] Show (via induction) that $\zeta^{(k)}$ satisfies (1).

(f) [3 pts] Explain (quickly) why (2) holds.

(g) [4 pts] Lower bound $||x^{(K)} - \bar{x}||^2$ as previously described. This involves applying the previous statements (which you should clearly mark) and some algebra (which you don't need to explain.)

(h) [3 pts] Using $||\bar{x}^{(k)}||^2 \leq \frac{1}{3}(k+1)$, lower bound $f(x^{(K)}) - \bar{f}$. Now help yourself to a tasty square: $\square$

3 Alternative formulations (Wooyoung)

As discussed in class, there are often more than one way to formulate an optimization problem. In this problem, we will go through examples in which you can set up your optimization in a few alternative ways.

3.1 Rank 1 approximation of matrices

In this problem, we will set up minimizing the squared error between a given $m \times n$ matrix A and a rank 1 matrix X.

$$\text{minimize}_{\text{rank}(X)=1} ||A - X||^2$$

(a) [1.5 pts] How can you reformulate the objective function and the constraint of the above optimization by writing the space of rank 1 matrices as outer products of two vectors?

(b) [1.5 pts] One possible disadvantage of writing the space of rank 1 matrices as outer products of two vectors is degeneracy of the solutions; if a pair of vectors $\mathbf{x}$ and $\mathbf{y}$ are solutions, for any non-zero scalar a, $a\mathbf{x}$ and $\frac{1}{a}\mathbf{y}$ would also be the solutions. How would you reformulate your objective functions and constraints to avoid this degeneracy?

(c) [3 pts] We started with one optimization variable X, but now we have more than one. How can you reduce the number of optimization variables in (b)? Show your work. *Hint:* try to show that one of the optimization variables is determined when others are fixed.

(d) [2 pts] Reformulate the optimization problem in (c) into an equivalent minimization problem with a bi-linear objective function.

3.2 Partial minimization of the lasso problem

In classical setting of the lasso problem, the sparsity of solutions are enforced by $L1$-norm of the optimization variables ($\mathbf{y} \in \mathcal{R}^d$, $A \in \mathcal{R}^{d \times n}$, $\mathbf{x} \in \mathcal{R}^n$, x_i: i th element of $\mathbf{x}$).

$$\min ||\mathbf{y} - A\mathbf{x}||_2 + \lambda ||\mathbf{x}||_1$$

In this problem, we derive the partial minimization of the lasso problem.

(a) [2 pts] How would you reformulate the optimization if we only care about the sparsity of the first k elements of $\mathbf{x}$: $x_1, x_2, \cdots, x_k$.

(b) [3 pts] Show how you can eliminate $x_{k+1}, x_{k+2}, \cdots, x_n$ from the objective function.

3.3 Equality constraint

Consider an convex optimization problem with equality constraint,

$$\begin{aligned}\text{minimize } f(x)\\ \text{subject to } Ax = b\end{aligned}$$

Assume that the feasible set is non-empty.

(a) [3 pts]Can you remove the equality constraint by reformulating x as a linear function of z. What conditions does the parameters of your linear function need to satisfy?

3.4 Building a box-shape structure

You are asked to build a box-shape structure. The area of the wall paper provided to you for the job is A_{wall}, and the area of the floor cannot exceed A_{floor}. Your picky boss even wants to control the ratio of height (h) to width (w) and also the ratio of width (w) to depth (d) of the wall structure; the ratio between h and w should be in the range of [α, β], the ratio between w and d in [γ,δ]. Your mission is to maximize the volume of the box-shape structure while satisfying all the constraints.

(a) [2 pts] Formulate your optimization problem as a maximization problem. Show your objective functions and constraints.

(b) [2 pts] A geometric program is an optimization problem of the form

$$\begin{aligned}\text{minimize} \quad & f_0(x)\\ \text{subject to} \quad & f_i(x) \le 1, \quad i = 1, \cdots, m\\ & g_i(x) = 1, \quad i = 1, \cdots, p\end{aligned}$$

where $g_i(x) = c_i x_1^{a_1} x_2^{a_2} \cdots x_n^{a_n}$, $(c_i > 0, a_j \in \mathcal{R})$ and $f_i(x) = \sum_{k=1}^{K} c_{i,k} x_1^{a_{1k}} x_2^{a_2} \cdots x_n^{a_{nk}}$, $(c_{i,k} > 0, a_{ik} \in \mathcal{R})$. Recently, efficient and robust algorithms have been developed to solve even large-scale geometric programs. Convert the optimization problem in (a) into a geometric programming problem.

4 Node by node (Kevin, Shiva)

Recall that a zero-mean, p-dimensional Gaussian distribution is defined by its covariance matrix Σ. So 'learning' such a distribution from some iid samples $T = \{X^{(1)}, \ldots, X^{(m)}\}$

amounts to an estimate $\widehat{\Sigma}$ of Σ. Since we have an iid sample from a known family of distributions, a natural approach is to pick the $\widehat{\Sigma}$ which maximizes the likelihood of the sample. However, if $m < p$, the estimate is not well-defined. Further, even if $m \leq p$ but m and p are of comparable size, the maximum likelihood estimate can have high variability, leading to bad predictive performance. Since p is big, we are inclined to restrict attention to sparse distributions, for some appropriate notion of sparsity. A tempting, but unrealistic, kind of sparsity is independence of a large number of the features i and j: $\Sigma_{ij} = E_{X \sim N(0,\Sigma)}(X_i X_j) = 0$. A more realistic kind of sparsity is conditional independence of features i and j given the other features. Since the distribution is Gaussian, this happens when $\Sigma^{-1}_{ij} = 0$. Since our sparsity belief/assumption concerns Σ^{-1}, let's orient our notation around that, starting with the log-likelihood function:

$$\ell_T(\Theta) = \log\, \det(\Theta) - \mathrm{tr}(S\Theta)$$

where Θ is a symmetric positive semidefinite $p \times p$ matrix meant to estimate Σ^{-1}, the empirical covariance matrix $S = \frac{1}{m-1}\sum_{i=1}^{m}(X_i - \bar{X})(X_i - \bar{X})^T$ and the sample mean $\bar{X} = \frac{1}{m}\sum_{i=1}^{m} X_i$. The maximizer of $\ell(\Theta)$ is generally ill-defined and non-sparse. As we saw in class, one way to obtain sparse solutions is via forward stagewise regression. A bit later, we saw this algorithm is equivalent to normalized steepest descent with the 1-norm. So let's use that for $-\ell_T(\Theta)$.

(a) [10 pts] Derive $-\nabla \ell_T(\Theta)$.

(b) Download the data file from `http://www.cs.cmu.edu/~ggordon/10725-F12/hws/hw1/` in either MATLAB or ASCII format. Both have a training set T and a test set U.

(c) [8 pts] Implement normalized steepest descent with the 1-norm yourself in Matlab or R. Use a constant stepsize $\gamma = 0.01$.

(d) [9 pts] Run your algorithm for $K = 5000$ iterations upon T starting from the identity matrix. For each $k \in \{1, \ldots, K\}$, plot the training likelihood $\ell_T(\Theta^{(k)})$ and test likelihood $\ell_U(\Theta^{(k)})$, where $\Theta^{(k)}$ is the kth iterate.

(e) [8 pts] Plot the matrix $\Theta^{(K)}$ in a way that emphasizes the sparsity pattern (e.g. `imagesc`).