The University of Texas at Austin
Department of Electrical and Computer Engineering

EE381K: Large Scale Optimization — Fall 2015

Problem Set Nine Solutions

Constantine Caramanis .

Matlab and Computational Assignments. Please provide a printout of the Matlab code you wrote to generate the solutions to the problems below.

1. **Low-rank matrix completion via projected gradient methods**

   (a) *Given a matrix X, how will you generate an element $Z \in \partial\|X\|_*$ using the* `svd` *function in matlab ?*

   Solution $Z = UV'$ is always a subgradient of $X = U\Sigma V'$.

   ```
   [U,~,V] = svd(X);
   Z = UV';
   ```

   (b) *Given a matrix X, how will you project it onto the feasible set (i.e. the set of matrices that satisfy the constraints) ?*

   Solution The matrix closest to X in the feasible set is given by $P(X)$ where

   $$P(X)_{ij} = \begin{cases} m_{ij} & \text{if } (i,j) \in \Omega, \\ x_{ij} & \text{if } (i,j) \notin \Omega \end{cases}$$

   We can obtain the matrix by replacing the values at the observed positions Ω with the known true values.

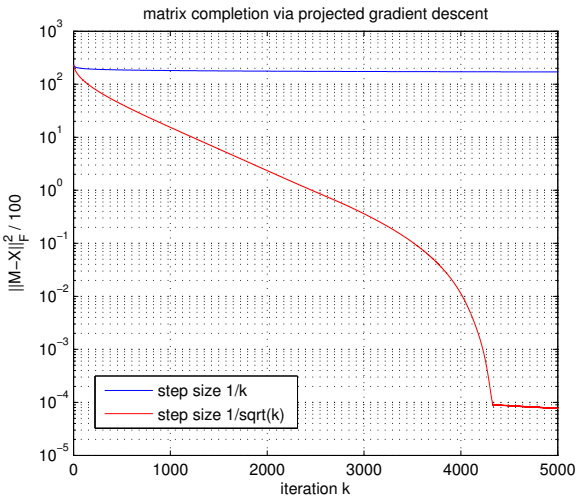
   ```
   X(O == 1) = M(O == 1);
   ```

   (c) *Implement projected sub gradient descent with two choices for step sizes: $\eta_k = \frac{1}{k}$ and $\eta_k = \frac{1}{\sqrt{k}}$. You will need to use the file* `ps9.mat`*, which contains two 100×100 matrices: a low-rank matrix M, and the matrix O that represents the set Ω by having entries that are 0 or 1 (in particular, $o_{ij} = 1$ means $(i,j) \in \Omega$).*

   Solution A sample code can be found in the file: `https://webspace.utexas.edu/dp24369/EE381V_matlab/ps9_1.m`.

   (d) *Plot the relative error $\frac{1}{100^2}\|M - X_k\|_F^2$ between the true matrix and the k^{th} iterate, as a function of k, for both step size choices; do so on one plot.*

   Solution The plot below shows the convergence results of the projected gradient algorithms with the two sequences of step sizes. As shown in the figure, the step size $1/k$ is too slow, while $1/\sqrt{k}$ is sufficiently fast.



(e) *What is the rank of the intermediate iterates ? Why is this the case ?*

Solution The iterates are always full rank. It is not guaranteed that some intermediate iterates are low-rank, though they converge to a low-rank matrix. They can have very small singular values, but we cannot tell that they are exactly zero. Also, it is often that an optimal solution to which the iterates converge may not be the true low-rank matrix.

Written Problems

1. *Prove that a matrix Z as described above (in the first computational problem) is indeed a sub gradient to the nuclear norm function at X. You can use the following fact about the nuclear norm: for any matrix $M \in \mathbb{R}^{m\times n}$, let $s = \min(m, n)$. Then for any matrices $A \in R^{m\times s}$ and $B \in R^{n\times s}$ that have orthonormal columns, we have that*

$$\|M\|_* \;\geq\; \langle M, AB'\rangle$$

Solution It is sufficient to show that for any $Y \in \mathbb{R}^{m\times n}$, we have

$$\|Y\|_* \geq \|X\|_* + \langle Y - X, UV' + W\rangle.$$

The nuclear norm is the dual norm of the spectral norm, defined as

$$\|M\|_* = \sup_{\|N\|\leq 1} \langle M, N\rangle. \tag{1}$$

We will prove (1) later. As the SVD of Z can be written as

$$Z = UV' + W = [U|U_W] \cdot \left[\begin{array}{c|c} I & 0 \\ \hline 0 & \Sigma_W \end{array}\right] \cdot [V|V_W]'$$

where $W = U_W \Sigma_W V_W'$ is the SVD of W. This shows that the spectral norm of Z is equal to one. Using (1), we have

$$\begin{aligned} \|Y\|_* &\geq \langle Y, UV' + W\rangle \\ &= \langle X, UV'\rangle + \langle X, W\rangle + \langle Y - X, UV' + W\rangle \\ &= \|X\|_* + \langle Y - X, UV' + W\rangle. \end{aligned}$$

because $\|X\|_* = \langle X, UV' \rangle$, and $\langle X, W \rangle = 0$. This completes the proof.

** Let us now show (1), which states that the nuclear norm is the dual norm of the spectral norm. Let the SVDs of $M, N \in \mathbb{R}^{m \times n}$ be written as

$$M = \sum_{i=1}^{\min(m,n)} \sigma_i u_i v_i^\top, \quad N = \sum_{i=1}^{\min(m,n)} \hat{\sigma}_i \hat{u}_i \hat{v}_i^\top$$

in the form of the sum of rank-one matrices. Since the spectral norm of N is equal to one, we have

$$\hat{\sigma}_i \leq 1, \quad i = 1, 2, \ldots, \min(m, n).$$

Now we have

$$\begin{aligned}
\langle M, N \rangle &= \sum_{i=1}^{\min(m,n)} \sum_{j=1}^{\min(m,n)} \sigma_i \hat{\sigma}_j \langle u_i v_i^\top, \hat{u}_j \hat{v}_j^\top \rangle \\
&\leq \sum_{i=1}^{\min(m,n)} \sigma_i \left(\sum_{j=1}^{\min(m,n)} |\hat{u}_j^\top u_i| \cdot |v_i^\top \hat{v}_j| \right) \\
&\leq \sum_{i=1}^{\min(m,n)} \sigma_i \left(\sum_{j=1}^{\min(m,n)} (\hat{u}_j^\top u_i)^2 \right)^{1/2} \cdot \left(\sum_{j=1}^{\min(m,n)} (v_i^\top \hat{v}_j)^2 \right)^{1/2} \\
&\leq \sum_{i=1}^{\min(m,n)} \sigma_i = \|M\|_*
\end{aligned}$$

where the second inequality is the Cauchy-Schwarz inequality. The third inequality holds because the square roots are the 2-norms of the projections of unit vectors u_i and v_i onto $\text{span}\{\hat{u}_1, \ldots, \hat{u}_{\min(m,n)}\}$ and $\text{span}\{\hat{v}_1, \ldots, \hat{v}_{\min(m,n)}\}$, respectively.

As the equality holds if $N = \sum_{i=1}^{\min(m,n)} u_i v_i^\top$, the inequality is tight, and thus (1) holds.

2. *For sub-gradient descent, we have shown that we have error $O(1/\sqrt{k})$ after k iterations,* when we take a fixed step size. *We can also use a step size that is decaying in the iteration. While using step size $t_k = 1/k$ is guaranteed to converge, it is not the optimal choice. In fact, there are simple functions where sub gradient descent with this sequence of step sizes might take exponential time to converge. Find such a function, and show that we need an exponential number of steps for ϵ-suboptimality. (You can find a simple 1-dimensional function, $f : [0, 1] \to \mathbb{R}$.)*

Solution $f(x) = x^3$ on $x \in [0, \infty)$ and other functions that are more flat around the optimum require an exponential number of steps to achieve ϵ-suboptimality.

We can informally check whether $f(x) = x^3$ is such a function. The subgradient descent update is given by

$$x_{k+1} - x_k = -\frac{1}{k} \left. \frac{df}{dx} \right|_{x=x_k}.$$

The decaying behavior of x_k for sufficiently large k will follow that of a continuous function $x(t)$ solving the differential equation

$$\frac{dx}{dt} = -\frac{1}{t}\frac{df}{dx}. \tag{2}$$

Plugging $df/dx = 3x^2$ in and solving (2), we get $x(t) = 1/(3\log t + c)$ where c is a constant. It follows that

$$\epsilon = f(x(t)) - f^* = f(x(t)) = \frac{1}{(3\log t + c)^3},$$

which implies informally that an exponential number of steps is required to achieve ϵ-optimality for sufficiently small ϵ.

A more rigorous proof is given as follows. Consider a sequence $\{x_k\}_k$ obtained from the subgradient descent method with step size $1/k$. Assume that x_1 is large enough to guarantee $x_k \geq 0$ for every k, define

$$k_n = \min\left\{k : x_k \leq \frac{1}{n}\right\},\ n = 1, 2, \cdots.$$

As the sequence always converges to zero, k_n exists for every $n = 1, 2, \cdots$.

We have

$$\begin{aligned}
\frac{1}{n(n+1)} = \frac{1}{n} - \frac{1}{n+1} &< x_{k_n - 1} - x_{k_{n+1}} \\
&= \sum_{i=k_n-1}^{k_{n+1}-1} \frac{1}{i} f'(x_i) \\
&< 3\left(\frac{1}{n-1}\right)^2 \cdot \sum_{i=k_n-1}^{k_{n+1}-1} \frac{1}{i} \\
&< 3\left(\frac{1}{n-1}\right)^2 \cdot \sum_{i=k_n-1}^{k_{n+1}-1} \frac{1}{k_n - 1} = \frac{1}{(n-1)^2(k_n-1)}(k_{n+1} - k_n).
\end{aligned}$$

where the first inequality follows from that $x_{k_n-1} > \frac{1}{n}$ and $\frac{1}{n+1} \geq x_{k_{n+1}}$, and the second inequality follows from that $f(x_i) = 3x_i^2 \leq 3(\frac{1}{n-1})^2$ for $i = k_n - 1, \ldots, k_{n+1} - 1$. Rearranging the terms, we get

$$\frac{k_{n+1} - 1}{k_n - 1} \geq 1 + \frac{(n-1)^2}{3n(n+1)}.$$

Fix n_0 to be a sufficiently large number. It follows from the telescoping product that

$$\begin{aligned}
k_n &\geq 1 + (k_{n_0} - 1) \cdot \prod_{i=n_0}^{n} \left(1 + \frac{(i-1)^2}{3i(i+1)}\right) \\
&> (k_{n_0} - 1) \cdot \prod_{i=n_0}^{n} \left(\frac{4}{3} - \frac{1}{i}\right) \\
&\geq (k_{n_0} - 1) \cdot \left(\frac{4}{3} - \frac{1}{n_0}\right)^{n - n_0 + 1},
\end{aligned}$$

which shows that k_n grows geometrically as n increases. Therfore, it takes exponential steps to achieve $x_k - x^* \leq 1/\epsilon$, and also to achieve $f(x_k) - f^* \leq 1/\epsilon$.

NOTE : The convergence analysis that we did in class is not appropriate for this problem because it gives worst-case analysis. It shows that if we use subgradient descent with step size $1/k$ for G-Lipschitz functions, an upper bound for suboptimality decays asymptotically as $O(1/\log k)$. This does not mean that the exact suboptimality decreases as $O(1/\log k)$ for any G-Lipschitz function. The exact suboptimality can decrease faster, which means that the required number of steps may not necessarily be exponential. We have to find which function is the "worst" in the sense that the upper bound is tight.

3. *Here you will do some work that helps compute the Mirror Descent update you need for the computational problem above. Mirror Descent is only computationally useful if we can easily solve the problem:*

$$\min_{u \in \mathcal{X}} : \ \langle z, u \rangle + \Phi(u).$$

In this problem, you will show that when $\mathcal{X}$ is the entire simplex, i.e., $\mathcal{X} = \Delta_n$, then this problem is indeed easy.

(a) *Consider the optimization problem:*

$$\begin{aligned} \min : \quad & \langle z, u \rangle + \Phi(u) \\ \text{s.t.} : \quad & \sum_i u_i = 1. \end{aligned}$$

(Note that the constraints $\{u_i \geq 0\}$ are implicitly included as they are part of $\mathrm{dom}\Phi$*.) Write the Lagrangian for the problem. The variables will be u and a single variable λ for the single constraint. Write the KKT conditions for the problem.*

Solution The Lagrangian is given by

$$L(u, \lambda) = \langle z, u \rangle + \sum_i u_i \ln u_i + \lambda(1 - \sum_i u_i).$$

Differentiating the Lagrangian, we obtain the KKT condition

$$z_i + 1 + \ln u_i - \lambda = 0, \ i = 1, 2, \ldots, n, \quad \sum_i u_i = 1.$$

(b) *Using the KKT conditions, derive a closed form expression for u as a function of z.*

Solution It follows from the KKT condition that

$$u_i = e^{-z_i}/e^{1-\lambda}, \ i = 1, 2, \ldots, n.$$

Since u_i is given by e^{-z_i} divided by a constant for every i, the division is nothing but the normalization to make u_i's summed to one. Therefore, we get

$$u_i = \frac{e^{-z_i}}{\sum_j e^{-z_j}}, \ i = 1, 2, \ldots, n.$$

(c) *Now go back to the Mirror Descent update and write explicitly the Mirror Descent update using your work above.*

Solution Replacing z with $tg - \nabla\Phi(x)$, we get the Mirror Descent update

$$(x_+)_i = \frac{x_i e^{-tg_i}}{\sum_j x_j e^{-tg_j}}, \ i = 1, 2, \ldots, n.$$

References

[1] Agarwal, A., Negahban, S., and Wainwright, M. J. Fast global convergence of gradient methods for high-dimensional statistical recovery, to appear in Annals of Statistics. Preprint: `http://arxiv.org/abs/1104.4824`