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

EE381V-11: Large Scale Optimization — Fall 2012

Problem Set Nine

Caramanis/Sanghavi Due: Thursday, November 29, 2012.

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

1. In this problem we investigate **low-rank matrix completion**, the problem of finding a low-rank matrix given only a few (randomly sampled entries). While this is (clearly) not possible in general, somewhat remarkably, it is possible once some additional assumptions are made on the problem setup (for example, for a "random" low-rank matrix and random samples). We will learn more about these assumptions next semester, but for now we will develop a projected subrgradient algorithm to solve such a problem.

   Suppose there is a true matrix $M \in \mathbb{R}^{m \times n}$ that we want to recover, but we are only given elements in the set $\Omega \subset [m] \times [n]$ (i.e. we know the value of m_{ij} if $(i,j) \in \Omega$). We want to solve the following constrained optimization problem

$$\begin{aligned} \min_{X} \quad & \|X\|_* \\ s.t. \quad & x_{ij} = m_{ij} \ \text{ for all } (i,j) \in \Omega \end{aligned}$$

   where the variable of optimization $X \in \mathbb{R}^{m \times n}$ is a matrix. Here $\|\cdot\|_*$ is the "nuclear" norm, equal to the sum of singular values of the matrix. This norm is a convex but not smooth function of X; we will implement projected sub gradient descent for this problem.

   The sub gradient of the $\|\cdot\|_*$ function is as follows: for any matrix X, if its SVD is $U\Sigma V'$, then a matrix $Z \in \partial\|X\|_*$ is in its sub gradient if and only if

$$Z = UV' + W$$

   where W is such that *(a)* the column and row spaces of W are perpendicular to the corresponding ones of X, and *(b)* the spectral norm $\|W\|_2 \leq 1$. Recall that the spectral norm of a matrix is its maximum singular value. Also recall that if X is rank r, then the matrices U, V are of sizes $m \times r$ and $n \times r$ respectively, and have orthonormal columns.

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

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

   (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 $(i,j) \in \Omega$).

(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.

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

2. In the very first problem set, we used CVX and OMP to solve a sparse regression problem. Here, you will use sub gradient descent, the proximal gradient method, and FISTA to solve the same problem. Go back to the first problem set, and recall the three problems of increasing size that we had given. These were all of the form:

$$\min : \ \|X\beta - y\|_2^2 + \lambda\|\beta\|_1.$$

Note that because X has more columns than rows, the first term in the objective is smooth, but not strongly convex. The second term is convex, but not smooth.

(a) Use sub gradient descent with fixed step size to solve the problem. Start from the smallest problem instance, and see if you can solve all three. Plot the error versus the iteration number.

(b) Compare this with projected sub-gradient descent for the related problem:

$$\begin{aligned} \min : \quad & \|X\beta - y\|_2^2 \\ \text{s.t.} : \quad & \|\beta\|_1 \leq R. \end{aligned}$$

Use a value R equal to the 1-norm of the actual sparse solution.

(c) Now use the proximal gradient method, with $f(\beta) = g(\beta)+h(\beta)$, where $g(\beta) = \|X\beta-y\|_2^2$ is your smooth function, and $h(\beta) = \|\beta\|_1$ is your non-smooth function with easy-to-compute prox operator. Plot the error versus the iteration number.

(d) Finally, use FISTA. Again, plot the error versus the iteration number.

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

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

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.* In the last homework, you investigated what happens when we take a step size that is not fixed, but 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, $f : [0, 1] \to \mathbb{R}$.)

3. (Warmup) In class on Tuesday we began discussing Mirror Descent. Recall that for the minimization problem:

$$\begin{aligned} \min : \quad & f(x) \\ \text{s.t.} : \quad & x \in \mathcal{X}, \end{aligned}$$

the Mirror Descent update is:

$$x_+ = \arg\min_{u \in \mathcal{X}} \{\langle tg - \nabla\omega(x), u\rangle + \omega(u)\}, \tag{1}$$

where t is the step size, $g \in \partial f(x)$ is any sub gradient of f at x, and ω is the *distance generating function*. Show that if we choose

$$\omega(u) = \frac{1}{2}\|u\|_2^2,$$

then we recover precisely the update for projected gradient descent:

$$x_+ = \mathrm{Proj}_{\mathcal{X}}(x - tg).$$

4. (More MD) In class on Tuesday, we will see that in some cases, there are advantages to choosing different distance generating functions $\omega(\cdot)$. In particular, when $\mathcal{X}$ is a subset of the simplex, $\mathcal{X} \subseteq \Delta_n$, where

$$\Delta_n = \{x \in \mathbb{R}^n \mid \sum_i x_i = 1 \; x_i \geq 0\},$$

we will see in class that it can be greatly beneficial (in terms of number of iterations) to choose

$$\omega(u) = \sum_i u_i \ln u_i.$$

Mirror Descent is only computationally useful, if the update (1) can be computed easily. That is, if we can easily solve the problem:

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

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:

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

(Note that the constraints $\{u_i \geq 0\}$ are implicitly included as they are part of $\mathrm{dom}\omega$.) 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.

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

(c) Now go back to the Mirror Descent update from (1), and write explicitly the Mirror Descent update using your work above.