Homework 1

Convex Optimization 10-725/36-725

Due Thursday January 29 at 4:00pm
submitted to Mallory Deptola in GHC 8001

1 Relaxing equality constraints

(a) This is Boyd & Vandenberghe's Exercise 4.6, copied here for convenience. Consider an optimization problem

$$\begin{array}{ll} \min & f_0(x) \\ \text{subject to} & f_i(x) \leq 0, \quad i = 1, \dots m \\ & h(x) = 0, \end{array} \tag{1}$$

where $f_0, \dots f_m, h$ are all convex with domain $\mathbb{R}^n$. Unless h is affine this is not a convex optimization problem. Consider the related problem

$$\begin{array}{ll} \min & f_0(x) \\ \text{subject to} & f_i(x) \leq 0, \quad i = 1, \dots m \\ & h(x) \leq 0, \end{array} \tag{2}$$

where the convex equality constraint has been relaxed to a convex inequality. This problem is, of course, convex.

Now suppose we can guarantee that at any optimal solution $x^\star$ of the convex problem (2), we have $h(x^\star) = 0$, i.e., the inequality $h(x) \leq 0$ is always active at the solution. Then we can solve the nonconvex problem (1) by solving the convex problem (2).

Show that this is the case if there is an index $j \in \{1, \dots n\}$ such that

- f_0 is increasing in x_j, and
- $f_1, \dots f_m$ are nondecreasing in x_j, and
- h is decreasing in x_j.

(b) Apply this logic to the maximum utility problem from Lecture 3 to argue that the equality constraint there can be relaxed to an inequality constraint, and this will still deliver the proper solutions. (Note: the maximum utility problem is also described in Boyd & Vandenberghe's Exercise 4.58, but they use slightly different notation.)

2 Partial optimization using ℓ_2 penalties

Consider the problem

$$\min_{\beta,\, \sigma \geq 0} \; f(\beta) + \frac{\lambda}{2} \sum_{i=1}^{n} g(\beta_i, \sigma_i), \tag{3}$$

for some convex f with domain $\mathbb{R}^n$, $\lambda \geq 0$, and

$$g(x,y) = \begin{cases} x^2/y + y & \text{if } y > 0 \\ 0 & \text{if } x = 0,\ y = 0 \\ \infty & \text{else} \end{cases}.$$

In other words, the problem (3) is really just the weighted ℓ_2 penalized problem

$$\min_{\beta,\,\sigma\geq 0}\ f(\beta) + \frac{\lambda}{2}\sum_{i=1}^n \Big(\frac{\beta_i^2}{\sigma_i} + \sigma_i\Big),$$

but being careful to treat the ith term in the sum as zero when $\beta_i = \sigma_i = 0$.

(a) Prove that g is convex. Hence argue that (3) is a convex problem. Note that this means we can perform partial optimization in (3) and expect it to return another convex problem.

(Hint: for convexity of g, use the second-order characterization when $y > 0$, and the definition of convexity when $y = 0$.)

(b) Argue that $\min_{y\geq 0}\ g(x,y) = 2|x|$.

(c) Argue that minimizing over $\sigma \geq 0$ in (3) gives the ℓ_1 penalized problem

$$\min_{\beta}\ f(\beta) + \lambda\|\beta\|_1.$$

3 Lipschitz gradients and strong convexity

Let f be twice differentiable.

(a) Show that the following statements are equivalent, for convex f:

- ∇f is Lipschitz with constant L;
- $\nabla^2 f(x) \preceq LI$ for all x;
- $f(y) \leq f(x) + \nabla f(x)^T(y-x) + \frac{L}{2}\|y-x\|_2^2$ for all x, y.

(b) Show that the following statements are equivalent, for convex f:

- f is strongly convex with constant m;
- $\|\nabla f(x) - \nabla f(y)\|_2 \geq m\|x-y\|_2$ for all x, y;
- $\nabla^2 f(x) \succeq mI$ for all x;
- $f(y) \geq f(x) + \nabla f(x)^T(y-x) + \frac{m}{2}\|y-x\|_2^2$ for all x, y.

4 Solving optimization problems with CVX

CVX is a fantastic Matlab package for disciplined convex programming. It's rarely the fastest tool for the job, but it's widely applicable, and so it's a great tool to be comfortable with.

(a) For Matlab users, install CVX from here: `http://cvxr.com/cvx/`, and read the user manual to get an idea of how to setup and solve optimization problems. For R users, you can call CVX from R. Use this package to so: `http://faculty.bscb.cornell.edu/~bien/cvxfromr.html`. (If you don't want to use Matlab or R, then this question will be more difficult, but you can try, e.g., using Julia. Find the Convex.jl package here: https://github.com/JuliaOpt/Convex.jl. This has similar, but not exactly the same syntax, as CVX).

Make sure that you can solve the least squares problem $\min_\beta \|y - X\beta\|_2^2$ for a vector y and matrix X, and check answers by comparing with the analytic least squares solution.

(b) Now download the following zip file from the course website: `hw1-q4.zip`. It contains starter code and data files meant for Matlab or for R, so use them according to your preferred programming language.

The file `TestSVM.m` / `TestSVM.R` reads in the data file `SVM_data.mat` / `SVM_data.RData`, and contains a code outline for implementing the primal and dual soft-margin linear SVM, plotting the separating hyperplanes, and plotting the curve of misclassification rate against the tuning parameter C. The formulation for the primal SVM is given in slide 8 of Lecture 3, which we copy here:

$$\begin{aligned}\underset{\beta\in\mathbb{R}^p,\beta_0\in\mathbb{R},\xi\in\mathbb{R}^n}{\text{minimize}}\ & \frac{1}{2}\|\beta\|^2 + C\sum \xi_i \\ \text{subject to } & 0 \le \xi_i \ge 0,\ \text{for } i = 1, ..., n \\ & y_i(x_i^T\beta + \beta_0) \ge 1 - \xi_i,\ \text{for } i = 1, ..., n.\end{aligned}$$

Here is a second, seemingly different problem, dual SVM:

$$\begin{aligned}\underset{\alpha\in\mathbb{R}^n}{\text{maximize}}\ & -\frac{1}{2}\sum_{i=1}^n\sum_{j=1}^n \alpha_i\alpha_j y_i y_j \langle x_i, x_j\rangle + 1^T\alpha \\ \text{subject to } & 0 \le \alpha \le C,\ \ y^T\alpha = 0.\end{aligned}$$

We'll learn later that this is equivalent in a precise sense to the first problem (primal SVM). You will see in this problem that they produce the same answer.

Fill in the CVX code needed to solve the SVM problems, and do the following:

1. Compare the optimal objective function values of the primal and dual problem for the last tuning parameter value C in the list `Clist`. You may either read them off the CVX log, or calculate them out directly.

2. Compare the the primal solution β and the resconstructed dual solution $\tilde{\beta} = X^T(\alpha \odot y)$, based on the dual solution α. (Here $a \odot b$ denotes the elementwise product between vectors a and b).

3. Plot the separating hyperplane:
$$\beta^T x + \beta_0 = 0.$$
Note that x is 2D, and this equation describes a line on the 2D plane.

4. Plot the misclassification rate on the provided test data as a function of C varying over the provided list `Clist`.

The file `TestLasso.m` / `TestLasso.R` reads the data file `LASSO_data.mat` / `LASSO_data.RData`, and contains a code outline for solving the lasso regression problem:

$$\underset{\beta\in\mathbb{R}^p}{\text{minimize}}\ \frac{1}{2}\|y - X\beta\|_2^2 + \lambda\|\beta\|_1,$$

Fill in the CVX code needed to solve the lasso problem, and collect the resulting solutions for each tuning parameter value in the provided list `lambdas`, into the columns of a matrix B. Then:

1. Plot the first 20 elements of each row of B versus the first 20 values of λ (so that we have one line for each variable, on the same figure). This is how we illustrate the solution path of lasso usually.
2. Calculate the mean square error of the solution $\hat{\beta}(\lambda)$ and the ground truth β (given in the .mat and .RData file) for each value of λ. Plot the MSE curve. Note that MSE is calculated as $\frac{1}{p}\|\hat{\beta}(\lambda) - \beta\|_2^2$, where p is the number of variables (columns of X).

Your solution to this problem should include the request plots and a short writeup that may include short code snippets, if you find them useful to share. Then append the full code at the end of the homework document.

(c) Using CVX is usually easy, but sometimes it can get tricky. Each of the following CVX code fragments describes a convex constraint on the scalar variables x, y, and z, but violates the CVX rule set. So if you code these snippets as is, CVX will throw an error.

Briefly explain why each snippet is invalid. Then rewrite each one in an equivalent form that conforms to the CVX rule set (and also explain why your reformulation is equivalent to the original statement).

Hint: Read the CVX documentation carefully. You may need to introduce additional variables, or use linear matrix inequalities. Note: this question is not really about learning the CVX lingo; it's about developing a better understanding of the various composition rules for convex functions.

1. `norm([x,y,z],2)^2 <= 1`
2. `norm( [x + 2*y, x-y] ) == 0`
3. `square( square ( 4x - y )) <= x-y`
4. `1/x + 2/y <= 5;  x>=0; y>=0`
5. `sqrt( x^2 + 1 ) <= 3*x + y`
6. `(x+z)*y >= 1; x+z >= 0; y >= 0`
7. `( x + y )^2 / sqrt( y ) <= x - y + 5`
8. `[2*x 2*y -z; 2*y -x+2*z x; -z x 3*z-x^2] == semidefinite(3)`

   (Hint: introduce a dummy variable so the constraint remains an LMI constraint. Justify that the reformulation is the same.)