ORIE 6300 Homework 7 midnight, December 4, 2020

We will use Gradescope for this homework as well. You will submit code and also output of your code as a separate PDF file. While you can collaborate on this assignment, what you submit (i.e. the code and its output) should be entirely your own.

1 Solving linear programing by IPM

. For this homework we will consider a sparse SVM problem, which is different from standard SVM problem is that it can be formulated as an LP (not a convex QP as a standard large margin SVM). Assume we are given a set of data-label pairs $\{(a_i, b_i)\, i = 1, \dots m\}$ with $a_i \in \mathbb{R}^n$ and $b \in \{+1, -1\}$. We want to find a separating hyperplane defined by $x \in \mathbb{R}^n$, which optimizes the following problem:

$$\min_x \|x\|_1 + \sum_{i=1}^{m} \max\{0, 1 - b_i(x^T a_i)\}$$

This problem can be reformulated as

$$\begin{aligned} \min_{x', x'', \xi} & \sum_{i=1}^{n} (x'_i + x''_i) + \sum_{i=1}^{m} \xi_i \\ s.t.\ & \xi_i - 1 + b_i a_i^T (x' - x'') \geq 0 \\ & x', x'', \xi \geq 0 \end{aligned}$$

Put it in the standard form and derive the dual, so that you can solve it by a primal-dual interior point method.

Reformulation: The primal problem in standard form is:

$$\begin{aligned} \min_{x', x'', \xi} & e^T [x'; x''; \xi] \\ s.t.\ & [I\ A\ -A\ -I][\xi; x'; x''; \beta] = e \\ & x', x'', \xi, \beta \geq 0 \end{aligned}$$

Where A is a matrix where each row is $b_i a_i^T$. The dual is thus:

$$\begin{aligned} \max_{y, s}\ & e^T y \\ s.t.\ & [I; A^T; -A^T; -I] y + s = e \\ & x', x'', \xi, \beta \geq 0 \end{aligned}$$

2 Generating problem instances

First generate your data by generating 50 points each in 2D, from two Gaussian distributions: $\mathcal{N}((1, 1), 0.5I)$ and $\mathcal{N}((-1, -1), 0.5I)$. Assign labels $b_i = 1$ to the points from the first distribution and $b_i = -1$ to those from the second.

After you make sure your code works on this problem, you should then try it on the "Digits" data set, where the data contains information about the pixels of an image of a digit. There are

10 digits, thus 10 labels. Your task is to construct a classifier that separates digit '5' from the rest, so you will need to create your own labels accordingly. You may want to model hyperplanes that include the intercept, by augmenting the data (which has dimension 64) with one additional 65th feature, that always equals 1 for all data. This way your data will be $a_i = (\bar{a}_i, 1)$ and your separating hyperplane is defined by $x^T a = (\bar{x}, z)^T(\bar{a}_i, 1) = \bar{x}^T \bar{a} + z$, where $z \in R$.

3 Optimization

Code two IPM algorithms, the short step path-following algorithm and Mehrotra's predictor corrector algorithm. Both are given below. For Mehrotra's predictor-corrector algorithm you can start with infeasible solutions, so you can choose initial primal and dual vectors to equal e - vector of all ones. However, for the short step method, you need to start with a feasible primal-dual pair that is not too far the the central path. Try your best to find such a solution, if you can't, just start with some feasible interior primal-dual pair of solutions and see what you get.

Both methods will solve the following system for some changes to the right hand side.

$$(IPM) \quad \begin{pmatrix} A & 0 & 0 \\ 0 & A^T & I \\ S & 0 & X \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta s \end{pmatrix} = \begin{pmatrix} r_1 \\ r_2 \\ r_3 \end{pmatrix}$$

Solve this system by reducing it to the form $A^T D^{-1} A \Delta y = r$. Try to be efficient with linear algebra. Run both algorithms until μ and the norm of the constraint violation are all $\leq 10^-8$ and print out the progress in terms of μ and the constraint violation.

I suggest you code the algorithm in Matlab an compare it to applying CVX. You should submit a ziped folder constaining your files and one simple run.m file that will run your algorithms and produce some comprehensible output. If you must use Python, you are allowed to do it, please discuss with Shijin, but I suggest the same simple format submission.

4 Short Path-Following IPM

The basic short-step path following algorithm that we discussed in class is as follows.

1. (Initialization) Choose convergence tolerance $\epsilon > 0$, parameters $\theta = 0.4$, $\sigma = 1 - 0.4/\sqrt{n}$ and starting point $(x_0, y_0, s_0) \in \mathcal{N}_2(\theta)$, set $k = 0$.

2. (Main iteration)

   (a) Set $\mu_k = \frac{s_k^T x_k}{n}$. If $\mu_k \leq \epsilon$, terminate.

   (b) Set $\sigma_k = \sigma$ and solve (IPM) with $r_1 = 0$, $r_2 = 0$ and $r_3 = -XSe + \sigma\mu e$ to obtain $(\Delta x_k, \Delta y_k, \Delta s_k)$.

   (c) Set $(x_{k+1}, y_{k+1}, s_{k+1}) = (x_k, y_k, s_k) + (\Delta x_k, \Delta y_k, \Delta s_k)$.

   (d) $k := k + 1$.

5 Predictor-Corrector IPM

1. (Initialization) Choose convergence tolerance $\epsilon > 0$, starting point (x_0, y_0, s_0), with $x_0 > 0$, $s_0 > 0$. set $k = 0$.

2. (Main iteration) Let $(x, y, s) = (x_k, y_k, s_k)$

   - Set $\mu = \frac{x^T s}{n}$. Compute $r_p = b - Ax$ and $r_d = c - A^T y - s$. If $\max\{\mu, \|r_p\|, \|r_d\|\} \leq \epsilon$, then terminate. Otherwise
   - Solve for $\sigma = 0$

$$\begin{aligned} S\Delta x^{pred} + X\Delta s^{pred} &= -Xs \\ A\Delta x^{pred} &= r_p \quad (r_p = b - Ax) \\ A^T \Delta y^{pred} + \Delta s^{pred} &= r_d \quad (r_d = c - A^T y - s) \end{aligned}$$

- Compute $\alpha = \min\{1, 0.99\alpha_{max}\}$, where α_{max} is maximum such that $x + \alpha_{max}\Delta x^{pred} \geq 0$ and $s + \alpha_{max}\Delta s^{pred} \geq 0$.

  Compute

$$\mu^{pred} = \frac{(x + \alpha\Delta x^{pred})^T(s + \alpha\Delta s^{pred})}{n}$$

- Solve for $\sigma = (\frac{\mu^{pred}}{\mu})^3$

$$\begin{aligned} S\Delta x + X\Delta s &= \sigma\mu e - Xs - \Delta X^{pred}\Delta s^{pred} \\ A\Delta x &= r_p \\ A^T\Delta y + \Delta s &= r_d \end{aligned}$$

- Compute $(x^{new}, y^{new}, s^{new})$ as

$$\begin{aligned} x^{new} &= x + \alpha\Delta x \\ y^{new} &= y + \alpha\Delta y \\ s^{new} &= s + \alpha\Delta s \end{aligned}$$

  where $\alpha = \min\{1, 0.99\alpha_{max}\}$, where α_{max} is maximum such that $x + \alpha_{max}\Delta x \geq 0$ and $s + \alpha_{max}\Delta s \geq 0$.

- Let $(x_{k+1}, y_{k+1}, s_{k+1})$=$(x^{new}, y^{new}, s^{new})$, repeat.