10725/36725 Optimization
Homework 5

Due December 6, 2012 at beginning of class

Instructions: There are four questions in this assignment. Please submit your homework as (up to) 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 and email your code to submission10725f12@gmail.com. Refer to the course webpage for policies regarding collaboration, due dates, and extensions.

New: For this problem set, you have a choice: you may submit answers for *any three* of the problems. You may choose to submit answers for all questions; if you do, we'll give you credit for the highest three scores.

1 Affine Scaling [Kevin, 25 points]

Affine scaling is a technique for solving problems of the form

$$\min_x \ c^T x, \text{ subject to:} \tag{1}$$

$$Ax = b \tag{2}$$

$$x \in \mathcal{X} \tag{3}$$

when we have a self-concordant barrier for $\mathcal{X}$ and a feasible interior point.

Given an interior point, $x \in \mathcal{X}$, we define the α-Dikin ellipsoid at x as

$$\text{Dikin}_\alpha(x) = \{y \mid (y-x)^T H(x)(y-x) \leq \alpha\}, \tag{4}$$

where $H(x)$ is the Hessian of a self-concordant barrier of $\mathcal{X}$ at x. Recall that if $\alpha \in (0, 1]$ then $\text{Dikin}_\alpha(x) \subseteq \mathcal{X}$.

The affine scaling algorithm iterates $x^{t+1} = x^t + d_\alpha(x^t)$, where $d_\alpha(x)$, the affine scaling

direction from x, is the solution to the following optimization problem:

$$d_\alpha(x) = \arg\min_{\Delta x} c^T(x + \Delta x), \text{ subject to:} \tag{5}$$

$$A(x + \Delta x) = b \tag{6}$$

$$x + \Delta x \in \text{Dikin}_\alpha(x). \tag{7}$$

Let's derive a procedure for this update using the affine invariance of the Dikin ellipsoid.

(a) [3 points] Write out the KKT conditions for the optimization problem at point $(x^*, \lambda^*, \gamma^*)$:

$$\min_x \ c^T x, \text{ subject to:} \tag{8}$$

$$Ax = 0 \tag{9}$$

$$x^T x \leq \alpha. \tag{10}$$

(b) [3 points] Derive a solution to the following problem from its KKT conditions:

$$z^* = \arg\min_z \ \|y - z\|_2^2/2, \text{ subject to:} \tag{11}$$

$$Az = 0 \tag{12}$$

That is, project y onto the null space of A. Your answer will be the solution to a system of linear equalities (which you do not have to solve).

The KKT conditions for (a) and (b) are necessary and sufficient by convexity and Slater's condition.

(c) [4 points] Use part (b) to find a point that satisfies (a)'s KKT conditions.

(d) [3 points] Write out the affine scaling direction program explicitly using the Dikin ellipsoid.

Recall that the Hessian of a self-concordant barrier is positive definite at an interior point and can be factored as $H(x) = PDP^T$, for positive diagonal D and orthonormal P (*i.e.*, $P^{-1} = P^T$).

(e) [4 points] Write the affine scaling update in the form of (a) (*i.e.*, what are c and A?).

(f) [4 points] Using (c) and (e), write $d_\alpha(x)$.

For linear programming, we take $\mathcal{X} = \mathbb{R}^n_+$ and use the self-concordant barrier $-\sum_{i=1}^n \log(x_i)$.

(g) [4 points] In this case, what are P and D? Write out the affine scaling direction for linear programming.

2 Gradient projected descent (Shiva)

Recall the notation of lecture 24, particularly for the potential reduction algorithm. Given the strictly feasible iterate (x, y, s), choose the direction $(\Delta x, \Delta y, \Delta s)$ which:

- attempts to minimize $p = p_1 + p_2$ by minimizing the quadratic upper bound

$$\bar{p}(x + \Delta x, y + \Delta y, s + \Delta s) =$$
$$\underbrace{\left(\overbrace{(m+k)\ln y^T s}^{p_1} + (m+k)\frac{1}{y^T s}(\Delta y^T s + \Delta s^T y) \right)}_{\bar{p}_1} +$$
$$\underbrace{\left(\overbrace{-\sum_i \ln(y_i) - \sum_i \ln(s_i)}^{p_2} - \Delta y^T \frac{1}{y} - \Delta s^T \frac{1}{s} + \Delta y^T Y^{-2} \Delta y + \Delta s^T S^{-2} \Delta s \right)}_{\bar{p}_2}$$

- is chosen from the region where the quadratic upper bound is valid: $\Delta y^T Y^{-2} \Delta y + \Delta s^T S^{-} 2 \Delta s \leq (2/3)^2$
- satisfies $A^T \Delta y = 0$, $\Delta s = A \Delta x$, so that $(x + \Delta x, y + \Delta y, s + \Delta s)$ remains feasible.

Since the feasible region is convex, a new strictly feasible iterate is chosen by $(x + c_x \Delta x, y + c_y \Delta y, s + c_s \Delta s)$ for some $c > 0$. As we discussed, the convergence rate of the algorithm is governed by the norm of $W^{-1} g$, where the diagonal elements of W are $\sqrt{y \circ s} / \min(\sqrt{y \circ s})$ and $g = (m+k)\frac{y \circ s}{y^T s} - 1$ is defined on slide 17 of lecture 24.

(a) [25 points] Lower bound $||W^{-1} g||$. For guidance, you may look at lemma 2.5 in the 1991 Mathematical Programming paper, "A polynomial-time algorithm for a class of linear complementary problems" by Kojima, Mizuno, and Yoshise.

3 2016 US Presidential Elections (Aadi)

Disclaimer This problem does not reflect the political views of the TAs or instructors.

Introduction The 2016 US presidential elections aren't so far away, and political strategists are already buckling their electoral college seat belts. Rumours are out that it's going to be between Hillary Clinton and Jeb Bush, and Bush wants the easiest and shortest route to victory. We are going to help him find the minimum number of votes that he needs to gain in order for him to become president.

How The Electoral College Works Well, firstly, it doesn't work, but we'll describe its broken rules. Each of the 50 states and the one non-state of Washington DC are assigned a certain number of electoral college seats, and the total number of seats in the US adds up

to 538. You become the president if you win at least 270 (majority) electoral college seats. Seems simple?

Not So Simple Some might guess that a state's electoral college seats are proportional to its population. However, Wyoming has a population of less than 600,000 people and is allotted 3 seats, whereas New Hampshire has more than a 1,200,000 people but has only 4 seats. Also, under the present system, whichever candidate wins a state gets ALL the electoral college seats that were allotted to that state, while the loser gets none. Hence, winning a state 51-49 (percent) is equivalent to 75-25 since in both cases you get all the electoral college votes of that state.

The 2000 Elections For example, in the 2000 elections, George Bush got only 537 more votes than Al Gore in Florida, and was awarded all 25 electoral college seats, making him the president. However, on counting the total number of votes ("popular vote count") in the entire country, Gore had actually received 543,895 votes more than Bush, but did not become president. This is the fourth such occurrence in US presidential history and Jeb Bush intends to make it the fifth, and do it in style.

Bush's Master Plan Let $x_1, ..., x_{51}$ be binary variables taking the values 0 or 1 where $x_i = 1$ represents Bush winning state i (DC is the 51st territory). Let $f_1..., f_{51}$ be real valued variables between 0 and 1 representing the fraction of votes that Bush wins in the corresponding state (of course, $f_i \geq 0.5$ iff $x_i = 1$). Let $P_1, ..., P_{51}$ be constants representing the projected populations of these states around the elections (we assume everybody votes, for simplicity). Let $E_1, ..., E_{51}$ be constants representing the electoral college seats for the corresponding states. Bush wants to calculate the minimum percentage of total votes that he needs to win, so that he still gets the 270 electoral college votes to become president.

Data Analysis [10 points] Download *data.xls*, which is a list of states with projected populations in 2015 (obtained from public US Census data) and electoral college seats (updated every 10 years, latest in 2010), which are the constants P and E. Which state has the largest projected population? What's the projected US total population? In which state does your vote count the most? (the lowest population to electoral college seats ratio) In which state does it count least? What's the national average ratio of population to electoral college seats?

Formulate IP [5 points] Write down an integer program (as mentioned in recitation or Wikipedia, this optimization problem has a linear objective and affine constraints with additional constraints enforcing that the domains of some variables have to be integers) to minimize the total number of votes that Bush needs to win subject to the constraint that

he becomes president. Do not use your intuition or knowledge of what the final solution might end up being like when framing the IP. Throughout this problem, you don't need to bother about the fact that a fraction (like 0.513) of the total votes in a state may not be integral number (it may correspond to 233,453.56 votes, and that's all right, votes are always represented in percentage without this confusion).

Solve IP [5 points] Code up the IP in Matlab (command *bintprog*), or any other commercially available integer programming software. What is the minimum percentage of the popular vote that future president Bush needs to gain?

LP relaxation [5 points] Solve the exact same IP that you formulated above but now relax the integer constraints on x and allow it take on any real value between 0 and 1. (the trivial LP relaxation of the original problem) How close is this to the true answer?

4 SVM and quadratic programming (Wooyoung)

SVM formulation can be interpreted as finding a hyperplane $f(x) = \mathbf{w}^T\mathbf{x} + b$ that minimizes the sum of the complexity of the hyperplane and the training error as below:

$$L(\mathbf{w}) = \frac{1}{2}||\mathbf{w}||^2 + C\sum_i^n l(f(\mathbf{x}_i), y_i)$$

where $\mathbf{x}_i \in \mathcal{R}^d, y_i \in \mathcal{R}, (i = 1, \cdots, n)$. C is a parameter which reflects the degree to which you care about the training errors.

When we adopt the hinge-loss $l(f(\mathbf{x}_i), y_i) = \max(0, 1 - f(\mathbf{x}_i)y_i)$ for computing training errors, minimizing the objective function $L(\mathbf{w})$ is equivalent to the following optimization problem:

$$\begin{aligned} L(\mathbf{w}) \quad &= \frac{1}{2}||\mathbf{w}||^2 + C\textstyle\sum_i^n \xi_i \\ \text{such that} \quad & \xi_i \geq 0 \\ & y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i \end{aligned}$$

The dual formulation of SVM is

$$\begin{aligned} D(\alpha) \quad &= -\frac{1}{2}\textstyle\sum_{i,j}^n \alpha_i\alpha_j y_i y_j \mathbf{x}_i^T\mathbf{x}_j + \sum_i^n \alpha_i \\ \text{such that} \quad & 0 \leq \alpha_i \leq C \\ & \textstyle\sum_{i=1}^n \alpha_i y_i = 0 \end{aligned}$$

Kernel trick allows you to project your data $\mathbf{x}_i$ into higher dimensional space $\Phi(\mathbf{x}_i)$ where your data points are hopefully better separated. With the kernel trick, the dual problem is as follows:

$$\begin{aligned} \text{maximize } D(\alpha) \quad &= -\frac{1}{2}\sum_{i,j}^{n} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) + \sum_i^n \alpha_i \\ \text{such that} \quad & 0 \leq \alpha_i \leq C \\ & \textstyle\sum_{i=1}^{n} \alpha_i y_i = 0 \end{aligned} \tag{13}$$

where $K(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i)^T \Phi(\mathbf{x}_j)$.

The optimal decision function is defined as $f(\mathbf{x}) = \sum_i^n \alpha_i y_i \Phi(\mathbf{x})^T \Phi(\mathbf{x}_i) + b$.

In this problem, we are going to solve the SVM problem using quadprog function implemented in MATLAB. We will use the radial basis function (RBF) kernel:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(\frac{-||\mathbf{x}_i - \mathbf{x}_j||^2}{2\sigma^2}\right)$$

Please download svmdata.mat from `http://www.cs.cmu.edu/~ggordon/10725-F12/hws/hw5/`.

(a) [4 points] $\mathsf{x} = \mathsf{quadprog}(\mathsf{H}, \mathsf{f}, \mathsf{A}, \mathsf{b}, \mathsf{Ae}, \mathsf{be})$ attemps to solve
$\min 0.5\mathsf{x}^\mathsf{T}\mathsf{H}\mathsf{x} + \mathsf{f}^\mathsf{T}\mathsf{x}$ subject to $\mathsf{A}\mathsf{x} \leq \mathsf{b}, \mathsf{Ae} * \mathsf{x} = \mathsf{be}$.

Show how you will arrange the terms in Eq.13 to solve the optimization problem with quadprog function. In other words, show how you will build $\mathsf{H}, \mathsf{f}, \mathsf{A}, \mathsf{b}, \mathsf{Ae}, \mathsf{be}$ from the terms in Eq.13.

(b) [7 points] Implement $\alpha = \mathsf{svmdual}(\mathsf{X}, \mathsf{Y}, \mathsf{C}, \sigma)$ ($\alpha \in \mathcal{R}^n$) that solves Eq.13. X is a $d \times n$ matrix whose columns correspond to one sample and i th element of Y corresponds to $y_i \in \{-1, 1\}$.

(c) [7 points] Implement $[\mathsf{ylabel}, \mathsf{yval}] = \mathsf{svmdecision}(\mathsf{x}, \alpha, \mathsf{X}, \mathsf{Y}, \mathsf{C}, \sigma, \mathsf{b})$ such that $\mathsf{yval} = f(\mathsf{x})$ and $\mathsf{ylabel} = \text{sign}(\mathsf{yval})$. Define b as the average of $(y_k - \sum_{i=1}^{n} \alpha_i y_i \Phi(\mathbf{x}_i)^T \Phi(\mathbf{x}_k))$ for all k that satisfy $0 < \alpha_k < C$.

(d) [7 points] Train SVM with $C = 0.01, 1, 100$ with $\sigma = 1$. For each condition, i) plot the training examples and the support vectors, ii) show the decision boundary, iii) report the training and the test error defined as number of correctly classified samples / number of total samples. Discuss which one of the three values of C you would choose. For plotting the decision boundary, first define a finely spaced grid by using meshgrid function. Then, compute $f(\mathbf{x})$ on the grid and plot the decision boundary using contourf function.