Please submit your answers to my courses before the given deadline.

Problem 4.1: Frank-Wolfe Method

Consider the following quadratic problem with a quadratic constraint

$$\min_{x} \ \frac{1}{2}||Ax - b||_2^2 \tag{1}$$

$$\text{subject to: } ||x||_1^2 \leq c \tag{2}$$

with variables $x \in \mathbb{R}^n$ and problem data are $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$, and $c \in \mathbb{R}$. This type of problem can be used in many applications such as regression analysis and model fitting. In some contexts, this problem is known as the least-absolute shrinkage and selection operator (LASSO) due to the effect of constraining the L_1-nom of the regression coefficients (represented by x).

Apply the Frank-Wolfe method to solve a randomly generated problem of the form (1) – (2) given in this Jupyter notebook file by filling in the missing parts of the implementation. All parameters and input data are given and are not to be changed. Only modify parts of the code where you see a commented line with `TODO`.

When you have completed the code, you can verify the correctness of your implementation by comparing it to the solution given in the notebook. **Remember to submit your completed Jupyter notebook file. Correct implementation is enough to get full points from this first problem.**

Problem 4.2: Interior-Point Method for Quadratic Problems

Consider the following quadratic optimization problem with equality constraints:

$$(P): \ \min. \ c^\top x + \frac{1}{2}x^\top Qx \tag{3}$$

$$\text{subject to: } Ax = b \tag{4}$$

$$x \geq 0 \tag{5}$$

with variables $x \in \mathbb{R}^n$. We assume that $Q \in \mathbb{R}^{n\times n}$ is a symmetric positive definite matrix, and that $c \in \mathbb{R}^n$, $A \in \mathbb{R}^{m\times n}$, and $b \in \mathbb{R}^m$. The dual problem of (3) – (5) is

$$(D): \ \max. \ b^\top v - \frac{1}{2}x^\top Qx \tag{6}$$

$$\text{subject to: } A^\top v + u - Qx = c \tag{7}$$

$$u \geq 0 \tag{8}$$

with dual variables $v \in \mathbb{R}^m$ and $u \in \mathbb{R}^n$. We want to solve the problem (3) – (5) with a similar primal-dual interior-point method as described in Lecture 10 for LP problems.

(a) Write the KKT conditions of the problem (3) – (5)

(b) Formulate the barrier problem for (3) – (5) using the logarithmic barrier function. Write the KKT conditions of the barrier problem using the same notation as in Lecture 10

(c) Write the Newton system based on the KKT conditions of part (b) similarly to Lecture 10. Derive update formulas for directions d_x, d_v, and d_u from the linear equations in this system. Notice that the starting point is not necessarily primal or dual feasible, so you have to also consider the primal and dual residuals (the RHS vector of the Newton system).

Hint: First derive a formula for d_v, then for d_x which uses the already computed d_v, and finally for d_u which uses the already computed d_v and d_x. You can start by first solving an expression for d_u from the 2nd KKT condition and then substituting this expression to the 3rd KKT condition. Then solve an expression for d_x, and, using the formula of the

1st KKT condition, multiply this expression with A. You will get an equation of the form $Ad_x = A * ... = -r_p$. By looking only at the part $A * ... = -r_p$, you can derive a formula for d_v. Then, you can solve d_x from the previously derived formula since you already computed d_v. Finally, you can solve d_u directly from the 2nd KKT condition since you already computed values of both d_v and d_x.

This allows us to solve the Newton directions efficiently without needing to invert the whole Newton system matrix. Notice that there are multiple ways to derive these formulas, but some may lead to premature execution of the method due to some matrices becoming ill-conditioned.

(d) Using the update rules derived in part (c), modify this Jupyter notebook to solve problems of the form (3) – (5) with the interior point method. Basically, you have to modify the function which computes the Newton directions based on the formulas derived in part (c). Solve the problem instance given in the skeleton code and include in your answer the picture that the code plots automatically.

(e) **Remember to submit both: (1) the completed Jupyter notebook file, and (2) a pdf file for your answers and the path plot.**

Problem 4.3: Sequential Quadratic Programming

Consider the quadratic optimization problem studied in Exercise 11.2:

$$\min_{x}.\ f(x) = 2x_1^2 + 2x_2^2 - 2x_1x_2 - 4x_1 - 6x_2 \tag{9}$$

$$\text{subject to: } g_1(x) = x_1^2 - x_2 \leq 0 \tag{10}$$

$$g_2(x) = x_1 + 5x_2 - 5 \leq 0 \tag{11}$$

$$g_3(x) = -x_1 \leq 0 \tag{12}$$

$$g_4(x) = -x_2 \leq 0 \tag{13}$$

(a) Solve the problem (9) – (13) using the l_1-SQP variant described at the end of Lecture 11. Use $x^1 = (0.5, 0.5)^\top$ as the initial primal solution and $u^1 = (0, 0, 0, 0)^\top$ as the initial dual solution. Implement the l_1-SQP variant presented at the end of Lecture 11 by modifying this Jupyter notebook. Try to find suitable values for the penalty parameter μ and the trust region parameter Δ^k to obtain convergence. You can keep the trust region parameter Δ^k constant even though it has the index k.

(b) Plot the feasible region defined by the inequalities (10) – (13) and the progress of the l_1-SQP iterations in the same plot. The skeleton code will do this automatically.

(c) **Remember to return your completed Jupyter notebook file when submitting your answers.**