

## Homework 6: Quadratic Programs and Regression

Due date: 11:59pm on Friday 24 March 2023

Submit pdf files via Canvas

### 1. [6 points]

In this problem, you are given a “random” polyhedron

$$P = \{x \in \mathbb{R}^n \mid Ax \leq b\},$$

where  $\sim$  is one of  $\leq, \geq$ , and  $b \in \mathbb{R}^m$ . You will create each  $P$  by choosing its parameters uniformly and randomly in the following intervals:

- $b_i \in [0, 100]$ ,  $i = 1, \dots, m$
- $a_{ij} \in [-2, 2]$ ,  $i = 1, \dots, m$ ,  $j = 1, \dots, n$ .

(Note that  $0 \in P$  because all the components of  $b$  are positive, so we know the polyhedron is nonempty.)

Let  $n = 100, m = 20$ .

- Generate 1000 random points  $\hat{x} \in \mathbb{R}^n$  whose components  $\hat{x}_j$  are chosen randomly from a uniform distribution over  $[-1, 1]$  for all  $j = 1, 2, \dots, n$ . By checking whether or not each of these points lies in  $P$ , estimate the probability `prob_in_P` of a random point of this type lying in  $P$ .
- Save one of the  $\hat{x}$  from part (a) that does *not* lie in  $P$ . For this value of  $\hat{x}$ , find the following quantities:
  - $z_2 := \min_{x \in P} \|x - \hat{x}\|_2$
  - $z_1 := \min_{x \in P} \|x - \hat{x}\|_1$
  - $z_\infty := \min_{x \in P} \|x - \hat{x}\|_\infty$

Recall that

$$\|x\|_2 = \sqrt{\sum_{j=1}^n x_j^2}, \quad \|x\|_1 = \sum_{j=1}^n |x_j|, \quad \|x\|_\infty = \max_{j=1, \dots, n} |x_j|.$$

Note that the first problem can be solved with convex quadratic programming (convex QP), and the second and third minimum distance problems can be solved with linear programming (LP). Ensure you report scalar outputs for the three projection problems.

Please use `set_silent(model_name)` in your code to avoid excessive output. Your code should print only the probability of the point being in  $P$ , and the three values of the norm distances to  $P$ .

### 2. [4 points] Quadratic form positivity. You're presented with the constraint:

$$2x^2 + 6y^2 + 2z^2 + 4xy - 6xz - 6yz \leq 1 \tag{1}$$

- Write the constraint (1) in the standard form  $v^T Q v \leq 1$ . Where  $Q$  is a symmetric matrix. What is  $Q$  and what is  $v$ ?
- It turns out the above constraint is *not* convex. In other words, the set of  $(x, y, z)$  satisfying the constraint (1) is not an ellipsoid. Explain why this is the case.

**Note:** you can perform an orthogonal decomposition of a symmetric matrix  $Q$  in Julia like this:

```
(Lambda,U) = eigen(Q)    # Lambda is the vector of eigenvalues and U is orthogonal
U * diagm(Lambda) * U'   # this is equal to Q (as long as Q was symmetric to begin with)
```

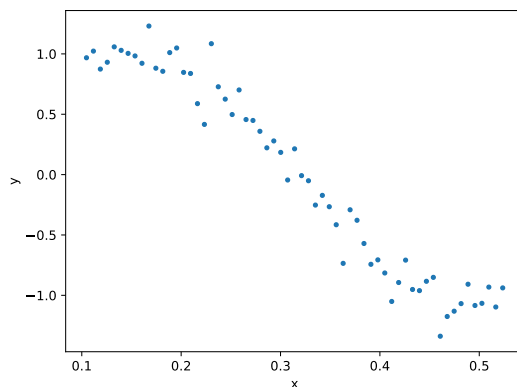
- c) We can also write the constraint (1) using norms by putting it in the form:

$$\|Av\|^2 - \|Bv\|^2 \leq 1$$

What is  $v$  and what are the matrices  $A$  and  $B$  that make the constraint above equivalent to (1)?

- d) Explain how to find a vector  $(x, y, z)$  that satisfies the constraint (1) and that has arbitrarily large magnitude (i.e.  $x^2 + y^2 + z^2$  is as large as you like).

3. [4 points] **Lasso regression.** Consider the data  $(x, y)$  plotted below, available in `lasso_data.csv`.



In this problem, we will investigate different approaches for performing polynomial regression.

- Perform ordinary polynomial regression. Make plots that show the data as well as the best fit to the data for polynomials of degree  $d = 5$  and  $d = 15$ . Also comment on the magnitudes of the coefficients in the resulting polynomial fits. Are they small or large?
- In order to get smaller coefficients, we will use ridge regression ( $L_2$  regularization). Re-solve the  $d = 15$  version of the problem using a regularization parameter  $\lambda = 10^{-6}$  and plot the new fit. How does the fit change compared to the non-regularized case of part a? How do the magnitudes of the coefficients in the resulting polynomial fit change?
- Our model is still complicated because it has so many parameters. One way to simplify our model is to look for a sparse model (where many of the parameters are zero). Solve the  $d = 15$  problem once more, but this time use the Lasso ( $L_1$  regularization). Start with a large  $\lambda$  and progressively make  $\lambda$  smaller until you obtain a model with a small number of parameters that fits the data reasonably well. **Note:** due to numerical inaccuracy in the solver, you may need to round very small coefficients (say less than  $10^{-5}$ ) down to zero. Plot the resulting fit.