Exercises: Penalty function approximation

The *Penalty function approximation* has the form

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^{m} \phi(r_m) \\ \text{subject to} & r = Ax - b. \end{array}$$

Note that for $\phi(x) = x^2$ this is equivalent to the standard linear regression problem.

$$\text{minimize} \quad \|Ax - b\|_2.$$

Your goal is to explain how one can implement the problems below using LP or SOCP constraints (the simpler, the better).

a) Least-squares: $\phi(x) = x^2$

b) ℓ_1-norm approximation: $\phi(x) = |x|$

c) deadzone-linear penalty function:

$$\phi(x) = \begin{cases} 0 & |x| \le a \\ |x| - a & |x| > a \end{cases}$$

d) log-barrier penalty function:

$$\phi(x) = \begin{cases} -a^2 \log(1 - (x/a)^2) & |x| < a \\ \infty & |x| \ge a \end{cases}$$

Note that for small $|x|$, log-barrier is close to x^2.

Hint: consider exp of the objective. Recall that in lab7 (Maximum volume rectangle) we expressed the geometric mean $(\prod_{i=1}^{n} x_i)^{1/n}$ as $O(n)$ SOCP constraints, you can use it here.

e) Huber loss function:

$$\phi(x) = \begin{cases} x^2 & |x| < M \\ M(2|x| - M) & |x| \ge M \end{cases}$$

Hint: Note that $|x| = \min\{|x|, M\} + \max\{|M| - x, 0\}$.

Note: in `cvxpy` there is a function `huber(x,M=1)`, so when in need, you don't need to implement it yourself (unless you want to use an efficient solver directly).