MATH 652: Optimization II

Lecturer: Prof. Wolfgang Bangerth
Blocker Bldg., Room 507D
(979) 845 6393
`bangerth@math.tamu.edu`
`http://www.math.tamu.edu/~bangerth`

Partial answers for homework assignment 1

Problem 1 (l_∞ minimization). In last week's homework, you formulated as a linear program the problem of finding a line of the form $y(t) = at + b$ using l_∞ minimization through this data set:

t_i	0	1	2	3
y_i	1.1	1.9	2.8	3.2

You should have ended up with a linear program in three variables and with eight constraints.

We have seen in class that among the solutions of a linear program is always at least one vertex. A simple way to find the solution is to find all basic solutions, and then

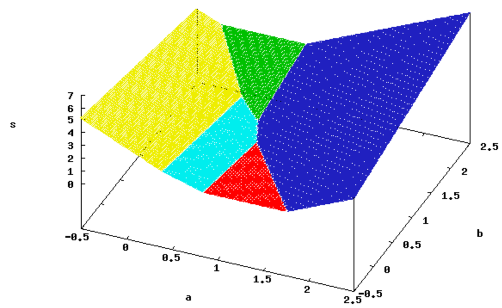
- eliminate those that are not feasible, i.e. select only the feasible basic solutions (we have seen that these are then the vertices of the feasible set)
- choose the feasible basic solution for which the objective function has the smallest value.

Perform this procedure by writing a program (or doing it by hand :-) that computes all basic solutions, tests for feasibility, and then checks their objective function values. Answer the following questions: (i) How many basic solutions exist for this problem? (ii) How many of these are feasible? (iii) What is the minimal objective function value? (iv) At how many vertices is this value attained?

Answer. By simply enumerating all possible active sets, we find the following answers:

(i) There are 48 basic solutions characterized by different active sets. Note first that this is less than the theoretical maximum of $\binom{8}{3=56}$ basic solutions one could get by choosing three constraints from the total of eight. The reason is that in some cases, the three selected constraints are not linearly independent and consequently intersect not at a single point but at a line or a plane. Furthermore, some of 48 basic solutions lie at identical locations, leaving only 30 unique basic solutions.

(ii) Of these, only 3 basic solutions are actually feasible with respect to the remaining constraints and are therefore vertices of the feasible set. This makes sense, given the feasible set you were supposed to show in last week's homework and that was the volume above the following manifold:



Note that the picture allows us to identify the three vertices. In fact, they are at the following locations:

a	b	s
0.4	1.55	0.45
0.7	1.25	0.15
0.85	0.875	0.225

(iii) At the second one of these vertices, the objective function is minimal. In fact, since the objective function is $f(a, b, s) = s$, the optimal value is 0.15 as can be seen from the table.

(iv) By the same token, it is obvious that the two other vertices of the feasible set have larger objective function values, and the optimum is therefore unique.

When computing whether the matrix whose rows contains the vectors a_i^T of the three selected constraints has full rank, one must take care of the fact that on a computer we can only operate with finite precision. For example, one way to test whether a matrix has full rank would be to compute its determinant and see if it is non-zero. However, in finite precision, a matrix whose determinant is 10^{-16} is, for all practical purposes, singular. Real-world programs therefore never test floating point numbers x for finiteness, but use a test of the form

$$|x| \geq \epsilon\xi$$

where ϵ is of the order of magnitude of round-off (for example 10^{-15} when using double precision) and ξ is a (positive) number that indicates the typical size and in the same physical units of numbers like x. For example, a practical test for singularity of a matrix could test whether

$$\det A \geq 10^{-15} \frac{1}{n^2} \sum_{i,j=1}^{n} |a_{ij}|.$$

Similarly, one has to take care when testing whether a constraint is satisfied by a vector x or not (i.e. whether it is feasible). In mathematical notation, we simply test whether

$$a_i^T x \geq b_i \qquad \forall i.$$

On the other hand, in finite computer arithmetic, a vector x may be so that after evaluating the left hand side it is, for all practical purposes, almost exactly equal to b_i, but happens to be less than b_i due to round-off. A practical test would be to use

$$a_i^T x \geq b_i - \epsilon\xi \qquad \forall i.$$

Here, one could choose ξ as $\xi = \|a_i\|\|x\|$.

Note that in both cases above we have not chosen a fixed prescribed tolerance $\epsilon = 10^{-15}$ but have instead made it relative to the expected size of objects on the left and right hand sides of comparisons. The reason is that we want this to work independent of the size of vector or matrix elements: if someone is solving linear problems from quantum mechanics, then the elements of a_i and x may be of the order 10^{-34} in which we case almost every vector would satisfy the inequality

$$a_i^T x \geq b_i - \epsilon \qquad \forall i$$

simply because all quantities except for ϵ are so small. Conversely, in linear problems from cosmology where $x \simeq 10^{20}$, vectors x for which a constraint is active may or may not satisfy this inequality because the round-off involved in evaluating the left hand side would be of the order of 10^{-16} times 10^{20} – far larger than the tolerance we allow on the right. The only way to avoid such trouble is to make the tolerance dependent on the expected size of round-off, which includes the expected size of the objects with which we compute.

Problem 2 (l_∞ minimization, again). Repeat the same problem with the following set of measured data:

t_i	0	1	2	3	4	5	6	7	8	9
y_i	1.1	1.9	2.8	3.2	4	5	6	7	8	9

Answer the same questions as before. What is likely going to be the problem in your algorithm if I kept giving you more and more data points? Can you estimate how the number of operations your algorithm needs to perform grows asymptotically if the number N of data points grows?

Answer. With these now 10 data points, we get 20 constraints on the slack variable s and consequently there are at most $\binom{20}{3=1140}$ basic solutions. Due to the potential of linearly dependent rows, there are in fact only 900 basic solutions, only 4 of which are feasible. The optimum lies at $x^* = (a, b, s)^T = (\frac{79}{90}, \frac{132}{180}, \frac{11}{36})$ where the objective function has value $\frac{11}{36}$. This solution is unique.

In general, to find the solution of this problem with N data points, we get a problem in three variables with $m = 2N$ constraints. Thus, we have to check for $\binom{m}{3} = \frac{1}{3}N(2N-1)(2N-2)$ matrices of size $3 times 3$ that they have full rank (which costs 3^3 operations), solve for the basic solution (which can be done at the same time as checking for full rank), compare the remaining $2N-3$ constraints for feasibility at a cost of $(2N-3)3$ multiplications, and evaluate the basic solution with the previously best (1 operation). This yields a total effort of order $\mathcal{O}(N^4)$ operations.

While this is certainly not an appealing prospect (the effort will be quite large if we have to, say, fit a line through 1,000 data points), it is also not a terrible complexity – at least the algorithm is still of polynomial order in N, and not exponential! The main reason is that here the number of possible vertices of the feasible set only grows polynomially with the number of variables and constraints, not exponentially.