Numerical optimization, lecture 1

W. Hebisch

October 12, 2021

1 Books

- Stephen Boyd, Lieven Vandenberghe, Convex Optimization, available online at `http://web.stanford.edu/~boyd/cvxbook/`
- David G. Luenberger, Yinyu Ye, Linear and Nonlinear Programming, Springer 2008
- Yurii Nesterov, Introductory lectures on convex optimization, Springer 2004
- Jorge Nocedal, Stephen J. Wright, Numerical Optimization, Springer 2006

2 Introduction

General optimization problem: given a set S and a function $f : S \to \mathbb{R}$ find $x_0 \in S$ such that

$$f(x_0) = \max_{x \in S} f(x).$$

In such case we write

$$x_0 = \operatorname{argmax} f(x).$$

Similarly, for minimal value we have min and argmin.

2.1 Why optimization?

Classical: decision making. We want maximal effect from given resources. Or to get desired effect at minimal cost.

This studies: most statistical estimation and machine learning uses optimization.

Approximation problem: if $y \notin S$ we want to find best approximation in S. If d measures quality of approximation we want to find

$$\operatorname{argmin}_{x \in S} d(y, x).$$

Least square linear regression: given $x_i \in \mathbb{R}^k, y_i \in \mathbb{R}^l$, $i = 1, \ldots n$ we want to find best linear approximation, that is find matrix A which minimizes

$$\sum_{i=1}^{n} \|y_i - Ax_i\|^2.$$

Regularization of ill posed problems: we want to solve equation

$$f(x) = y.$$

When f is not invertible or inverse is badly behaved (for example discontinuous) natural approach requires some regularity of solution. If P measures regularity we minimize

$$d(f(x), y) + P(x).$$

where d is a distance function in the image. In simplest case we use square of euclidean norm

$$d(x, y) = \|x - y\|^2,$$
$$P(x) = \|x\|^2.$$

Important special case: LASSO. Given $\lambda \in \mathbb{R}$, $\lambda > 0$, $x_i \in \mathbb{R}^k, y_i \in \mathbb{R}$ for $i = 1, \ldots, n$ find

$$\text{argmin}_{b,c} \sum |y_i - c - (b, x_i)|^2 + \lambda \|b\|_1$$

where $b \in \mathbb{R}^k$, $c \in R$ and $\|b\|_1 = \sum_{j=1}^{k} |b_j|$. Using $\|b\|_1$ tends to promote sparse solutions.

Typically there are ready to use implementations of optimization methods, why we can not just use them?

- different methods have different properties, we need to choose one good for specific problem
- we need understanding to combine/tweak methods in useful ways
- we need understanding for troubleshooting (when creating complex systems something *will* go wrong)
- we may need to change inner working to get gain from properties of specific problem (like sparsity)

2.2 Narrowing problem

General optimization problem stated before is too general. For example, let A be any logical formula on S. Let $f(x) = 1$ when $A(x, y)$ is true and $f(x, y) = 0$ otherwise. Clearly $\max_x f(x, y) = 1$ if and only if for given y there exists x such that $A(x, y)$ is true. In other words, using optimization procedure on f we can solve validity of formula $B(y) = \exists_x A(x, y)$. Such problem may be arbitrarily

hard, in particular there is computable A such that B is not computable. Easy A may lead to NP-complete problem for B.

In the future we will work mostly with rather regular subsets of $\mathbb{R}^k$ and reasonably regular functions. In particular we will assume that functions are almost everywhere differentiable, but we allow nondifferentiability like $|x|$.

However, there exists polynomials P and Q with integer coefficients such that parametric problem

$$\operatorname{argmin}_x P(x,y) + Q(x,y) \prod_{i=1}^{k} \sin^2(\pi x_i)$$

is unsolvable. Above, difficulty appears because $\|x\|$ may be huge and there is no computable bound on $\|x\|$. If we limit x to a bounded set one can easily get NP-hard problem.

Remark. Difficulty above is related to difficulty of finding integer solutions. Namely, for given polynomial S we can build polynomial Q such that problem above with $P = S^2$ for any integer vector y has minimum 0 if and only if there is integer vector x such that $P(x,y) = 0$. Yuri Matiyasevich showed that finding integer solutions to polynomial equations is uncomputable, so also our optimization problem is uncomputable.

We can also get NP hard problems.

Example: Let

$$s_1 = x_1 + x_2 + x_3,$$

$$s_2 = x_1x_2 + x_1x_3 + x_2x_3,$$

$$s_3 = x_1x_2x_3,$$

$$Q = 1 - (s_1^2 - 3s_2 + s_3).$$

One can check that for $x \in [0,1]^3$ we have $0 \le Q \le 1$ with equality only at vertices of the cube. Moreover $Q(0,0,0) = 1$ and at other vertices $Q = 0$.

Consider boolean formulas in variables $x_1, \ldots, x_k$. Let x_{i+k} be negation of x_i (this is to avoid explicitly writing negations). Given boolean formula

$$B = (x_{j_{1,1}} \vee x_{j_{2,1}} \vee x_{j_{3,1}}) \wedge \cdots \wedge (x_{j_{1,m}} \vee x_{j_{2,m}} \vee x_{j_{3,m}})$$

we build polynomial f in $y_1, \ldots, y_k$ as

$$f = Q(y_{j_{1,1}}, y_{j_{2,1}}, y_{j_{3,1}}) + \cdots + Q(y_{j_{1,m}}, y_{j_{2,m}}, y_{j_{3,m}})$$

where $y_{i+k} = 1 - y_i$. Let S be unit hypercube, that is set of y such that $0 \le y_i \le 1$ for $i = 1, \ldots, k$. One can show that $\min f$ on S is zero if and only if there is substitution of truth values for variables in B which makes B true.

Discrete problem above is usually called 3-SAT. It is NP-complete problem. However, many instances of 3-SAT are easily solvable. Trying projected gradient descent (which is one of methods that we will study later) on easily solvable 3-SAT instances sometimes gives correct answer, but frequently converges to non-optimal point (local minimum).

In practice local minimum may be good enough, in particular this is the case in neural network training.

We need to limit to more special problems. One condition which assures good properties is convexity. On nonconvex problems there may be difference between local minimum and global minimum and we typically must be satisfied with local minimum.

2.3 Desired features of solution method

We need following feature from solution method:

- efficiently handle problems of high dimension
- in many cases low accuracy solution is good enough
- robust (can cope with errors in input data)

Sometimes we need methods which tolerate points with no derivative.

2.4 Discrete optimization

For optimization on discrete sets we need different methods, most is outside scope for this course. Some discrete methods are similar to or use continuous methods. If time permits we will have some examples.

2.5 Equivalent problems

We frequently transfer problems to different, more convenient form in such a way that we can easily recover solution of original problem from solution of transformed problem. In such case we say that the two problems (original one and transformed problem) are equivalent.

For example, problem of maximizing f may be replaced by problem of minimizing $-f$. More generally, we may replace f by composition with a monotonic function.

Or we may add extra variables and rewrite problem in terms of new variables.

We do not give precise definition of equivalent problems. Any simple definition risk missing some way of transforming problems or allowing too general transformations. Rather, we will give several examples of transformations.

3 Linear programming

General form: Minimize or maximize linear function $\langle c, x\rangle$ on a set S defined by system of linear equations and inequalities.

Standard form:

$$\operatorname{argmin}_{x\in S}\langle c, x\rangle$$

where $S = \{x : Ax = b, x \geq 0\}$. Here $(x_1, \ldots, x_k) \geq 0$ means that $x_i \geq 0$ for $i = 1, \ldots, k$.

The set S above is called feasible set (or feasible region), elements of S are called feasible points (or feasible vectors).

For theoretical purposes we can transform any linear programming problem to standard form.

- if original problem requests maximum we replace c by $-c$
- we can replace inequality $\sum_{i=1}^{m} a_i x_i \leq b$ by equality $\sum_{i=1}^{m} a_i x_i + x_{m+1} = b$ introducing extra variable
- when original variable x_i has value in $\mathbb{R}$ we replace it by pair t_i, s_i putting in all equation and in goal function $t_i - s_i$ instead of x_i

Example of transformation to standard form:

$$\begin{array}{ll} \text{maximize} & x + y \\ \text{subject to} & -2 \leq x + 2y \leq 7 \\ & \text{and } x \geq -1. \end{array}$$

First, we change to minimization of $-x - y$. We have 3 inequalities so we introduce 3 extra variables z_1, z_2, z_3. We rewrite

$$-2 \leq x + 2y$$

as

$$x + 2y - z_1 = -2.$$

We rewrite

$$x + 2y \leq 7$$

as

$$x + 2y + z_2 = 7.$$

We rewrite

$$x \geq -1$$

as

$$x - z_3 = -1.$$

This gives problem in form:

$$\begin{array}{ll} \text{minimize} & -x - y \\ \text{subject to} & x + 2y - z_1 = -2 \\ & x + 2y + z_2 = 7 \\ & x - z_3 = -1 \\ & \text{and } x \in \mathbb{R},\ y \geq 0,\ z_1 \geq 0,\ z_2 \geq 0,\ z_3 \geq 0 \end{array}$$

To get standard form we need to replace x by $z_4 - z_5$ where z_4 and z_5 are new variables. So finally we get problem in standard form

$$\begin{array}{ll} \text{minimize} & -z_4 + z_5 - y \\ \text{subject to} & z_4 - z_5 + 2y - z_1 = -2 \\ & z_4 - z_5 + 2y + z_2 = 7 \\ & z_4 - z_5 - z_3 = -1 \\ & \text{and } y \geq 0,\ z_1 \geq 0,\ z_2 \geq 0,\ z_3 \geq 0,\ z_4 \geq 0,\ z_5 \geq 0 \end{array}$$

Example: diet problem. x represents various foods (ingredients), matrix A represent nutritional content of foods, equation $Ax = b$ means that nutritional requirements are satisfied. c represent cost of foods and we want to minimize total cost.

Example: Manufacturing problem. x represents various products. Ax gives resources needed to make x. Assuming maximal resource use is limited by b we get inequality $Ax \leq b$. c represents prices, we want to maximize total value of products.

Example: Transportation problem. We have k sources with given capacities s_i and l destinations with demands d_j and need to move a single product from sources do destinations. $x_{i,j}$ is amount of product moved from source i to destination j. Demand should be satisfied, that is

$$d_j = \sum_{i=1}^{k} x_{i,j}.$$

We should not exhaust source capacity, that is

$$s_i \geq \sum_{j=1}^{l} x_{i,j}.$$

$c_{i,j}$ represents cost of moving unit of product from source i to destination j, so

$$\sum_{i=1}^{k} \sum_{j=1}^{l} c_{i,j} x_{i,j}$$

is total cost that we want to minimize.

In theory we usually assume that feasible set is nonempty (otherwise linear programming problem has no solution). However, in practice it may happen that all we need is to know if feasible set is nonempty and possibly find some feasible point. Also finding out if feasible set is nonempty is essentially as hard as solving linear programming problem. On the other hand, we can add extra variables to linear programming problem and modify equations so that for new problem there is obvious feasible point. By assigning high cost to extra variable we ensure that new variables are zero in optimal solution.

3.1 Simplex algorithm, basic idea

Lemma 3.1 *One of the following holds*

- *the feasible set S is empty*
- *the goal function $\langle c, x \rangle$ is unbounded from below*
- *$\langle c, x \rangle$ attains minimal value at a boundary point of S*

Remark: For problem in standard form by definition boundary points have one of coordinates equal to 0.

Example: Let $S = \{(x, y) : x \geq 0, y \geq 0\}$. When $f = 0$ then all points in S are optimal, in particular boundary points. When $f = x$ then all points in S with $x = 0$ are optimal. When $f = x + y$ then $(0, 0)$ is unique optimal point. When $f = -x$ then goal function is unbounded from below.

Remark: proof below is for linear programming problem in standard form, but transforming problem to standard form we get result in general case.

Proof of the lemma: If goal function is unbounded from below the claim is true, so we may assume that is is bounded from below. If S is empty or consists of one point then the claim is clearly true. If goal function is constant, then any point in S is an optimal solution, so in this case it is enough to show existence of boundary point. To show existence of boundary point we may replace goal function by one of coordinates, so we may assume in the sequel that goal function is not a constant.

Let y and z be two points in S giving different value of goal function. Consider line L passing trough y and z. Clearly for $x \in L$ we have $Ax = b$ and intersection I of L with S is an interval or a halfline (it can not be a line because at least one coordinate must change and it would became negative at some point of I, which is impossible).

The goal function restricted to I is linear, so since it is bounded from below it attains minimum at endpoint of I. So we have found boundary point t of S such that $\langle c, t\rangle \leq \langle c, x\rangle$ and one of coordinates of t is zero.

It remains to show that minimal value is attained. We do this by induction on number of variables. Clearly, if number of variables is small enough then S is at most one dimensional and then $S = I$ and minimal value is attained. Otherwise, let $S_j = \{x \in S : x_j = 0\}$. S_j is feasible set of linear programming problem with smaller number of variables (we remove x_j from set of variables), so by inductive assumption minimal value on S_j is attained, so also minimal value on $H = \bigcup_{j=1}^{k} S_j$ is attained. But, by previous part minimal value on H is also minimal value on S. □

Now, in view of the lemma it is natural to seek solutions so that as many as possible coordinates of x are zero. Let $J \subset \{1, \ldots, k\}$ and $\mathrm{card}(J) = l = \mathrm{rank}(A)$. We say that solution to $Ax = b$ is basic solution corresponding to J if $x_j \neq 0$ implies $j \in J$ and columns A_j of A for $j \in J$ are linearly independent (so they form basis of image of A).

Coordinates x_j with $j \in J$ are called basic coordinates. Note that linear independence of columns A_j means that in basic solution basic coordinates are uniquely determined by the equation $Ax = b$. So, if $x_j > 0$ for $j \in J$ we can not zero any more coordinates without loosing property of having basic solution corresponding to J.

It may happen that in basic solution corresponding to J for some basic coordinate is zero. Such basic solution is called degenerate.

We say that solution is basic solution if it is basic solution corresponding to some J.

We say that feasible point x is basic feasible point if it is basic solution.

We say that optimal solution is basic optimal solution if it is it is basic solution.

Lemma 3.2 *If linear programming problem has optimal solution, then it also has basic optimal solution. If linear programming problem has feasible solution, then it also has basic feasible solution.*

Proof: Let x be feasible solution and let $I = \{j : x_j > 0\}$. If columns of A with numbers from I are linearly independent, then x is basic solution corresponding to some $J \supset I$, which gives the claim. Otherwise consider a new problem when we restrict coordinates to I (that is we set all other coordinates to zero). For the new problem x is not a boundary point of feasible set, so if x is optimal solution by previous lemma we get new optimal solution such that more coordinates are zero, that is with smaller I, which gives claim by induction. When dealing with nonoptimal feasible solution, we can use one of coordinates as new goal function. Since coordinates of feasible points are nonnegative, new goal function is bounded from below, so new problem a has basic optimal solution. This solution is basic feasible solution to original problem. □

So, given basic feasible point we would like to check if it is optimal solution or find better feasible point.

Second idea: for given J basic coordinates are uniquely determined by nonbasic coordinates. This allows rewriting goal function so that new function is equal to old on S and basic coordinates of new c are zero. If all nonbasic coordinates of c are nonnegative, then corresponding basic solution is optimal. If some nonbasic coordinate c_j of c is negative and solution is nondegenerate, then we can increase x_j decreasing goal function till one of basic coordinates becomes zero. Degeneracy complicates problem, but modified version still works.