Numerical optimization, lecture 1

W. Hebisch

Februray 22, 2023

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

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

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. Easy example

$$S(x,y) = x_1^2 - yx_2^2 - 1$$

already shows that for moderate y solution may be quite large.

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).

We will later study convex and concave function. The Q above is concave. But we could use different building block Q, namely

$$Q = 1 - (s_1 - s_2 + s_3)$$

Experiments show that for purpose of finding solution to 3-SAT concave version behaves worse the second one.

Actually, using non-smooth function one can do much better than the nice Q-s that we gave.

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.

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.

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.

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.

In particular, by adding extra variable and changing set we can find equivalent problem with linear goal function. Namely, put $T = \{(t, x) : x \in S, t \geq f(x)\}$. Then

$$\min_{x \in S} f(x) = \min_{(t,x) \in T} t,$$

so we replaced problem of minimizing f on S by problem of minimizing t on T.