

# CS 8803-MDM Lecture 24

## *Parameter Optimization III: Convex*

Alexander Gray

`agray@cc.gatech.edu`

Georgia Institute of Technology

# Today

1. Convex Optimization Problems
2. Convex Optimization Methods

# Convex Optimization Problems

Problems with a convex objective function and, if there are constraints, convex constraints.

# Convex Functions

It used to be that optimization was divided into linear and nonlinear objective functions. Now the distinction is between convex and non-convex functions.

A *convex set*  $C$  has the property that for any two points  $x_1$  and  $x_2$  in  $C$ ,  $0 \leq \alpha \leq 1$ , a line segment

$$\alpha x_1 + (1 - \alpha)x_2 \in C. \quad (1)$$

A *convex function*  $f$  has the property that  $\text{dom } f$ , the domain of  $f$ , is a convex set and for any two points in  $\text{dom } f$ ,

$$f(\alpha x_1 + (1 - \alpha)x_2) \leq \alpha f(x_1) + (1 - \alpha)f(x_2). \quad (2)$$

This inequality is often called *Jensen's inequality*.  $f$  is *concave* if  $-f$  is convex.

# Convex Functions

Examples of convex functions:

- affine functions  $f(x) = Ax + b$
- quadratic functions
- exponential  $f(x) = e^{ax}$
- powers  $f(x) = x^a$ ,  $a \geq 1$  or  $a \leq 0$
- logarithm  $f(x) = \log x$
- all norms
- maximum  $f(x) = \max(x_1, \dots, x_N)$
- nonnegative weighted sum of convex functions  
 $f = \alpha_1 f_1 + \dots + \alpha_K f_m$ ,  $\alpha_k \geq 0$
- maximum of convex functions  $f(x) = \max(f_1(x), f_2(x))$

# Convex Functions

Suppose  $f$  is differentiable (its gradient  $\nabla f$  exists at each point in  $\text{dom} f$ ). Then  $f$  is convex iff  $\text{dom} f$  is convex and for any two points in  $\text{dom} f$ ,

$$f(x_2) \geq f(x_1) + \nabla f(x_1)^T (x_2 - x_1). \quad (3)$$

Thus, if  $\nabla f(x) = 0$ , then for all  $\tilde{x} \in \text{dom} f$ ,  $f(\tilde{x}) \geq f(x)$ , i.e.  $x$  is a *global* minimizer of  $f$ . This is what makes convex functions special.

If  $f$  is twice differentiable,  $f$  is convex iff  $\text{dom} f$  is convex and its Hessian is positive definite (for all  $x \in \text{dom} f$ ,  $\nabla^2 f(x) \geq 0$ ), or in 1 dimension,  $f''(x) \geq 0$ .

# Least-Squares

Consider the least-squares problem

$$\text{Find } x^* = \arg \min_{x \in \mathbb{R}^D} \|Ax - b\|_2^2 = x^T A^T Ax - 2b^T Ax + b^T b. \quad (4)$$

This is quadratic and unconstrained.

It can be solved analytically as  $x^* = A^{-1}b$ . So this is a special easy case.

# Unconstrained Convex Optimization

The conditions for optimality of a convex function, in the unconstrained case, boil down to the necessary and sufficient condition

$$\nabla f(x) = 0. \quad (5)$$

So for convex functions without constraints, unconstrained optimization methods like Newton's method find the *global* minimum.

If we use  $L_1$  or  $L_\infty$  minimization, we end up with a constrained optimization problem.

# General Form

From now on, “convex optimization” refers to problems having convex objective functions and convex constraints. For such problems, we can find the global minimum, in polynomial time.

Recall the general form for an optimization problem:

$$\text{Find } x^* = \arg \min_{x \in \mathbb{R}^D} f(x) \quad (6)$$

$$\text{subject to } c_i(x) \geq 0, \quad i = 1, \dots, M \quad (7)$$

$$d_i(x) = 0, \quad i = 1, \dots, N. \quad (8)$$

The domain is the intersection of the domains of the constraint functions. A point is *feasible* if it satisfies the constraints.

# Linear Programming

When the objective and constraint functions are all affine, the problem is called a *linear program* (LP), which has the form

$$\text{Find } x^* = \arg \min_{x \in \mathbb{R}^D} c^T x + d \quad (9)$$

$$\text{subject to } Gx \leq h \quad (10)$$

$$Ax = b. \quad (11)$$

# Quadratic Programming

When the objective function is quadratic and the constraint functions are affine, the problem is called a *quadratic program* (QP), which has the form

$$\text{Find } x^* = \arg \min_{x \in \mathbb{R}^D} \frac{1}{2} x^T P x + q^T x + r \quad (12)$$

$$\text{subject to } Gx \leq h \quad (13)$$

$$Ax = b. \quad (14)$$

If the constraints are also quadratic, the problem is called a *quadratically constrained quadratic program* (QCQP):

$$\text{Find } x^* = \arg \min_{x \in \mathbb{R}^D} \frac{1}{2} x^T P x + q^T x + r \quad (15)$$

$$\text{subject to } \frac{1}{2} x^T P_i x + q_i^T x + r_i \leq 0, \quad i = 1, \dots, M \quad (16)$$

$$Ax = b. \quad (17)$$

# Second-order Cone Programming

A closely related problem is called a *second-order cone program* (SOCP), which has the form

$$\text{Find} \quad x^* = \arg \min_{x \in \mathbb{R}^D} f^T x \quad (18)$$

$$\text{subject to} \quad \|A_i x + b_i\|_2 \leq c_i^T x + d_i, \quad i = 1, \dots, M \quad (19)$$

$$F x = g. \quad (20)$$

A constraint of this form is called a *second-order cone constraint*.

# Geometric Programming

A *geometric program* (GP) is a problem of the form

$$\text{Find } x^* = \arg \min_{x \in \mathbb{R}^D} f(x) \quad (21)$$

$$\text{subject to } c_i(x) \leq 1, \quad i = 1, \dots, M \quad (22)$$

$$d_i(x) = 1, \quad i = 1, \dots, N \quad (23)$$

$$(24)$$

where  $f$  and the  $c_i$  have the form

$$\log \left( \sum_k e^{a_{1k}^T x + b_{1k}} \right) \quad (25)$$

and the  $d_i$  have the form  $e^{g_i^T x + h_i}$ .

# Semidefinite Programming

A *semidefinite program* (SDP) has the form

$$\text{Find } x^* = \arg \min_{x \in \mathbb{R}^D} c^T x \quad (26)$$

$$\text{subject to } x_1 F_1 + \dots + x_n F_n + G \leq 0 \quad (27)$$

$$Ax = b. \quad (28)$$

$$(29)$$

# Convex Optimization Methods

The interior-point method.

# Lagrangian Duality

Consider again the general form for an optimization problem:

$$\text{Find } x^* = \arg \min_{x \in \mathbb{R}^D} f(x) \quad (30)$$

$$\text{subject to } c_i(x) \geq 0, \quad i = 1, \dots, M \quad (31)$$

$$d_i(x) = 0, \quad i = 1, \dots, N. \quad (32)$$

# Lagrangian Duality

We can rewrite the original problem as the unconstrained optimization problem

$$\text{Find } x^* = \arg \min_{x \in \mathbb{R}^D} f(x) + \sum_i^M I_\infty(c_i(x)) + \sum_i^N I_\infty(d_i(x)). \quad (33)$$

where  $I_\infty$  is the indicator-like function which takes the value 0 if the constraint function is satisfied by  $x$  and  $\infty$  otherwise.

# Lagrangian Duality

The basic idea of Lagrangian duality is to soften the problem, by replacing  $I_\infty(c_i(x))$  by  $\lambda_i c_i(x)$  and  $I_\infty(d_i(x))$  by  $\eta_i d_i(x)$ , where the  $\lambda_i$  and  $\eta_i$  are positive weights, obtain the *Lagrangian* of the problem:

$$L(x, \lambda, \eta) = f(x) + \sum_i^M \lambda_i c_i(x) + \sum_i^N \eta_i d_i(x). \quad (34)$$

The  $\lambda_i$  and  $\eta_i$  are called the *Lagrange multipliers*, and the  $\lambda$  and  $\eta$  vectors are called the *dual variables* or *Lagrange multiplier vectors* associated with the problem.

# Lagrangian Duality

We define the *dual function*  $g$  as the minimum value of the Lagrangian over  $x$ :

$$g(\lambda, \eta) = \inf_x L(x, \lambda, \eta). \quad (35)$$

Since the dual function is the pointwise infimum of a family of affine functions of  $(\lambda, \eta)$ , it is concave, even if the original problem is not convex. The dual function yields a lower bound on the optimal value:  $g(\lambda, \eta) \leq x^*$ .

# Lagrangian Duality

The dual function value  $g(\lambda, \eta)$  is its optimal value over  $x$ . Now we'd like to find the best lower bound that can be obtained from the dual function:

$$\text{Find } \lambda^*, \eta^* = \arg \max_{\lambda, \eta} g(\lambda, \eta) \quad (36)$$

$$\text{subject to } \lambda \geq 0. \quad (37)$$

This is called the *dual problem*, while the original problem is called the *primal problem*. This is always a convex optimization problem because the objective function is concave and the constraint is convex, even if the primal problem is not convex.

# Lagrangian Duality

If the primal problem is convex, usually maximizing the dual problem is the same as minimizing the primal problem. We call this *strong duality*.

Suppose we have strong duality, and all the functions are differentiable. Since  $x^*$  minimizes  $L(x, \lambda^*, \eta^*)$  over  $x$ , it follows that its gradient is zero at  $x^*$ :

$$\nabla f(x^*) + \sum_i^M \lambda_i^* \nabla c_i(x^*) + \sum_i^N \eta_i^* \nabla d_i(x^*) = 0. \quad (38)$$

# KKT Optimality Conditions

Thus we have these conditions which must hold at the optimum:

$$c_i(x^*) \leq 0 \quad (39)$$

$$d_i(x^*) = 0 \quad (40)$$

$$\lambda_i^* \geq 0 \quad (41)$$

$$\lambda_i^* c_i(x^*) = 0 \quad (42)$$

$$\nabla f(x^*) + \sum_i^M \lambda_i^* \nabla c_i(x^*) + \sum_i^N \eta_i^* \nabla d_i(x^*) = 0. \quad (43)$$

These are called the *Karush-Kuhn-Tucker* (KKT) conditions. They are necessary and sufficient at the optimum. We can thus formulate optimization as solving these equations.

# Hierarchy of Problems

There is a kind of hierarchy of problems:

1. Simple quadratic problems can be solved analytically.
2. Newton's method solves a sequence of quadratic problems.
3. Newton's method can be applied to equality-constrained optimization problems by removing the equality constraints to create unconstrained problems.
4. Interior-point methods solve a sequence of equality-constrained optimization problems using Newton's method.

# Logarithmic Barrier Idea

We'll now do something related but slightly different with the constraints. We rewrite the general problem as

$$\text{Find } x^* = \arg \min_{x \in \mathbb{R}^D} f(x) + \sum_i^M I_\infty(c_i(x)) \quad (44)$$

$$\text{subject to } Ax = b. \quad (45)$$

We approximate the indicator function by

$$\hat{I}(u) = -\frac{1}{t} \log(-u) \quad (46)$$

where  $t$  is a parameter which increases the accuracy of the approximation as it increases.  $\hat{I}(u)$  goes to  $\infty$  as  $u$  increases to zero, but is differentiable, and convex.

# Logarithmic Barrier Idea

We call the function

$$\phi(x) = - \sum_i^M \log(-c_i(x)) \quad (47)$$

the *logarithmic barrier* for the problem. We'll optimize  $f(x) + \frac{1}{t}\phi(x)$ .

The *barrier method* solves a sequence of such problems (each of which is convex), increasing  $t$  on each iteration, using Newton's method. The solution  $x^*(t)$  is the starting point for the next value of  $t$ .

It can be shown that the error for each iteration is bounded by  $M/t$ , and thus the error goes to zero.

# Interior-Point Method

The barrier method is an example of an *interior-point* method.

Given feasible  $x$ ,  $t > 0$ ,  $\mu > 0$ , tolerance  $\epsilon > 0$ , repeat:

1. Find  $x^*(t)$  by using Newton's method to minimize  $tf + \phi$  subject to  $Ax = b$ , starting at  $x$ .
2.  $x = x^*(t)$ .
3. Quit if  $M/t < \epsilon$ .
4.  $t = \mu t$ .

Methods called *phase I* methods are used to choose the starting  $x$ .

# Interior-Point Method

The interior-point method can be used for all of the constrained convex optimization problems.

In practice, despite convergence analysis which relates the number of iterations to  $M$ , it always takes about 10-20 iterations.

There is a modification of the barrier method called the *primal-dual* interior-point method, which is often a bit faster, and is what is used in practice.

# SVM Quadratic Program

Recall that the formulation of the support vector machine results in a quadratic program.

Though generally effective, interior-point is not sufficient to handle the large number of variables and constraints in an SVM problem. An idea called *chunking* is often used for large-scale convex optimization problems, which solves smaller problems using subsets of the constraints, while adding more constraints until all of them are satisfied.

# SMO: Extreme Chunking

The *sequential minimization optimization* algorithm was developed specifically for SVM's. It is a special case of chunking which considers only two constraints at a time, the minimum number.

While it can be shown to converge, it is based on several heuristic steps. Empirically it is much more efficient than interior-point or general chunking.

The exact reason it has an advantage in this setting has not yet been elucidated.