CSE 6740 Lecture 21

How Do I Optimize With Constraints?
(Constrained Optimization)

Ravi Sastry and Alexander Gray
agrav@cc.gatech.edu

Georgia Institute of Technology
Today

1. Convex (Constrained) Optimization Problems
2. Convex (Constrained) Optimization Methods
3. SMO: An Active Set Method
Convex (Constrained) Optimization Problems

Problems with a convex objective function and, if there are constraints, convex constraints.
When the objective and constraint functions are all affine, the problem is called a *linear program* (LP), which has the form

$$\text{Find } x^* = \arg \min_{x \in \mathbb{R}^D} c^T x + d$$  \hspace{1cm} (1)

subject to

$$Gx \leq h$$  \hspace{1cm} (2)

$$Ax = b.$$  \hspace{1cm} (3)
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

Find \( x^* = \arg \min_{x \in \mathbb{R}^D} \frac{1}{2} x^T P x + q^T x + r \) \hspace{1cm} (4)

subject to

\( Gx \leq h \) \hspace{1cm} (5)

\( Ax = b. \) \hspace{1cm} (6)
Quadratically Constrained QP

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

Find $x^* = \arg \min_{x \in \mathbb{R}^D} \frac{1}{2} x^T P x + q^T x + r$  \hspace{1cm} (7)

subject to $\frac{1}{2} x^T P_i x + q_i^T x + r_i \leq 0, \quad i = 1, \ldots, M$  \hspace{1cm} (8)

$Ax = b.$  \hspace{1cm} (9)
Second-order Cone Programming

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

Find \( x^* = \arg \min_{x \in \mathbb{R}^D} f^T x \) \hspace{1cm} (10)
subject to \( \|A_i x + b_i\|_2 \leq c_i^T x + d_i, \quad i = 1, \ldots, M \) \hspace{1cm} (11)
\[ F x = g. \] \hspace{1cm} (12)

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

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

\[
\text{Find } \quad x^* = \arg \min_{x \in \mathbb{R}^D} f(x) \\
\text{subject to } \quad c_i(x) \leq 1, \quad i = 1, \ldots, M \\
\quad \quad d_i(x) = 1, \quad i = 1, \ldots, N
\]  

(13) \quad (14) \quad (15) \quad (16)

where \( f \) and the \( c_i \) have the form

\[
\log \left( \sum_k e^{a_{1k}^T x + b_{ik}} \right)
\]

(17)

and the \( d_i \) have the form \( e^{g_i^T x + h_i} \).
Semidefinite Programming

A *semidefinite program* (SDP) has the form

\[
\text{Find} \quad x^* = \arg \min_{x \in \mathbb{R}^D} c^T x \\
\text{subject to} \quad x_1 F_1 + \ldots + x_n F_n + G \leq 0 \\
A x = b.
\]
Relationships

Generality:
LP < QP < QCQP < SOCP < SDP.

Computational cost:
LP < QP < QCQP < SOCP < SDP.
Convex (Constrained) Optimization Methods

The interior-point method.
Lagrangian Duality

Consider again the general form for an optimization problem:

\[ \text{Find } x^* = \arg \min_{x \in \mathbb{R}^D} f(x) \]

subject to
\[ c_i(x) \geq 0, \quad i = 1, \ldots, M \]
\[ d_i(x) = 0, \quad i = 1, \ldots, N. \]
Lagrangian Duality

We can rewrite the original problem as the unconstrained optimization problem

Find \( x^* = \arg \min_{x \in \mathbb{R}^D} f(x) + \sum_{i}^{M} I_{\infty}(c_i(x)) + \sum_{i}^{N} I_{\infty}(d_i(x)). \) (24)

where \( I_{\infty} \) is the indicator-like function which takes the value 0 if the constraint function is satisfied by \( x \) and \( \infty \) otherwise.
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:

$$
L(x, \lambda, \eta) = f(x) + \sum_{i}^{M} \lambda_i c_i(x) + \sum_{i}^{N} \eta_i d_i(x). \tag{25}
$$

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

$$g(\lambda, \eta) = \inf_{x} L(x, \lambda, \eta).$$  \hspace{1cm} (26)

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:

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

subject to

$$\lambda \geq 0.$$ 

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.
If the primal problem is convex, usually maximizing the dual problem is the same as minimizing the primal problem. We call this \textit{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 $x^*$:

$$\nabla f(x^*) + \sum_{i}^{M} \lambda^*_i \nabla c_i(x^*) + \sum_{i}^{N} \eta^*_i \nabla d_i(x^*) = 0.$$ \hfill (29)
Optimality Conditions

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

\[ c_i(x^*) \leq 0 \]  
(30)

\[ d_i(x^*) = 0 \]  
(31)

\[ \lambda_i^* \geq 0 \]  
(32)

\[ \lambda_i^* c_i(x^*) = 0 \]  
(33)

\[ \nabla f(x^*) + \sum_{i}^{M} \lambda_i^* \nabla c_i(x^*) + \sum_{i}^{N} \eta_i^* \nabla d_i(x^*) = 0. \]  
(34)

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

\begin{align}
\text{Find} \quad & x^* = \operatorname*{arg\,min}_{x \in \mathbb{R}^D} f(x) + \sum_{i=1}^{M} I_\infty(c_i(x)) \tag{35} \\
\text{subject to} \quad & Ax = b. \tag{36}
\end{align}

We approximate the indicator function by

\[ \hat{I}(u) = -\frac{1}{t} \log(-u) \tag{37} \]

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

\[ \phi(x) = - \sum_{i}^{M} \log(-c_i(x)) \]  

(38)

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

The \textit{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.
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 $x$. 
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.
The SMO Algorithm

An active set method for support vector machines.
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. The approach of *active set* methods, or *working set* methods *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.
SVM Quadratic Program

The SVM optimization problem in dual:

$$\max_{\alpha \in \mathbb{R}^n} W(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j k(x_i, x_j)$$

subject to

$$0 \leq \alpha_i \leq \frac{C}{m} \quad \forall i = 1, 2 \ldots m$$

$$\sum_{i=1}^{m} \alpha_i y_i = 0$$

Optimization over a vector of size $m$. # of constraints=$m$. Can be expensive if $m$ is large.
Most $\alpha_i$ will be at their extreme values i.e. either at $\frac{C}{m}, 0$. So, decompose to subproblem over variables not at their extremes.

This is the intuition behind working set algorithms. We shall discuss SMO which is an extreme form of working set algorithm.
Barebones of SMO

Roughly the SMO works as follows:

1. Heuristically picks 2 lagrangians to update, say $\alpha_i, \alpha_j$ and freezes the other variables
2. Analytically update $\alpha_i, \alpha_j$
3. Iterate until KKT gap=0

Two issues still remain:

1. How do we select $\alpha_i, \alpha_j$?
2. How do we update $\alpha_i, \alpha_j$?
Heuristically picking $\alpha_i, \alpha_j$

Every lagrangian makes a contribution to the KKT gap. Intuitively we would like to pick lagrangian which makes the largest contribution to the KKT gap. To pick $\alpha_i$:

1. Loop over lagrangians which are neither at the lower or upper boundary.

2. Once all these are satisfied we loop over all patterns violating the KKT, to ensure self consistency over complete datasets

To pick $\alpha_j$: $\alpha_j = \arg \max_j |(f(x_i) - y_i) - (f(x_j) - y_j)|$. This ensures maximum change in the lagrangians $\alpha_i, \alpha_j$. 
Second Choice Heuristics

What if the heuristic for $\alpha_j$ fails (first choice)? Use second choice heuristic.

1. All indices $j$ corresponding to non-bound examples are looked at, searching for an example to make progress on

2. In case the first heuristic was unsuccessful, all other examples are analyzed until an example is found where progress can be made

3. If both examples fail SMO proceeds to find another $i$
Updating $\alpha_i, \alpha_j$

1. Basic Idea: Solve the SVM problem with $m-2$ components of $\alpha$ fixed.

2. 2-variable optimization, hence eliminate one variable and solve for the other
2-variable Optimization

For classification we need to solve the following 2-variable optimization problem:

$$\min_{\alpha_i,\alpha_j} \frac{1}{2} (\alpha_i^2 K_{ii} + \alpha_j^2 K_{jj} + 2\alpha_i \alpha_j K_{ij}) + c_i \alpha_i + c_j \alpha_j$$  \hspace{1cm} (42)$$

subject to

$$s\alpha_i + \alpha_j = \gamma$$
$$0 \leq \alpha_i, \alpha_j \leq C$$

where

$$c_i = y_i (f(x_i) - b_i - y_i) - \alpha_i K_{ii} - \alpha_j s K_{ij}$$
$$s = y_i y_j$$
Main Things You Should Know

- The different types of constrained optimization problems
- The idea of the interior point method
- The idea of the SMO method
1. (T/F) A quadratic program is a type of optimization algorithm.
2. (T/F) A quadratic program can have linear constraints.
3. (T/F) SMO uses second-order (Hessian) information.