Convex Optimization Overview

Zico Kolter (updated by Honglak Lee)

October 17, 2008

1 Introduction

Many situations arise in machine learning where we would like to ***optimize*** the value of some function. That is, given a function $f : \mathbb{R}^n \to \mathbb{R}$, we want to find $x \in \mathbb{R}^n$ that minimizes (or maximizes) $f(x)$. We have already seen several examples of optimization problems in class: least-squares, logistic regression, and support vector machines can all be framed as optimization problems.

It turns out that, in the general case, finding the global optimum of a function can be a very difficult task. However, for a special class of optimization problems known as ***convex optimization problems***, we can efficiently find the global solution in many cases. Here, "efficiently" has both practical and theoretical connotations: it means that we can solve many real-world problems in a reasonable amount of time, and it means that theoretically we can solve problems in time that depends only *polynomially* on the problem size.

The goal of these section notes and the accompanying lecture is to give a very brief overview of the field of convex optimization. Much of the material here (including some of the figures) is heavily based on the book *Convex Optimization* [1] by Stephen Boyd and Lieven Vandenberghe (available for free online), and EE364, a class taught here at Stanford by Stephen Boyd. If you are interested in pursuing convex optimization further, these are both excellent resources.

2 Convex Sets

We begin our look at convex optimization with the notion of a ***convex set***.

Definition 2.1 *A set C is convex if, for any $x, y \in C$ and $\theta \in \mathbb{R}$ with $0 \leq \theta \leq 1$,*

$$\theta x + (1 - \theta)y \in C.$$

Intuitively, this means that if we take any two elements in C, and draw a line segment between these two elements, then every point on that line segment also belongs to C. Figure 1 shows an example of one convex and one non-convex set. The point $\theta x + (1 - \theta)y$ is called a ***convex combination*** of the points x and y.

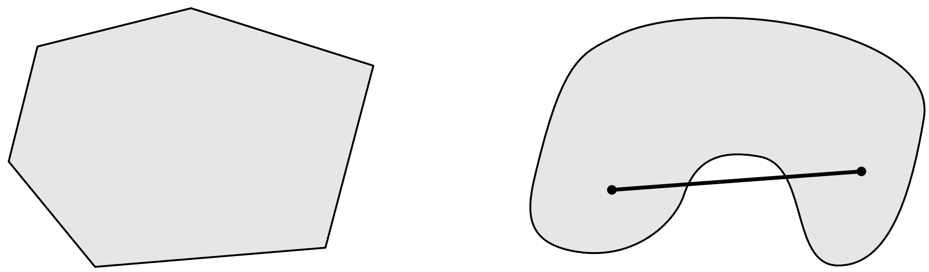

Figure 1: Examples of a convex set (a) and a non-convex set (b).

2.1 Examples

- **All of $\mathbb{R}^n$.** It should be fairly obvious that given any $x, y \in \mathbb{R}^n$, $\theta x + (1-\theta)y \in \mathbb{R}^n$.

- **The non-negative orthant, $\mathbb{R}^n_+$.** The non-negative orthant consists of all vectors in $\mathbb{R}^n$ whose elements are all non-negative: $\mathbb{R}^n_+ = \{x : x_i \geq 0 \;\; \forall i = 1, \ldots, n\}$. To show that this is a convex set, simply note that given any $x, y \in \mathbb{R}^n_+$ and $0 \leq \theta \leq 1$,

$$(\theta x + (1-\theta)y)_i = \theta x_i + (1-\theta)y_i \geq 0 \;\; \forall i.$$

- **Norm balls.** Let $\|\cdot\|$ be some norm on $\mathbb{R}^n$ (e.g., the Euclidean norm, $\|x\|_2 = \sqrt{\sum_{i=1}^n x_i^2}$). Then the set $\{x : \|x\| \leq 1\}$ is a convex set. To see this, suppose $x, y \in \mathbb{R}^n$, with $\|x\| \leq 1, \|y\| \leq 1$, and $0 \leq \theta \leq 1$. Then

$$\|\theta x + (1-\theta)y\| \leq \|\theta x\| + \|(1-\theta)y\| = \theta\|x\| + (1-\theta)\|y\| \leq 1$$

where we used the triangle inequality and the positive homogeneity of norms.

- **Affine subspaces and polyhedra.** Given a matrix $A \in \mathbb{R}^{m \times n}$ and a vector $b \in \mathbb{R}^m$, an affine subspace is the set $\{x \in \mathbb{R}^n : Ax = b\}$ (note that this could possibly be empty if b is not in the range of A). Similarly, a polyhedron is the (again, possibly empty) set $\{x \in \mathbb{R}^n : Ax \preceq b\}$, where '$\preceq$' here denotes componentwise inequality (i.e., all the entries of Ax are less than or equal to their corresponding element in b).[1] To prove this, first consider $x, y \in \mathbb{R}^n$ such that $Ax = Ay = b$. Then for $0 \leq \theta \leq 1$,

$$A(\theta x + (1-\theta)y) = \theta Ax + (1-\theta)Ay = \theta b + (1-\theta)b = b.$$

Similarly, for $x, y \in \mathbb{R}^n$ that satisfy $Ax \leq b$ and $Ay \leq b$ and $0 \leq \theta \leq 1$,

$$A(\theta x + (1-\theta)y) = \theta Ax + (1-\theta)Ay \leq \theta b + (1-\theta)b = b.$$

[1] Similarly, for two vectors $x, y \in \mathbb{R}^n$, $x \succeq y$ denotes that each element of x is greater than or equal to the corresponding element in y. Note that sometimes '$\leq$' and '$\geq$' are used in place of '$\preceq$' and '$\succeq$'; the meaning must be determined contextually (i.e., both sides of the inequality will be vectors).

- **Intersections of convex sets.** Suppose $C_1, C_2, \ldots, C_k$ are convex sets. Then their intersection

  $$\bigcap_{i=1}^{k} C_i = \{x : x \in C_i \ \ \forall i = 1, \ldots, k\}$$

  is also a convex set. To see this, consider $x, y \in \bigcap_{i=1}^{k} C_i$ and $0 \leq \theta \leq 1$. Then,

  $$\theta x + (1-\theta)y \in C_i \ \ \forall i = 1, \ldots, k$$

  by the definition of a convex set. Therefore

  $$\theta x + (1-\theta)y \in \bigcap_{i=1}^{k} C_i.$$

  Note, however, that the *union* of convex sets in general will not be convex.

- **Positive semidefinite matrices.** The set of all symmetric positive semidefinite matrices, often times called the *positive semidefinite cone* and denoted $\mathbb{S}_{+}^{n}$, is a convex set (in general, $\mathbb{S}^n \subset \mathbb{R}^{n\times n}$ denotes the set of symmetric $n \times n$ matrices). Recall that a matrix $A \in \mathbb{R}^{n\times n}$ is symmetric positive semidefinite if and only if $A = A^T$ and for all $x \in \mathbb{R}^n$, $x^T A x \geq 0$. Now consider two symmetric positive semidefinite matrices $A, B \in \mathbb{S}_{+}^{n}$ and $0 \leq \theta \leq 1$. Then for any $x \in \mathbb{R}^n$,

  $$x^T(\theta A + (1-\theta)B)x = \theta x^T A x + (1-\theta)x^T B x \geq 0.$$

  The same logic can be used to show that the sets of all positive definite, negative definite, and negative semidefinite matrices are each also convex.

3 Convex Functions

A central element in convex optimization is the notion of a ***convex function***.

Definition 3.1 *A function $f : \mathbb{R}^n \to \mathbb{R}$ is convex if its domain (denoted $\mathcal{D}(f)$) is a convex set, and if, for all $x, y \in \mathcal{D}(f)$ and $\theta \in \mathbb{R}$, $0 \leq \theta \leq 1$,*

$$f(\theta x + (1-\theta)y) \leq \theta f(x) + (1-\theta)f(y).$$

Intuitively, the way to think about this definition is that if we pick any two points on the graph of a convex function and draw a straight line between then, then the portion of the function between these two points will lie below this straight line. This situation is pictured in Figure 2.[2]

We say a function is ***strictly convex*** if Definition 3.1 holds with strict inequality for $x \neq y$ and $0 < \theta < 1$. We say that f is ***concave*** if $-f$ is convex, and likewise that f is ***strictly concave*** if $-f$ is strictly convex.

[2] Don't worry too much about the requirement that the domain of f be a convex set. This is just a technicality to ensure that $f(\theta x + (1-\theta)y)$ is actually defined (if $\mathcal{D}(f)$ were not convex, then it could be that $f(\theta x + (1-\theta)y)$ is undefined even though $x, y \in \mathcal{D}(f)$).

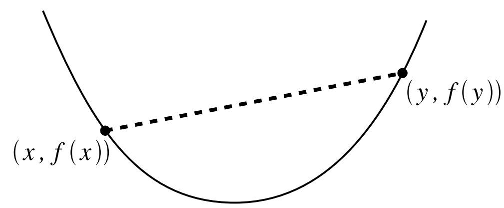

Figure 2: Graph of a convex function. By the definition of convex functions, the line connecting two points on the graph must lie above the function.

3.1 First Order Condition for Convexity

Suppose a function $f : \mathbb{R}^n \to \mathbb{R}$ is differentiable (i.e., the gradient[3] $\nabla_x f(x)$ exists at all points x in the domain of f). Then f is convex if and only if $\mathcal{D}(f)$ is a convex set and for all $x, y \in \mathcal{D}(f)$,

$$f(y) \geq f(x) + \nabla_x f(x)^T (y - x).$$

The function $f(x) + \nabla_x f(x)^T (y - x)$ is called the ***first-order approximation*** to the function f at the point x. Intuitively, this can be thought of as approximating f with its tangent line at the point x. The first order condition for convexity says that f is convex if and only if the tangent line is a global underestimator of the function f. In other words, if we take our function and draw a tangent line at any point, then every point on this line will lie below the corresponding point on f.

Similar to the definition of convexity, f will be strictly convex if this holds with strict inequality, concave if the inequality is reversed, and strictly concave if the reverse inequality is strict.

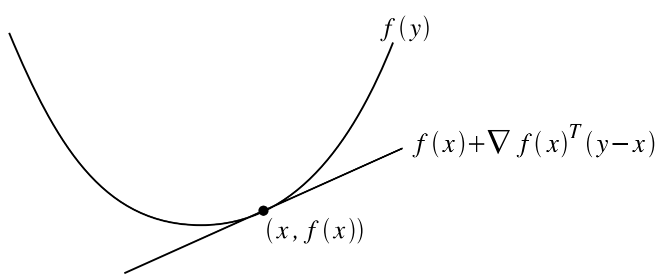

Figure 3: Illustration of the first-order condition for convexity.

[3]Recall that the gradient is defined as $\nabla_x f(x) \in \mathbb{R}^n, (\nabla_x f(x))_i = \frac{\partial f(x)}{\partial x_i}$. For a review on gradients and Hessians, see the previous section notes on linear algebra.

3.2 Second Order Condition for Convexity

Suppose a function $f : \mathbb{R}^n \to \mathbb{R}$ is twice differentiable (i.e., the Hessian[4] $\nabla_x^2 f(x)$ is defined for all points x in the domain of f). Then f is convex if and only if $\mathcal{D}(f)$ is a convex set and its Hessian is positive semidefinite: i.e., for any $x \in \mathcal{D}(f)$,

$$\nabla_x^2 f(x) \succeq 0.$$

Here, the notation '$\succeq$' when used in conjunction with matrices refers to positive semidefiniteness, rather than componentwise inequality. [5] In one dimension, this is equivalent to the condition that the second derivative $f''(x)$ always be non-negative (i.e., the function always has positive non-negative).

Again analogous to both the definition and the first order conditions for convexity, f is strictly convex if its Hessian is positive definite, concave if the Hessian is negative semidefinite, and strictly concave if the Hessian is negative definite.

3.3 Jensen's Inequality

Suppose we start with the inequality in the basic definition of a convex function

$$f(\theta x + (1-\theta)y) \leq \theta f(x) + (1-\theta) f(y) \quad \text{for} \quad 0 \leq \theta \leq 1.$$

Using induction, this can be fairly easily extended to convex combinations of more than one point,

$$f\left(\sum_{i=1}^{k} \theta_i x_i\right) \leq \sum_{i=1}^{k} \theta_i f(x_i) \quad \text{for} \quad \sum_{i=1}^{k} \theta_i = 1,\ \theta_i \geq 0 \ \ \forall i.$$

In fact, this can also be extended to infinite sums or integrals. In the latter case, the inequality can be written as

$$f\left(\int p(x) x dx\right) \leq \int p(x) f(x) dx \quad \text{for} \quad \int p(x) dx = 1,\ p(x) \geq 0 \ \ \forall x.$$

Because $p(x)$ integrates to 1, it is common to consider it as a probability density, in which case the previous equation can be written in terms of expectations,

$$f(\mathbf{E}[x]) \leq \mathbf{E}[f(x)].$$

This last inequality is known as *Jensen's inequality*, and it will come up later in class.[6]

[4] Recall the Hessian is defined as $\nabla_x^2 f(x) \in \mathbb{R}^{n\times n}$, $(\nabla_x^2 f(x))_{ij} = \frac{\partial^2 f(x)}{\partial x_i \partial x_j}$

[5] Similarly, for a symmetric matrix $X \in \mathbb{S}^n$, $X \preceq 0$ denotes that X is negative semidefinite. As with vector inequalities, '$\leq$' and '$\geq$' are sometimes used in place of '$\preceq$' and '$\succeq$'. Despite their notational similarity to vector inequalities, these concepts are very different; in particular, $X \succeq 0$ does not imply that $X_{ij} \geq 0$ for all i and j.

[6] In fact, all four of these equations are sometimes referred to as Jensen's inequality, due to the fact that they are all equivalent. However, for this class we will use the term to refer specifically to the last inequality presented here.

3.4 Sublevel Sets

Convex functions give rise to a particularly important type of convex set called an ***α-sublevel set***. Given a convex function $f : \mathbb{R}^n \to R$ and a real number $\alpha \in \mathbb{R}$, the α-sublevel set is defined as

$$\{x \in \mathcal{D}(f) : f(x) \leq \alpha\}.$$

In other words, the α-sublevel set is the set of all points x such that $f(x) \leq \alpha$.

To show that this is a convex set, consider any $x, y \in \mathcal{D}(f)$ such that $f(x) \leq \alpha$ and $f(y) \leq \alpha$. Then

$$f(\theta x + (1-\theta)y) \leq \theta f(x) + (1-\theta)f(y) \leq \theta\alpha + (1-\theta)\alpha = \alpha.$$

3.5 Examples

We begin with a few simple examples of convex functions of one variable, then move on to multivariate functions.

- **Exponential.** Let $f : \mathbb{R} \to \mathbb{R}$, $f(x) = e^{ax}$ for any $a \in \mathbb{R}$. To show f is convex, we can simply take the second derivative $f''(x) = a^2 e^{ax}$, which is positive for all x.

- **Negative logarithm.** Let $f : \mathbb{R} \to \mathbb{R}$, $f(x) = -\log x$ with domain $\mathcal{D}(f) = \mathbb{R}_{++}$ (here, $\mathbb{R}_{++}$ denotes the set of strictly positive real numbers, $\{x : x > 0\}$). Then $f''(x) = 1/x^2 > 0$ for all x.

- **Affine functions.** Let $f : \mathbb{R}^n \to \mathbb{R}$, $f(x) = b^T x + c$ for some $b \in \mathbb{R}^n$, $c \in \mathbb{R}$. In this case the Hessian, $\nabla_x^2 f(x) = 0$ for all x. Because the zero matrix is both positive semidefinite and negative semidefinite, f is both convex and concave. In fact, affine functions of this form are the *only* functions that are both convex and concave.

- **Quadratic functions.** Let $f : \mathbb{R}^n \to \mathbb{R}$, $f(x) = \frac{1}{2}x^T A x + b^T x + c$ for a symmetric matrix $A \in \mathbb{S}^n$, $b \in \mathbb{R}^n$ and $c \in \mathbb{R}$. In our previous section notes on linear algebra, we showed the Hessian for this function is given by

  $$\nabla_x^2 f(x) = A.$$

  Therefore, the convexity or non-convexity of f is determined entirely by whether or not A is positive semidefinite: if A is positive semidefinite then the function is convex (and analogously for strictly convex, concave, strictly concave); if A is indefinite then f is neither convex nor concave.

  Note that the squared Euclidean norm $f(x) = \|x\|_2^2 = x^T x$ is a special case of quadratic functions where $A = I$, $b = 0$, $c = 0$, so it is therefore a strictly convex function.

- **Norms.** Let $f : \mathbb{R}^n \to \mathbb{R}$ be some norm on $\mathbb{R}^n$. Then by the triangle inequality and positive homogeneity of norms, for $x, y \in \mathbb{R}^n$, $0 \leq \theta \leq 1$,

$$f(\theta x + (1-\theta)y) \leq f(\theta x) + f((1-\theta)y) = \theta f(x) + (1-\theta)f(y).$$

This is an example of a convex function where it is *not* possible to prove convexity based on the second-order or first-order conditions because norms are not generally differentiable everywhere (e.g., the 1-norm, $||x||_1 = \sum_{i=1}^n |x_i|$, is non-differentiable at all points where any x_i is equal to zero).

- **Nonnegative weighted sums of convex functions.** Let $f_1, f_2, \ldots, f_k$ be convex functions and $w_1, w_2, \ldots, w_k$ be nonnegative real numbers. Then

$$f(x) = \sum_{i=1}^k w_i f_i(x)$$

is a convex function, since

$$\begin{aligned} f(\theta x + (1-\theta)y) &= \sum_{i=1}^k w_i f_i(\theta x + (1-\theta)y) \\ &\leq \sum_{i=1}^k w_i(\theta f_i(x) + (1-\theta)f_i(y)) \\ &= \theta \sum_{i=1}^k w_i f_i(x) + (1-\theta)\sum_{i=1}^k w_i f_i(y) \\ &= \theta f(x) + (1-\theta)f(x). \end{aligned}$$

4 Convex Optimization Problems

Armed with the definitions of convex functions and sets, we are now equipped to consider ***convex optimization problems***. Formally, a convex optimization problem in an optimization problem of the form

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & x \in C \end{array}$$

where f is a convex function, C is a convex set, and x is the optimization variable. However, since this can be a little bit vague, we often write it as

$$\begin{array}{lll} \text{minimize} & f(x) & \\ \text{subject to} & g_i(x) \leq 0, & i = 1, \ldots, m \\ & h_i(x) = 0, & i = 1, \ldots, p \end{array}$$

where f is a convex function, g_i are convex functions, and h_i are affine functions, and x is the optimization variable.

Is it imporant to note the direction of these inequalities: a convex function g_i must be *less* than zero. This is because the 0-sublevel set of g_i is a convex set, so the feasible region, which is the intersection of many convex sets, is also convex (recall that affine subspaces are convex sets as well). If we were to require that $g_i \geq 0$ for some convex g_i, the feasible region would no longer be a convex set, and the algorithms we apply for solving these problems would no longer be guaranteed to find the global optimum. Also notice that only affine functions are allowed to be equality constraints. Intuitively, you can think of this as being due to the fact that an equality constraint is equivalent to the two inequalities $h_i \leq 0$ and $h_i \geq 0$. However, these will both be valid constraints if and only if h_i is both convex and concave, i.e., h_i must be affine.

The ***optimal value*** of an optimization problem is denoted $p^\star$ (or sometimes $f^\star$) and is equal to the minimum possible value of the objective function in the feasible region[7]

$$p^\star = \min\{f(x) : g_i(x) \leq 0, i = 1, \ldots, m, h_i(x) = 0, i = 1, \ldots, p\}.$$

We allow $p^\star$ to take on the values $+\infty$ and $-\infty$ when the problem is either *infeasible* (the feasible region is empty) or *unbounded below* (there exists feasible points such that $f(x) \to -\infty$), respectively. We say that $x^\star$ is an ***optimal point*** if $f(x^\star) = p^\star$. Note that there can be more than one optimal point, even when the optimal value is finite.

4.1 Global Optimality in Convex Problems

Before stating the result of global optimality in convex problems, let us formally define the concepts of local optima and global optima. Intuitively, a feasible point is called ***locally optimal*** if there are no "nearby" feasible points that have a lower objective value. Similarly, a feasible point is called ***globally optimal*** if there are no feasible points at all that have a lower objective value. To formalize this a little bit more, we give the following two definitions.

Definition 4.1 *A point x is locally optimal if it is feasible (i.e., it satisfies the constraints of the optimization problem) and if there exists some $R > 0$ such that all feasible points z with $\|x - z\|_2 \leq R$, satisfy $f(x) \leq f(z)$.*

Definition 4.2 *A point x is globally optimal if it is feasible and for all feasible points z, $f(x) \leq f(z)$.*

We now come to the crucial element of convex optimization problems, from which they derive most of their utility. The key idea is that ***for a convex optimization problem all locally optimal points are globally optimal***.

Let's give a quick proof of this property by contradiction. Suppose that x is a locally optimal point which is not globally optimal, i.e., there exists a feasible point y such that

[7] Math majors might note that the min appearing below should more correctly be an inf. We won't worry about such technicalities here, and use min for simplicity.

$f(x) > f(y)$. By the definition of local optimality, there exist no feasible points z such that $\|x - z\|_2 \le R$ and $f(z) < f(x)$. But now suppose we choose the point

$$z = \theta y + (1-\theta)x \quad \text{with} \quad \theta = \frac{R}{2\|x-y\|_2}.$$

Then

$$\begin{aligned}
\|x - z\|_2 &= \left\| x - \left(\frac{R}{2\|x-y\|_2} y + \left(1 - \frac{R}{2\|x-y\|_2}\right) x \right) \right\|_2 \\
&= \left\| \frac{R}{2\|x-y\|_2} (x - y) \right\|_2 \\
&= R/2 \le R.
\end{aligned}$$

In addition, by the convexity of f we have

$$f(z) = f(\theta y + (1-\theta)x) \le \theta f(y) + (1-\theta) f(x) < f(x).$$

Furthermore, since the feasible set is a convex set, and since x and y are both feasible $z = \theta y + (1-\theta)$ will be feasible as well. Therefore, z is a feasible point, with $\|x - z\|_2 < R$ and $f(z) < f(x)$. This contradicts our assumption, showing that x cannot be locally optimal.

4.2 Special Cases of Convex Problems

For a variety of reasons, it is oftentimes convenient to consider special cases of the general convex programming formulation. For these special cases we can often devise extremely efficient algorithms that can solve very large problems, and because of this you will probably see these special cases referred to any time people use convex optimization techniques.

- **Linear Programming.** We say that a convex optimization problem is a ***linear program*** (LP) if both the objective function f and inequality constraints g_i are affine functions. In other words, these problems have the form

$$\begin{array}{rl}
\text{minimize} & c^T x + d \\
\text{subject to} & Gx \preceq h \\
& Ax = b
\end{array}$$

  where $x \in \mathbb{R}^n$ is the optimization variable, $c \in \mathbb{R}^n$, $d \in \mathbb{R}$, $G \in \mathbb{R}^{m \times n}$, $h \in \mathbb{R}^m$, $A \in \mathbb{R}^{p \times n}$, $b \in \mathbb{R}^p$ are defined by the problem, and '$\preceq$' denotes elementwise inequality.

- **Quadratic Programming.** We say that a convex optimization problem is a ***quadratic program*** (QP) if the inequality constraints g_i are still all affine, but if the objective function f is a convex quadratic function. In other words, these problems have the form,

$$\begin{array}{rl}
\text{minimize} & \frac{1}{2} x^T P x + c^T x + d \\
\text{subject to} & Gx \preceq h \\
& Ax = b
\end{array}$$

where again $x \in \mathbb{R}^n$ is the optimization variable, $c \in \mathbb{R}^n$, $d \in \mathbb{R}$, $G \in \mathbb{R}^{m \times n}$, $h \in \mathbb{R}^m$, $A \in \mathbb{R}^{p \times n}$, $b \in \mathbb{R}^p$ are defined by the problem, but we also have $P \in \mathbb{S}^n_+$, a symmetric positive semidefinite matrix.

- **Quadratically Constrained Quadratic Programming.** We say that a convex optimization problem is a ***quadratically constrained quadratic program*** (QCQP) if both the objective f and the inequality constraints g_i are convex quadratic functions,

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}x^TPx + c^Tx + d \\ \text{subject to} & \frac{1}{2}x^TQ_ix + r_i^Tx + s_i \leq 0, \quad i = 1, \ldots, m \\ & Ax = b \end{array}$$

  where, as before, $x \in \mathbb{R}^n$ is the optimization variable, $c \in \mathbb{R}^n$, $d \in \mathbb{R}$, $A \in \mathbb{R}^{p \times n}$, $b \in \mathbb{R}^p$, $P \in \mathbb{S}^n_+$, but we also have $Q_i \in \mathbb{S}^n_+$, $r_i \in \mathbb{R}^n$, $s_i \in \mathbb{R}$, for $i = 1, \ldots, m$.

- **Semidefinite Programming.** This last example is more complex than the previous ones, so don't worry if it doesn't make much sense at first. However, semidefinite programming is becoming more prevalent in many areas of machine learning research, so you might encounter these at some point, and it is good to have an idea of what they are. We say that a convex optimization problem is a ***semidefinite program*** (SDP) if it is of the form

$$\begin{array}{ll} \text{minimize} & \text{tr}(CX) \\ \text{subject to} & \text{tr}(A_iX) = b_i, \quad i = 1, \ldots, p \\ & X \succeq 0 \end{array}$$

  where the symmetric matrix $X \in \mathbb{S}^n$ is the optimization variable, the symmetric matrices $C, A_1, \ldots, A_p \in \mathbb{S}^n$ are defined by the problem, and the constraint $X \succeq 0$ means that we are constraining X to be positive semidefinite. This looks a bit different than the problems we have seen previously, since the optimization variable is now a matrix instead of a vector. If you are curious as to why such a formulation might be useful, you should look into a more advanced course or book on convex optimization.

It should be obvious from the definitions that quadratic programs are more general than linear programs (since a linear program is just a special case of a quadratic program where $P = 0$), and likewise that quadratically constrained quadratic programs are more general than quadratic programs. However, what is not obvious is that semidefinite programs are in fact more general than all the previous types, that is, any quadratically constrained quadratic program (and hence any quadratic program or linear program) can be expressed as a semidefinte program. We won't discuss this relationship further in this document, but this might give you just a small idea as to why semidefinite programming could be useful.

4.3 Examples

Now that we've covered plenty of the boring math and formalisms behind convex optimization, we can finally get to the fun part: using these techniques to solve actual problems.

We've already encountered a few such optimization problems in class, and in nearly every field, there is a good chance that someone has applied convex optimization to solve some problem.

- **Support Vector Machines (SVM).** One of the most prevalent applications of convex optimization methods in machine learning is the support vector machine classifier. As discussed in class, finding the support vector classifier (in the case with slack variables) can be formulated as the optimization problem

$$\begin{array}{rll} \text{minimize} & \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^m \xi_i & \\ \text{subject to} & y^{(i)}(w^T x^{(i)} + b) \geq 1 - \xi_i, & i = 1, \ldots, m \\ & \xi_i \geq 0, & i = 1, \ldots, m \end{array}$$

  with optimization variables $w \in \mathbb{R}^n$, $\xi \in \mathbb{R}^m$, $b \in \mathbb{R}$, and where $C \in \mathbb{R}$ and $x^{(i)}, y^{(i)}, i = 1, \ldots m$ are defined by the problem. This is an example of a quadratic program, which we shall show by putting the problem into the form described in the previous section. In particular, if we define $k = m + n + 1$, let the optimization variable be

$$x \in \mathbb{R}^k \equiv \begin{bmatrix} w \\ \xi \\ b \end{bmatrix}$$

  and define the matrices

$$P \in \mathbb{R}^{k\times k} = \begin{bmatrix} I & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad c \in \mathbb{R}^k = \begin{bmatrix} 0 \\ C \cdot \mathbf{1} \\ 0 \end{bmatrix},$$

$$G \in \mathbb{R}^{2m\times k} = \begin{bmatrix} -\text{diag}(y)X & -I & -y \\ 0 & -I & 0 \end{bmatrix}, \quad h \in \mathbb{R}^{2m} = \begin{bmatrix} -\mathbf{1} \\ 0 \end{bmatrix}$$

  where I is the identity, $\mathbf{1}$ is the vector of all ones, and X and y are defined as in class,

$$X \in \mathbb{R}^{m\times n} = \begin{bmatrix} x^{(1)T} \\ x^{(2)T} \\ \vdots \\ x^{(m)T} \end{bmatrix}, \quad y \in \mathbb{R}^m = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(m)} \end{bmatrix}.$$

  You should convince yourself that the quadratic program described in the previous section, when using these matrices defined above, is equivalent to the SVM optimization problem. In reality, it is fairly easy to see that there the SVM optimization problem has a quadratic objective and linear constraints, so we typically don't need to put it into standard form to "prove" that it is a QP, and we would only do so if we are using an off-the-shelf solver that requires the input to be in standard form.

- **Constrained least squares.** In class we have also considered the least squares problem, where we want to minimize $\|Ax - b\|_2^2$ for some matrix $A \in \mathbb{R}^{m\times n}$ and $b \in \mathbb{R}^m$. As we saw, this particular problem can be solved analytically via the normal equations. However, suppose that we also want to constrain the entries in the solution x to lie within some predefined ranges. In other words, suppose we wanted to solve the optimization problem,

$$\begin{array}{rl} \text{minimize} & \frac{1}{2}\|Ax - b\|_2^2 \\ \text{subject to} & l \preceq x \preceq u \end{array}$$

  with optimization variable x and problem data $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$, $l \in \mathbb{R}^n$, and $u \in \mathbb{R}^n$. This might seem like a simple additional constraint, but it turns out that there will no longer be an analytical solution. However, you should convince yourself that this optimization problem is a quadratic program, with matrices defined by

$$P \in \mathbb{R}^{n\times n} = \frac{1}{2}A^TA, \quad c \in \mathbb{R}^n = -b^TA, \quad d \in \mathbb{R} = \frac{1}{2}b^Tb,$$

$$G \in \mathbb{R}^{2n\times 2n} = \left[\begin{array}{cc} -I & 0 \\ 0 & I \end{array}\right], \quad h \in \mathbb{R}^{2n} = \left[\begin{array}{c} -l \\ u \end{array}\right].$$

- **Maximum Likelihood for Logistic Regression.** For homework one, you were required to show that the log-likelihood of the data in a logistic model was concave. The log likehood under such a model is

$$\ell(\theta) = \sum_{i=1}^{n}\left\{y^{(i)}\ln g(\theta^Tx^{(i)}) + (1 - y^{(i)})\ln(1 - g(\theta^Tx^{(i)}))\right\}$$

  where $g(z)$ denotes the logistic function $g(z) = 1/(1 + e^{-z})$. Finding the maximum likelihood estimate is then a task of maximizing the log-likelihood (or equivalently, minimizing the negative log-likelihood, a convex function), i.e.,

$$\text{minimize} \quad -\ell(\theta)$$

  with optimization variable $\theta \in \mathbb{R}^n$ and no constraints.

  Unlike the previous two examples, it is not so easy to put this problem into a "standard" form optimization problem. Nevertheless, you have seen on the homework that the fact that ℓ is a concave function means that you can very efficiently find the global solution using an algorithm such as Newton's method.

4.4 Implementation: Linear SVM using CVX

Many convex optimization problems can be solved by several off-the-shelf software packages including CVX, Sedumi, CPLEX, MOSEK, etc. Thus, in many cases, once you identify the convex optimization problem, you can solve it without worrying about how to implement the algorithm yourself. This is particularly useful for a rapid prototyping.[8]

Among these software packages, we introduce CVX [2] as an example. CVX is a free MATLAB-based software package for solving generic convex optimzation problems; it can solve a wide variety of convex optimization problems such as LP, QP, QCQP, SDP, etc. As an illustration, we conclude this section by implementing a linear SVM classifier for the binary classification problem using the data given in the Problem Set #1. For more general setting using other non-linear kernels, the dual formulation can be solved using CVX as well.

```
% load data
load q1x.dat
load q1y.dat

% define variables
X = q1x;
y = 2*(q1y-0.5);

C = 1;
m = size(q1x,1);
n = size(q1x,2);

% train svm using cvx
cvx_begin
    variables w(n) b xi(m)
    minimize 1/2*sum(w.*w) + C*sum(xi)
    y.*(X*w + b) >= 1 - xi;
    xi >= 0;
cvx_end

% visualize
xp = linspace(min(X(:,1)), max(X(:,1)), 100);
yp = - (w(1)*xp + b)/w(2);
yp1 = - (w(1)*xp + b - 1)/w(2); % margin boundary for support vectors for y=1
yp0 = - (w(1)*xp + b + 1)/w(2); % margin boundary for support vectors for y=0

idx0 = find(q1y==0);
idx1 = find(q1y==1);

plot(q1x(idx0, 1), q1x(idx0, 2), 'rx'); hold on
```

[8]However, depending on the optimization problem, these off-the-shelf convex optimization solvers can be much slower compared to the best possible implementation; therefore, sometimes you may have to use more customized solvers or implement your own.

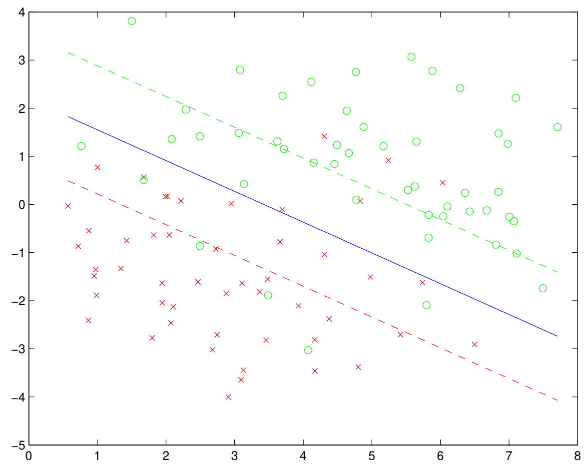

Figure 4: Decision boundary for a linear SVM classifier with $C = 1$.

```
plot(q1x(idx1, 1), q1x(idx1, 2), 'go');
plot(xp, yp, '-b', xp, yp1, '--g', xp, yp0, '--r');
hold off
title(sprintf('decision boundary for a linear SVM classifier with C=%g', C));
```

References

[1] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge UP, 2004. Online: `http://www.stanford.edu/~boyd/cvxbook/`

[2] M. Grant and S. Boyd. *CVX: Matlab software for disciplined convex programming* (web page and software). `http://stanford.edu/~boyd/cvx/`, September 2008.