

凸优化基础(Convex Optimization)

Zico Kolter (updated by Honglak Lee, translated by WiiGe)

October 17, 2008(updated)/January 1,2019(translation)

1 导引

许多情况下，我们做机器学习会遇到“如何**优化**某函数的值”这一问题。即有给定函数 $f: \mathbb{R}^n \rightarrow \mathbb{R}$ ，要找能使 $f(x)$ 取最小值/最大值的 $x \in \mathbb{R}^n$ 。机器学习领域已经知的几个优化问题的例子是：最小二乘，逻辑回归和支持向量机。

实践证明，在一般情况下找到函数的全局最优解可能是一项非常困难的任务。然而，对于被称为凸优化(convex optimization)问题的一类特殊优化问题，常常可以有效地找到其全局最优解。这里的“有效”具有实践和理论两重内涵：它意味着我们可以以合理的时间开销解决许多现实世界中的问题，而且意味着——理论上问题的求解是受问题规模影响的多项式时间(译者：亦即NP问题中的P问题)。

2 凸集(Convex Sets)

我们先从**凸集**的概念开始学习凸优化。

定义 2.1 对任意 $x, y \in C$ ， $\theta \in \mathbb{R}$ 且 $0 \leq \theta \leq 1$ ，有 $\theta x + (1 - \theta)y \in C$ ，则集合 C 为凸集。

这很符合直觉——它意味着如果我们在 C 中任取两个元素，并在这两个元素之间绘制一条线段，则该线段上的每个点都属于 C 。图 1 展示了一个关于凸集和非凸集的示例。点 $\theta x + (1 - \theta)y$ 被称为点 x 和 y 的**凸组合**。

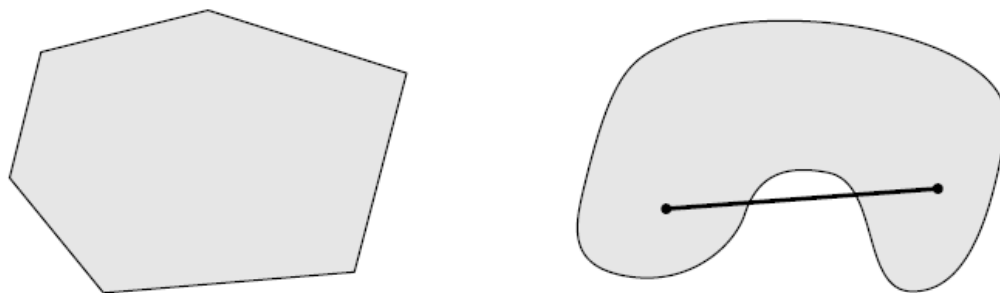


图 1: 凸集(左)和非凸集(右)的示例

2.1 一些凸集的例子

- 全体 $\mathbb{R}^n$ 。给定任意 $x, y \in \mathbb{R}^n$ ，则 $\theta x + (1 - \theta)y \in \mathbb{R}^n$ 是显然的。
- 非负象限 $\mathbb{R}_+^n$ 。非负象限由 $\mathbb{R}^n$ 中的全部非负向量组成，这些向量其元素均为非负： $\mathbb{R}_+^n = \{x: x_i \geq 0, \forall i = 1, \dots, n\}$ 。要说明它是一个凸集，只需简单记：给定任意 $x, y \in \mathbb{R}_+^n$ 和 $0 \leq \theta \leq 1$ ，均有 $(\theta x + (1 - \theta)y)_i = \theta x_i + (1 - \theta)y_i \geq 0, \forall i$ 。
- 范数曲面 (Norm balls)。令 $\|\cdot\|$ 为 $\mathbb{R}^n$ 上的范数(例如，欧氏范数 $\|x\|_2 = \sqrt{\sum_{i=1}^n x_i^2}$)，则集合 $\{x: \|x\| \leq 1\}$ 为一个凸集。要证明该结论，只需令 $x, y \in \mathbb{R}^n$ ，其中 $\|x\| \leq 1, \|y\| \leq 1$ 且 $0 \leq \theta \leq 1$ ，则有

$$\|\theta x + (1 - \theta)y\| \leq \|\theta x\| + \|(1 - \theta)y\| = \theta\|x\| + (1 - \theta)\|y\| \leq 1$$
 ，这里利用了三角不等式和范数的非负性。
- 仿射子空间 (Affine subspaces) 和多面体 (polyhedra)。对给定的矩阵 $A \in \mathbb{R}^{m \times n}$ 和向量 $b \in \mathbb{R}^m$ ，仿射子空间即为集合 $\{x \in \mathbb{R}^n: Ax = b\}$ (注意如果 b 不在 A 的值域中时， x 有可能为空集)。类似地，多面体即为(要注意也有可能为空)集合 $\{x \in \mathbb{R}^n: Ax \leq b\}$ ，这里的“ $\leq$ ”是分量不等式(componentwise inequality)的不等号(亦即， Ax 中任一向量都小于等于 b 中的对应元素) 注1。要证明这一点，先令 $x, y \in \mathbb{R}^n$ 使得 $Ax = Ay = b$ 。因此对于 $0 \leq \theta \leq 1$ ，有

$$A(\theta x + (1 - \theta)y) = \theta Ax + (1 - \theta)Ay = \theta b + (1 - \theta)b = b。$$
 类似地，对满足 $Ax \leq b, Ay \leq b$ 且 $0 \leq \theta \leq 1$ 的 $x, y \in \mathbb{R}^n$ ，亦有

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

注1：类似地，对任意两个向量 $x, y \in \mathbb{R}^n$ ， $x \geq y$ 表示 x 中的任一元素都大于等于 y 中的对应元素。有时候会用“ $\geq$ ”和“ $\leq$ ”代替“ $\geq$ ”和“ $\leq$ ”，所以其具体意义要根据上下文判断。(比如不等式两边都是向量的时候)

- 凸集的交集。记 $C_1, C_2, \dots, C_k$ 为凸集，则其交集

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

也为凸集。要证这一点，记 $x, y \in \bigcap_{i=1}^k C_i$ 且 $0 \leq \theta \leq 1$ ，依凸集定义易有 $\theta x + (1 - \theta)y \in C_i, \forall i = 1, \dots, k$ ，所以可得：

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

- 半正定矩阵 (Positive semidefinite matrices)。所有对称半正定矩阵的集合均为凸集，这种集合一般被称为半正定凸锥 (positive semidefinite cone) 并记为 $\mathbb{S}_+^n$ 。(一般来说， $\mathbb{S}^n \in \mathbb{R}^{n \times n}$ 表示 $n \times n$ 对称方阵的集合)。我们知道当且仅当 $x \in \mathbb{R}^n$ 且 $x^T A x \geq 0$ 时， $A \in \mathbb{R}^{n \times n}$ 为对称半正定矩阵。现有两个对称半正定矩阵 $A, B \in \mathbb{S}_+^n$ 且 $0 \leq \theta \leq 1$ ，则对于任意 $x \in \mathbb{R}^n$ ，有

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

这一证明思路不仅可用于正半定矩阵，也可以用于证明所有的正定，负定和负

半定矩阵均为凸的。

3 凸函数(Convex Functions)

凸优化中的核心概念就是是凸函数。

定义 3.1 对函数 $f: \mathbb{R}^n \rightarrow \mathbb{R}$ ，其定义域(记为 $D(f)$)为凸集，其中任意 $x, y \in D(f)$ ， $\theta \in \mathbb{R}$ ， $0 \leq \theta \leq 1$ ，且有

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

，则函数 f 为凸函数。

直观来看，这个定义意味着如果我们在凸函数的图形上选取任意两个点绘制一条直线，那么两点之间部分的函数图形将位于直线下方，如图 2 所示^{注2}。

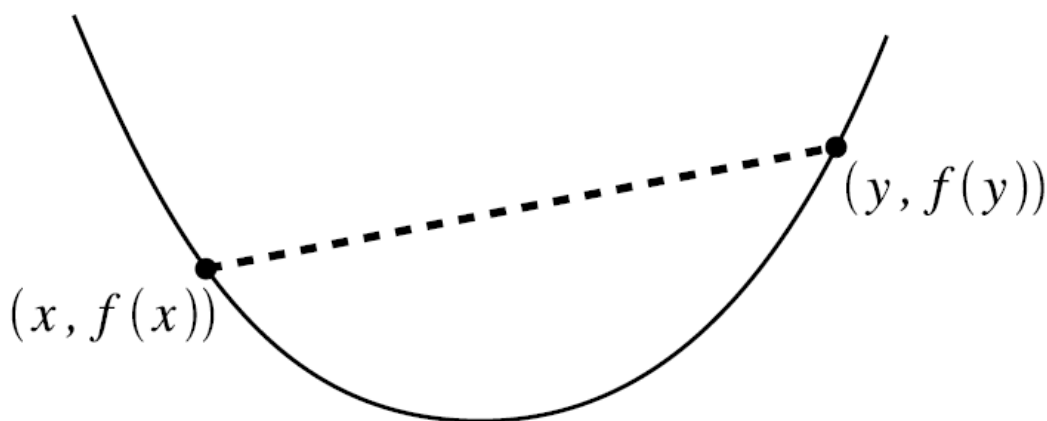


图 2:凸函数的图像。由凸函数的定义可知，连接图像上两点的线段必大于函数值

当定义3.1中的函数满足更严格的条件： $x \neq y$ 且 $0 < \theta < 1$ 时，我们说函数是严格凸的。当 $-f$ 为凸函数时则原函数 f 为凹(concave)函数，类似可知当 $-f$ 严格凸时则 f 为严格凹(strictly concave)函数。

注2：不必过分关注定义域为凸这一条件。这只是为了确保 $f(\theta x + (1 - \theta)y)$ 有定义的一个技巧(如果 $D(f)$ 非凸，那么即使 $x, y \in D(f)$ 也有可能导致 $f(\theta x + (1 - \theta)y)$ 无定义)。

3.1 一阶条件下的凹凸性(Convexity)

设函数 $f: \mathbb{R}^n \rightarrow \mathbb{R}$ 可微（如函数梯度^{注3} $\nabla_x f(x)$ 在 f 定义域内的 x 上处处存在），当且仅当 $D(f)$ 为凸集且对任意 $x, y \in D(f)$ 有

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

时， f 为凸函数。

函数 $f(x) + \nabla_x f(x)^T (y - x)$ 被称为 f 在点 x 的一阶近似。可以直观地认为它是

用切线在点 x 处对 f 的逼近。一阶条件下的凹凸性指出，当且仅当切线为函数的全局下估计(global underestimator)时 f 为凸函数。换句话说，如果我们在函数 f 的任一点作切线，切线上的每一点都位于对应函数 f 值的下方。

同凹凸性定义类似，当 f 满足严格不等式的时候 f 严格凸；若不等式反向则凹，反向后为严格不等式则严格凹。

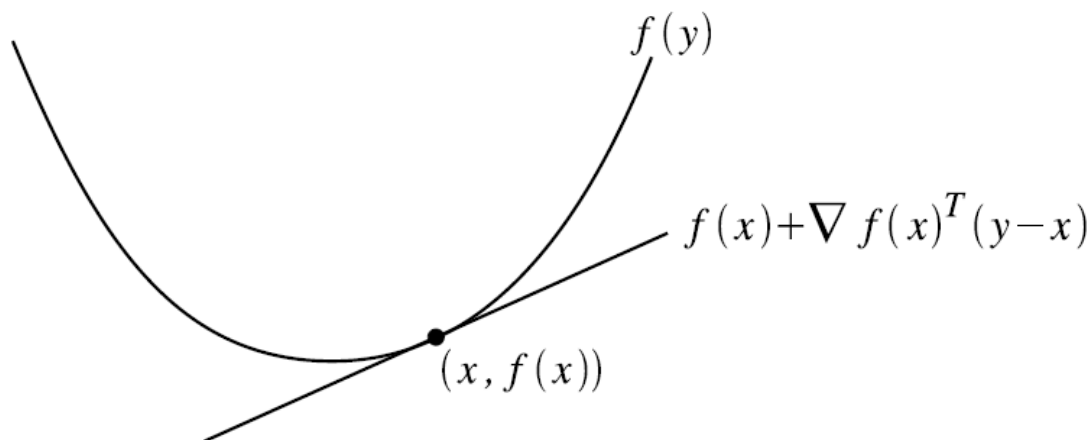


图 3:一阶条件下凸函数的图像.

注3：梯度定义为 $\nabla_x f(x)$, $(\nabla_x f(x))_i = \frac{\partial f(x)}{\partial x_i}$

3.2 二阶条件下的凹凸性(Convexity)

设函数 $f: \mathbb{R}^n \rightarrow \mathbb{R}$ 二阶可微(Hessian矩阵^{注4} $\nabla_x^2 f(x)$ 在 f 定义域内的 x 上处处存在)，当且仅当 $D(f)$ 为凸集且半正定时， f 为凸函数。例如，对任意 $x \in D(f)$ 有 $\nabla_x^2 f(x) \succeq 0$ 。

这里的“ $\succeq$ ”符号连接两个矩阵的时候，表示左侧矩阵半正定而非一个分量不等式^{注5}。一维情况下，它等价于二阶导数 $f''(x)$ 非负这一条件。

也类似于凹凸性定义和一阶条件，若Hessian矩阵正定则 f 严格凸；若Hessian矩阵负半定则凹，若Hessian矩阵负定 f 则严格凹。

注4：Hessian矩阵定义为 $\nabla_x^2 f(x) \in \mathbb{R}^{n \times n}$, $(\nabla_x^2 f(x))_{ij} = \frac{\partial^2 f(x)}{\partial x_i \partial x_j}$

注5：对于对称矩阵 $X \in \mathbb{S}^n$ ，记号 $X \preceq 0$ 表示 X 为负半定矩阵。同向量不等式一样，有时候“ $\geq$ ”、“ $\leq$ ”会和“ $\succeq$ ”、“ $\preceq$ ”混用。虽然记法上和向量不等式相似，但它们的概念却截然不同；特别地， $X \succeq 0$ 并不意味着对所有 i 和 j 都有 $X_{ij} \geq 0$ 。

3.3 Jensen's 不等式

凸函数基本定义的不等式中有

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

且 $0 \leq \theta \leq 1$ 。可以简单推知，对于任意 i ， $\sum_{i=1}^k \theta_i = 1$ 且 $\theta_i \geq 0$ ，则多点的凸组合有

$$f\left(\sum_{i=1}^k \theta_i x_i\right) \leq \sum_{i=1}^k \theta_i f(x_i)$$

事实上它也可以拓展到无穷级数或积分上。对积分形式，当 x 任意， $p(x) \geq 0$ 且 $\int p(x)dx = 1$ ，不等式可写为

$$f\left(\int p(x)x dx\right) \leq \int p(x)f(x)dx$$

因为 $p(x)$ 的积分为 1，所以常将其视为概率密度函数，因此上式可以写成期望的形式

$$f(E[x]) \leq E[f(x)]$$

，本式就是 Jensen's 不等式^{注6}。

注6：这四个不等式都是 Jensen's 不等式的等价形式。

3.4 子水平集(Sublevel Sets)

凸函数和一类特别重要的凸集有关，其为 α -子水平集(α -sublevel set)。给定凸函数 $f: \mathbb{R}^n \rightarrow \mathbb{R}$ 和实数 $\alpha \in \mathbb{R}$ ， α -子水平集即定义为

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

。换句话说， α -子水平集即是所有使得 $f(x) \leq \alpha$ 的 x 的集合。

如何证明它是个凸集呢？考虑使得 $f(x) \leq \alpha$ ， $f(y) \leq \alpha$ 的 $x, y \in D(f)$ ，则有

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

，得证。

3.5 凸函数示例

先看一些简单的一元函数然后再过渡到复杂一些的多元函数。

- 指数函数(Exponential)。对任意 $a \in \mathbb{R}$ ，有函数 $f: \mathbb{R} \rightarrow \mathbb{R}$ ， $f(x) = e^{ax}$ 。只要对 $f(x)$ 求二阶导就能证明指数函数为凸函数，因为其二阶 $f''(x) = a^2 e^{ax}$ 导对任意 x 恒为正。
- 负对数函数(Negative logarithm)。有函数 $f: \mathbb{R} \rightarrow \mathbb{R}$ ， $f(x) = -\log x$ ，其

$D(f) = \mathbb{R}_{++}$ (这里 $\mathbb{R}_{++}$ 表示严格正实数, $\{x : x > 0\}$)。因此对于全体 x 有 $f''(x) = 1/x^2 > 0$ 。

- **仿射函数(Affine functions)**。函数 $f: \mathbb{R}^n \rightarrow \mathbb{R}$, 对某些 $b \in \mathbb{R}^n$ 和 $c \in \mathbb{R}$ 有 $f(x) = b^T x + c$ 。这里, 对全体 x , 其Hessian矩阵为 $\nabla_x^2 f(x) = 0$ 。因为零矩阵既是半正定矩阵亦是半负定矩阵, 所以 f 既是凹函数亦是凸函数。事实上, 该形式的仿射函数是**唯一**既凹又凸的函数。

- **二次函数(Quadratic functions)**。令 $f: \mathbb{R}^n \rightarrow \mathbb{R}$, $f(x) = \frac{1}{2}x^T A x + b^T x + c$, 其中对称矩阵 $A \in \mathbb{S}^n$, $b \in \mathbb{R}^n$, $c \in \mathbb{R}$ 。其Hessian矩阵为 $\nabla_x^2 f(x) = A$ 。因此 f 是否为凸函数完全由 A 是否半正定来决定: 如果 A 半正定那么 f 为凸(类似地可推严格凸, 凹和严格凹); 如果 A 不定那么 f 为既非凹也非凸。欧氏范数平方函数 $f(x) = \|x\|_2^2 = x^T x$ 是二次函数的特例, 它的 $A = I$, $b = 0$, $c = 0$, 因此它是严格凸函数。

- **范数**。令 $f: \mathbb{R}^n \rightarrow \mathbb{R}$ 为某个在 $\mathbb{R}^n$ 上的范数。对 $x, y \in \mathbb{R}^n$ 和 $0 \leq \theta \leq 1$, 由三角不等式可得

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

。这个凸函数不可能用一阶导或者二阶导来证明其凹凸性, 因为它并非处处可导(比如, 1-范数 $\|x\|_1 = \sum_{i=1}^n |x_i|$, 它所有 x_i 为0的点都不可导)。

- **权值非负的凸函数加权和**。令 $f_1, f_2, \dots, f_k$ 为凸函数且 $w_1, w_2, \dots, w_k$ 为非负实数, 则

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

为凸函数。因为

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

4 凸优化(Convex Optimization)问题

了解了凸函数和凸集的概念之后, 我们现在可以来研究凸优化问题了。凸优化问题的优化问题形式是:

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

其中 f 为凸函数, C 为凸集, 而 x 为决策变量。但这样写有些抽象, 我们通常写成如

下形式：

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

其中 f 为凸函数， g_i 为一组凸函数， h_i 为一组仿射函数，而 x 为决策变量。

一定要注意这些不等式的方向：凸函数 g_i 一定是小于0的。因为 g_i 的0-子水平集为凸集，所以本身也是许多凸集的交集的可行域，也是个凸集。如果令某些 $g_i \geq 0$ ，则可行域将不再是凸集，求解问题的算法将无法保证找到全局最优解。还要注意，只有仿射函数可以做等式约束。可以把等式约束等价理解为两个不等式

$h_i(x) \leq 0$ 和 $h_i(x) \geq 0$ ，所以当且仅当 h_i 既是凸函数又是凹函数的时候它才是一条有效约束，这使得 h_i 只能是一个仿射函数。

优化问题的**最优值**记为 p^* (有时候记为 f^*)，它等于目标函数在可行域中的最小值^{注7}

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

如果问题没有**无可行解**(可行域为空)则 $p^* = +\infty$ ，而问题**无下界**(现有可行解使得 $f(x) \rightarrow -\infty$)则记为 $p^* = -\infty$ 。如果 x^* 使得 $f(x^*) = p^*$ 则 x^* 即为**最优解**。注意有时最优值一定的时候，最优解也会不止一个。

注7：数学系也把min写在括号右下角，其实这不重要。

4.1 凸问题中的全局最优性(Global Optimality)

在陈述凸问题的全局最优性结论之前，让我们形式化地定义局部最优解和全局最优解的概念。直观地，如果“附近”没有目标函数值更低的可行点，则该可行点即是局部最优解。而与之类似，如果可行域内完全没有使目标函数值更低的可行点，则这个使目标函数值最低的可行点即是全局最优解。为了更形式化一些，我们给出以下两个定义。

定义 4.1 若点 x 是可行的（即满足优化问题的约束的点）且存在 $R > 0$ 使得所有满足 $\|x - z\|_2 \leq R$ 的可行点 z 满足 $f(x) \leq f(z)$ ，则点 x 为局部最优解。

定义 4.2 如果点 x 是可行且对任意可行点 z 有 $f(x) \leq f(z)$ ，则点 x 为全局最优解。

我们现在来到了使得凸优化问题最为有用的关键：**凸优化问题中局部最优解即是全局最优解**。

可以通过反证法来为该性质作一个简明的证明。令 x 为局部最优解而非全局最优解，即存在可行点 y 使得 $f(x) > f(y)$ 。由局部最优性的定义可知，不存在可行点 z ，使得 $\|x - z\|_2 \leq R$ 且 $f(z) < f(x)$ 。但当令 $z = \theta y + (1 - \theta)x$ 其中 $\theta = \frac{R}{2\|x - y\|_2}$ 时，则有

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

$$= \left\| \frac{R}{2\|x - y\|_2} (x - y) \right\|_2 = R/2 \leq R$$

此外，由 f 的凹凸性可知 $f(z) = f(\theta y + (1 - \theta)x) \leq \theta f(y) + (1 - \theta)f(x) < f(x)$ 。不仅如此，由于可行域为凸集且由于 x 和 y 均可行，因此 $z = \theta y + (1 - \theta)x$ 也是可行的。所以可知 z 也是可行点且有 $\|x - z\|_2 \leq R$ 且 $f(z) < f(x)$ 。这与我们的假设相矛盾，说明 x 不仅是局部最优解，它也是全局最优解。

4.2 凸问题中的特例(Special Cases)

由于各种原因，通常考虑一般凸规划公式的特殊情况比较方便。对于这些特殊情况，我们通常可以设计出能够解决非常大问题的极其有效的算法，因此您可能会在人们使用凸优化技术的任何时候看到这些特殊情况。

- **线性规划**。若目标函数 f 和不等式约束 g_i 均为仿射函数，则该凸优化问题是一个线性规划(LP)问题。换句话说，即是有如下形式的问题：

$$\begin{aligned} & \text{minimize} && c^T x + d \\ & \text{subject to} && G(x) \preceq h \end{aligned}$$

$$Ax = b$$

其中决策变量 $x \in \mathbb{R}^n$, $c \in \mathbb{R}^n$, $d \in \mathbb{R}$, $G \in \mathbb{R}^{m \times n}$, $h \in \mathbb{R}^m$, $A \in \mathbb{R}^{p \times n}$, $b \in \mathbb{R}^p$, “ $\preceq$ ”表示元素(对应元素间的)不等式。

- **二次规划**。若不等式约束 g_i 为仿射函数而目标函数 f 为凸二次函数，则该凸优化问题是一个二次规划(QP)问题。换句话说，即是有如下形式的问题：

$$\begin{aligned} & \text{minimize} && \frac{1}{2} x^T P x + c^T x + d \\ & \text{subject to} && G(x) \preceq h \end{aligned}$$

$$Ax = b$$

其中决策变量 $x \in \mathbb{R}^n$, $c \in \mathbb{R}^n$, $d \in \mathbb{R}$, $G \in \mathbb{R}^{m \times n}$, $h \in \mathbb{R}^m$, $A \in \mathbb{R}^{p \times n}$, $b \in \mathbb{R}^p$, 且对称半正定矩阵 $P \in \mathbb{S}_+^n$ 。

- **带二次约束的二次规划**。若不等式约束 g_i 和目标函数 f 均为凸二次函数，则该凸优化问题是一个二次约束的二次规划(QCQP)问题。其形式为：

$$\begin{aligned} & \text{minimize} && \frac{1}{2} x^T P x + c^T x + d \\ & \text{subject to} && \frac{1}{2} x^T Q_i x + r_i^T x + s_i, \quad i = 1, \dots, m \\ & && Ax = b \end{aligned}$$

和前文类似，其中决策变量 $x \in \mathbb{R}^n$, $c \in \mathbb{R}^n$, $d \in \mathbb{R}$, $G \in \mathbb{R}^{m \times n}$, $h \in \mathbb{R}^m$, $A \in \mathbb{R}^{p \times n}$, $b \in \mathbb{R}^p$, 对称半正定矩阵 $P \in \mathbb{S}_+^n$, 且对 $i = 1, \dots, m$ 有 $Q_i \in \mathbb{S}_+^n$, $r_i \in \mathbb{R}^n$, $s_i \in \mathbb{R}$ 。

- **半定规划**。最后这个例子比前面几个示例要复杂，如果一开始没能理解它也大可不必担心。不过，半定规划在机器学习的许多领域变得越来越常见，所以你可能会

在以后遇到它，了解它仍对我们有所帮助。若凸优化问题有如下形式则为半定规划 (SDP)：

$$\begin{aligned} & \text{minimize} \quad \text{tr}(CX) \\ & \text{subject to} \quad \text{tr}(A_i X) = b_i, \quad i = 1, \dots, p \\ & \quad \quad \quad X \succeq 0 \end{aligned}$$

其中对称矩阵 $X \in \mathbb{S}^n$ 为决策变量，对称矩阵 $C, A_1, \dots, A_p \in \mathbb{S}^n$ ，约束 $X \succeq 0$ 表示 X 半正定。这看起来与之前几个凸优化问题略有不同，因为决策变量是矩阵而非向量。如果你好奇为什么这样的公式会有用，那么你可以研究一下关于凸优化的专著。

从上述定义中可以明显看出，二次规划比线性规划更一般（因为线性规划只是 $P = 0$ 这一特殊情形下的二次规划），同样地，带二次约束的二次规划比二次规划更一般。然而，不直观的事实是半定规划实际上比前几类规划更一般，因为任何带二次约束的二次规划（包含全部二次规划和线性规划）都可以表示为半定规划。

4.3 凸优化应用实例

既然我们已经了解了凸优化问题背后大量乏味的数学原理和公式，现在终于到了有趣的部分：用他们来解决实际问题。

• **支持向量机 (SVM)**。凸优化方法在机器学习中最普遍的应用之一就是支持向量机分类器。正如课堂上讲到的，得到（带有松弛变量的）支持向量机分类器可以建模为如下为优化问题：

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

其中决策变量 $w \in \mathbb{R}^n$ ， $\xi \in \mathbb{R}^m$ ， $b \in \mathbb{R}$ ， $C \in \mathbb{R}$ ，且 $x^{(i)}$ ， $y^{(i)}$ 中 $i = 1, \dots, m$ 。可以将问题变形为上一节中描述过的形式来证明这是一个二次规划。特别地，若定义

$$k = m + n + 1, \text{ 令决策变量为 } x \in \mathbb{R}^k \equiv \begin{bmatrix} w \\ \xi \\ b \end{bmatrix}, \text{ 且定义 } P \in \mathbb{R}^{k \times k} = \begin{bmatrix} I & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, c \in$$

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

$$\text{位矩阵, } \mathbf{1} \text{ 为元素全为1的向量, } X \text{ 和 } y \text{ 的定义为: } X \in \mathbb{R}^{m \times n} = \begin{bmatrix} x^{(1)T} \\ x^{(2)T} \\ \vdots \\ x^{(m)T} \end{bmatrix}, y \in \mathbb{R}^m =$$

$$\begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(m)} \end{bmatrix}.$$

可以发现当使用上面定义的这些矩阵时，上一节中描述的二次规划等价于SVM优化问题。在实践中，很容易发现SVM优化问题是由二次目标函数和线性约束构成的，

因此我们通常不需要将其改写为标准形式来“证明”它是一个二次规划问题。只有当使用现成的求解器，而它的输入要求采用标准二次规划形式的时候我们才会这样做。

• **带约束的最小二乘问题。**最小二乘问题中，我们希望对矩阵 $A \in \mathbb{R}^{m \times n}$ 和 $b \in \mathbb{R}^m$ ，求出 $\|Ax - b\|_2^2$ 的最小值。正如我们所看到的，这一问题可以通过正规方程 $(\theta = (X^T X)^{-1} X^T Y)$ 求出其解析解。但如果我们还希望将解 x 中的元素限定在某些预先定义的范围内，即求解下述优化问题：

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

其中 x 为决策变量， $A \in \mathbb{R}^{m \times n}$ ， $b \in \mathbb{R}^m$ ， $l \in \mathbb{R}^n$ ， $u \in \mathbb{R}^n$ 。这似乎仅仅是增加了一个约束，但它却会导致问题不再有解析解。当问题参数有如下定义时，这个优化问题确实是一个二次规划问题： $P \in \mathbb{R}^{n \times n} = \frac{1}{2} A^T A$ ， $c \in \mathbb{R}^n = -b^T A$ ， $d \in \mathbb{R} = \frac{1}{2} b^T b$ ， $G \in \mathbb{R}^{2n \times 2n} = \begin{bmatrix} -I & 0 \\ 0 & I \end{bmatrix}$ ， $h \in \mathbb{R}^{2n} = \begin{bmatrix} -l \\ u \end{bmatrix}$ 。

• **逻辑(logistic)回归的最大似然。**该模型的对数似然函数是

$$\ell(\theta) = \sum_{i=1}^n \{y^{(i)} \ln g(\theta^T x^{(i)}) + (1 - y^{(i)}) \ln [1 - g(\theta^T x^{(i)})]\}$$

其中 $g(z)$ 表示逻辑函数 $g(z) = 1/(1 + e^{-z})$ 。要最大化对数似然函数(等价于最小化负对数似然函数，该函数为凸函数)就要求出其最大似然估计，即

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

其中决策变量 $\theta \in \mathbb{R}^n$ 且不带任何约束。

与前两个例子不同，此问题化为优化问题的“标准”形并不容易。然而 ℓ 是凹函数的事实意味着可以用像牛顿法这样的算法来快速找到全局最优解。

4.4 Implementation: Linear SVM using CVX

许多凸优化问题可以通过几个现成的软件包解决，包括 CVX，Sedumi，CPLEX，MOSEK 等。因此在大多数情况下，一旦明确了凸优化问题的类型，你就不用考虑自己实现算法而直接求解它。这对于快速构建原型尤其有用^{注8}。

以上述软件包中的 CVX^[2] 为例。CVX 是一个基于 MATLAB 的免费软件包，用于解决一般凸优化问题；它可以求解各种凸优化问题，包括 LP，QP，QCQP，SDP 等。

如图所示，通过实现线性 SVM 分类器可以对已有数据进行二元分类。对使用其他非线性核的更一般情况，用 CVX 也可以解决对偶问题。

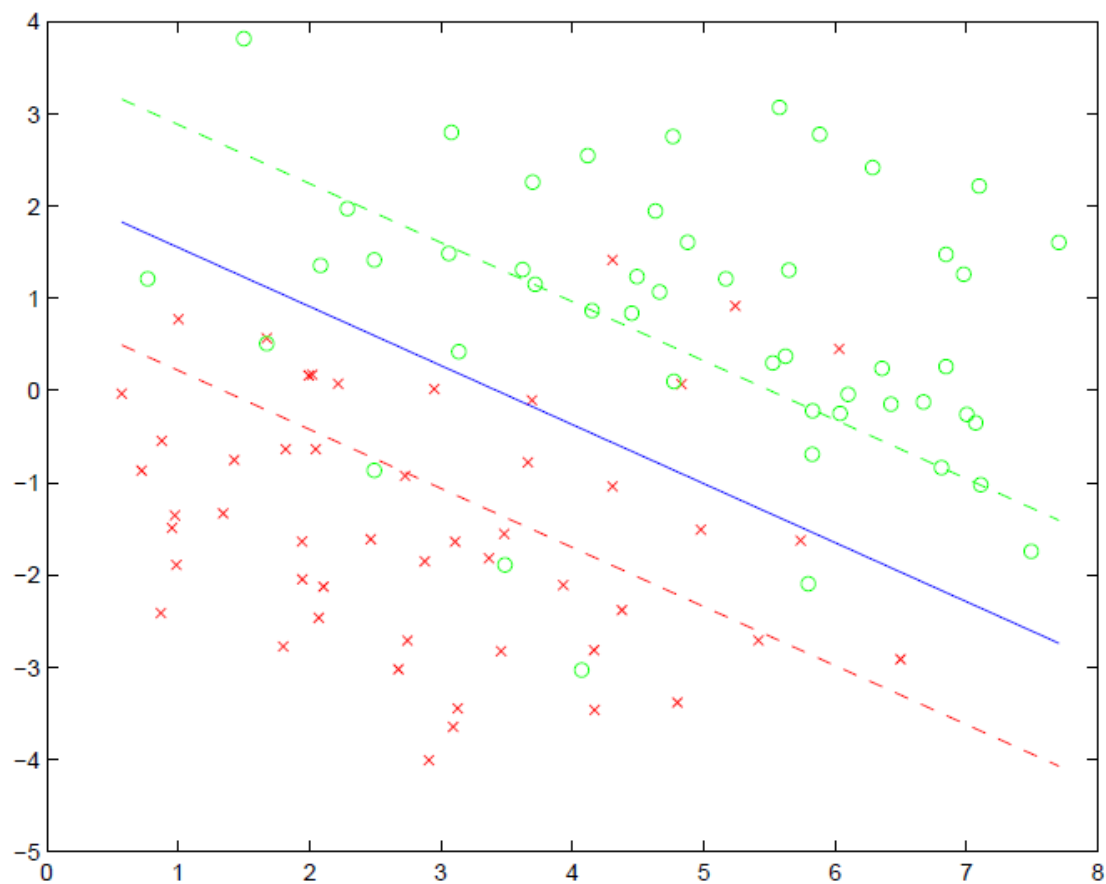


图 4: $C = 1$ 时线性SVM分类器的决策边界

```
% 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
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));

```

注8：然而根据优化问题的不同，这些现成的凸优化求解器相较于最佳实现来说可能要慢得多；因此，有时你可能需要更多地使用自定义求解器或者自己实现求解器。

参考文献

- [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://cvxr.com/>, September 2008.