第九章 优化基础

第 28 讲 优化

黄定江

DaSE @ ECNU
djhuang@dase.ecnu.edu.cn

引言

- 根据第 1 章提及的统计学习理论中的经验风险最小化准则，我们知道数据科学、人工智能和机器学习的很多问题都归结为一个优化问题。
- 对优化问题的求解已然成为大部分数据分析和机器学习算法的核心组成部分。
- 而且机器学习算法都是在计算机上操作的，其数学公式就表示为数值优化算法。
- 因为来源于实际应用的优化问题是如此的多样和复杂，所以在我们介绍各种具体的数值优化算法之前，
- 我们将安排两章内容，也即本章和下一章，来理清我们所面对的各种优化问题以及其可解的条件，
- 然后在第 12 章，我们会详细介绍各种具体的数值优化求解算法。

- 本章主要介绍优化的基础理论。我们在第 5 章已经看到，普通的最小二乘问题可以用标准线性代数工具求解。
- 在这种情况下，最小化问题的解可以被有效找到并且是整体最优解，也即，除了最小二乘最优解外没有其他更优的解。这些令人满意的特性实际上可扩展到一类更广泛的优化问题，而实现优化的关键特性就是所谓的“凸性”性质。
- 因此在本章中，将主要介绍：
  - 优化问题的定义、优化问题的分类、数据科学中常见的优化问题
  - 凸集和凸函数的定义和判别方法以及保凸运算、凸优化问题的定义和标准形式
  - 具有特定结构的某些类型的凸优化模型，例如线性、凸二次或凸二次曲线模型等等。
- 并介绍数据科学中常见的典型凸优化问题。

词向量相关的优化问题

- **词向量**:

$$\arg\max_{\mathbf{w},b} \prod_{i=1}^{m} (h_{\mathbf{w},b}(x_i)^{y_i} * (1 - h_{\mathbf{w},b}(x_i)^{1-y_i}))$$

或

$$\arg\min_{\mathbf{w},b} -\sum_{i=1}^{m} (y_i \log h_{\mathbf{w},b}(x_i)) + (1-y_i)\log(1-h_{\mathbf{w},b}(x_i))$$

- **连续词袋模型**

$$\min_{u,v} -u_c^T \hat{v} + \log \sum_{j=1}^{|V|} \exp\left(u_j^T \hat{v}\right)$$

- **跳格模型**

$$\min - \sum_{j=0, j\neq m}^{2m} u_{c-m+j}^T v_c + 2m \log \sum_{k=1}^{|V|} \exp\left(u_k^T v_c\right)$$

推荐系统中的优化问题

- 我们讨论了在推荐系统中的优化问题

$$\begin{aligned} \min_{\boldsymbol{X}} \quad & \operatorname{rank}(\boldsymbol{X}) \\ s.t. \quad & \boldsymbol{X}_{ij} = \boldsymbol{M}_{ij} \quad \forall i, j \in \mathbb{E} \end{aligned}$$

或者转化为限定在秩为 r 的条件下，求矩阵使得观测到的评分与预测的评分最接近：

$$\begin{aligned} \min_{\boldsymbol{X}} \quad & \sum_{ij} (\boldsymbol{X}_{ij} - \boldsymbol{M}_{ij})^2 \quad \forall i, j \in \mathbb{E} \\ s.t. \quad & \operatorname{rank}(\boldsymbol{X}) = r \end{aligned}$$

与正交矩阵相关的优化问题

- 正交 Procrustes 问题

$$\arg\max_{\boldsymbol{Q}\in\mathcal{O}(n)} \mathsf{Tr}(\boldsymbol{S}\boldsymbol{Q})$$

- Wahba 问题

$$\arg\min_{\boldsymbol{R}\in\mathcal{SO}(n)} \sum_{t=1}^{T} \frac{1}{2}\left\|\boldsymbol{R}\boldsymbol{x}_t - \boldsymbol{y}_t\right\|^2 = \arg\max_{\boldsymbol{R}\in\mathcal{SO}(n)} \mathsf{Tr}(\boldsymbol{S}\boldsymbol{R})$$

低秩矩阵相关优化问题

- 鲁棒 PCA

$$\min_{\boldsymbol{A},\boldsymbol{E}} \|\boldsymbol{A}\|_* + \lambda\|\boldsymbol{E}\|_1,\ s.t.\boldsymbol{X} = \boldsymbol{A} + \boldsymbol{E}$$

- 低秩矩阵补全

$$\min_{\boldsymbol{A}} \|\boldsymbol{A}\|_* s.t. P_\Omega(\boldsymbol{A}) = P_\Omega(\boldsymbol{D})$$

- 低秩矩阵表示

$$\min_{\boldsymbol{Z}} \|\boldsymbol{Z}\|_* s.t.\boldsymbol{D} = \boldsymbol{B}\boldsymbol{Z}$$

以及

$$\min_{\boldsymbol{Z},\boldsymbol{E}} \|\boldsymbol{Z}\|_* + \lambda\|\boldsymbol{E}\|_{2,1} s.t.\boldsymbol{D} = \boldsymbol{D}\boldsymbol{Z} + \boldsymbol{E}$$

最小二乘问题相关优化问题

- 最小二乘问题

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} \|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}\|_2 \tag{1}$$

- 加权最小二乘

$$\min \|\boldsymbol{A}_w\boldsymbol{x}-\boldsymbol{y}_w\|_2^2$$

- 约束最小二乘

$$\min_{\boldsymbol{x}} \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}\|_2^2$$
$$s.t. \quad \boldsymbol{B}\boldsymbol{x}=\boldsymbol{f}$$

- 总体最小二乘

$$\text{TLS:} \quad \min_{\Delta\boldsymbol{A},\Delta\boldsymbol{b},\boldsymbol{x}} \|\Delta\boldsymbol{A}\|_{\mathsf{F}}^2+\|\Delta\boldsymbol{b}\|_2^2$$
$$\text{subject to} \quad (\boldsymbol{A}+\Delta\boldsymbol{A})\boldsymbol{x}=\boldsymbol{b}+\Delta\boldsymbol{b}$$

机器学习中的优化问题

- 逻辑回归：

$$\min_{\boldsymbol{w}} \sum_{i=1}^{N} [y_i(\boldsymbol{w}^T \boldsymbol{x}_i) - \log(1 + \exp(\boldsymbol{w}^T \boldsymbol{x}_i))]$$

- 线性回归模型

$$\min_{(\mathbf{w}, b)} \sum_{i=1}^{m} (\mathbf{w}^T \mathbf{x}_i + b - y_i)^2$$

- 感知机

$$\min_{\mathbf{w}, b} - \sum_{x_i \in M} y_i(\mathbf{w}^T \mathbf{x}_i + b)$$

- **支持向量机**

$$\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N} L_{0/1}(y_i(\mathbf{w}^T\mathbf{x}_i + b) - 1),$$

$$\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N} \max(0, 1 - y_i(\mathbf{w}^T\mathbf{x}_i + b))$$

$$\begin{aligned} &\min_{\mathbf{w},b,\xi} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N} \xi_i \\ &s.t. \quad y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i, i = 1, 2, \ldots, N \\ &\xi_i \geq 0, i = 1, 2, \ldots, N \end{aligned}$$

- **非线性支持向量机**

$$
\min_{\alpha} \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j K(x_i, x_j) - \sum_{i=1}^{N}\alpha_i
$$

$$
s.t. \quad \sum_{i=1}^{N}\alpha_i y_i = 0, 0 \leq \alpha_i \leq C, i = 1, 2, \ldots, N
$$

- PCA

$$\min_{\boldsymbol{W}} \mathsf{Tr}(-\boldsymbol{W}^T\boldsymbol{X}\boldsymbol{X}^T\boldsymbol{W})$$
$$s.t. \boldsymbol{W}^T\boldsymbol{W} = \boldsymbol{I}$$

- k 均值聚类

$$C^* = \arg\min_C W(C) = \arg\min_C \sum_{l=1}^{k} \sum_{C(i)=l} \| Vx_i - \bar{\mathbf{x}}_l\|^2$$

- 谱聚类

$$\min_{\boldsymbol{x}} \boldsymbol{x}^T\boldsymbol{L}\boldsymbol{x}$$
$$\text{s.t.} \boldsymbol{x}^T\mathbf{1} = 0$$

优化简介

- 在标准算法理论中，设计一个有效的算法来解决手头的问题是算法设计者的主要责任。自计算机科学引入的几十年以来，人们为各种任务设计了很多优美的算法，这些任务包括查找图中的最短路径、计算网络中的最佳流、压缩包含由数码相机拍摄的图像的计算机文件以及替换文本文档中的字符串等。
- 这些设计方法虽然对许多任务都很有用，但并没有解决更复杂的问题，例如在位图格式的图像中识别特定的人，或者将文本从英语翻译成中文。对于上述任务，可能有一个很好的算法，但是算法设计方案可能是不容易扩展的。
- 正如图灵在他的论文中所提倡的那样，我们要教计算机学习如何解决一个任务，而不是教给它特定任务的解决方案。
- 实际上，这就是我们在学校中所做的，教会大家如何学习。我们希望教会计算机如何学习。这就是人工智能的思想，其核心是机器学习，并且主要就是从数据中来进行学习。

- 解决问题的机器学习方法有一个自动学习算法的机制。
- 比如，我们考虑一个图像数据，将图像分为两类的问题：包含汽车的图像和包含椅子的图像（假设世界上只有两种类型的图像）。
- 在机器学习中，我们训练（教导）一台机器以实现所需的功能，同一台机器可以潜在地解决任何算法任务，并且不同于一个任务到另一个任务只能由一组参数来决定机器的功能。
- 这很像计算机芯片中的电线决定了它的功能。事实上，目前最流行的机器学习方法之一是人工神经网络。
- 机器学习的数学优化方法是将机器训练过程看作一个优化问题。如果我们把 $\theta \in \mathbb{R}^d$ 作为机器的参数（也即模型，确定了参数就确定了模型），它被限制在某个集合 $\mathcal{K} \subseteq \mathbb{R}^d$ 中，如果函数 f 成功地度量了将实例映射到它们的正确标签，那么这个训练过程可以用数学优化问题来描述：

$$\min_{\theta \in \mathcal{K}} f(\theta) \tag{2}$$

- 这是本书关注的主要问题，并且将特别强调机器学习中出现的具有特殊结构的函数，以便设计有效的算法。

- 事实上，根据度量的准则不同和参数模型的不一样，机器学习中会有很多各种具有特殊结构的优化问题。
- 例如，我们在第 1 章中提到，在确定了训练集 $\mathcal{D}$、假设空间 $\mathcal{F}$ 以及学习准则后，如何找到最优的模型 $f(x,\theta)$ 就成了一个最优化（Optimization）问题。
- 注意这里 $f(x,\theta)$ 类似于优化问题(2) 中的 θ，其中 $\mathbf{x}$ 是输入实例，它对应的输出记为 $\mathbf{y}$，它们一起形成训练数据集

$$T=\{(x_1,y_1),(x_2,y_2),\cdots,(x_N,y_N)\}.$$

根据模型是否含有概率以及学习准则的不同，我们有如下四大类优化问题有待求解。

- 首先是经验风险最小化问题，求最优模型就是求解最优化问题：

$$\min_{f\in\mathcal{F}}\frac{1}{N}\sum_{i=1}^{N}L(y_i,f(x_i)),\tag{3}$$

其中，$\mathcal{F}$ 是假设空间，L 是损失函数，如平方损失函数等。

- 有时，为了避免过拟合，需引入结构风险最小化。结构风险最小化的策略认为结构风险最小的模型是最优的模型，也就是要求解最优化问题：

$$\min_{f\in\mathcal{F}} \frac{1}{N}\sum_{i=1}^{N} L(y_i, f(x_i)) + \lambda J(f), \tag{4}$$

- 其中 $J(f)$ 为模型的复杂度，是定义在假设空间 $\mathcal{F}$ 上的泛函。
- 上面两类优化主要针对函数类模型，当我们使用概率分布来为实际问题建模，我们会求解最大似然估计，也即最小化如下负对数似然问题：

$$\min_{\theta} \mathcal{L}(\theta), \tag{5}$$

其中 $\mathcal{L}(\theta) = -\log p(\mathbf{y}|\mathbf{X}, \theta) = -\sum_{n=1}^{N} \log p(y_n|\mathbf{x}_n, \theta)$。

- 如果我们有关于参数 θ 的分布的先验知识，则我们会求解一个最大后验估计，也即最小化如下负对数后验问题：

$$\min_{\theta} -\log p(\theta|\mathbf{x}), \tag{6}$$

- 其中 $p(\theta|\mathbf{x}) = \frac{p(\mathbf{x}|\theta)p(\theta)}{p(\mathbf{x})} \propto p(\mathbf{x}|\theta)p(\theta)$。
- 机器学习的训练过程其实就是最优化问题的求解过程。在机器学习中，优化又可以分为参数优化和超参数优化。
- 模型 $f(x;\theta)$ 中的 θ 称为模型的参数，可以通过优化算法进行学习，上面提及的四类优化问题都属于参数优化问题。
- 除了可学习的参数 θ 之外，还有一类参数是用来定义模型结构或优化策略的，这类参数叫做超参数（Hyper-Parameter）。在贝叶斯方法中，超参数可以理解为参数的参数，即控制模型参数分布的参数。

- 常见的超参数包括：聚类算法中的类别个数、梯度下降法的步长、正则项的系数、神经网络的层数、支持向量机中的核函数等。
- 超参数的选取一般都是组合优化问题，很难通过优化算法来自动学习。
- 因此，超参数优化是机器学习的一个经验性很强的技术，通常是按照人的经验设定，或者通过搜索的方法对一组超参数组合进行不断试错调整。

优化问题的一般形式

下面我们给出数学优化问题的一般形式以及相关的概念。

优化问题的一般形式

最优化问题或者说**优化问题**的一般形式表示为:

$$\begin{array}{rl} \text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) \le b_i, i = 1, \cdots, m \\ & h_j(x) = 0, j = 1, \cdots, p \end{array} \tag{7}$$

其中，向量 $\mathbf{x} = (x_1, \cdots, x_n)$ 称为问题的优化变量，函数 $f_0 : \mathbb{R}^n \to \mathbb{R}$ 称为目标函数，函数 $f_i : \mathbb{R}^n \to \mathbb{R}$，被称为不等式约束函数，常数 $b_1, \cdots, b_m$ 称为约束上限或者约束边界，$f_i(x) \le b_i (i = 1, \cdots, m)$ 称为不等式约束，函数 $h_j : \mathbb{R}^n \to \mathbb{R}$，被称为等式约束函数，$h_j(x) = 0, j = 1, \cdots, p$ 称为等式约束。

称满足所有约束条件的向量 $\mathbf{x}$ 为可行解或可行点，全体可行点的集合称为可行集，记为 D，其表示为：

$$D = \{\mathbf{x} | h_j(x) = 0, j = 1, \cdots, p, f_i(x) \leq b_i, i = 1, \cdots, m\}$$

若 $h_j(x)$ 和 $f_i(x)$ 是连续函数，则 D 是闭集。
在可行集中找一点 x^*，使目标函数 $f_0(x)$ 在该点取最小值，即满足：
$f_0(x^*) = \min f_0(x)$，使得 $f_i(x^*) \leq b_i$ 和 $h_j(x^*) = 0$ 的过程即为最优化的求解过程。x^* 称为问题的最优点或最优解，$f_0(x^*)$ 称为最优值。

- 优化问题(7)可以看成在向量空间 $\mathbb{R}^n$ 的一集备选解中选择最好的解。用 x 表示备选解，$f_i(x) \leq b_i$ 和 $h_j(x) = 0$ 表示 x 必须满足的条件，目标函数 $f_0(x)$ 表示选择 x 的成本 (同理也可以认为 $-f_0(x)$ 表示选择 x 的效益或者效用)。
- 优化问题(7)的解即为满足约束条件的所有备选解中成本最小 (或者效用最大) 的那个解。
- 在数据拟合中，人们需要在一族候选模型中选择最符合观测数据与先验知识的模型。此时，变量为模型中的参数，约束可以是先验知识以及参数限制 (比如说非负性)。
- 目标函数可能是与真实模型的偏差或者是观测数据与估计模型的预测值之间的偏差，也有可能是参数值的似然度和置信度的统计估计。
- 优化问题(7)此时即为寻找合适的模型参数值，使之符合先验知识，且与真实模型之间的偏差或者预测值与观测值之间的偏差最小 (或者在统计意义上更加相似)。

局部最优和整体最优

定义 1

整体（全局）最优解：若 $x^* \in D$，对于一切 $x \in D$，恒有 $f_0(x^*) \leq f_0(x)$，则称 x^* 是最优化问题(7)的整体最优解。

定义 2

局部最优解：若 $x^* \in D$，存在某个领域 $N_\varepsilon(x^*)$，使得对于一切 $x \in N_\varepsilon(x^*) \cap D$，恒有 $f_0(x^*) \leq f_0(x)$，则称 x^* 是最优化问题(7)的局部最优解。其中
$N_\varepsilon(x^*) = \{x | \|x - x^*\| < \varepsilon, \varepsilon > 0\}$
严格最优解：当 $x \neq x^*$，有 $f_0(x^*) < f_0(x)$ 则称 x^* 为优化问题(7)的严格最优解。

- 由上述定义可知，局部最优解 x^* 使 f_0 最小，但仅对可行集上的邻近点。
- 此时目标函数的值不一定是问题的（全局）最优值。局部最优解可能对用户没有实际意义。
- 因此局部最优解的存在在一般优化问题中是一个挑战，因为大多数算法往往被困在局部极小，如果存在的话，从而不能产生期望的全局最优解。

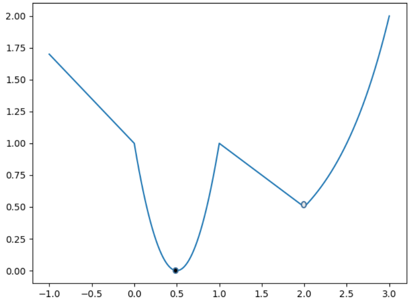

图 1: 局部（灰色）与全局（黑色）最小值。最佳集是单重态 $X_{opt} = 0.5$，点 x=2 是局部最小值。

易处理优化问题和不易处理优化问题

- 并非所有的优化问题都是平等的。一些问题类，如寻找一组有限的线性等式或不等式的解，可以用有效可靠的方法数值求解。相反，对于其他的一些问题，没有可靠有效的求解算法。
- 不讨论优化问题的计算复杂性，在这里，我们称之为“可处理的”所有那些优化模型，对于这些模型，可以用可靠的方式（在任何问题实例中）在数值上找到全局最优解，并且随着问题的大小（非正式地，问题的大小由模型中的决策变量和/或约束的数量来衡量）。其他问题被称为“困难”，然而对于其他问题，计算复杂性是未知的。
- 这本书的重点是可处理的模型，一个关键的信息是，可以以线性代数问题的形式或以凸的形式来表达的模型，通常是可处理的。此外，如果凸模型具有一些特殊结构，那么，可以使用一些现有的可靠的数值求解器进行求解。

问题变换

优化问题(7)的形式是非常灵活的，并允许许多变换，这就有利于我们将一个给定的问题处理为一个易于处理的问题。例如，优化问题

$$\min_x \sqrt{(x_1+1)^2+(x_2-2)^2} \qquad \text{s.t.}: x_1 \geq 0$$

与

$$\min_x (x_1+1)^2+(x_2-2)^2 \qquad \text{s.t.}: x_1 \geq 0$$

是等价的，而第二个优化问题的目标函数是可微的。有些情况下，还可以使用变量替换。例如，给定一个优化问题

$$\max_x \quad x_1 x_2^3 x_3 \qquad \text{s.t.}: x_i \geq 0, i=1,2,3, x_1 x_2 \leq 2, x_2^3 x_3 \leq 1$$

令新变量 $z_i = \log x_i, i=1,2,3$，在取目标的对数之后，该问题可以等价地写为

$$\max_z \quad z_1+3z_2+z_3 \quad z_1+z_2 \leq log2, 2z_2+z_3 \leq 0.$$

优点是替换后的目标函数和约束函数都是线性的。

优化问题的分类

- 优化问题种类繁多，因而分类的方法也有许多。可以按变量的性质分类，按有无约束条件分类，按目标函数的个数分类等等。
- 一般来说，变量可以分为确定性变量，随机变量和系统变量等等，相对应的最优化问题分别称为：普通最优化问题，统计最优化问题和系统最优化问题。
- 根据输入变量 x 的值域是否连续，数学优化问题可以分为离散优化问题和连续优化问题。

离散优化

离散优化（Discrete Optimization）问题是目标函数的输入变量为离散变量，比如为整数或有限集合中的元素。离散优化问题主要有三个分支：

1. 整数规划（Integer Programming）：输入变量 $\boldsymbol{x} \in \mathbb{Z}^d$ 为整数向量。常见的整数规划问题通常为整数线性规划（Integer Linear Programming，ILP）。整数线性规划的一种最直接的求解方法是：（1）去掉输入必须为整数的限制，将原问题转换为一般的线性规划问题，这个线性规划问题为原问题的松弛问题；（2）求得相应松弛问题的解；（3）把松弛问题的解四舍五入到最接近的整数。但是这种方法得到的解一般都不是最优的，因为原问题的最优解不一定在松弛问题最优解的附近。另外，这种方法得到的解也不一定满足约束条件。

2. 混合整数规划（Mixed Integer Programming, MIP），即自变量既包含整数也有连续变量。
2. 组合优化（Combinatorial Optimization）：其目标是从一个有限集合中找出使得目标函数最优的元素。在一般的组合优化问题中，集合中的元素之间存在一定的关联，可以表示为图结构。典型的组合优化问题有旅行商问题、最小生成树问题、图着色问题等。很多机器学习问题都是组合优化问题，比如特征选择、聚类问题、超参数优化问题以及结构化学习（Structured Learning）中标签预测问题等。

从这个意义上讲，组合优化是整数规划的子集。的确，绝大多数组合优化问题都可以被建模成（混合）整数规划模型来求解。
离散优化问题的求解一般都比较困难，优化算法的复杂度都比较高。

连续优化

连续优化（Continuous Optimization）问题的目标函数的输入变量为连续变量 $\boldsymbol{x} \in \mathbb{R}^d$，即目标函数为实函数。一般认为，在深度学习或机器学习中，模型中要学习的参数是连续变量。本书中主要讲解连续优化问题内容为主。

在连续优化问题中，根据是否有变量的约束条件，可以将优化问题分为无约束优化问题和约束优化问题。

1. 无约束优化问题（Unconstrained Optimization）的可行域为整个实数域 $D=\mathbb{R}^d$。在优化问题(7)中，当我们把不等式约束 $f_i(x)\le b_i$ 和等式约束 $h_j(x)=0$ 去掉时，即退化为无约束优化问题

$$\min_{x} f(x) \tag{8}$$

其中 $\boldsymbol{x}\in\mathbb{R}^d$ 为输入变量，$f:\mathbb{R}^d\to\mathbb{R}$ 为目标函数。我们前面提到的最小二乘问题和低秩近似问题都属于无约束优化问题。

2. 约束优化问题（Constrained Optimization）中变量 $\boldsymbol{x}$ 需要满足一些等式或不等式的约束。在优化问题(7)中，当不等式约束 $f_i(x)\le b_i$ 和等式约束 $h_j(x)=0$ 只要有一个成立，其即被称为约束优化问题。我们之前提到的最大方差问题属于约束优化问题。

此外，在连续优化问题中，根据函数的线性性质，可以将优化问题分为线性规划（线性优化）和非线性规划（非线性优化）。

1. 在优化问题(7)中，当目标函数和所有的约束函数都为线性函数，则该问题为线性规划问题（Linear Programming）。
   线性规划问题的解并没有一个简单的解析表达形式 (和最小二乘问题不同), 然而，存在很多非常有效的求解线性规划问题的方法，这当中包括 Dantzig 的单纯形法以及上世纪 80 年代发展起来的内点法。和最小二乘问题一样，处理极大规模的线性规划问题或者在很短时间内实时解决线性规划问题还是具有一定难度的。但是和最小二乘问题的情况类似，我们可以说求解 (大部分) 线性规划问题是一项成熟的技术。线性规划的求解程序可以 (并已经) 嵌入到很多工具箱和应用软件中。虽然一些应用可以直接表述为线性规划的形式，但是在很多情况下，原始的优化问题并不是线性规划问题的标准形式，这可以利用技巧转化为一个等价的线性规划问题 (然后进行求解)。

2. 在优化问题(7)中，如果目标函数或任何一个约束函数为非线性函数，则该问题为非线性规划问题（Nonlinear Programming）。
   对于一般的非线性规划问题，目前还没有有效的求解方法。有时看似简单的问题，变量个数可能不到 10，却非常难以求解，更不用说上百变量的非线性优化问题。因此，现有的用于求解一般非线性规划的问题都是在放宽某些指标条件下，采取不同的途径进行求解。

更进一步，根据函数的凸性，也即是否为凸函数，我们还可以把优化问题分为凸优化（Convex Programming）和非凸优化。

1. 在凸优化问题中，变量 $\mathbf{x}$ 的可行域为凸集，即对于集合中任意两点，它们的连线全部位于集合内部。目标函数 f 也必须为凸函数，即满足

$$f(\alpha\mathbf{x}+(1-\alpha)\mathbf{y}) \leq \alpha f(\mathbf{x})+(1-\alpha)f(\mathbf{y}), \forall\alpha \in [0,1]$$

   凸优化问题是一种特殊的约束优化问题，需满足目标函数为凸函数，并且等式约束函数为线性函数，不等式约束函数为凸函数。
   最小二乘问题和线性规划问题实质上都是凸优化问题的特殊形式。在非线性规划问题中，有些是凸优化问题，有些是非凸优化问题。

- 凸优化问题的解并没有一个解析表达式.
- 但是，和线性规划问题类似，存在很多有效的算法求解凸优化问题，如内点法，它可以在多项式时间内以给定精度求解这些凸优化问题。内点法几乎总可以在 10 步到 100 步之间解决凸优化问题。
- 如果不考虑特殊结构的凸优化问题 (如稀疏结构)，每一步需要的操作次数和下述变量成正比： $\max\{n^3, n^2m, F\}$，其中 F 是计算目标函数和约束函数 $f_0, \cdots, f_m$ 的一阶导数和二阶导数所需要的计算量。
- 我们可以使用目前的台式计算机轻易地解决包含数百变量、数千约束的凸优化问题，计算时间不超过一百秒。
- 如果问题本身具有一些特殊结构 (如稀疏结构)，则可以解决含有数千变量以及约束的更大规模的问题。
- 如果是一般的非线性凸优化问题，则并不容易，目前只对几类重要问题，如二阶锥规划和几何规划问题等，内点法正在逐渐成为一项成熟的技术。

- 类似于线性规划问题，理论上，对于大部分优化问题只要能把问题表述为凸优化问题，我们就能迅速有效地进行求解。
- 然而，不像判断某个问题是否为最小二乘问题或线性规划问题非常直接，凸优化问题的识别比较困难。
- 此外，较之线性规划问题，转换为凸优化问题的过程中存在更多的技巧。
- 因此，判断某个问题是否属于凸优化问题或识别那些可以转换为凸优化问题的问题是具有挑战性的工作。

2. 非凸优化对应于标准形式(7)中的一个或多个目标函数或约束函数不具有凸性的问题。一般来说，这样的问题很难解决。实际上，这个类包含组合优化：如果一个变量 x_i 必须是布尔型的（即 $x_i \in \{0\ 1\}$），我们可以将其建模为一对约束，其中第二个约束包含一个非凸函数。一般的非凸问题很难解决的原因之一是它们可能出现局部极小值

- 然而，需要注意的是，并不是每一个非凸优化问题都难以求解。例如，最大方差和低秩逼近问题是非凸问题，可以使用线性代数的特殊算法可靠地求解。
- 除了上述分类，还有按目标函数的个数分类：单目标最优化问题，多目标最优化问题；以及按约束条件和目标函数是否是时间的函数分类：静态最优化问题和动态最优化问题（动态规划）。
- 由上述讨论可知，目前对于线性规划和凸优化问题有成熟的求解方法，对于其它问题的求解思路主要想办法把它转化为线性规划或凸优化问题。
- 因此接下来我们重点介绍凸优化问题有关的凸分析基础，包括凸集、凸函数以及凸优化问题的性质等等。