机器学习：凸优化(Convex Optimization)

求解一般的最优化问题的全局最优解通常有一定困难，至少会面临局部极值点或者是鞍点的问题。但是如果对于优化问题加以约束条件限定，则可以有效的避免这些问题，保证算法一定能求得全局极值点。

典型的限定问题为凸优化问题。前面我们学习过的线性回归(第四章)，$Logistic$ 回归(第五章)，带约束的支撑向量机(第六章)，以及神经网络中常用的 $Softmax$ 回归(第五章)，都是典型的凸优化问题。

1. 函数凹凸性

首先复习一下第三章的内容：

对于函数 $f(x)$，对于其定义域内的任意两点 x, y，以及任意的实数 $0 \leqslant \theta \leqslant 1$，都有：

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

则函数为凸函数。如果上式满足 $<$，则为严格凸函数。(曲线/曲面向下凸)

对于函数 $f(x)$，对于其定义域内的任意两点 x, y，以及任意的实数 $0 \leqslant \theta \leqslant 1$，都有：

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

则函数为凹函数。如果上式满足 $>$，则为严格凹函数。(曲线/曲面向上凸)

对多元函数，假设 $f(x)$ 二阶可导，$f(x)$ 为凸函数的充要条件是：$f''(x) \geqslant 0$。

对多元函数，假设 $f(x)$ 二阶可导，如果对应的黑塞矩阵半正定($xAx^T \geqslant 0$；或者 A 的特征值均非负)，则为凸函数；如果黑塞矩阵正定($xAx^T > 0$；或者 A 的特征值均为正)，则为严格凸函数。凹函数为对应的黑塞矩阵半负定...

2. 常规数值优化算法的问题

基于导数的数值优化算法判断收敛的依据是梯度为 0 ，但是梯度为 0 只是函数取得局部极值的必要条件而非充分条件，更不是函数取得全局极值的充分条件。因此，这类算法面临了如下问题：

1. 无法收敛到梯度为 0 的点，此时算法不收敛。
2. 能够收敛到梯度为 0 的点，但该点的黑塞矩阵为非正定。因此这只是个鞍点，不是局部极值点。
3. 能够收敛到梯度为 0 的点，该点的黑塞矩阵正定。这是局部极值点，但不一定是全局的极值点。

对于 $-x^2+y^2$ ，如果以 $(0,4)$ 作为初始迭代点，最后会陷入鞍点 $(0,0)$ 。该点梯度为 0 ，黑塞矩阵为 $\begin{pmatrix}-2 & 0\\ 0 & 2\end{pmatrix}$ ，该矩阵特征值为 $-2,2$ ，黑塞矩阵不定，不是极值点。

相比鞍点，判断一个局部极值点是否是全局极值点更为困难，目标函数可能存在多个局部极值。需要找到所有的局部极值然后比较，通常这是一个 NP 难的问题。梯度下降法及其变种，具有一定的摆脱局部极小值和鞍点的能力。

3. 凸集($Convex\ Set$)

对于 n 维空间中的点集 C ，如果对该集合中的任何两点 x,y ，以及实数 $0\leqslant\theta\leqslant 1$ ，都有：

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

则该集合为凸集。直观上而言，凸集的形状都是凸的，没有凹进去的地方。把集合中的任何两点用直线连接，线段上的所有点都属于该集合。

$\theta x+(1-\theta)y$ 称为点 x,y 的凸组合。下面列举一些常见的凸集：

1.n 维实向量空间 $\mathbb{R}^n$ 是凸集。显然如果 $x,y\in\mathbb{R}^n$ ，则有：

$$\theta x+(1-\theta)y\in\mathbb{R}^n$$

2.给定 $A_{m\times n}$ 和 b_m ，仿射子空间是非齐次线性方程组的解，也是凸集：

$$\{Ax=b,x\in\mathbb{R}^n\}$$

3.由一组线性等式约束条件定义的可行域是凸集。假设 $x,y\in\mathbb{R}^n$ 并且 $Ax=b,Ay=b$ ，对于任意的 $0\leqslant\theta\leqslant 1$ ，有：

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

4.多面体也是凸集，定义为如下线性不等式组定义的向量的集合：

$$\{Ax\leqslant b,x\in\mathbb{R}^n\}$$

对于 $\forall x,y\in\mathbb{R}^n$ 并且 $Ax\leqslant b,Ay\leqslant b,0\leqslant\theta\leqslant 1$ ，有：

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

5.由 **4.** 推导得到：由线性不等式约束条件定义的可行域是凸集。

实际优化问题中，等式和不等式的约束通常是线性的，因此它们确定的可行域是凸集。

6.多个凸集的交集也是凸集。假设凸集 $C_1, \cdots, C_k$ 的交集为 $\bigcap_{i=1}^{k} C_i$ 。对于任意的点 $x, y \in \bigcap_{i=1}^{k} C_i, 0 \leqslant \theta \leqslant 1$ ，有：

$$\theta x + (1-\theta)y \in C_i, \quad \forall i = 1, \cdots, k$$
$$\rightarrow \theta x + (1-\theta)y \in \bigcap_{i=1}^{k} C_i$$

这个结论意味着，如果每个等式或者不等式的约束条件的集合如果都是凸集，那么这些条件联合起来定义的集合也是凸集。

ps: 凸集的并集不是凸集。

7.给定一个凸函数 $f(x)$ 以及实数 α ，此函数的 α 下水平集($Sub-level\ Set$)定义为函数值小于等于 α 的点构成的集合：

$$\{f(x) \leqslant \alpha, x \in D(f)\}$$

$D(f)$ 为 $f(x)$ 定义域。对于 $\forall x, y$ 满足 $f(x) \leqslant \alpha, f(y) \leqslant \alpha$ 。对于 $0 \leqslant \theta \leqslant 1$ ，根据凸函数定义有：

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

即 $\theta x + (1-\theta)y$ 也属于该下水平集，因此下水平集是凸集。例如：对于凸函数 $f(x, y) = x^2 + y^2$ ，如果 $\alpha = 1$ ，则下水平集 $x^2 + y^2 \leqslant 1$ 为单位圆，凸集。

如果 $f(x)$ 不是凸函数，则不能保证下水平集是凸集。对于下面的凹函数 $f(x, y) = -x^2 - y^2$ ，如果 $\alpha = 1$ ，则下水平集 $-x^2 - y^2 \leqslant 1$ 为二维空间除去单位圆的区域，为非凸集。

这一结论用于，我们需要确保优化问题中的一些不等式约束条件定义的可行域，是凸集。

4. 凸优化问题

如果一个最优化问题的可行域是凸集且目标函数是凸函数，则称该问题为凸优化问题。凸优化问题要求目标函数是凸函数，且优化变量的可行域是凸集，其形式为：

$$\min_x f(x),\ x \in C$$

其中 x 为优化变量，f 为凸目标函数，C 是优化变量的可行域，为凸集。其另一种表达形式为：

$$\begin{aligned} &\min_x f(x) \\ &g_i(x) \leqslant 0,\ \ i = 1, \cdots, m \\ &h_i(x) = 0,\ \ i = 1, \cdots, p \end{aligned}$$

支撑向量机的原问题就采用了这样的表达方式。

其中 $g_i(x)$ 是不等式约束函数，为凸函数；$h_i(x)$ 是等式约束函数，为仿射(线性函数。$g_i(x)$ 的不等式方向非常重要，前文已经提到，一个凸函数的 0 下水平集为凸集，一个凹函数则不成立。

这些不等式共同定义的可行域是一组凸集的交集，仍然是凸集。通过将大于或等于号形式的不等式同时乘以 -1 ，可以把不等式统一写成小于或等于的形式。前面已经证明仿射空间是凸集，因此加上这些等式约束后还是凸集，需要强调的是，如果等式约束不是仿射函数，那么通常无法保证其定义的可行域是凸集。例如等式约束 $x^2+y^2+z^2=1$ 确定的可行域是三维空间的球面，显然不是凸集。

上面的定义也给出了证明一个最优化问题是凸优化问题的一般性方法，即证明目标函数是凸函数，证明目标丽数是凸函数的方法前文已经介绍。

对于凸优化问题，所有局部最优解一定是全局最优解。这个特性可以保证在求解时不会陷入局部极值问题， 如果找到了问题的一个局部最优解，则它一定也是全局最优解，这极大地简化了问题的求解。下面采用反证法证明此结论。

假设一个解是局部最优解但不是全局最优解，则存在一个可行解 y ，满足

$$f(x)>f(y)$$

根据局部最优解的定义，对于给定的邻域半径 δ ，不存在满足 $||x-z||_2<\delta$ 并且 $f(z)<f(x)$ 的点 z 。选择一个点，令：

$$z=\theta y+(1-\theta x)$$

其中，$\theta=\frac{\delta}{2||x-y||_2}$ ，则有：

$$||x-z||_2=\left|\left|x-\left(\frac{\delta}{2||x-y||_2}y+\left(1-\frac{\delta}{2||x-y||_2}\right)x\right)\right|\right|_2=\left|\left|\frac{\delta}{2||x-y||_2}(x-y)\right|\right|_2=\frac{\delta}{2}<\delta$$

即该点在 x 的 δ 邻域内。根据凸函数的性质以及前面的假设 $f(x)>f(y)$ 有

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

这与 x 是局部最优解矛盾。即如果一个局部最优解不是全局最优解，在它的任何邻域内还可以找到函数值比该点函数值更小的点，这与该点是局部最优解矛盾。

之所以凸优化问题的定义要求且标函数是凸函数，并且优化变量的可行域是凸集，是因为缺少其中任何一个条件都不能保证局部最优解是全局最优解。下面来看两个反例。

- 可行域是凸集，目标函数不是凸函数。显然，此非凸函数存在多个局部极小值点，但只有一个是全局极小值点。
- 可行域不是凸集，目标函数是凸函数。 可行域不是凸集，中间有断裂，目标函数是凸函数。左边和右边的曲线各有一个局部极小值点，分别为 $x=-1,x=1$ ，不能保证局部极小值就是全局极小值。可以很容易把这个例子推广到三维空间里的二元函数(曲面)。

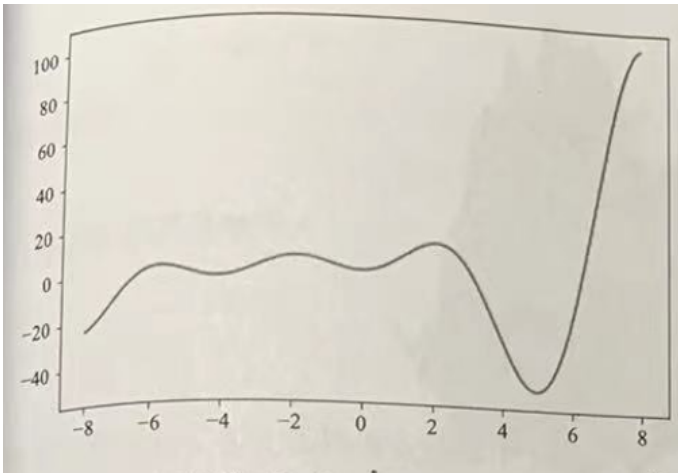

图 4.19　可行域是凸集，目标函数不是凸函数

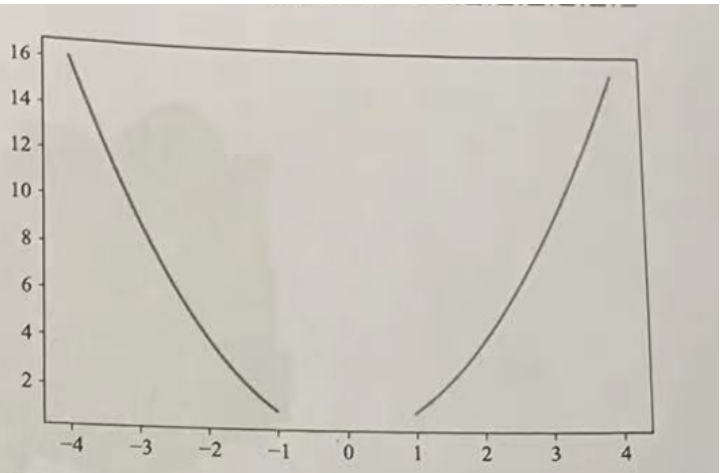

图 4.20　可行域不是凸集，目标函数是凸函数

由于凸函数的黑塞矩阵是半正定的，不存在鞍点，因此凸优化问题也不会出现鞍点问题。

5. 机器学习中的凸优化问题

下面介绍机器学习中典型的凸优化问题。对于这些问题，优化算法可以保证找到全局极值点，因此训练时的收敛性是有保证的。

这些凸优化问题前文已经详细阐述过，此处不再赘述。

线性回归：

$$\min_{w} \frac{1}{2l} \sum_{i=1}^{l} (y_i - w^T x_i)^2$$

可以对该损失函数求二阶导，证明其黑塞矩阵为半正定函数，因此这是一个凸函数。

$Logistic$ **回归：**

$$\min_{w} -\sum_{i=1}^{l} (y_i \ln h(x_i) + (1 - y_i) \ln(1 - h(x_i)))$$

$$h(x) = \frac{1}{1 + \exp(-w^T x)}$$

可以证明，交叉熵函数的二阶黑塞矩阵半正定，该目标函数是凸函数。

支撑向量机：

软间隔支撑向量机(带约束修正)的公式如下：

$$\begin{aligned} \min \frac{1}{2} w^T w + \ & C \sum_{i=1}^{l} \xi_i \quad (C > 0) \\ s.t \quad & y^{(i)}(w^T x^{(i)} + b) \geqslant 1 \ - \xi_i \\ & \xi_i \geqslant 0 \quad i = 1, \cdots, l \end{aligned}$$

在支撑向量机一章中详细分析过，目标函数的黑塞矩阵是 n 阶单位矩阵 I，是严格正定矩阵，因此目标函数是凸函数。可行域是线性不等式围成的区域，因此是一个凸集。这个最优化问题是一个凸优化问题。

$Softmax$ **回归：**

$$\min_{\theta_i} -\sum_{i=1}^{l}\sum_{j=1}^{k}\left(y_{ij}\ln\frac{\exp(\theta_j^T x_i)}{\sum_{t=1}^{k}\exp(\theta_t^T x_i)}\right)$$

非凸优化问题：神经网络

在常用的机器学习算法中，目标函数不是凸函数的典型代表是神经网络。

假设神经网络使用的是均方误差：

$$L(w)=\frac{1}{2l}\sum_{i=1}^{l}(y_i-h(x_i))^2$$

$h(x_i)$ 是神经网络实现的非线性映射，w 是所有层的权重和偏置的参数集合，这是一个不带约束条件的优化问题，训练时无法保证收敛到局部极值点，更无法保证收敛到全局最优解，会面临局部极值和鞍点问题。

常见的神经网络中的优化算法变种可参考Lesson 3 优化方法基础。

更详细的凸优化知识请自行翻阅相关资料。