最优化学习笔记 1(p1-p12)

BD S

2021 年 7 月 2 日

目录

1 优化问题概括

优化问题的形式

$$\begin{aligned} &minimize\ f_0(x) \\ &subject\ to\ f_k(x) \le 0 \\ &\qquad h_i(x) = 0 \end{aligned} \tag{1}$$

$f_0(x)$ 的最大最小值可以不存在，但是上确界 $(sup\ f)$ 下确界 $(inf\ f)$ 必须存在

类型有很多： 线性规划（目标函数和约束函数都是线性函数），非线性规划（有一个不是线性函数），二次规划（目标函数二次函数约束函数是线性函数），无倒数优化（不能求导

的问题），整数规划（变量只能取整数），半定规划，稀疏优化（最优解只要少量非 0 元素）低秩矩阵优化，分布式优化。

不同的建模会给出不同的问题，极小化收益的方差就是二次规划，极小化风险价值函数就是混合整数规划，极小化条件风险价值，是非光滑优化

2 稀疏优化

考虑这样一个方程组

$$Ax = b \tag{2}$$

其中 $x \in \mathbb{R}^n$ $b \in \mathbb{R}^m$ $A \in \mathbb{R}^{m*n}$ ，其中 $m << n$。问题一般是已知向量 b 和矩阵 A，想要重构 x。前置条件是原始信号中有很多 0 元素，也就是**稀疏解**. 显然是有无穷多解的，但是大多数是我们不感兴趣的，有用的是**稀疏解**。这被广泛用于**压缩感知**。常见的数据矩阵 A 通常由离散余弦变换，小波变换，傅里叶变换生成。可以进一步写成

$$\begin{aligned} &minimize\ ||x||_1 \\ &subject\ to\ Ax = b \end{aligned} \tag{3}$$

LASSO 问题，带 l_1 范数正则化的优化问题

$$min\ \mu||x||_1 + \frac{1}{2}||Ax - b||_2^2 \tag{4}$$

显然，把问题转换成这个形式会简单很多。

3 低秩矩阵恢复优化

这个和稀疏性恢复有点像。低秩矩阵恢复可以用于推荐系统以及相似性的分析，比如说下面的这个推荐系统。

一直矩阵的一部分值，来获得矩阵的其他值（就是猜测其他用户对于电影的喜欢程度）。如图 1 所示，已知了一部分的用户的评分，问号部分是需要去预测的。

	电影 1	电影 2	电影 3	电影 4	…	电影n
用户 1	4	?	?	3	⋯	?
用户 2	?	2	4	?	⋯	?
用户 3	3	?	?	?	⋯	?
用户 4	2	?	5	?	⋯	?
⋮	⋮	⋮	⋮	⋮		⋮
用户m	?	3	?	4	⋯	?

图 1: 用户电影评级系统

列出优化问题

$$\begin{aligned} &min\ rank(X) \\ &s.t.\ X_{ij} = M_{ij} \end{aligned} \tag{5}$$

其中 X 预测的结果矩阵，M 是已知的矩阵值。仅有的约束条件就是 X 的部分值要与已知的值相等，并且秩越低（越相似）越好。

但是求秩并不简单，$rank(X)$ 可以替换为非 0 奇异值个数。根据稀疏优化的思想，可以更换为奇异值之和，也就是矩阵的核范数 $||X||_*$。问题 (5) 就变成了 (6)

$$\begin{aligned} &min\ ||X||_* \\ &s.t.\ X_{ij} = M_{ij}\ (i,j) \in \omega \end{aligned} \tag{6}$$

很容易证明，问题 (6) 是一个凸优化的问题，也是半定规划问题。它的二次罚函数形式为

$$min\ \mu||X||_* + \frac{1}{2}\sum_{(i,j)\in\omega}(X_{ij} - M_{ij})^2 \tag{7}$$

4 深度学习——多层感知机

多层感知机（前馈神经网络、深度前馈网络）。这里举例一个图片分类的神经网络，比如说要区分是猫或者狗，就是在逼近一个从图片到 $\{0,1\}$ 的函数。

具体的，给定一个训练集 $D = \{\{a_1,b_1\},\{a_2,b_2\},...,\{a_m,b_m\}\}$，假设 $a_i \in \mathbb{R}^p, b_i \in \mathbb{R}^q$，并且设 a_i 的第一项是 1，也就是 $a_{i1} = 1$。a_i 是输入，是输入层 $l = 0$，b_i 是输出，是输出层 $l = L + 1$，中间还有 L 个隐藏层。

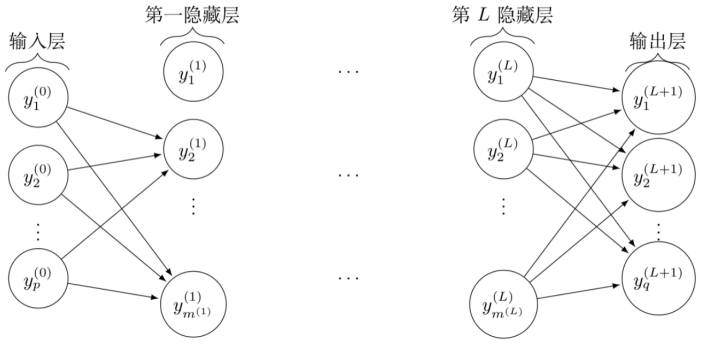

图 2: 神经网络结构

多层感知机的实质就是由上一层的数值做线性组合再逐分量作线性变换得到的。设第一层的元素为 $y^{(0)}$，第二层的元素就是 $y^{(1)}$，令它们的中间变量是 $z^{(1)}$。变换过程为：

$$z_{(i)}^{(1)} = \sum_{k=1}^{p} x_{i,k}^{(l)} y_k^{(0)} \tag{8}$$

这一步的 $z^{(1)}$ 就是 $y^{(0)}$ 中元素的简单加权求和。权重的选择未知。下一步是把 $z^{(1)}$ 变成 $y^{(1)}$，这一步使用一个激活函数 $t(\cdot)$ 也就是

$$y_i^{(1)} = t(z^{(1)}) \tag{9}$$

激活函数 $t(\cdot)$ 可以由很多函数，比如说 Sigmoid 函数

$$t(z) = \frac{1}{1 + exp(-z)} \tag{10}$$

Heaviside 函数

$$t(z) = \begin{cases} 1, & z \geq 0 \\ 0, & z < 0 \end{cases} \tag{11}$$

整个过程可以描述为 $y^{(0)} \to z^{(1)} \to y^{(1)} \to z^{(2)} \to y^{(2)} \to ... \to z^{(l)} \to y^{(l)} \to z^{(l+1)} \to y^{(l+1)}$

那么多层感知机的优化模型就迟来了，为

$$\min_x \sum_{i=1}^{m} ||h(a_i; x) - b_i||_2^2 + \lambda r(x) \tag{12}$$

其中 $h(a_i; x)$ 是输出的结果，b_i 是实际的结果，$r(x)$ 是正则项，用来刻画解的某些性质，比如说稀疏性和光滑性。

5 深度学习——卷积神经网络

CNN，卷积神经网络，是一种深度前馈人工神经网络（前馈和反馈的区别），可以用于时间序列和图像等网格数据的处理。

区别于上文中的多层感知机，卷积神经网络并没有采取全连接（即采用上一层中的所有数据），仅选取部分数据，不过总体思想是一样的。如图 3 所示，K 是一个权重矩阵，取矩

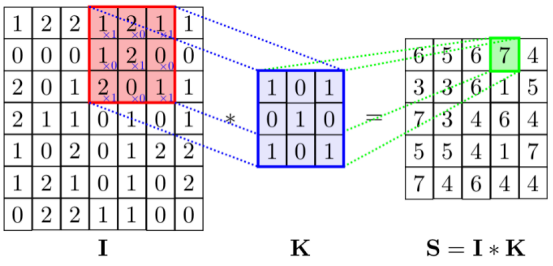

图 3: 卷积操作

阵 I 中对应位置对应元素与权重矩阵 K 进行逐元素的相乘。也就是说，给定一个二维图像

$I \in \mathbb{R}^{n\times n}$ 以及卷积核 $K \in \mathbb{R}^{k\times k}$。有这样的一种卷积操作 $S = I * K$，并且每个元素满足式子

$$S_{i,j} =< I(i : i + k - 1, j : j + k - 1), K >$$

与多层感知机相似，整个流程也是输入图片，卷积（权重的赋值），激活函数，得到下一层。以此类推，得到了卷积神经网络的结构，如图 4 所示。

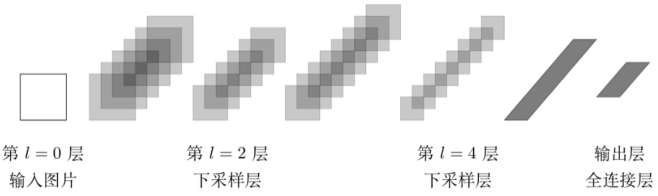

图 4: 卷积神经网络示意图

6 最优化基本概念

主要任务 (3 个)：**构造模型，确定优化问题设计算法，实现算法求解**。

连续和离散优化问题：离散优化，变量在离散集合上取值（离散点集、整数集），比如说整数规划。求解比较困难，一般需要把离散变成连续来求解（定界方法）。

有约束和无约束优化问题：可以将约束罚到目标函数上去。常见方法：**增广拉格朗日函数法、罚函数法**。

随机性和确定性优化问题：随机优化问题：目标或者约束函数中涉及随机变量而带有不确定性的问题（只会这些参数的估计）。在机器学习、强化学习有广泛的应用。参数未知，所以实际中是通过足够大的样本，来逼近目标函数，得到一个有限和的目标函数。以目标函数为多项求和的优化问题为例，使用确定性优化算法，会引入昂贵的复杂度。但是对于随机优化问题，每次只能计算和式的一项或者几项，节约了计算时间。

线性规划和非线性规划问题：46-47 年单纯形法，79 年线性规划问题多项式时间算法的存在，84 年多项式时间的内点法。

凸和非凸优化问题：

全局最优解和局部最优解：全局极小解、严格局部最小解、非严格局部最小解。

7 最优化基本概念——优化算法

能用代数表达式给出最优解——显式解，但是一般没法得到，就要用迭代算法。

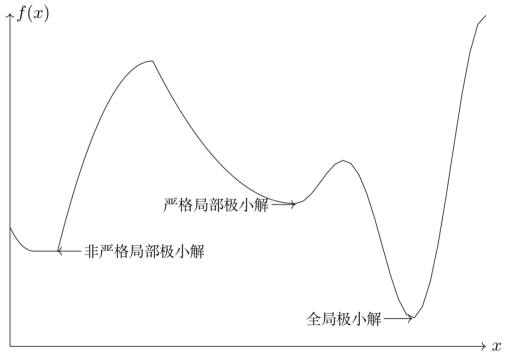

图 5: 全局极小解、严格局部最小解、非严格局部最小解

那么迭代算法一定要有收敛准则，一般的收敛准则是

$$\frac{f(x^k) - f^*}{max\{|f^*|, 1\}} \leq \epsilon_1,\ ||\nabla f(x^k)|| \leq \epsilon_2 \tag{13}$$

ϵ_1 和 ϵ_2 都是很小的正数，$||*||$ 代表某种范数，f^* 代表最优值。也就是一般是用函数值或者梯度来判断收敛的。

除了收敛性还有违反约束度的分析。

还有与最优解之间的距离的分析。（此处省略）。

接下来是一些收敛的概念 除了点列和函数值的收敛，还会使用到如无约束优化问题中

表 1: 一些概念

依点列收敛到局部（全局）极小解	从 x 角度来看收敛
全局依点列收敛到局部（全局）极小解	任意的初值 $x0$
全局依函数值收敛到局部（全局）极小解	从 $f(x)$ 角度来看收敛

的梯度范数，有约束优化问题中的最优性违反度等.

7.1 优化问题处理的操作技巧

泰勒展开：对于一个非线性的目标函数或者约束函数，通过泰勒展开来用线性函数或者二次函数来逼近。但是简化问题只在小邻域内逼近原始问题，需要不断根据迭代点来重新构造简化问题。

对偶：通过求解对偶问题来同时求解原始问题和对偶问题。（拉格朗日？）

拆分：如 $\min_x h(x) + g(x)$ 可以拆分为 $\min_{x,y} g(x) + h(y)\ \ s.t.\ x = y$ ，这样通过交替求解来得到原问题的解。（ADMM 算法）

块坐标下降：对于一个 n 维（n 很大）空间的优化问题，通过逐步求解分量的方式来算。先固定 $x_2 - x_{100}$ 去求解 x_1，再固定 x_1 和 $x_3 - x_{100}$，求解 x_2，以此类推。

7.2 收敛速度 Q 和 R

对于充分大的 k,Q 收敛的定义为

表 2: Q 收敛

Q-线性收敛的	$\frac{\|\|x^{k+1}-x^*\|\|}{\|\|x^k-x^*\|\|} \leq a, \quad a \in (0,1)$
Q-超线性收敛的	$\lim\limits_{k\to\infty} \frac{\|\|x^{k+1}-x^*\|\|}{\|\|x^k-x^*\|\|} = 0$
Q-次线性收敛的	$\lim\limits_{k\to\infty} \frac{\|\|x^{k+1}-x^*\|\|}{\|\|x^k-x^*\|\|} = 1$
Q-二次线性收敛的	$\frac{\|\|x^{k+1}-x^*\|\|}{\|\|x^k-x^*\|\|^2} \leq a, \quad a \in (0,1)$

R 收敛定义：以一个点列 $\{x^k\}$ 为例，设 $\{x^k\}$ 最终收敛到最优值 x^*，并且存在一个非负序列 t_k，使得 $|x^k - x^*| \leq t_k$，那么就称算法为 **R 线性收敛**，同理还有 **R 超线性收敛**和 **R 二次收敛**。同时，这个非负序列还用于收敛速度计算上，成为 $\mathbb{O}(t_k)$。此外，优化算法的复杂度 $N(\epsilon)$，定义为计算出给定精度 ϵ 的解所需的迭代次数或浮点运算次数，比如 $f(x^k) - f(x^*) \leq \epsilon$ 中的 k 的大小。复杂度与收敛速度之间息息相关，一个是从 x 的角度来看，一个是从 $f(x)$ 的角度来看。

举例一个迭代序列 $\{x^k\}$，满足 $f(x^k) - f(x^*) \leq \frac{c}{\sqrt{k}}$, $\forall k > 0$。得到 $N(\epsilon) = \mathbb{O}(\frac{1}{\epsilon^2})$。

8 小结

本章主要讲了优化问题的基本概念以及分类、热点问题等。