

数学建模

杨周旺

中国科学技术大学
数学科学学院

2017年2月

Outline

绪论

微分方程建模

常微分方程模型

偏微分方程模型

最优化建模

统计建模

决策分析

机器学习

人工神经网络

深度学习

总结

绪论

微分方程建模

常微分方程模型

偏微分方程模型

最优化建模

统计建模

决策分析

机器学习

人工神经网络

深度学习

总结

数学

为了解决问题所需要的数学大多已被研究发展起来了，但“数学家们”只注重研究数学的概念以及某些可被证明的结果，而几乎不考虑如何将这些数学结果应用于实际问题。

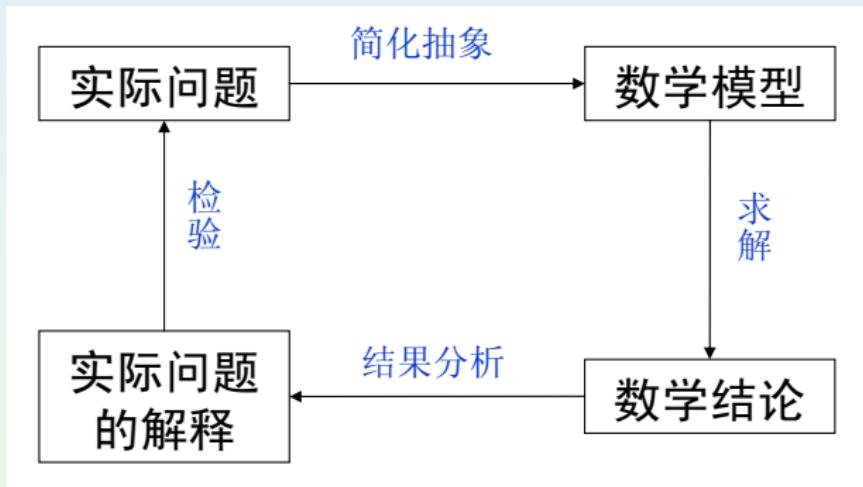
应用数学

数学的另一重要内容就是探索和解决实际问题。为了解决来自工业等领域的实际问题，应用数学(Applied Mathematics)也就应运而生了。

应用数学并不是说哪些数学知识特别有用而哪些又不怎么有用。所谓应用数学可理解为运用数学及其方法去解决实际问题。

数学建模

数学建模属于运用数学解决实际问题的范畴，强调建立数学模型的整个过程。



建模步骤

1. 分析问题
2. 引进合理的假设
3. 建立模型
4. 模型求解及结果的分析
5. 模型的检验
6. 模型的改进

数学模型

数学模型概括地说，就是用数学语言和方法对实际问题的描述和抽象。

依据所用数学方法手段的分类：

- ▶ 微分方程模型
- ▶ 最优化模型
- ▶ 组合图论模型
- ▶ 概率统计模型
- ▶ 几何计算模型
- ▶ 计算机模拟
- ▶

刻画变化

可以用数学有效地刻画现实世界中许多变化的现象，并对它们“数学地”进行研究。这时，我们就能大致或精确地解释、预测、控制、…

我们观察变化时，有兴趣且必要去了解以何种方式发生变化；为什么这样变化；分析不同条件的影响；预测将来的发展趋势；也许还想如何有效控制之。

数学模型常常能帮助我们很好地理解事物的性态，同时它还能使我们对影响这种性态的不同条件数学地进行实验。

例子：酵母的增长

根据酵母培养物增长实验中采集到的数据（见附表），试对酵母生物量的变化过程建模。附表中 n 为时间刻度； p_n 表示 n 个时间刻度后的酵母生物量。

n	0	1	2	3	4	5	6	7	8	9
p_n	9.6	18.3	29.0	47.2	71.1	119.1	174.6	257.3	350.7	441.0
n	10	11	12	13	14	15	16	17	18	
p_n	513.3	559.7	594.8	629.4	640.8	651.1	655.9	659.6	661.8	

例子：酵母的增长（模型一）

假设群体的改变量与当前群体数量是成比例的，即

$$\Delta p_n = (p_{n+1} - p_n) \propto p_n. \quad (1)$$

例子：酵母的增长（模型一）

假设群体的改变量与当前群体数量是成比例的，即

$$\Delta p_n = (p_{n+1} - p_n) \propto p_n. \quad (1)$$

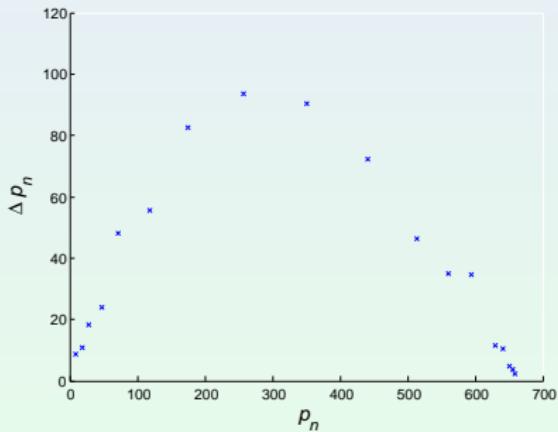
作为假设的一个“合理”解释：设一个酵母细胞分裂成两个要用 k 个时间刻度，那么酵母群体在任一时间刻度区间内平均有 $\frac{1}{k}$ 会发生分裂（即为增长率），在良好的情况下酵母细胞不会死亡，所以 $\frac{1}{k}p_n$ 就是纯增长。因而酵母群体变化量可写成

$$p_{n+1} - p_n = \frac{1}{k} p_n.$$

例子：酵母的增长（模型一）

n	0	1	2	3	4	5	6	7	8	9
p_n	9.6	18.3	29.0	47.2	71.1	119.1	174.6	257.3	350.7	441.0
Δp_n	8.7	10.7	18.2	23.9	48.0	55.5	82.7	93.4	90.3	72.3

n	10	11	12	13	14	15	16	17	18
p_n	513.3	559.7	594.8	629.4	640.8	651.1	655.9	659.6	661.8
Δp_n	46.4	35.1	34.6	11.4	10.3	4.8	3.7	2.2	



例子：酵母的增长（模型二）

如果在一段时间内群体出生数大于死亡数且都与群体数量成比例，则群体将是永远增长的。但是群体的增长受某些资源的限制，当接近极大群体数时，增长就会减缓。在此，酵母培养物在受到制约的区域内生长，从群体数关于时间的图形可见群体数趋于一个极限容量。我们假设极限容量的一个估计值为665，于是考虑以下模型：

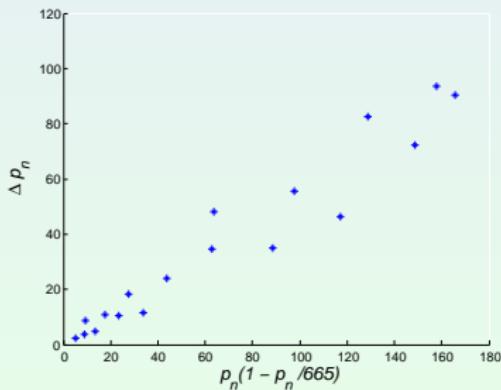
$$\Delta p_n = (p_{n+1} - p_n) = r \cdot p_n \left(1 - \frac{p_n}{665}\right). \quad (2)$$

直观地看，当 p_n 趋于665时，因 $\left(1 - \frac{p_n}{665}\right) \rightarrow 0$ ，从而导致 $\Delta p_n \downarrow$.

例子：酵母的增长（模型二）

为了检测这个模型并得到相应比例 r , 我们画 Δp_n 关于 $q_n = p_n(1 - \frac{p_n}{665})$ 的图形来看它们是否呈现比例关系。

n	0	1	2	3	4	5	6	7	8	9
Δp_n	8.7	10.7	18.2	23.9	48.0	55.5	82.7	93.4	90.3	72.3
q_n	9.46	17.80	27.74	43.85	63.50	97.77	128.76	157.75	165.75	148.55
n	10	11	12	13	14	15	16	17	18	
Δp_n	46.4	35.1	34.6	11.4	10.3	4.8	3.7	2.2		
q_n	117.09	88.63	62.79	33.69	23.32	13.61	8.98	5.36	3.18	



例子：酵母的增长（模型二）

模型求解：

如果我们接受这个模型，可以用数据直线的斜率作为 r 的估计，约为0.5389（最小二乘法），于是就得到模型的方程式

$$\Delta p_n = (p_{n+1} - p_n) = 0.5389 \cdot p_n \left(1 - \frac{p_n}{665}\right). \quad (3)$$

利用迭代式 $p_{n+1} = p_n + 0.5389 \cdot p_n \left(1 - \frac{p_n}{665}\right)$ ，在给定初值 $p_0 = 9.6$ 条件下，我们可以计算该模型的数值解，列于表中作为预测值。

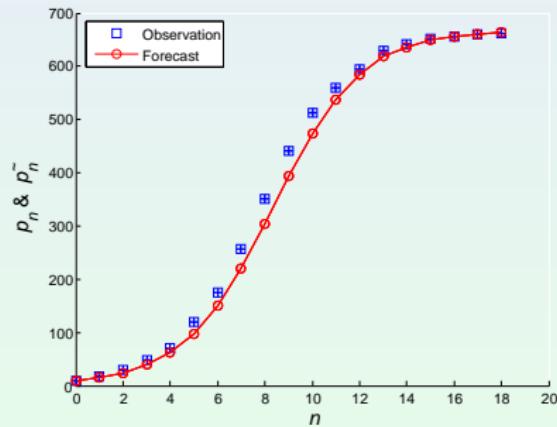
n	0	1	2	3	4	5	6	7	8	9
p_n	9.6	18.3	29.0	47.2	71.1	119.1	174.6	257.3	350.7	441.0
\tilde{p}_n	9.6	14.7	24.3	39.2	62.9	97.1	149.8	219.2	304.2	393.5

n	10	11	12	13	14	15	16	17	18
p_n	513.3	559.7	594.8	629.4	640.8	651.1	655.9	659.6	661.8
\tilde{p}_n	473.5	536.6	584.4	618.2	636.4	648.9	656.3	661.1	664.0

例子：酵母的增长（模型二）

结果检验：

关于时间的预测值和观测值作图示，这样可以验证所建立模型的合理性。



例子：酵母的增长（模型二）

由最小二乘法

$$\min \sum_n (\Delta p_n - r \cdot q_n)^2, \quad (4)$$

可求得 $r = \frac{\sum_n \Delta p_n q_n}{\sum_n q_n^2}.$

本讲作业

作业： 英文单词记忆

有打算出国的同学都要考TOEFL和GRE, 其中背单词是考前一项重要的强化内容。试建立适当的数学模型来描述英文单词的记忆过程，以便给出一些可实施的建议或者执行方案。

绪论

微分方程建模

常微分方程模型

偏微分方程模型

最优化建模

统计建模

决策分析

机器学习

人工神经网络

深度学习

总结

Outline

绪论

微分方程建模

常微分方程模型

偏微分方程模型

最优化建模

统计建模

决策分析

机器学习

人工神经网络

深度学习

总结

常微分方程

一阶微分方程

$$\frac{dy}{dt} = f(t, y) \quad (5)$$

一阶微分方程组

$$\begin{cases} y'_1 &= f_1(t, y_1, y_2, \dots, y_m) \\ y'_2 &= f_2(t, y_1, y_2, \dots, y_m) \\ \vdots & \\ y'_m &= f_m(t, y_1, y_2, \dots, y_m) \end{cases} \quad (6)$$

高阶微分方程

$$y^{(n)} = f(t, y, y', y'', \dots, y^{(n-1)}) \quad (7)$$

高阶微分方程转换为一阶微分方程组

假设给定高阶微分方程

$$y^{(n)} = f(t, y, y', y'', \dots, y^{(n-1)})$$

我们令

$$y_1 = y, y_2 = y', y_3 = y'', \dots, y_n = y^{(n-1)}$$

则引进的新变量 y_1, y_2, \dots, y_n 满足如下一阶微分方程组

$$\begin{cases} y'_1 &= y_2 \\ y'_2 &= y_3 \\ \vdots & \\ y'_n &= f(t, y_1, y_2, \dots, y_n) \end{cases}$$

初值问题

考虑以下初值问题

$$\begin{cases} \frac{dy}{dt} = f(t, y) \\ y(t_0) = y_0 \end{cases} \quad (8)$$

这里 y 是 t 的未知函数，我们希望从(8)中给出的信息去构造它。

第一个等式给出 $y(t)$ 在任意 t 处的斜率 $f(y, t)$ ，其中 f 是指定的；
第二个等式给出 $y(t)$ 在 t_0 处的一个特定值。

例如

$$\begin{cases} y' = (t + \sin y)^2 \\ y(0) = 3 \end{cases}$$

存在性/唯一性

定理（存在性）：若 f 在 $D = \{(t, y) \mid |t - t_0| \leq \alpha, |y - y_0| \leq \beta\}$ 内连续，则对于 $|t - t_0| \leq \min(\alpha, \beta/M)$ 的初值问题(8)有解 $y(t)$ ，其中 M 是 $|f(t, y)|$ 在矩形 D 内的最大值。

定理（唯一性）：若 f 和 $\frac{\partial f}{\partial y}$ 在矩形 D 内连续，则初值问题(8)在区间 $|t - t_0| \leq \min(\alpha, \beta/M)$ 上的解唯一。

定理（第二存在性）：若 f 在 $a \leq t \leq b, -\infty < y < \infty$ 上连续并满足 $|f(t, y_1) - f(t, y_2)| \leq L|y_1 - y_2|$ ，则初值问题(8)在 $[a, b]$ 上有唯一解。

求解方法

变量分离法：

只适用于能将自变量与因变量分离的情形，这一类微分方程可以写成如下形式

$$q(y)dy = p(t)dt. \quad (9)$$

由于方程的左边是仅含 y 的函数，而右边是仅含 t 的函数，因此通过对两边直接积分就得到解。这种方法称为变量分离法，是解析求解微分方程的一种最基本的方法。

数值解

在实际中很少能直接得到微分方程的公式解，给出 $y(t)$ 关于 t 的函数。代之，通常构造下列形式的函数值表：

t_0	t_1	t_2	t_3	\cdots	t_n
y_0	y_1	y_2	y_3	\cdots	y_n

即所谓的数值解。

Taylor级数法

举个例子

$$\begin{cases} y' = t^2 + \cos t - \sin y \\ y(-1) = 3 \end{cases} \quad (10)$$

考虑 y 的Taylor展开级数，我们记之为

$$y(t+h) = y(t) + hy'(t) + \frac{h^2}{2!}y''(t) + \frac{h^3}{3!}y'''(t) + \dots$$

上式中的导数可从微分方程中依据链式法则得到，它们是

$$\begin{aligned} y'' &= 2t - \sin t - y' \cos y \\ y''' &= 2 - \cos t - y'' \cos y + (y')^2 \sin y \\ &\vdots \end{aligned}$$

Taylor级数法

下面是求解初值问题的一个算法，它从 $t_0 = -1$ 开始，步长取 $h = 0.01$. 我们求 $y(t)$ 在区间 $[-1, 1]$ 上的数值解，要执行 $N = 200$ 步。

Input t_0, y_0, h

for $k = 1$ **to** N **do**

$$y' \leftarrow t^2 + \cos t - \sin y$$

$$y'' \leftarrow 2t - \sin t - y' \cos y$$

$$y''' \leftarrow 2 - \cos t - y'' \cos y + (y')^2 \sin y$$

$$y \leftarrow y + h(y' + \frac{h}{2}(y'' + \frac{h}{3}(y''')))$$

$$t \leftarrow t + h$$

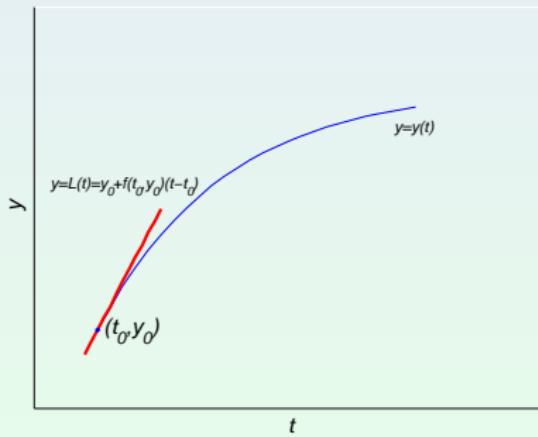
Output $t(k), y(k)$

end do

Euler方法

上述Taylor级数法中，特殊地只取到一阶近似即称为Euler方法，它具有如下形式

$$y(t+h) \approx y(t) + hy'(t) = y(t) + hf(t, y). \quad (11)$$



Runge-Kutta法

从 $y(t + h)$ 的Taylor级数入手

$$y(t + h) = y(t) + hy'(t) + \frac{h^2}{2!}y''(t) + \frac{h^3}{3!}y'''(t) + \dots$$

根据微分方程我们有

$$\begin{aligned} y' &= f \\ y'' &= f_t + f_y y' = f_t + f_y f \\ y''' &= f_{tt} + f_{ty} f + (f_t + f_y f) f_y + f(f_{yt} + f_{yy} f) \\ &\vdots \end{aligned}$$

取展开式中前三项可写成如下形式

$$\begin{aligned} y(t + h) &= y + hf + \frac{h^2}{2}(f_t + f_y f) + O(h^3) \\ &= y + \frac{1}{2}hf + \frac{1}{2}h(f + hf_t + hff_y) + O(h^3) \end{aligned} \tag{12}$$

Runge-Kutta法

又由于

$$f(t + h, y + hf) = f + hf_t + hff_y + O(h^2), \quad (13)$$

$y(t + h)$ 的二阶展开式(12)可改写成

$$y(t + h) = y + \frac{1}{2}hf + \frac{1}{2}hf(t + h, y + hf) + O(h^3) \quad (14)$$

因此，步进求解的迭代公式是

$$y(t + h) = y(t) + \frac{1}{2}hf(t, y) + \frac{1}{2}hf(t + h, y + hf(t, y)) \quad (15)$$

Runge-Kutta法

等价地可写成

$$\begin{cases} F_1 = hf(t, y) \\ F_2 = hf(t + h, y + F_1) \\ y(t + h) = y(t) + \frac{1}{2}(F_1 + F_2) \end{cases} \quad (16)$$

反复地使用这个公式，每次就前进一步求解。它被称为二阶Runge-Kutta法。

Runge-Kutta法

下面是经典的四阶Runge-Kutta方法：

$$y(t+h) = y(t) + \frac{1}{6}(F_1 + 2F_2 + 2F_3 + F_4), \quad (17)$$

其中

$$\begin{cases} F_1 = hf(t, y), \\ F_2 = hf\left(t + \frac{1}{2}h, y + \frac{1}{2}F_1\right), \\ F_3 = hf\left(t + \frac{1}{2}h, y + \frac{1}{2}F_2\right), \\ F_4 = hf(t + h, y + F_3). \end{cases}$$

Matlab数值求解微分方程

The van der Pol equation

$$y'' = \mu * (1 - y^2) * y' + y, \quad (18)$$

where $\mu > 0$ is a scalar parameter.

Rewrite the problem as a system of first-order ODEs. Let

$$y_1 = y, y_2 = y',$$

The resulting system of first-order ODEs is

$$\begin{cases} y'_1 &= y_2 \\ y'_2 &= \mu(1 - y_1^2)y_2 - y_1 \end{cases} \quad (19)$$

Matlab数值求解微分方程

Code the system of first-order ODEs.

The code below represents the van der Pol system in the function, myvdp1. The myvdp1 function is saved as m-file ‘myvdp1.m’ and assumes that $\mu = 1$.

The variables y_1 and y_2 are the entries $y(1)$ and $y(2)$ of a two-element vector.

‘myvdp1.m’:

```
function dydt = myvdp1(t,y)
dydt = [y(2); 1*(1-y(1)*y(1))*y(2)-y(1)];
```

Matlab数值求解微分方程

Apply a solver to the problem.

For the van der Pol system, you can use `ode45` (solver) on time interval [0 20] with initial values $y(0) = y_1(0) = 2$ and $y'(0) = y_2(0) = 0$.

```
[tt,yy] = ode45(@myvdp1,[0 20],[2; 0]);
```

Matlab数值求解微分方程

View the solver output.

You can simply use the plot command to view the solver output.

```
figure(11)
hold on;
plot(tt,yy(:,1),'-',tt,yy(:,2),'- -');
title('Solution of van der Pol Equation, \mu = 1');
xlabel('time t');
ylabel('solution y');
legend('y_1','y_2',2);
```

Matlab数值求解微分方程

Evaluate the Solution at Specific Points.

The numerical methods implemented in the ODE solvers produce a continuous solution over the interval of integration $[a,b]$. You can evaluate the approximate solution, $S(x)$, at any point in $[a,b]$ using the function 'deval' and the structure 'sol' returned by the solver.

```
sol = ode45(@myvdp1,[0 20],[2; 0]);
tv = 1:5;
sv = deval(sol,tv);
```

例子：传染病的传播

传染病传播涉及的因素很多，如已感染病人的多少，易受传染者的多少，传染率的大小，潜伏期的长短，传染途径如何。如果还要考虑人群的流动情况、疾病预防与卫生医疗条件等影响，那么传染病的传播就变得非常复杂。

例子：传染病的传播

传染病传播涉及的因素很多，如已感染病人的多少，易受传染者的多少，传染率的大小，潜伏期的长短，传染途径如何。如果还要考虑人群的流动情况、疾病预防与卫生医疗条件等影响，那么传染病的传播就变得非常复杂。

注：一开始就把众多的因素考虑在内将会使建模者陷入困境而无所适从，倒不如抓住主要因素，简化问题，建立初步的数学模型。将初步模型所得结果与实际情况分析比较，找出不足之处，修改原有假设，逐步建立与实际相吻合的模型。

模型一

假设1 每个已感染病人在单位时间内传染的人数是常数 α_0 ;

假设2 一个人得病后经久不愈，并在传染期内不会死亡。

记 $i(t)$ 为 t 时刻病人数， $i(0) = i_0$ 是最初的染病者个数。那么在 Δt 时间内增加的病人数为

$$i(t + \Delta t) - i(t) = \alpha_0 i(t) \Delta t,$$

于是得到微分方程

$$\begin{cases} \frac{di(t)}{dt} = \alpha_0 i(t) \\ i(0) = i_0 \end{cases} \quad (20)$$

其解为 $i(t) = i_0 e^{\alpha_0 t}$.

模型一

模型结果表明，传染病的传播是按指数函数发展的。

这个结果可能与传染病传播的初期比较吻合，但由其解可以推出当 $t \rightarrow \infty$ 时 $i(t) \rightarrow \infty$ ，这显然是不合实际情况的。问题在于假设不合理，特别是假设1（每个病人单位时间内传染的人数是常数）。在传播初期，传染病人少未被传染者众多，而在传播中后期，传染病人增多未被传染者逐渐减少，因而在不同时期的传染率是不同的。为了吻合实际情况，我们要修改原有假设并建立新的模型。

模型二

假设1 $i(t), s(t)$ 分别表示已感染病人数和未被传染人数，总人数为 $N = i(t) + s(t)$ ；

假设2 每个已感染病人在单位时间内传染的人数与此时未被传染的人数成正比，即 $\alpha_0 = \alpha s(t)$ ；

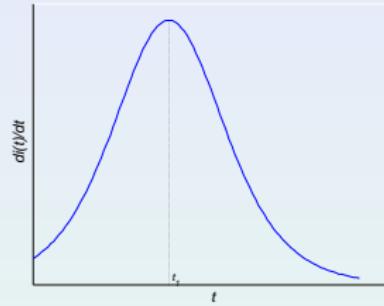
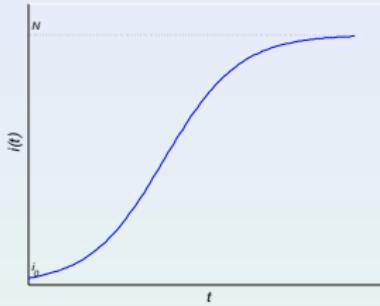
假设3 一个人得病后经久不愈，并在传染期内不会死亡。

基于以上假设可建立微分方程

$$\begin{cases} \frac{di(t)}{dt} = \alpha s(t)i(t) \\ s(t) + i(t) = N \\ i(0) = i_0 \end{cases} \quad (21)$$

由分离变量法解得 $i(t) = \frac{N}{1 + (N/i_0 - 1)e^{-\alpha N t}}$.

模型二



$$\frac{di(t)}{dt} = \frac{\alpha N^2(N/i_0 - 1)e^{-\alpha Nt}}{[1 + (N/i_0 - 1)e^{-\alpha Nt}]^2} \quad (22)$$

令 $\frac{d^2i(t)}{dt^2} = 0$, 得 $\frac{di(t)}{dt}$ 的极大值点 $t_1 = \frac{\ln(N/i_0 - 1)}{\alpha N}$.

模型二

优点：

当传染病的传染强度 α 或总人数 N 增加时， t_1 都将变小，即传染病的高峰来得快，这与实际情况相吻合。同时，如果知道了传染病强度（ α 可由统计数据得出），即可预报传染病高峰到来的时间 t_1 ，这对于传染病的防治工作是有益处的。

模型二

优点：

当传染病的传染强度 α 或总人数 N 增加时， t_1 都将变小，即传染病的高峰来得快，这与实际情况相吻合。同时，如果知道了传染病强度（ α 可由统计数据得出），即可预报传染病高峰到来的时间 t_1 ，这对于传染病的防治工作是有益处的。

缺点：

当 $t \rightarrow \infty$ 时， $i(t) \rightarrow N$ ，即最后人人都要得病。这显然不合实际情况，造成不合理的原因是假设了“人得病后经久不愈”。

模型三

假设患过传染病而痊愈的人具有较长期免疫力，不考虑反复受传染的情形。另外设传染病的潜伏期很短，可忽略不计，也就是说一个人患了病后立即成为传染者。于是我们把居民分为三类：

第一类 由能够把疾病传染给别人的那些染病者组成，用 $i(t)$ 表示 t 时刻的第一类人数。

第二类 由未被传染但可能得病成为传染者的那些人组成，用 $s(t)$ 表示 t 时刻的第二类人数。

第三类 包括患病死去的人，病愈后具有长期免疫力的人，以及被隔离起来的人，用 $r(t)$ 表示 t 时刻的第三类人数。

模型三

具体假设服从如下规则：

- (1) 在所考察的时期内人口总数保持不变，即不考虑出生，其它原因引起的死亡，以及迁入迁出等情况；
- (2) 易受传染者人数 $s(t)$ 的变化率正比于第一类人数 $i(t)$ 与第二类人数 $s(t)$ 的乘积；
- (3) 由第一类向第三类转变的速率与第一类人数成正比。

基于这些规则得微分方程组

$$\begin{cases} \frac{ds(t)}{dt} = -\alpha s(t)i(t) \\ \frac{di(t)}{dt} = \alpha s(t)i(t) - \beta i(t) \\ \frac{dr(t)}{dt} = \beta i(t) \end{cases} \quad (23)$$

其中 α 是传染率， β 是排除率，两者皆为比率常数。

模型三

方程组(23)的三个方程相加得

$$\frac{d}{dt} [s(t) + i(t) + r(t)] = 0,$$

故有

$$s(t) + i(t) + r(t) = \text{常数} = N(\text{人口总数}).$$

模型三

由方程组的前两个方程

$$\begin{cases} \frac{ds(t)}{dt} = -\alpha s(t)i(t) \\ \frac{di(t)}{dt} = \alpha s(t)i(t) - \beta i(t) \end{cases} \quad (24)$$

可得

$$\frac{di}{ds} = \frac{di(t)}{dt} / \frac{ds(t)}{dt} = \frac{\alpha si - \beta i}{-\alpha si} = -1 + \frac{\beta}{\alpha s},$$

解出*i*关于*s*的函数关系

$$i(s) = -s + \frac{\beta}{\alpha} \ln s + C.$$

模型三

当 $t = t_0$ 时, $i(t_0) = i_0, s(t_0) = s_0$. 记 $\rho = \frac{\beta}{\alpha}$, 则有

$$i(s) = i_0 + s_0 - s + \rho \ln \frac{s}{s_0}. \quad (25)$$

因为

$$i'(s) = -1 + \frac{\rho}{s} \begin{cases} < 0, & \text{when } s > \rho \\ = 0, & \text{when } s = \rho \\ > 0, & \text{when } s < \rho \end{cases}$$

所以, 当 $s > \rho$ 时 $i(s)$ 是 s 的减函数, $s < \rho$ 时 $i(s)$ 是 s 的增函数。

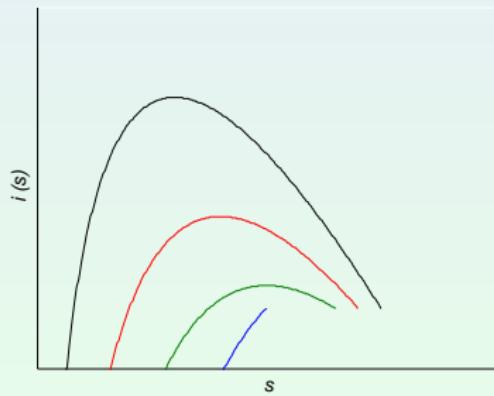
又因为 $i(s=0) = -\infty, i(s_0) = i_0 > 0$, 由中值定理及单调性知:

存在唯一的点 $0 < s_\infty < s_0$, 使得 $i(s_\infty) = 0$.

而对于 $s_\infty < s < s_0$ 时, 有 $i(s) > 0$.

模型三

从方程组中知 $i = 0$ 时, $\frac{ds}{dt} = 0, \frac{di}{dt} = 0$. 所以, $(s_\infty, 0)$ 为方程组的平衡点。如图所示, 当 t 从 t_0 变到 ∞ 时, 点 $(s(t), i(t))$ 沿 s 减少的方向在 $i(s)$ 曲线上移动。如果 $s_0 < \rho$, 则 $i(t)$ 单调减小到 0, $s(t)$ 单调减小到 s_∞ . 如果 $s_0 > \rho$, 则随着 $s(t)$ 减小到 ρ , $i(t)$ 增加, 且当 $s = \rho$ 时 $i(t)$ 达到最大值, 而当 $s(t) < \rho$ 时 $i(t)$ 开始减小。



模型三

由以上分析可得出结论：只当居民中易受传染者的人数超过阀值时，即 $s > \rho$ ，传染病才会蔓延。

我们可以用常识来检验上述结论是否符合实际。当人口密度高，卫生医疗条件不好，隔离不良导致排除率低时，阀值 ρ 较小，传染病就会很快蔓延；反之，人口密度低，社会条件好，有良好的公共卫生医疗条件而排除率高时，阀值 ρ 较大，传染病在有限范围内出现便会很快被扑灭。

模型三

在传染病发生的过程中，很难准确地调查每一天得病的人数。因为只有那些来医院就医的人才能被获知他们得病了，并把他们隔离起来防止传染。因此，统计的记录是每一天新排除者的人数，而不是新得病的人数。为了把数学模型所预示的结果同传染病传播的实际情况进行比较，必须解出(23)中的第三个方程。

$$\frac{dr(t)}{dt} = \beta i(t) = \beta(N - r - s)$$

模型三

因为 $\frac{ds}{dr} = \frac{ds}{dt}/\frac{dr}{dt} = \frac{-\alpha si}{\beta i} = -\frac{s}{\rho}$,

所以 $s(r) = s_0 e^{-r/\rho}$.

于是有

$$\frac{dr(t)}{dt} = \beta(N - r - s_0 e^{-r/\rho})$$

模型三

当 r/ρ 是小量，取Taylor级数 $e^{-r/\rho} = 1 - r/\rho - \frac{1}{2}(r/\rho)^2 + \dots$

近似得

$$\begin{aligned}\frac{dr}{dt} &= \beta \left(N - r - s_0 [1 - r/\rho - \frac{1}{2}(r/\rho)^2] \right) \\ &= \beta \left[N - s_0 + \left(\frac{s_0}{\rho} - 1 \right) r - \frac{s_0}{2\rho^2} r^2 \right]\end{aligned}$$

其解为

$$r(t) = \frac{\rho^2}{s_0} \left[\left(\frac{s_0}{\rho} - 1 \right) + \phi \tanh \left(\frac{1}{2} \phi \beta t - \psi \right) \right], \quad (26)$$

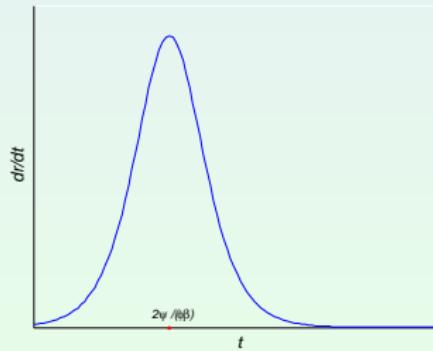
其中 $\phi = \left[\left(\frac{s_0}{\rho} - 1 \right)^2 + \frac{2s_0(N-s_0)}{\rho} \right]^{1/2}$, $\psi = \tanh^{-1} \left(\frac{\frac{s_0}{\rho} - 1}{\phi} \right)$.

模型三

疾病传染曲线：

$$\frac{dr}{dt} = \frac{\beta \rho^2 \phi^2}{2s_0} \cosh^{-2}\left(\frac{1}{2}\phi\beta t - \psi\right). \quad (27)$$

它在 $(t, \frac{dr}{dt})$ 坐标平面上定义了一条对称钟形曲线（称为疾病传染曲线），很好地说明了传染病实际发生的情况。每天报告新病案的数量逐步上升到峰值，然后又减少下来。



Outline

绪论

微分方程建模

常微分方程模型

偏微分方程模型

最优化建模

统计建模

决策分析

机器学习

人工神经网络

深度学习

总结

室内温度振荡

在这里，我们来讨论一个关于室内温度振荡的例子。

室内温度振荡

请问，当代建筑物（砖混结构）的室内温度振荡滞后于外部温度振荡多少个小时？

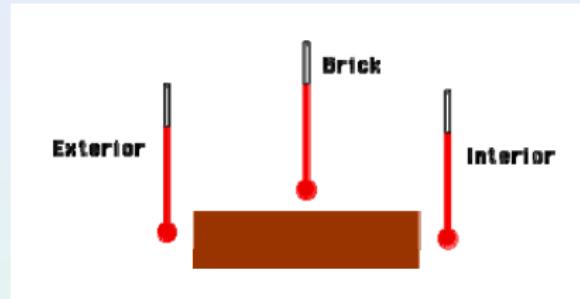
室内温度振荡

Newton's law of cooling:

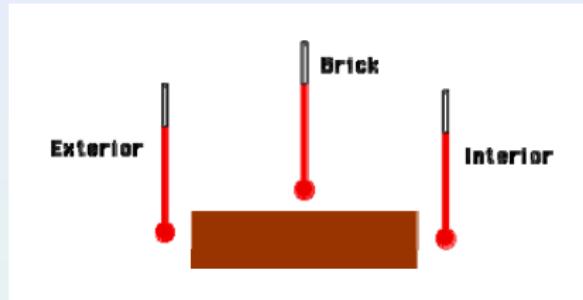
$$T'(t) = k(A(t) - T), \quad (28)$$

其中 $T(t)$ 是物体温度, $A(t)$ 是环境温度, k 是一个正常数。

室内温度振荡

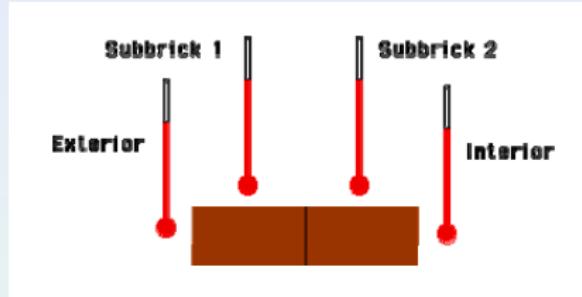


室内温度振荡

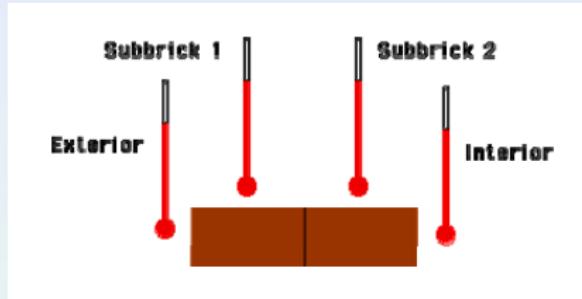


$$\begin{aligned} E(t) &= \sin\left(\frac{\pi t}{12}\right), \\ B'(t) &= k_1(E(t) - B(t)) + k_2(I(t) - B(t)), \\ I'(t) &= k_3(B(t) - I(t)). \end{aligned} \quad (29)$$

室内温度振荡

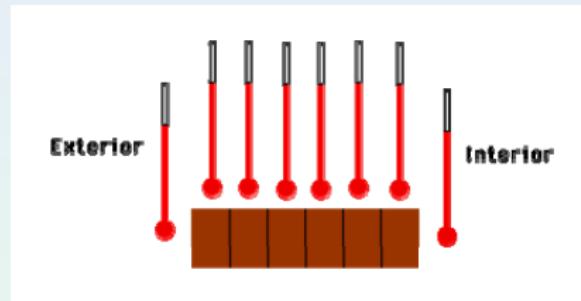


室内温度振荡



$$\begin{aligned} E(t) &= \sin\left(\frac{\pi t}{12}\right), \\ B'_1(t) &= k_1(E(t) - B_1(t)) + k_2(B_2(t) - B_1(t)), \\ B'_2(t) &= k_2(B_1(t) - B_2(t)) + k_3(I(t) - B_2(t)), \\ I'(t) &= k_4(B_2(t) - I(t)). \end{aligned} \quad (30)$$

室内温度振荡



本讲作业

1: 体温速测

体温测量过程大致是一个热传导过程，但由于受到各种环境因素的影响，并不是理想状态下的热传导方程。基于实测数据，对体温测量过程进行数学建模，从而达到体温速测目的（30秒以内），并使误差满足医学精度要求（ ± 0.1 度）。

2: 室内温度振荡

用模型的解来回答一开始提出的问题，即室内温度振荡滞后于外部温度振荡多少个小时？

另外，如何设计墙体能使人在室内生活更舒服？

绪论

微分方程建模

常微分方程模型

偏微分方程模型

最优化建模

统计建模

决策分析

机器学习

人工神经网络

深度学习

总结

最优化

在很多实际应用问题中，从数学上看都是非适定的(ill-posed)，即解不唯一。对于这样的实际问题，人们往往通过制定相应的目标准则，然后从众多的解中选出在一定条件下最好的解。

这些正是最优化理论与方法所研究的内容，本节对最优化的基本概念作一些简要的介绍，并给出最优化建模方法的相关知识。

概述

最优化是一门应用性很强的学科，它讨论的是为找出实际问题的最优解决策略而采取的模型化及其方法。

其过程是，先把待解决的问题用最优化形式描述为在给定的约束条件下找出使某个目标函数达到最大（小）的解，然后再采用数学上严密的算法来求解。

在这里我们强调最优化方法作为数学工具在现实中被广泛应用的事实，大多数有代表性的最优化算法都已有程序库和软件包。但是，有效利用这些成果是以有待解决的问题已被模型化成最优化问题的形式为前提的，而这一过程并非简单。

应用

最优化方法在工业工程、经济管理、交通控制、金融、国防等重要领域中有广泛的应用，并已受到政府部门、科研机构和产业界的高度重视。

数学描述

最优化(optimization)一般是指在某种条件下作出最好的决策，或者是从众多的方案中选出最好的。这种问题经常用下面的数学模型描述：

在给定的约束条件(*constraint*)下，找出一个决策变量(*decision variable*)的值，使得被称为目标函数(*objective function*)的表达愿望尺度的函数值达到最小或最大。

最优化问题

一般说来决策变量有多个，因此用 n 维向量 $\mathbf{x} = (x_1, \dots, x_n)^T$ 来表示，于是把问题写成如下形式

$$\begin{aligned} & \min \quad f(\mathbf{x}) \\ \text{s.t.} \quad & \mathbf{x} \in S \subset \mathbb{R}^n. \end{aligned} \tag{31}$$

在此，目标函数 f 是定义在包含 S 的适当集合上的实值函数。进一步， S 是该问题变量 \mathbf{x} 的可取值之集合，称为问题的可行域(feasible region)。

最优化问题的解

满足约束条件 $\mathbf{x} \in S$ 的 \mathbf{x} 称为问题的可行解(feasible solution), 如果可行解 $\mathbf{x}^* \in S$ 进一步满足

$$f(\mathbf{x}^*) \leq f(\mathbf{x}), \forall \mathbf{x} \in S. \quad (32)$$

则称 \mathbf{x}^* 为问题(31)的全局最优解(global optimal solution). 另外, 在包含可行解 $\mathbf{x}^* \in S$ 的适当邻域 $U(\mathbf{x}^*)$ 里, 成立

$$f(\mathbf{x}^*) \leq f(\mathbf{x}), \forall \mathbf{x} \in S \cap U(\mathbf{x}^*). \quad (33)$$

此时称 \mathbf{x}^* 为问题(31)的局部最优解(local optimal solution).

不少问题的目标函数或约束条件可能很复杂, 要找出全局最优解非常困难, 这时我们的目标就会是求出局部最优解。

最优化问题的分类

- 如果可行域 $S = \mathbb{R}^n$, 则称为无约束最优化问题:

$$\min_{\mathbf{x} \in \mathbb{R}^n} f(\mathbf{x}) \quad (34)$$

- 带约束最优化问题通常可写为:

$$\begin{aligned} & \min f(\mathbf{x}) \\ \text{s.t. } & c_i(\mathbf{x}) = 0, \quad i \in E, \\ & c_i(\mathbf{x}) \geq 0, \quad i \in I. \end{aligned} \quad (35)$$

这里 $c_i(\mathbf{x})$ 是约束函数, E 和 I 分别是等式约束的指标集和不等式约束的指标集。

最优化问题的分类

- 当目标函数和约束函数均为线性函数时，问题称为线性规划(Linear Programming);
- 当目标函数与约束函数中至少有一个是变量 x 的非线性函数时，问题称为非线性规划(Nonlinear Programming)。
- 此外，根据决策变量、目标函数和要求的不同，最优化还分成了整数规划、动态规划、网络规划、非光滑规划、随机规划、几何规划、多目标规划等分支。

最优化条件

最优化条件：问题的最优解所满足的必要或者充分条件。它为最优化问题求解算法的设计、分析提供必不可少的理论基础。

无约束问题的极值条件

一阶必要条件：设目标函数 $f(\mathbf{x})$ 在点 $\bar{\mathbf{x}}$ 处可微，若 $\bar{\mathbf{x}}$ 是局部极小点，则 $\nabla f(\bar{\mathbf{x}}) = 0$.

二阶必要条件：设目标函数 $f(\mathbf{x})$ 在点 $\bar{\mathbf{x}}$ 处二次可微，若 $\bar{\mathbf{x}}$ 是局部极小点，则 $\nabla f(\bar{\mathbf{x}}) = 0$, 并且Hessian矩阵 $\nabla^2 f(\bar{\mathbf{x}}) \geq 0$.

二阶充分条件：设目标函数 $f(\mathbf{x})$ 在点 $\bar{\mathbf{x}}$ 处二次可微，若 $\nabla f(\bar{\mathbf{x}}) = 0$ 且 $\nabla^2 f(\bar{\mathbf{x}}) > 0$, 则 $\bar{\mathbf{x}}$ 是局部极小点。

充要条件：设 $f(\mathbf{x})$ 是定义在 \mathbb{R}^n 上的可微凸函数，则 $\bar{\mathbf{x}}$ 是整体极小点（全局最优解）的充要条件是 $\nabla f(\bar{\mathbf{x}}) = 0$.

最优化条件

约束问题的最优化条件

Kuhn-Tucker必要条件：设 $\bar{\mathbf{x}}$ 为约束问题(35)的可行点，记 $I_e = \{i \in I \mid c_i(\bar{\mathbf{x}}) = 0\}$, f 和 $c_i(i \in I_e)$ 在点 $\bar{\mathbf{x}}$ 可微， $c_i(i \in I \setminus I_e)$ 在点 $\bar{\mathbf{x}}$ 连续， $c_i(i \in E)$ 在点 $\bar{\mathbf{x}}$ 连续可微，向量集 $\{\nabla c_i(\bar{\mathbf{x}}) \mid i \in E \cup I_e\}$ 线性无关。如果 $\bar{\mathbf{x}}$ 是局部最优解，则存在数 $\lambda_i \geq 0(i \in I)$ 和 $\mu_i(i \in E)$ 使得

$$\nabla f(\bar{\mathbf{x}}) - \sum_{i \in I} \lambda_i \nabla c_i(\bar{\mathbf{x}}) - \sum_{i \in E} \mu_i \nabla c_i(\bar{\mathbf{x}}) = 0. \quad (36)$$

最优化条件

定义Lagrange函数 $L(\mathbf{x}, \lambda, \mu) = f(\mathbf{x}) - \sum_{i \in I} \lambda_i c_i(\mathbf{x}) - \sum_{i \in E} \mu_i c_i(\mathbf{x}).$

若 $\bar{\mathbf{x}}$ 为问题局部最优解，则存在乘子向量 $\bar{\lambda} \geq 0, \bar{\mu}$ 使得 $\nabla_{\mathbf{x}} L(\bar{\mathbf{x}}, \bar{\lambda}, \bar{\mu}) = 0.$

此时，一阶必要条件可以表达为

$$(K-T) \begin{cases} \nabla_{\mathbf{x}} L(\mathbf{x}, \lambda, \mu) = 0, \\ c_i(\mathbf{x}) = 0, i \in E, \\ c_i(\mathbf{x}) \geq 0, i \in I, \\ \lambda_i c_i(\mathbf{x}) = 0, i \in I, \\ \lambda_i \geq 0, i \in I. \end{cases} \quad (37)$$

求解算法

理论上可以用最优化条件求“非线性规划”的最优解，但在实践中并不切实可行。

一般情况下，求解最优化问题所用的计算方法最常见的是“迭代下降算法”。

算法映射

算法映射：算法 \mathcal{A} 是定义在空间 \mathcal{X} 上的点到集的映射，即对每一点 $\mathbf{x}^{(k)} \in \mathcal{X}$ ，经算法 \mathcal{A} 作用后产生一个点集 $\mathcal{A}(\mathbf{x}^{(k)}) \subset \mathcal{X}$ ，任意选择一个点 $\mathbf{x}^{(k+1)} \in \mathcal{A}(\mathbf{x}^{(k)})$ 作为 $\mathbf{x}^{(k)}$ 的后续点。

解集合

为研究算法的收敛性，首先要明确解集合的概念。

在许多情况下，要使算法产生的点列收敛于全局最优解是非常困难的。因此，一般把满足某些条件的点集定义为解集合。当迭代点属于这个集合时，就停止迭代。例如，在无约束最优化问题中，可以定义解集合为 $\Omega = \{\bar{x} \mid \|\nabla f(\bar{x})\| = 0\}$ ；在约束最优化问题中，解集合取为 $\Omega = \{\bar{x} \mid \bar{x} \text{是K-T点}\}$.

下降函数

下降算法总是与某个函数在迭代中函数值的减小联系在一起，因此需要给出下降函数的概念。

设 $\Omega \subset \mathbf{X}$ 为解集合， \mathcal{A} 为 \mathbf{X} 上的一个算法映射， $\psi(\mathbf{x})$ 是定义在 \mathbf{X} 上的连续实函数，若满足

当 $\mathbf{x} \notin \Omega$ 且 $\mathbf{y} \in \mathcal{A}(\mathbf{x})$ 时， $\psi(\mathbf{y}) < \psi(\mathbf{x})$

当 $\mathbf{x} \in \Omega$ 且 $\mathbf{y} \in \mathcal{A}(\mathbf{x})$ 时， $\psi(\mathbf{y}) \leq \psi(\mathbf{x})$

则称 ψ 是关于解集合 Ω 和算法 \mathcal{A} 的下降函数。

算法收敛性

1. 算法收敛
2. 实用收敛准则
3. 收敛速度
4. 时间复杂性

迭代方法

最优化方法通常采用迭代方法求问题的最优解，其基本思想是：给定一个初始点 $\mathbf{x}^{(0)} \in \mathbb{R}^n$ ，按照某一迭代规则产生一个点列 $\{\mathbf{x}^{(k)}\}$ ，使得当 $\{\mathbf{x}^{(k)}\}$ 是有穷点列时，其最后一个点是最优化模型问题的最优解，当 $\{\mathbf{x}^{(k)}\}$ 是无穷点列时，它有极限点且其极限点是最优化模型问题的最优解。

一个好的算法应具备的典型特征是：迭代点 $\mathbf{x}^{(k)}$ 能稳定地接近局部极小点 \mathbf{x}^* 的小领域，然后迅速收敛于 \mathbf{x}^* 。一般地，我们要证明迭代点列 $\mathbf{x}^{(k)}$ 的聚点（即子列的极限点）为一局部极小点。在实际计算中，当指定的某种收敛准则满足时，迭代即终止。

搜索方向与步长因子

设 $\mathbf{x}^{(k)}$ 为第 k 次迭代点， $\mathbf{d}^{(k)}$ 为第 k 次搜索方向， α_k 为第 k 次步长因子，则第 $k+1$ 次迭代为：

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha_k \mathbf{d}^{(k)}. \quad (38)$$

从上述迭代格式可以看出，不同的搜索方向和不同的步长因子构成不同的方法。在最优化方法中，搜索方向 $\mathbf{d}^{(k)}$ 一般选取的是函数 f 在 $\mathbf{x}^{(k)}$ 处的下降方向，即 $\mathbf{d}^{(k)}$ 满足

$$\nabla f(\mathbf{x}^{(k)})^T \mathbf{d}^{(k)} < 0. \quad (39)$$

迭代算法的基本结构

- (a) 给定初始点 $\mathbf{x}^{(0)}$
- (b) 计算搜索方向 $\mathbf{d}^{(k)}$, 即构造函数 f 在 $\mathbf{x}^{(k)}$ 点处的下降方向作为搜索方向;
- (c) 确定步长因子 α_k , 使目标函数值有某种程度的下降;
- (d) 迭代, 令 $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha_k \mathbf{d}^{(k)}$.
- (e) 判断停机准则, 若 $\mathbf{x}^{(k+1)}$ 满足某种终止条件, 则停止迭代, 得到近似最优解 $\bar{\mathbf{x}} = \mathbf{x}^{(k+1)}$. 否则, 返回(b)重复以上步骤。

构造下降方向

例子: $\min_{\mathbf{x} \in \mathbb{R}^n} f(\mathbf{x})$

$$f(\mathbf{x}) = f(\mathbf{x}^{(k)}) + \nabla f(\mathbf{x}^{(k)})^T (\mathbf{x} - \mathbf{x}^{(k)}) + O(\|\mathbf{x} - \mathbf{x}^{(k)}\|^2) \quad (40)$$

取负梯度方向

$$\mathbf{d}^{(k)} = -\nabla f(\mathbf{x}^{(k)}),$$

则当 α_k 足够小时, 总能使

$$f(\mathbf{x}^{(k)} + \alpha_k \mathbf{d}^{(k)}) < f(\mathbf{x}^{(k)}).$$

步长策略

一维搜索(Line Search):

步长因子的确定一般归结为解一维最优化问题

$$\min_{\alpha \geq 0} \psi(\mathbf{x}^{(k)} + \alpha \mathbf{d}^{(k)}) \quad (41)$$

梯度法及其方法类

构造的搜索方向为

$$\mathbf{d}^{(k)} = -H_k \nabla f(\mathbf{x}^{(k)}),$$

牛顿法与拟牛顿法

$$\begin{aligned} f(\mathbf{x}) = & \quad f(\mathbf{x}^{(k)}) + \nabla f(\mathbf{x}^{(k)})^T (\mathbf{x} - \mathbf{x}^{(k)}) \\ & + \frac{1}{2} (\mathbf{x} - \mathbf{x}^{(k)})^T \nabla^2 f(\mathbf{x}^{(k)}) (\mathbf{x} - \mathbf{x}^{(k)}) + O(\|\mathbf{x} - \mathbf{x}^{(k)}\|^3) \end{aligned} \tag{42}$$

取搜索方向

$$\mathbf{d}^{(k)} = -H_k^{-1} \nabla f(\mathbf{x}^{(k)}),$$

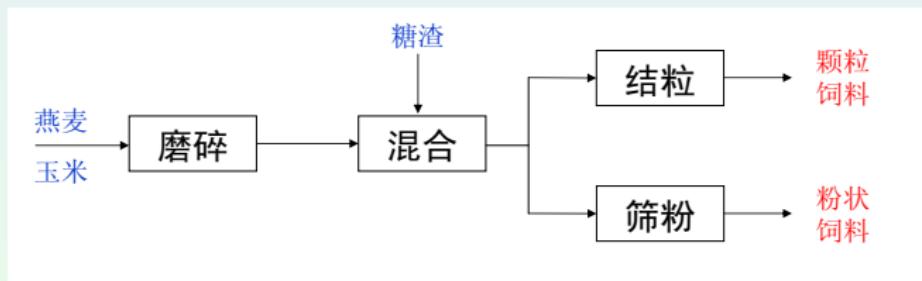
其中 $H_k = \nabla^2 f(\mathbf{x}^{(k)})$ 为函数 f 在 $\mathbf{x}^{(k)}$ 点处的 Hesse 矩阵.

信赖域方法

$$\begin{aligned} \min \quad & Q^k(\mathbf{d}) = \nabla f(\mathbf{x}^{(k)})^T \mathbf{d} + \frac{1}{2} \mathbf{d}^T \nabla^2 f(\mathbf{x}^{(k)}) \mathbf{d} \\ \text{s.t.} \quad & \mathbf{d}^T \mathbf{d} \leq e_k^2. \end{aligned} \quad (43)$$

饲料配方

某饲料公司要生产两种类型的动物饲料：粉状饲料和颗粒饲料。生产这些饲料所需的原料由燕麦、玉米和糖渣组成。首先需要将这些原料（除糖渣外）磨碎，然后将所有原料混合形成饲料半成品。在最后一个生产工序中，将半成品制成颗粒状或粉末状，从而得到最终的饲料产品。



饲料配方

饲料产品都要满足一些营养成分需求。表1列出了各种原料的蛋白质，脂肪和纤维含量百分比，以及最终产品的相应含量要求。

Table: 营养成分含量百分比

原料	蛋白质	脂肪	纤维素
燕麦	13.6	7.1	7.0
玉米	4.1	2.4	3.7
糖渣	5.0	0.3	25.0
含量要求	≥ 9.5	≥ 2.0	≤ 6.0

饲料配方

各种原料的可用量也有限制。表2给出了各种原料的可用量以及对应的价格。

Table: 原料可用量与价格

原料	可用量（千克）	价格（元/千克）
燕麦	11900	1.3
玉米	23500	1.7
糖渣	750	1.2

饲料配方

表3列出了各道工序的成本。

Table: 加工成本

磨碎	混合	结粒	筛粉
2.5	0.5	4.2	1.7

如果每天需求量为9吨颗粒饲料，12吨粉状饲料，则各种原料应分别使用多少进行混合才能使得总成本最低？

模型表达

在这里，燕麦、玉米、糖渣的使用量为决策变量，并分别设为 x_1 千克， x_2 千克， x_3 千克。

饲料配制最优化模型：

$$\begin{aligned} \min \quad & (1.3x_1 + 1.7x_2 + 1.2x_3) + 2.5(x_1 + x_2) + 0.5(x_1 + x_2 + x_3) \\ \text{s.t.} \quad & 13.6x_1 + 4.1x_2 + 5.0x_3 \geq 9.5(x_1 + x_2 + x_3), \\ & 7.1x_1 + 2.4x_2 + 0.3x_3 \geq 2.0(x_1 + x_2 + x_3), \\ & 7.0x_1 + 3.7x_2 + 25.0x_3 \leq 6.0(x_1 + x_2 + x_3), \\ & x_1 + x_2 + x_3 \geq 9000 + 12000, \\ & 0 \leq x_1 \leq 11900, \\ & 0 \leq x_2 \leq 23500, \\ & 0 \leq x_3 \leq 750. \end{aligned} \tag{44}$$

线性规划[linprog]

Solve a linear programming problem

$$\begin{aligned} \min_x \quad & f^T x \\ \text{s.t.} \quad & A x \leq b \\ & A_{eq} x = b_{eq} \\ & lb \leq x \leq ub \end{aligned} \tag{45}$$

where f , x , b , b_{eq} , lb , and ub are vectors and A and A_{eq} are matrices.

```
[x,fval,exitflag,output,lambda] =  
linprog(f,A,b,Aeq,beq,lb,ub,x0,options)
```

模型求解

$f = [4.3; 4.7; 1.7];$

$A = [-4.1, 5.4, 4.5; -5.1, -0.4, 1.7; 1, -2.3, 19; -1, -1, -1];$

$b = [0; 0; 0; -21000];$

$lb = zeros(3, 1);$

$ub = [11900; 23500; 750];$

[x,fval,exitflag,output,lambda] = linprog(f,A,b,[],[],lb,ub);

模型结果

生产出需求的9吨颗粒饲料和12吨粉状饲料，最低成本为150868元。详细的结果如下所示：

Table: 饲料生产的最佳方案

原料	燕麦（千克）	玉米（千克）	糖渣（千克）
使用量	11897	8678.9	424.47

无约束最优化模型求解

Find a minimum of an unconstrained multi-variable function

$$\min_x f(x) \quad (46)$$

where x is a vector and $f(x)$ is a function that returns a scalar.

fminunc

```
[x,fval,exitflag,output,grad,hessian] =  
fminunc(fun,x0,options)
```

约束最优化模型求解

Find a minimum of a constrained nonlinear multi-variable function

$$\begin{aligned} \min_x \quad & f(x) \\ \text{s.t.} \quad & c(x) \leq 0 \\ & c_{eq}(x) = 0 \\ & Ax \leq b \\ & A_{eq}x = b_{eq} \\ & lb \leq x \leq ub \end{aligned} \tag{47}$$

where x , b , b_{eq} , lb , and ub are vectors, A and A_{eq} are matrices, $c(x)$ and $c_{eq}(x)$ are functions that return vectors, and $f(x)$ is a function that returns a scalar. $f(x)$, $c(x)$, and $c_{eq}(x)$ can be nonlinear functions.

fmincon

```
[x,fval,exitflag,output,lambda,grad,hessian] =  
fmincon(fun,x0,A,b,Aeq,beq,lb,ub,nonlcon,options)
```

Matlab程序实例

实例程序（附件）：

1. 'opt_linprog.m'
2. 'opt_fminunc.m', 'myobjfun.m'
3. 'opt_fmincon.m', 'myfun.m', 'myconf.m'

本讲作业

作业：电梯分组运行的优化

在建构高楼时，应该考虑电梯的安装运行问题。对多个电梯实行分组运行（把楼层分为若干组，每个电梯只服务于某组楼层），一般可提高电梯的服务效率。

在电梯分组设置中应该考虑哪些主要因素？试针对某大楼采集相关数据并建立电梯分组运行的优化模型。

绪论

微分方程建模

常微分方程模型

偏微分方程模型

最优化建模

统计建模

决策分析

机器学习

人工神经网络

深度学习

总结

回归分析

Logistic回归

聚类分析

物以类聚。现实中，我们通常按照事物所具有的特性进行分类。分类学是人类认识世界的基础科学。

早期的分类学中，人们主要靠经验和专业知识进行定性的分类。随着科学技术的发展，分类的要求越来越高，以致只凭经验和专业知识无法确切地进行分类。于是数学工具逐渐被引用到分类学中，形成了定量分类学。

聚类分析

聚类分析是用多元统计理论进行分类的一种方法。

聚类分析的发展起步较晚，在方法和理论上尚有诸多不完善之处。但由于它能解决很多实际问题，因而受到人们的格外重视。现在已广泛应用于生物，地质，图像识别，信息处理，行为分析等领域。

聚类分析

聚类分析是通过分类对象的各种属性指标进行分类的。对象的属性指标一般如下三种描述尺度之一：

- ▶ 1. 名义尺度(Nominal)

用这种指标表示时既没有数量也没有次序关系，如性别中的男和女，天气的晴、阴、雨等。

- ▶ 2. 有序尺度(Ordinal)

用这种指标表示时没有明确的数量，只有次序关系，如产品质量的等级划分，一等品、二等品、三等品...

- ▶ 3. 间隔尺度(Interval)

该指标用连续实值表示，如长度、重量等。

聚类分析

当我们对研究对象进行分类时，总要选定一种度量，用以衡量两个对象间的接近程度或相似程度。把相互接近的、类似的对象放在一起，归结为一类，而把相互疏远的、差异较大的对象分在不同的类中。

一般地，这种度量需要根据具体问题来决定。

聚类分析

在实际问题中，如果有 n 个样品，每个样品测量了 p 个属性指标。以 x_{ij} 表示第 i 个样品的第 j 个指标值。

Table: 样品指标数据表

样品	指标			
	1	2	…	p
1	x_{11}	x_{12}	…	x_{1p}
2	x_{21}	x_{22}	…	x_{2p}
⋮	⋮	⋮		⋮
n	x_{n1}	x_{n2}	…	x_{np}

聚类分析

我们需要根据 p 个指标值对 n 个样品进行分类。

可将每个样品看成 p 维空间中的一个点，然后在 p 维空间中定义适当的距离。

聚类分析

为了消除各个指标所用量纲的影响，以保证各指标量在分析中处于同等地位，常对数据进行标准化变换。

$$\bar{x}_{\cdot j} = \frac{1}{n} \sum_{i=1}^n x_{ij}, \quad j = 1, 2, \dots, p$$

$$s_j = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_{ij} - \bar{x}_{\cdot j})^2}, \quad j = 1, 2, \dots, p$$

分别表示第 j 指标的均值和标准差。那么，基于标准差的标准化变换是指

$$\tilde{x}_{ij} = \frac{x_{ij} - \bar{x}_{\cdot j}}{s_j}, \quad i = 1, 2, \dots, n, \quad j = 1, 2, \dots, p.$$

聚类分析

距离定义

我们以 d_{ik} 表示第 i 个样品 $\mathbf{x}_i = (x_{i1}, x_{i2}, \dots, x_{ip})$ 和第 k 个样品 $\mathbf{x}_k = (x_{k1}, x_{k2}, \dots, x_{kp})$ 之间的距离。

对标准化后的指标量（仍记为 x_{ij} ），最常用的距离有以下几种：

Minkowski 距离：

$$d_{ik}(q) = \left[\sum_{j=1}^p |x_{ij} - x_{kj}|^q \right]^{1/q}, \quad q > 0.$$

聚类分析

距离定义

我们以 d_{ik} 表示第 i 个样品 $\mathbf{x}_i = (x_{i1}, x_{i2}, \dots, x_{ip})$ 和第 k 个样品 $\mathbf{x}_k = (x_{k1}, x_{k2}, \dots, x_{kp})$ 之间的距离。

对标准化后的指标量（仍记为 x_{ij} ），最常用的距离有以下几种：

Mahalanobis 距离：

$$d_{ik}^2(M) = (\mathbf{x}_i - \mathbf{x}_k) \Sigma^{-1} (\mathbf{x}_i - \mathbf{x}_k)^T,$$

$$\text{其中 } \Sigma = (\sigma_{\ell j})_{p \times p}, \sigma_{\ell j} = \frac{1}{n-1} \sum_{i=1}^n (x_{i\ell} - \bar{x}_{.\ell})(x_{ij} - \bar{x}_{.j}).$$

聚类分析

当前最常用的是系统聚类法，其基本思想是逐步将距离近的类合并在一起。

首先将 n 个样品看成 n 类（类的初始化，一个样品是一类），然后将性质最接近的两类合并为一个新类，我们即得到 $n - 1$ 类，再从中找出最接近的两类加以合并变成了 $n - 2$ 类，如此下去，最终所有的样品全在一类。

把上述合并类的过程画成一张聚类图，便可决定分多少类，每类各包含什么样品。

聚类分析

系统聚类法的步骤：

- (1) 计算 n 个样品两两之间的距离 $\{d_{ik}\}$, 记作 $D = (d_{ik})_{n \times n}$. 构造 n 个初始类, 每类只包含一个样品。
- (2) 合并距离最近的两类为一个新类, 类的数目减一。
- (3) 若类的数目等于1, 转到步骤(4); 否则, 计算当前各类之间的距离, 返回到步骤(2).
- (4) 画出聚类图。
- (5) 决定类的个数和类。

聚类分析

一个简单例子：设有5个样品，其指标只有一个变量，分别是{1, 2, 4.5, 6, 8}，试将它们分类。

本讲作业

作业：聚类分析在图像分割中的应用

请运用聚类分析方法对给定图像的目标物体进行分割。在这里，推荐使用教室背景的班级学生上课照片或者一段交通车辆监控视频来测试所提方法的有效性。

注：建模过程应包括问题描述、算法和编程实现等。

绪论

微分方程建模

常微分方程模型

偏微分方程模型

最优化建模

统计建模

决策分析

机器学习

人工神经网络

深度学习

总结

决策分析

人们时常要在不确定的条件下，从多个可选方案中决定自己的抉择，以期获得某种最佳效果。如投资规划，在多个项目中选择一种或者几种进行投资，以期获得最大的经济效益；

决策分析

随着社会的发展，人们遇到的决策问题越来越复杂，凭决策者个人的学识和经验做决定，已不能适应当前的情况了。对于一些影响重大的问题，单靠原有的经验进行决策往往易产生不良后果。为了避免这种情况发生，人们逐渐归纳和总结了一些相对客观而科学的决策思想和方法。

决策分析

决策分析是提供一种思考模式和理论指导，通过表达和交流的思维过程，帮助决策者解释和分析所面临的问题，特别是把复杂问题分解成若干单独因素分别进行定性和定量研究。

决策分析的基本概念

状态空间

状态是指决策者所面临的决策问题中不可预知或无法控制的不确定因素。比如，在防震抗灾计划制定中，地震和不发生地震是决策者不能完全预知的两个状态；在决定是否开发一种新产品投放市场时，市场对该新产品的需求量是无法控制的不确定变量，可把它简单地归结为“需求量大”、“需求量一般”、“需求量小”三个状态。

状态的全体称为状态空间。假定有且仅有一个状态会发生（为真），即两个不同状态不可能同时发生。状态空间的子集称为事件，事件发生（为真）是指该事件中的一个状态发生。状态的发生独立于决策者的意志，不依赖决策者的选择。

决策分析的基本概念

策略空间

在决策问题中，决策者往往面临着众多的不确定因素，可供选择的方案（策略、抉择、行动）也不唯一。所有可供选择的方案全体称为策略空间，决策的目的就是从策略空间中选出某些个“最佳”的方案。

决策分析的基本概念

后果空间

决策者在采取了一项策略后，无论哪个状态发生，都会导致某一后果。所有可能产生的后果全体叫做后果空间。

由于决策者不能预知哪个状态会发生，因而也无法知道究竟哪个后果会出现。决策者知道的是在某个状态发生的条件下选择某项策略将会对应什么样的后果。

决策分析的基本概念

决策理论的表达方式是把状态和后果作为两个基本元素，而把策略作为一种作用，定义为从状态空间到后果空间的一个映射。

记 S 为状态空间， X 为后果空间，则映射 $f : S \rightarrow X$ 看作是一项策略， $F = \{f \mid f : S \rightarrow X\}$ 就是策略空间。

决策分析的基本概念

主观概率

决策者要在不确定的环境下作出决策，但怎样描述不确定性呢？人们往往用概率来研究自然和社会中的不确定性，但必须扩展概率的内涵。

在决策分析中度量不确定性的概率与普通概率统计中随机事件的概率涵义有所不同（这里的事件一般不能在相同条件下重复试验），而是将它看做决策者在所掌握的信息条件下对有关事件发生的主观信任程度，因此称为主观概率。

决策分析的基本概念

偏好关系

偏好是指决策者衡量策略和后果优劣的价值观，用数学中的一个二元关系来描述，称为偏好关系。

决策分析的基本概念

效用函数

设 D 是一个非空集， \succ 是 D 上的一个序关系，若有实函数 u 满足

$$A \succ B \Leftrightarrow u(A) > u(B), \forall A, B \in D$$

则称 u 为 D 上的效用函数。如果 D 是策略空间， u 就是关于策略的效用函数。如果 D 是后果空间， u 就是关于后果的效用函数。

效用函数不仅能定性地反映决策者对于策略或者后果的偏好关系，而且还定量地刻画了偏好的相对强度。

决策分析的基本概念

期望效用

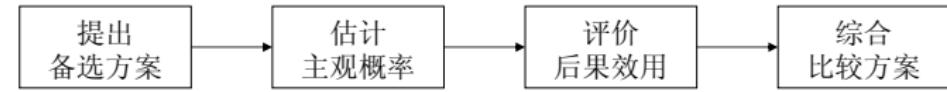
设 S, X, F 分别代表状态空间，后果空间和策略空间， u 是 X 上的效用函数， P 是 S 上的概率测度，则对于策略 f ，

$$U(f) = \int_S u(f(s))dP$$

称为策略 f 的期望效用， U 又看做是 F 上的效用函数。

决策分析的步骤

决策分析是一个复杂的系统分析过程。为了强调其基本原理，可将决策分析过程简单地分解为四个步骤：形成决策问题；度量主观概率；评价决策后果的效用；计算期望效用比较策略的优劣。



决策分析的应用

例子：对于某项新技术要在生产中投入使用，现有两个方案可供选择：

方案A 从国外引进成套设备，若输入为高质量的原材料，可获利3000万元；若是低质量的原材料，将会亏损1000万元。

方案B 采用国产设备，若输入为高质量的原材料，可获利1200万元；若是低质量的原材料，还能获利300万元。

已知原材料的两种状态概率为： $P_r(S_1) = 0.35$, $P_r(S_2) = 0.65$, 其中 S_1, S_2 分别表示高质量原料和低质量原料。

决策分析的应用

为了更确切地了解将用到的原料质量，以便对设备A和B的选择进行决策，有关方面拟抽样检验原材料。

检验结果有下列两种情况： O_1 质量高； O_2 质量低。而检验结果的可信赖度由以下条件概率给出：

$$P_r(O_1|S_1) = 0.95, P_r(O_1|S_2) = 0.05;$$

$$P_r(O_2|S_1) = 0.05, P_r(O_2|S_2) = 0.95.$$

另外，进行抽样检验的费用是30万元，请问应该如何决策该新技术投入生产的方式。

期望效用理论

决策分析的基本原理是决策者根据自己的价值观念对后果和策略的优劣进行比较选择。决策者的价值观念具有两层涵义：一方面是定性的比较，例如对于方案 f, g, h , 喜欢 f 甚于 g , 喜欢 g 甚于 h ；另一方面是相对定量的比较，例如相对于 h 来说，喜欢 f 的程度比喜欢 g 的程度强烈。习惯上，前者叫“偏好”，可用称为“序”的二元关系来描述；后者叫“效用”，可用称为“效用函数”的实值函数来刻画。

期望效用理论

设 D 是一个非空集合， $D \times D = \{(x, y) \mid x, y \in D\}$ 的子集 R 称为 D 上的一个二元关系。对于 $x, y \in D$ ，若 $(x, y) \in R$ ，则称 x 与 y 有关系，记作 xRy ；若 $(x, y) \notin R$ ，则称 x 与 y 没有关系，记作 $x\bar{R}y$ 。

期望效用理论

二元关系的常见性质：

1. 自反性： xRx
2. 非自反性： $x\bar{R}x$
3. 对称性：若 xRy , 则 yRx
4. 非对称性：若 xRy , 则 $y\bar{R}x$
5. 传递性：若 xRy 且 yRz , 则 xRz
6. 负传递性：若 $x\bar{R}y$ 且 $y\bar{R}z$, 则 xRz
7. 连通性： xRy 或者 yRx
8. 弱连通性：若 $x \neq y$, 则有 xRy 或者 yRx

期望效用理论

一个二元关系只能有上述性质的其中一部分性质。把具有传递性的二元关系称为序，在此基础上可以定义各种序关系。对具有非对称性的二元关系，常用记号“ \succ ”或“ \prec ”表示。

期望效用理论

定义 设 R 是集合 D 上的二元关系，

- 1) 如果 R 是非对称和负传递的，则称 R 是一个弱序；
- 2) 如果 R 是弱连通的弱序，则称 R 是一个严格弱序；
- 3) 如果 R 是自反、对称和传递的，则称 R 是一个等价关系。

例如， D 是实数集，“大于”关系是个弱序，也是严格弱序，而“等于”关系是个等价关系。

期望效用理论

决策分析中通常用弱序来描写决策者的基本偏好。

设 \succ 是一个弱序，如果 $x \succ y$ 称 x 优于 y 或者 y 劣于 x ；如果既非 $x \succ y$ 也非 $y \succ x$ 称 x 与 y 无差别，记为 $x \sim y$.

此外，如果 $x \succ y$ 或者 $x \sim y$ ，则称 x 不劣于 y 或者 y 不优于 x ，记为 $x \succcurlyeq y$.

期望效用理论

命题 设 \succ 是 D 上的弱序，有

- (1) 对任意的 $x, y \in D$, 三种关系 $x \succ y, y \succ x, x \sim y$ 有且仅有一种成立；
- (2) \succ 是传递的；
- (3) \sim 是等价关系；
- (4) 由 $x \succ y, y \sim z \Rightarrow x \succ z$; 以及 $x \sim y, y \succ z \Rightarrow x \succ z$;
- (5) \succsim 是传递和连通的；
- (6) 在等价类 $D / \sim \triangleq \{a \subset D \mid \forall x, y \in a, x \sim y\}$ 上的关系 \succ 定义如下：设对某个 $x \in a$ 和 $y \in b$, 有 $a \succ b \Leftrightarrow x \succ y$, 则 \succ 是严格弱序。

期望效用理论

决策分析中，最终希望在策略空间 F 上定义决策者的偏好关系，因而有必要分析策略空间的结构。

描述策略的一种常用方法是，把每项策略 f 看作是从状态空间 S 到后果空间 X 的一个映射。当决策者的偏好关系满足某些公理假设时，其蕴涵着存在决策者关于状态空间上的主观概率测度 P^* 以及存在后果空间 X 上的效用函数 u ，使得决策者关于策略空间的偏好可由期望效用值来表征。

期望效用理论

对策略 $f \in F$, 定义 $P_f(A) = P^*(\{s \in S \mid f(s) \in A\})$, 其中 $A \in \mathcal{Y}$,
 \mathcal{Y} 是 X 上的某个 σ -代数, 则 P_f 是后果空间 X 上的一个概率测度。

因而, 每项策略可以看作是后果空间 X 上的一个概率测度。

期望效用理论

定义 设 \mathcal{Y} 为 X 的全体子集类，若映射 $P : \mathcal{Y} \rightarrow \mathbb{R}$ 满足

- 1) $P(A) \geq 0, \forall A \in \mathcal{Y}$,
- 2) $P(X) = 1$,
- 3) $P(A \cup B) = P(A) + P(B), \forall A, B \in \mathcal{Y}$ and $A \cap B = \emptyset$,
- 4) 存在 X 的有限子集 A 使 $P(A) = 1$,

则称 P 为 X 上的简单概率测度。

期望效用理论

对于 X 上的概率测度 P, Q 以及实数 $\alpha \in [0, 1]$, 定义

$$[\alpha P + (1 - \alpha)Q](A) = \alpha P(A) + (1 - \alpha)Q(A), \forall A \in \mathcal{Y}$$

则 $\alpha P + (1 - \alpha)Q$ 也是 X 上的一个概率测度。

对于简单概率测度 P , 由定义知, 存在有限子集 $\{x_1, \dots, x_n\} \subset X$ 使得 $P(\{x_i\}) = P(x_i) = p_i > 0, i = 1, \dots, n$, 且 $\sum_{i=1}^n p_i = 1$, 因此 P 可以分解为 $P = \sum_{i=1}^n p_i \delta_{x_i}$, 称其为一个抽奖(lottery). 它可以理解为以概率 p_i 得到后果 $x_i (i = 1, \dots, n)$.

期望效用理论

定义 非空集合 F 称为混合集是指，对任意的 $P, Q \in F$ 及 $\alpha \in [0, 1]$ ，对应元素 $\alpha P + (1 - \alpha)Q \in F$ ，且对任意的 $P, Q \in F$ 以及 $\alpha, \beta \in [0, 1]$ 有

- 1) $1P + 0Q = P,$
- 2) $\alpha P + (1 - \alpha)Q = (1 - \alpha)Q + \alpha P,$
- 3) $\alpha[\beta P + (1 - \beta)Q] + (1 - \alpha)Q = \alpha\beta P + (1 - \alpha\beta)Q.$

期望效用理论

效用理论的根本出发点是先建立一组具有实际背景的偏好公理，并以此推导出决策者的策略偏好能够用关于后果的期望效用值定量地表达。

经典效用理论的基本思想可精炼为下面三条公理假设：

- A1 (弱序公理) \succ 是定义在策略空间 F 上的弱序；
- A2 (Archimedean 公理) 若 $P \succ Q, Q \succ T$, 则存在 $\alpha, \beta \in [0, 1]$, 使 $\alpha P + (1 - \alpha) T \succ Q, Q \succ \beta P + (1 - \beta) T$;
- A3 (独立性公理) $P \succ Q$, 则对任意的 $\alpha \in (0, 1)$ 及 T , 有 $\alpha P + (1 - \alpha) T \succ \alpha Q + (1 - \alpha) T$.

期望效用理论

定理： 设 F 是一个混合集，偏好关系 \succ 在 F 上满足公理假设 A1, A2, A3 的充要条件是

- (1) 存在 F 上的实值函数 U , 对任意的 $P, Q \in F$ 有 $P \succ Q \Leftrightarrow U(P) > U(Q)$;
- (2) 对任意的 $P, Q \in F$ 及 $\alpha \in [0, 1]$, 有 $U(\alpha P + (1 - \alpha)Q) = \alpha U(P) + (1 - \alpha)U(Q)$.

此外, U 在正线性变换意义下是唯一的: 若 F 上的实函数 V 也满足条件(1),(2), 则存在常数 $c > 0$ 和 d , 使 $V(P) = cU(P) + d, \forall P \in F$.

期望效用理论

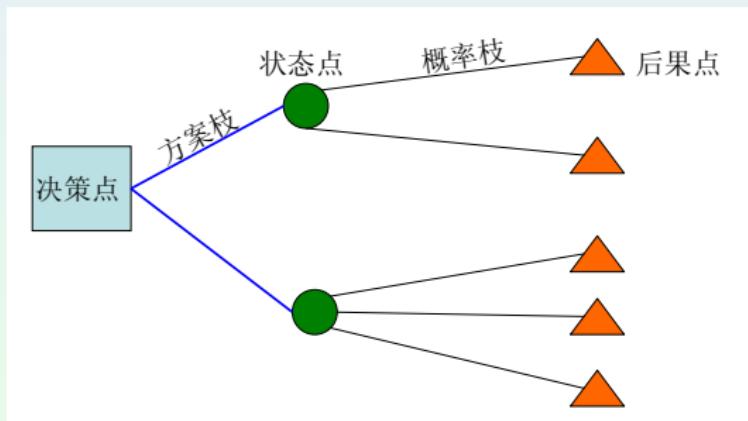
函数 U 称为 F 上的效用函数。若混合集 F 看作是后果空间 X 上的简单概率测度全体，特别地，对任何 $x \in X$ 有 $\delta_x \in F$ ，则利用效用函数 U 的线性性质，很容易地把关于策略的效用转化为关于后果的期望效用。

定理： 设 F 为 X 上的简单概率测度全体，若 F 上的偏好关系 \succ 满足公理假设 A1, A2, A3，则存在 X 上的实值函数 u 使得

- (1) $\forall P, Q \in F$, 有 $P \succ Q \Leftrightarrow EU(P) > EU(Q)$;
- (2) $u(x)$ 在正线性变换意义下是唯一的。

决策树方法

决策树是将方案、状态、后果、效用、概率等用一株树来表示。如图所示，图中的方块叫决策点；由决策点画出若干条线，每条线代表一个方案，叫做方案枝；方案枝的末端有圆圈，叫状态点；从它引出的线代表不同的状态，叫概率枝；在概率枝末端画的小三角叫后果点，在后果点列出后果的效用。



本讲作业

作业：

出国留学申请

根据你的综合情况及所掌握的相关信息，试制定出国留学申请策略并给出决策分析过程。

绪论

微分方程建模

常微分方程模型

偏微分方程模型

最优化建模

统计建模

决策分析

机器学习

人工神经网络

深度学习

总结

Outline

绪论

微分方程建模

常微分方程模型

偏微分方程模型

最优化建模

统计建模

决策分析

机器学习

人工神经网络

深度学习

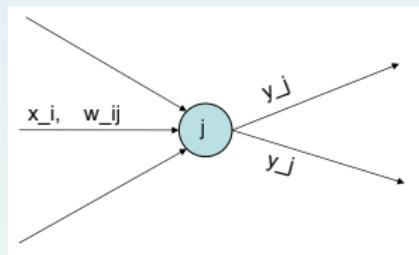
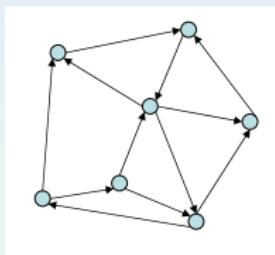
总结

人工神经网络

人工神经网络是模仿人脑结构进行高度平行分散处理的数学模型。人脑神经网络的基本原理是构造人工神经网络模型的一个基本依据。本讲主要介绍人工神经网络中的前向神经网络，内容包括模型、原理和应用。

人工神经网络

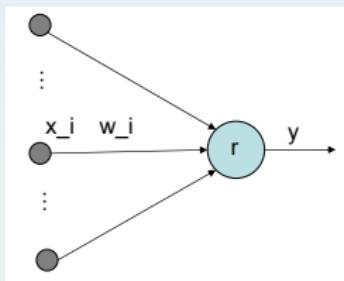
神经网络是大脑的一个重要成分，它由相互关联的神经元组成。神经网络可用连结着神经细胞的回路网抽象模型化后的有向图来表示。



神经网络的各节点（神经元）是处理要素，该处接受其它节点的输出信号作为输入，进一步对这些输入作适当变换后决定该节点的输出，并传递给其它节点。所有这些操作都是各节点以平行分散的方式进行处理。

人工神经网络

模拟上述原理的一个简单人工神经网络是如图所示的基本感知网络（感知器）。



在基本感知网络里，网络的输入是 m 维向量 \mathbf{x} ，其各分量 x_i 由节点*i*送到节点*r*，汇总后通过激活函数作用后输出 y .

人工神经网络

人工神经网络根据构造方法的不同，可以分为两类主要模型：前向型神经网络和反馈型神经网络。

前向型人工神经网络是将神经元分为层，每一层的层内神经元之间没有信息交流，并且一层一层地同步计算。

反馈型神经网络则将整个网络看成一个整体，神经元相互作用，其计算是整体性的。

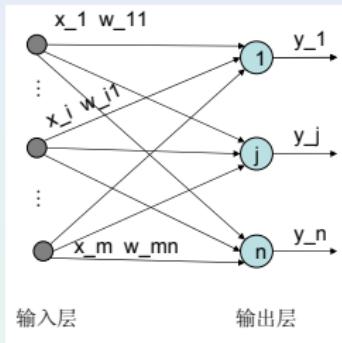
人工神经网络

人工神经网络的计算：

- ▶ 学习阶段
- ▶ 应用阶段

人工神经网络

单层前向神经网络



在单层前向神经网络中，只有输入层和输出层，层内的元素无权数相连。该网络通过对输入层的输入数据学习并同理想输出结果比较，确定其权数，最后将确定了权数的神经网络应用到实际问题中。这类神经网络主要在模式识别中有较强的能力。

人工神经网络

若记输入变量 $\mathbf{x} = (x_1, \dots, x_m)^T$, 输出结果 $\mathbf{y} = (y_1, \dots, y_n)^T$, 从输入*i*到输出节点(神经元)*j*的权数为 w_{ij} , 则该神经网络的输出可表示为

$$y_j = G\left(\sum_{i=1}^m w_{ij}x_i - \theta_j\right), \quad j = 1, \dots, n \quad (48)$$

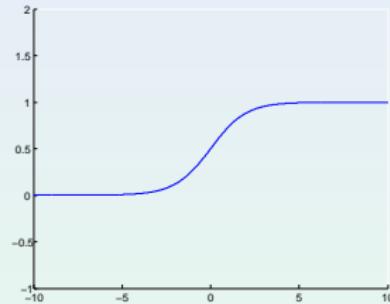
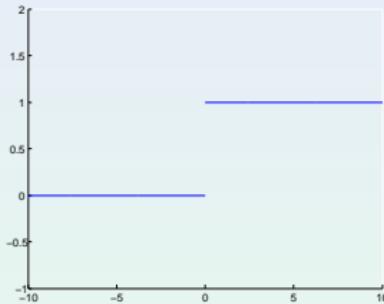
其中 G 是激活函数, θ_j 为神经元*j*的阀值。

给定一组数据样本, 通过学习来确定网络参数:

$$W = (w_{ij})_{m \times n}, \quad \Theta = (\theta_1, \dots, \theta_n)^T.$$

人工神经网络

经常用到的激活函数：



$$G_T(z) = \begin{cases} 1, & \text{if } z > 0 \\ 0, & \text{otherwise} \end{cases}$$

$$G_S(z) = \frac{1}{1 + e^{-z}}$$

人工神经网络

示例：模式识别

现给定 N 个模式向量 $\mathbf{p}^1, \mathbf{p}^2, \dots, \mathbf{p}^N \in \mathbb{R}^m$, 不失一般性, 设前 M 个属于某一类而剩下的 $N - M$ 个不属于该类。如果模式集合线性可分, 那么存在 m 维向量 $\mathbf{w} = (w_1, w_2, \dots, w_m)^T$ 及实数 θ 使得

$$\begin{cases} \mathbf{w}^T \mathbf{p}^s - \theta > 0, & s = 1, \dots, M \\ \mathbf{w}^T \mathbf{p}^s - \theta < 0, & s = M + 1, \dots, N \end{cases}$$

成立。

若记 $\mathbf{q}^s = \begin{pmatrix} \mathbf{p}^s \\ -1 \end{pmatrix}$, $\mathbf{v} = \begin{pmatrix} \mathbf{w} \\ \theta \end{pmatrix}$, 则上式可写成

$$\begin{cases} \mathbf{v}^T \mathbf{q}^s > 0, & s = 1, \dots, M, \\ \mathbf{v}^T \mathbf{q}^s < 0, & s = M + 1, \dots, N. \end{cases} \quad (49)$$

人工神经网络

为达到模式识别的目的，我们须设定权系数 \mathbf{v} 使得输入是 $\mathbf{q}^1, \dots, \mathbf{q}^M$ 时输出 $y = 1$, 输入是 $\mathbf{q}^{M+1}, \dots, \mathbf{q}^N$ 时输出 $y = 0$, 即满足式(49).

为求得权系数 \mathbf{v} , 考虑用改错学习的方法。首先，适当设定神经网络的初始权系数，然后逐次输入待识别的模式，等到出现了错误输出时再修改权系数。

人工神经网络

改错学习算法：

- (0) (初始化) 给定适当的初始权系数向量 $\mathbf{v}(0)$ 及正常数 η ,
令 $t := 0$.
- (1) (计算输出) 令 $s := t(\bmod N) + 1$, 计算模式向量 \mathbf{q}^s 的输出 $y = G_T(\mathbf{v}(0)^T \mathbf{q}^s)$. 若连续 N 次都得到正确输出, 则停止。
- (2) (修正权系数) 若 $1 \leq s \leq M$, 则

$$\mathbf{v}(t+1) := \begin{cases} \mathbf{v}(t), & \text{if } y = 1 \text{ 正确输出时} \\ \mathbf{v}(t) + \eta \mathbf{q}^s, & \text{if } y = 0 \text{ 错误输出时} \end{cases}$$

若 $M + 1 \leq s \leq N$, 则

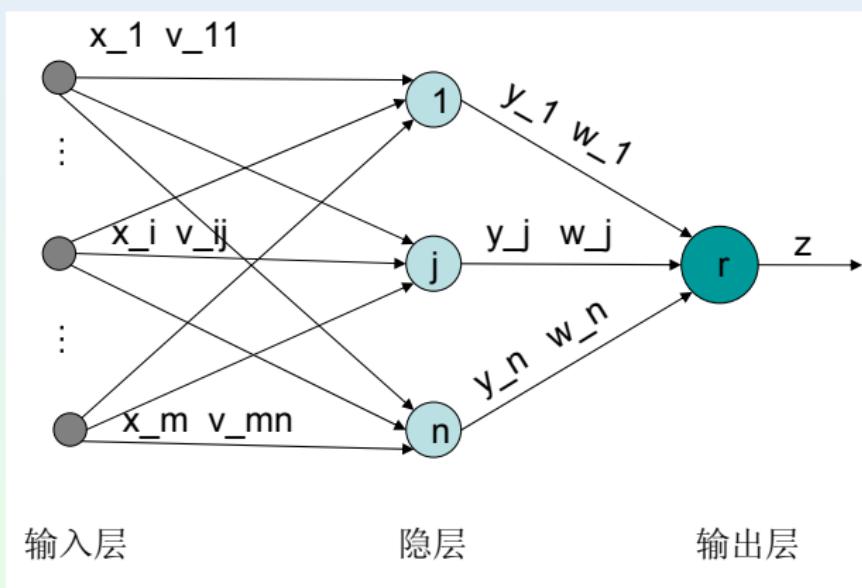
$$\mathbf{v}(t+1) := \begin{cases} \mathbf{v}(t) - \eta \mathbf{q}^s, & \text{if } y = 1 \text{ 错误输出时} \\ \mathbf{v}(t), & \text{if } y = 0 \text{ 正确输出时} \end{cases}$$

令 $t := t + 1$, 返回第(1)步。

人工神经网络

多层前向神经网络

为简单起见，我们考虑如图所示的三层网络。



人工神经网络

第一层（输入层）各节点 $i \in \{1, \dots, m\}$ 把输入 x_i 信号传递到第二层（隐层）的节点 $j = 1, 2, \dots, n$. 若输入层节点*i*到隐层节点*j*的权系数为 v_{ij} , 则隐层节点*j*有输入总和 $\sum_{i=1}^m v_{ij}x_i$, 并通过激活函

数 G 将其变换成 $y_j = G(\sum_{i=1}^m v_{ij}x_i)$ 后, 再传送到第三层（输出层）的节点*r*. 若隐层节点*j*到输出层节点*r*的权系数为 w_j , 则输出层节点*r*有输入总和 $\sum_{j=1}^n w_jy_j$, 并由激活函数 G 变换成 $z = G(\sum_{j=1}^n w_jy_j)$ 后成为网络的输出。

人工神经网络

记待识别的 N 个模式向量为 $\mathbf{x}^s = (x_1^s, \dots, x_m^s)^T, s = 1, 2, \dots, N$, 各自所对应的理想输出为 $d^s, s = 1, 2, \dots, N$. 若设网络的权系数为 $V = (v_{ij})_{m \times n}, \mathbf{w} = (w_1, \dots, w_n)^T$, 网络输入模式向量 \mathbf{x}^s 时对应的第二层输出记为 $\mathbf{y}^s(V) = (y_1^s(V), \dots, y_n^s(V))^T$, 第三层(节点 r)给出网络的最终输出记为 $z^s(V, \mathbf{w})$.

此时我们的目标是, 确定权系数的值使得对所有输入模式向量, 网络都能给出正确的输出。即求解下列以权系数 (V, \mathbf{w}) 为变量的非线性方程组:

$$\left\{ \begin{array}{rcl} z^1(V, \mathbf{w}) & = & d^1 \\ z^2(V, \mathbf{w}) & = & d^2 \\ & \vdots & \\ z^N(V, \mathbf{w}) & = & d^N \end{array} \right.$$

人工神经网络

一般地，在多层神经网络中我们采用S-型函数 G_S 作为各节点的激活函数。于是有

$$y_j^s(V) = G_S(V_j^T \mathbf{x}^s), \quad j = 1, \dots, n, \quad s = 1, \dots, N,$$

$$z^s(V, \mathbf{w}) = G_S(\mathbf{w}^T \mathbf{y}^s), \quad s = 1, \dots, N.$$

考虑输入模式 \mathbf{x}^s 对应的实际输出与正确输出的二乘误差

$$e^s(V, \mathbf{w}) = (z^s(V, \mathbf{w}) - d^s)^2,$$

我们可以建立如下最小化模型

$$\min E(V, \mathbf{w}) = \frac{1}{2} \sum_{s=1}^N e^s(V, \mathbf{w}) \quad (50)$$

Outline

绪论

微分方程建模

常微分方程模型

偏微分方程模型

最优化建模

统计建模

决策分析

机器学习

人工神经网络

深度学习

总结

深度学习

深度学习基本概念

本讲作业

作业：

课程大作业

课程大作业：

考核要求包括：

- (1) 建模文档（6月25日之前提交）
- (2) 算法实现的源代码（6月27日之前提交）
- (3) 测试报告（6月22日发布测试数据，6月27日提交报告）