



中国科学技术大学

University of Science and Technology of China

数学建模

Mathematical Modeling

刘利刚

中国科学技术大学

1. 课程介绍



中国科学技术大学
University of Science and Technology of China

数学建模 Mathematical Modeling

历史上的数学家



欧几里得



欧拉



高斯

Siméon Denis Poisson (泊松)

- 泊松的老师
 - *Laplace, Lagrange, ...*
- 以Poisson命名的词汇
 - Poisson's equation
 - Poisson's integral
 - Poisson distribution
 - Poisson brackets
 - Poisson's ratio
 - Poisson's constant



1781-1840, France

Siméon Denis Poisson (泊松)

“Life is good for only two things:

*to **study** mathematics
and to **teach** it.”*



1781-1840, France

华罗庚老前辈

- 中科大数学系首任主任
- “中国最伟大的数学家，当代世界上最重要的数学家之一”



华老的贡献

- **纯粹数学**

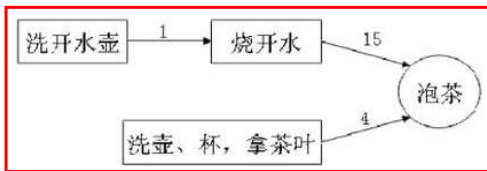
- 解析数论、函数论
- 代数几何、矩阵几何学

- **应用数学与计算数学**

- 数学技术、计算技术

- **数学的应用：数学的普及与推广**

- 数学工程
- “统筹法”
- “优选法”



华老推动数学的广泛应用



数学的“哲学三问”



你是谁？



你从哪来？

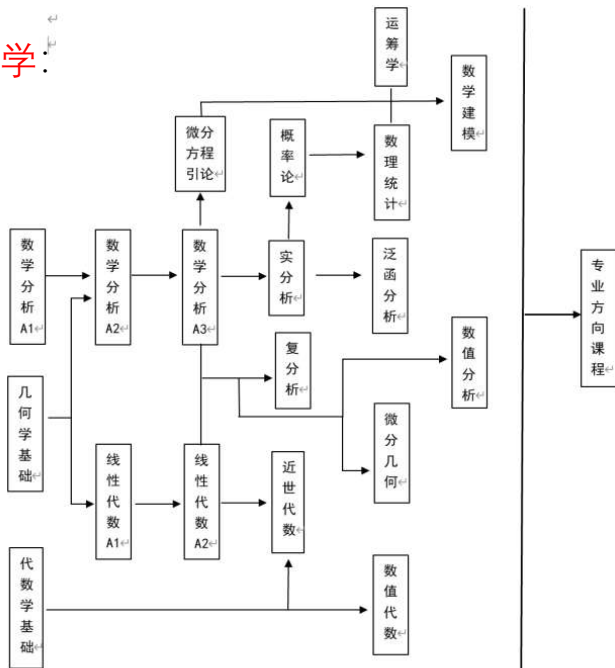


你到哪儿去？

- 数学是什么？
- 数学好玩吗？
- 数学有用吗？

数学是什么？

我们学了很多数学：



数学好玩吗？

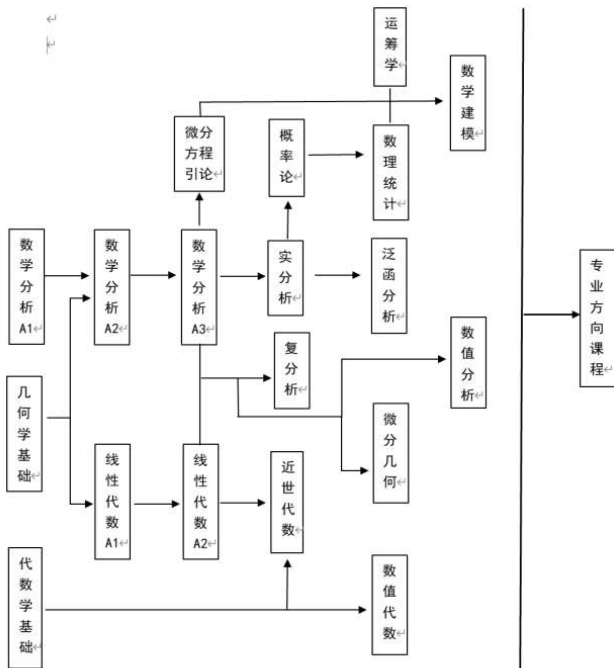
数学有用吗？



弗兰西斯·培根

“知识就是力量”

数学知识:





弗兰西斯·培根

“用知识武装的**技能**才是力量”

— 培小根

我的亲身体会

数学不是没有用，而是不够用！



中国科学技术大学
University of Science and Technology of China

数学建模 Mathematical Modeling

目标：数学思维和建模能力

数学是一种语言和思维

数学建模是解决实际问题的
技能和方法

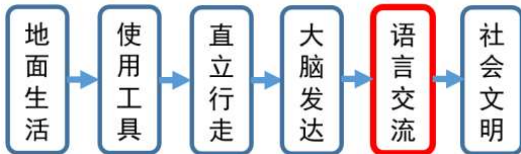
数学建模的核心能力： “抽象”能力

1. 透过表象看本质：第一性原理
2. 逻辑推理形递归：演绎法

人类发展的驱动力

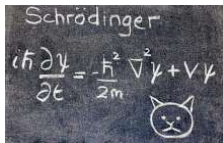
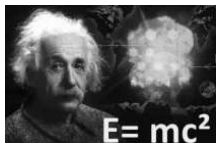
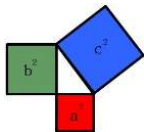
- 好奇心与想象力：

- 探索宇宙规律
- 改造现实世界
- 让生活更美好



数学：自然科学的语言及文字

一种对客观世界规律进行描述的符号系统（文字）



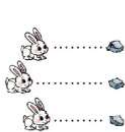
数学来源于生活、并应用于生活！

“事物”的抽象：文字

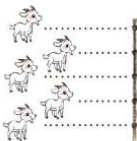


人: 𠤎	禾: 禾	孕: 𠬞	鸟: 𠬞	尿: 𠬞
牛: 𠬞	羊: 𠬞	目: 目	屎: 𠬞	水: 𠬞
尾: 𠬞	马: 𠬞	鹿: 𠬞	网: 𠬞	门: 𠬞
栅: 𠬞	山: 𠬞	火: 𠬞	酒: 𠬞	雨: 𠬞
包: 𠬞	囚: 𠬞	日: 日	月: 月	象: 𠬞
鱼: 𠬞	伏: 𠬞	射: 𠬞	舟: 𠬞	饮: 𠬞

“计数”的抽象： 数字



石子计数



结绳计数



1世纪的婆罗门		—	=	≡	+	ℵ	④	7	5	7
印度梵文	०	१	२	३	४	५	६	७	८	९
东支阿拉伯至今	.	١	٢	٣	٤	٥	٦	٧	٨	٩
11世纪的西支阿拉伯		١	٢	٣	٤	٥	٦	٧	٨	٩
欧洲15世纪	0	I	2	3	℥	④	6	7	8	9
16世纪至今	0	1	2	3	4	5	6	7	8	9
罗马数字		I	II	III	IV	V	VI	VII	VIII	IX
中国算筹		Ⅰ	Ⅱ	Ⅲ	Ⅳ	Ⅴ	Ⅵ	Ⅶ	Ⅷ	Ⅸ

“对象”的抽象：变量与函数

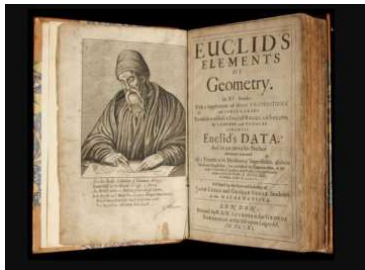
- 幼儿园：2个苹果 + 3个苹果 = 5个苹果
- 小学： $2+3=5$
- 中学： $ax^2 + bx + c = 0$; $y = f(x)$
- 大学： $\Delta f = 0$

“数量”的抽象：运算与数

$+$ $-$ $\times$ $\div$

“几何”的抽象

• 丈量土地

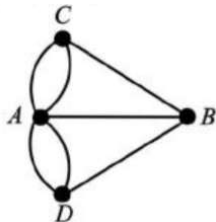
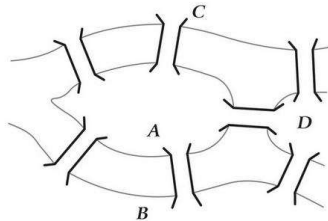
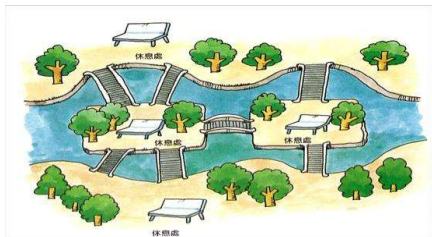


《几何原本》

欧几里得几何学的五条公设（公理）：

1. 任意两个点可以通过一条直线连接；
2. 任意线段能无限延伸成一条直线；
3. 给定任意线段，可以以其一个端点作为圆心，该线段作为半径作一个圆；
4. 所有直角都全等；
5. 若两条直线都与第三条直线相交，并且在同一边的内角之和小于两个直角，则这两条直线在这一边必定相交。

“空间关系”的抽象：图及图论

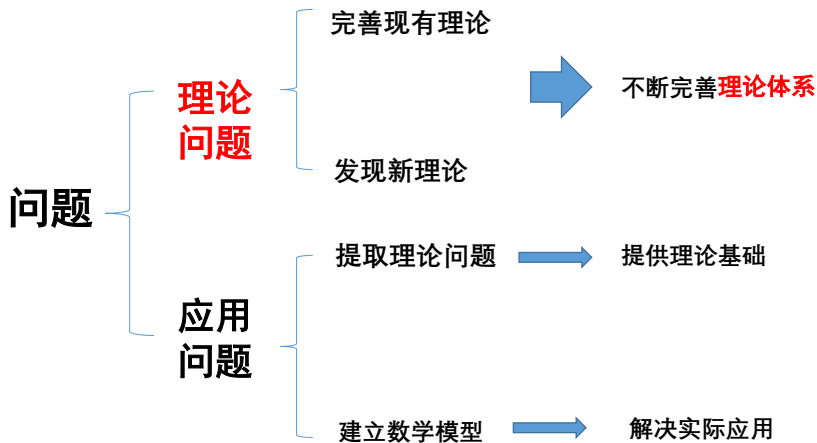


第一性原理（公理）
+
演绎法
=
逻辑（理性）系统

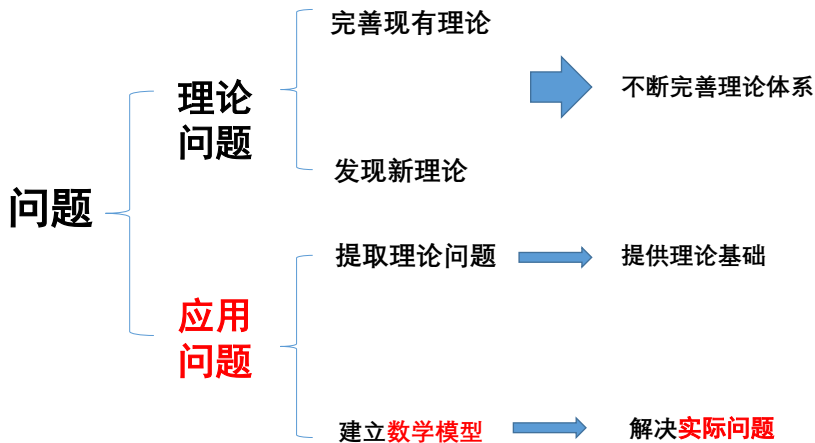
数学的产生与发展

- 起源于非常实际的目的：符号系统
 - 计数、土地测量、灌溉系统…
- 再到推理演绎和逻辑体系：逻辑系统
 - 抽象总结共性规律：公理体系
 - 例子：欧几里得《几何原本》…
- 表达世界规律的符号系统：科学的语言与文字
 - 物理、化学、生物、经济…
 - 工程技术
- 辅助发现新科学规律
 - 相对论、黑洞、引力波…
 - （科学能解释的现象仍然有限）

工作（学习、研究）范式



工作（学习、研究）范式



数学的应用与计算

- 数学有没有用?
 - 数学不是没有用，而是不够用
 - 现有的数学工具不能解决所有实际问题
 - 我在专业研究中遇到很多数学问题而受阻
- 怎么用？如何解决实际问题？
 - 将问题进行抽象：数学建模
 - 求解模型并验证：数值计算

数学技能： 数学建模技能

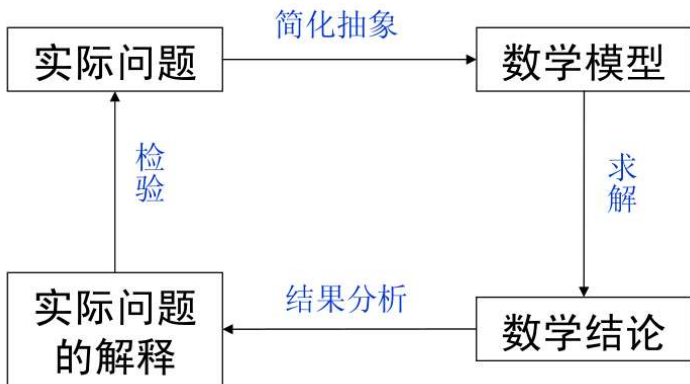
- 数学是一种**思维模式**
 - “数学方式的理性思维”
- 数学不仅是一些知识，也是一种素质
- 解决实际问题的能力与素质
 - 很重要！

数学是一种“语言”！

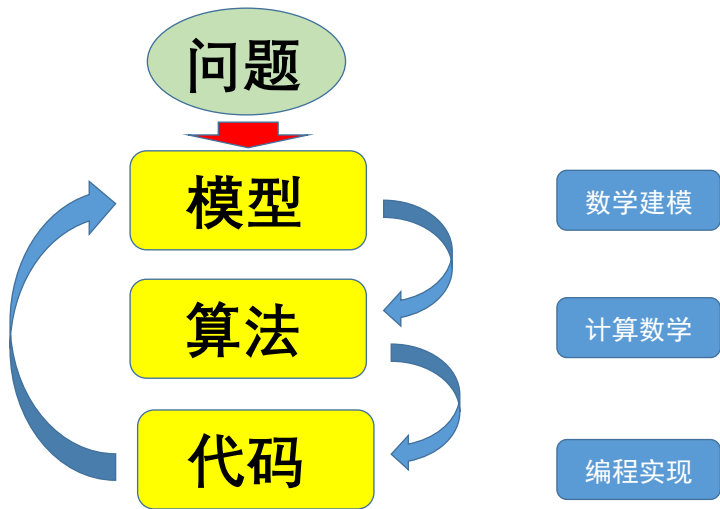
数学建模的“定义”

- 定义1: 关于部分现实世界为一定目的而作的抽象、简化的数学结构, 即, 用数学术语对部分现实世界的描述
- 定义2: 用数学的语言和方法, 通过抽象、简化建立能近似刻画并“解决”实际问题的一种数学工具

数学建模的过程



解决问题的关键在于**数学建模**能力



即：科学研究的过程

简单的数学模型

数学建模早就知

- 我们从小就接触过数学模型与数学应用：
 - 应用题
 - “甲乙两地相距750公里，船从甲到乙顺水航行需30小时，从乙到甲逆水航行需50小时，问航速，水速若干？”
 - 物体
 - “从平静湖面的小船上仍一块石头至水中，湖面是上涨还是下降？”
 - 数学竞赛
 - ...

例1. 手机电话卡的选择

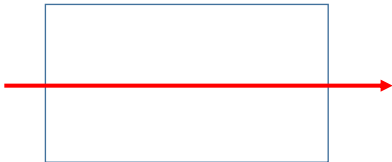
- 已知：入网电话卡每分钟0.4元，每月25元租金；神州行卡每分钟0.6元，不用月租金
- 问：选择哪种卡比较省钱？

例2. 电梯运行策略

- 楼房有30层，有2个电梯
- 你在10楼
- 一个电梯在1层，另一个电梯在19层
- 问哪个电梯来接你合理？

例3. 摘桃时机

- 有一片桃林，其中的桃子有大有小
- 你从桃林的一头走到另一头，只能走一次（只能往前走，不能往回走）
- 你只有一次机会摘桃子
- 问：你什么时候摘桃子最合适（能摘到可能最大的桃子）？



例4. 从科大到合肥站最快的线路



课程信息与要求

课程主页

http://staff.ustc.edu.cn/~lgliu/Courses/MathModeling_2023_spring-summer/default.htm

Prof. Dr. Ligang Liu (刘利刚)



Graphics&Geometric Computing Laboratory
School of Mathematical Sciences
University of Science and Technology of China
Hefei 230026, China

Phone/Fax: +86-551-63600985

Email: lgliu@ustc.edu.cn

通讯地址: 安徽省合肥市金寨路96号中国科技大学数学学院

邮政编码: 230026

办公室: 中科大东区管理科研楼1217室



Short Bio

Research

Publications

Professionals

Teaching

Resources



- Mathematical Modeling (数学建模), Undergraduates, 001139
 - Spring-Summer, 2021
 - Spring-Summer, 2023

Navigation icons for presentation software.

课程助教

- 陈泽豪 (czh15967722357@mail.ustc.edu.cn)
- 施羽翔 (shiyuxiang@mail.ustc.edu.cn)
- 莫易凡 (moyf@mail.ustc.edu.cn)

在该课上，你们将能学到...

- 抽象思维与数学建模的方法
 - 阅读文献
 - 科研的初步方法：从数学建模到算法实现
 - 论文的写作
- 常用算法
 - 算法设计与实现
 - 改进与推广
- 编程工具
 - C++面向对象编程/Matlab/Mathematica/Grasshopper...
 - 图像图形编程、使用程序库
- 学以致用！！

预备知识：数学

- 微积分
- 线性代数
- 微分方程
- 数值方法与计算
- 最优化（运筹学）
- 统计学
- ...

还没有学没有关系：数学在使用的过程中学得更快，能更深刻地深刻理解和掌握

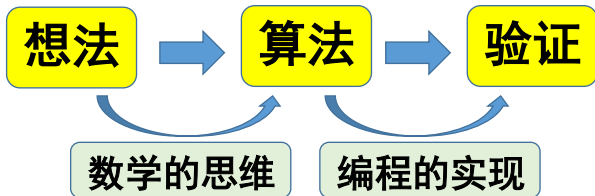
预备知识：编程

- 编程能将你脑中的想法得到实现并看到、应用
 - 从C到C++（面向对象编程）
 - Python
 - ...
- 算法：严谨的逻辑思维
- 数学工具
 - Matlab
 - Mathematica/Maple
 - Lingo/Lindo...
- 其他工具
 - Grasshopper, Photoshop...

工欲善其事必先利其器

数学与编程的关系

- **数学**是一种思考方式!
 - 数学不是没有用，而是不够用!
 - 在应用的过程中体会数学的美妙
- **编程**能帮助你实现自己的想法
 - 编程比学习数学容易得多!
 - 较强的动手能力



“Everybody in this country should learn how to program a computer... because it teaches you how to think.”

- Steve Jobs

@36氪

- 乔布斯：“每个人都应学习编程，因为它教你如何思考”

交叉学科

- 图像处理
- 物理
- 化学
- 经济
- 生物
- 计算机
- 信息
- ...

教材：非教材，参考为主

- 《数学模型》第2版，谭永基、蔡志杰编著，复旦大学出版社，2011



数学建模不是做应用题

- 书中的大部分案例靠自学
- 课程过程中的实际实例剖析
 - 课堂内容及作业更重要
 - 动手做，“做中学”

课程QQ群： 736711705

(入群验证：ustc)

- 昵称命名规则：“姓名”
 - 姓名用实名



课程作业和考试

- 平时作业： 75%
 - 每2周一个
 - 课程强度中等
- 期末大作业： 25%
 - 模拟全国/美国建模竞赛
 - 3人合作完成
- 无期末考试

程序作业递交

- 作业递交是通过CES云作业管理系统来完成
 - 主页上有链接及详细的使用说明
- 递交内容：
 - 源代码+作业报告
- 所有文件压缩打包成一个文件(zip或rar)
 - 系统限制文件大小为30M
 - 如文件大小超过30M而无法上传，可以挂到其他云空间（建议校内的睿客网），然后将链接写到*.txt文件上传

建议：学习使用Github

- 构建自己的Github
 - 设为private，不要public：避免其他同学能访问到你的作业代码
 - 管理和维护自己的文档、代码
- 作业提交：将Github的工程目录（去除不必要的文件）压缩提交

作业要求 (1)

- 模型报告书写
 - 符合论文规范
 - 文字，图表清晰
 - 数据说明
- 代码提交
 - 文档说明：运行参数，操作等
 - 程序能运行：编译通过
 - 压缩打包
 - 去除不必要的文件，如\debug目录

作业要求 (2)

- 通过CES系统来递交作业，不接受email递交作业
- 必须在规定的最后期限(ddl)之前递交
- 超过最后期限系统将自动不再接受作业

作业要求 (3)

- 独立完成
- 相互帮助
- 团队合作
- 绝不允许抄袭!

作业评判维度

95+	方法新颖巧妙，非常好
85+	模型建立求解合理有创新，书写很好
80	模型建立求解合理，书写规范
60	模型建立求解基本合理，但书写一般
40	模型建立求解有问题，书写一般
20	模型建立不正确，书写糟糕，态度有问题
0	态度有问题，很遗憾 ☹

对于出色的作业，我们将给予其展示的机会！

作业布置：2周一次

- 周五课上布置作业
- 次周日晚递交
- 次次周五课上讲解作业

课程目标

- 让同学们真正能
 - 提高发现问题和解决问题的能力
 - 运用知识和寻找知识的能力
 - 学有所用，增强兴趣和信心
- 方法
 - 多思考分析
 - 实践、实践、实践

数学的应用与计算

- 数学来源于生活，而应用于生活中
- 宇宙之大，粒子之微，火箭之速，化工之巧，地球之变，日用之繁，无处不用数学

从数学研究到数学应用

- 信息时代（“**大数据时代**”）
 - 数学应用的**黄金时代**
- 数学应用的几大优势
 - 信息获取即时(internet)，能跟上前沿
 - 计算能力大大提高
 - 数据量大大增加
 - 高科技投入增多
 - 沟通相对容易：手机，电话，网络...

Siméon Denis Poisson (泊松)

“Life is good for only two things:

*to **study** mathematics
and to **teach** it.”*



1781-1840, France

*Life is good for only two things:
to **study** mathematics
and to **apply** it.*

泊小松



中国科学技术大学
University of Science and Technology of China

谢谢！



中国科学技术大学
University of Science and Technology of China

数学建模 Mathematical Modeling

刘利刚

中国科学技术大学

2. 抽象

方法论

- 抽象
 - 简化非本质的因素
 - 透过现象看本质
- 迭代
 - 将问题转化为已有问题
 - 组合、递归
- 执行方法
 - Divide-and-conquer
 - Coarse-to-fine

第一性原理

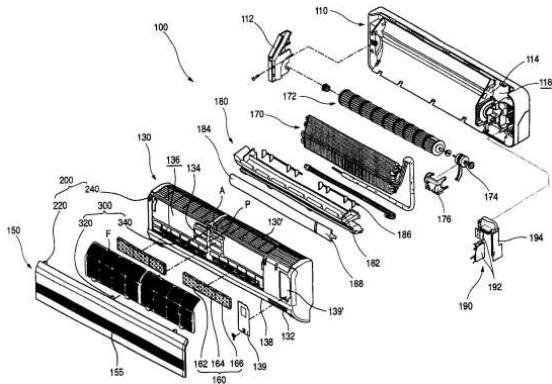
- 第一性原理：就是看透事物本质的根本方法
- 最早由古希腊哲学家亚里士多提出：“在每个系统探索中都存在第一性原理。第一性原理是基本的命题和假设，不能被省略和删除，也不能被违反”
- 大道至简：本质、道、底层逻辑
- 举例
 - 几何学的第一性原理
 - 物理学的第一性原理
 - 人际交往的第一性原理
 - ...

(1) 物体对象

什么是“空调”？



从设计者/生产者角度看空调



- 运行原理

从用户的角度看空调



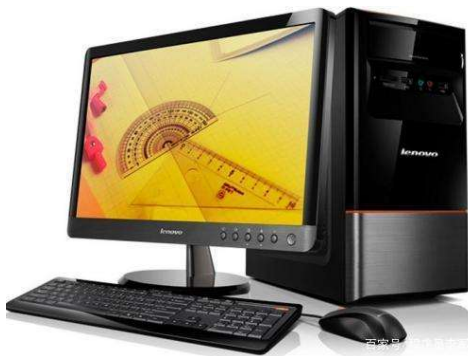
- 用户只关心“对象”的**功能**及使用功能的**接口**
- 用户界面：人机（物）接口

用户思维：从用户角度来看对象

- 什么是椅子？
- 什么是手机？
- 什么是激光笔？
- 什么是汽车？
- ...

(2) 计算机系统

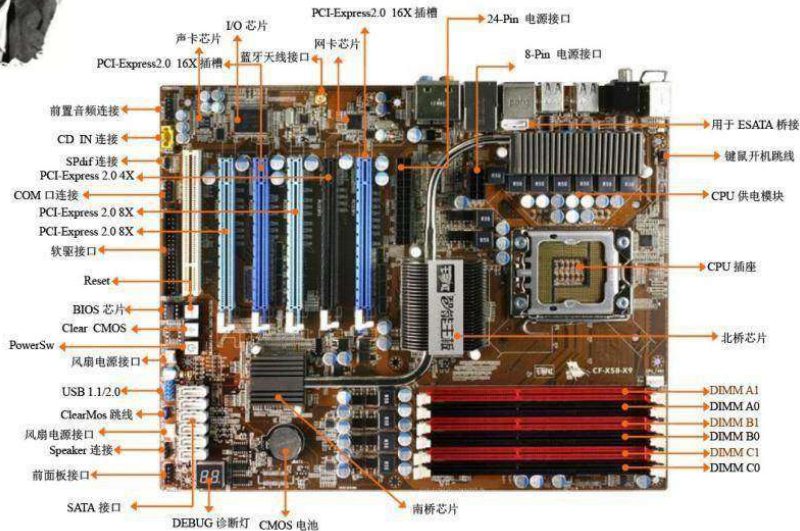
什么是计算机？



- 功能需求：解决一切可以用“计算”来解决的问题



冯.诺依曼计算机的体系结构

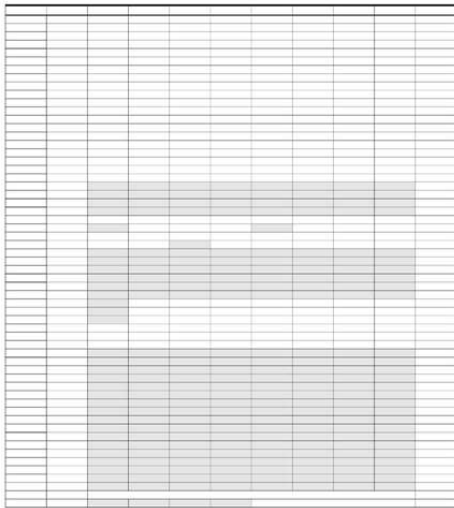


计算机结构的抽象



存储器：内存

- byte (字节) : 8 bits, 表示256个状态



	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
00000000h:	CA	FE	BA	BE	00	00	00	33	00	22	0A	00	06	00	14	09
00000010h:	00	15	00	16	08	00	17	0A	00	18	00	19	07	00	1A	07
00000020h:	00	1B	01	00	06	3C	69	6E	69	74	3E	01	00	03	28	29
00000030h:	56	01	00	04	43	6F	64	65	01	00	0F	4C	69	6E	65	4E
00000040h:	75	6D	62	65	72	54	61	62	6C	65	01	00	12	4C	6F	63
00000050h:	61	6C	56	61	72	69	61	62	6C	65	54	61	62	6C	65	01
00000060h:	00	04	74	68	69	73	01	00	1F	4C	63	6F	6D	2F	70	61
00000070h:	64	64	78	2F	74	65	73	74	2F	61	73	6D	2F	48	65	6C
00000080h:	6C	6F	57	6F	72	6C	64	3B	01	00	04	6D	61	69	6E	01
00000090h:	00	16	28	5B	4C	6A	61	76	61	2F	6C	61	6E	67	2F	53
000000a0h:	74	72	69	6E	67	3B	29	56	01	00	04	61	72	67	73	01
000000b0h:	00	13	5B	4C	6A	61	76	61	2F	6C	61	6E	67	2F	53	74
000000c0h:	72	69	6E	67	3B	01	00	0A	53	6F	75	72	63	65	46	69
000000d0h:	6C	65	01	00	0F	48	65	6C	6C	6F	57	6F	72	6C	64	2E
000000e0h:	6A	61	76	61	0C	00	07	00	08	07	00	1C	0C	00	1D	00
000000f0h:	1E	01	00	0C	48	65	6C	6C	6F	2C	57	6F	72	6C	64	21
00000100h:	07	00	1F	0C	00	20	00	21	01	00	1D	63	6F	6D	2F	70
00000110h:	61	64	64	78	2F	74	65	73	74	2F	61	73	6D	2F	48	65
00000120h:	6C	6C	6F	57	6F	72	6C	64	01	00	10	6A	61	76	61	2F
00000130h:	6C	61	6E	67	2F	4F	62	6A	65	63	74	01	00	10	6A	61
00000140h:	76	61	2F	6C	61	6E	67	2F	53	79	73	74	65	6D	01	00
00000150h:	03	6F	75	74	01	00	15	4C	6A	61	76	61	2F	69	6F	2F
00000160h:	50	72	69	6E	74	53	74	72	65	61	6D	3B	01	00	13	6A
00000170h:	61	76	61	2F	69	6F	2F	50	72	69	6E	74	53	74	72	65
00000180h:	61	6D	01	00	07	70	72	69	6E	74	6C	6E	01	00	15	28
00000190h:	4C	6A	61	76	61	2F	6C	61	6E	67	2F	53	74	72	69	6E
000001a0h:	67	3B	29	56	00	21	00	05	00	06	00	00	00	00	00	02
000001b0h:	00	01	00	07	00	08	00	01	00	09	00	00	00	2F	00	01
000001c0h:	00	01	00	00	00	05	2A	B7	00	01	B1	00	00	00	02	00
000001d0h:	0A	00	00	00	06	00	01	00	00	00	03	00	0B	00	00	00
000001e0h:	0C	00	01	00	00	00	05	00	0C	00	0D	00	00	00	09	00
000001f0h:	0E	00	0F	00	01	00	09	00	00	00	37	00	02	00	01	00
00000200h:	00	00	09	B2	00	02	12	03	B6	00	04	B1	00	00	00	02
00000210h:	00	0A	00	00	00	0A	00	02	00	00	05	00	08	00	00	06
00000220h:	00	0B	00	00	00	0C	00	01	00	00	09	00	10	00	11	
00000230h:	00	00	00	01	00	12	00	00	00	02	00	13				

CPU（运算器）

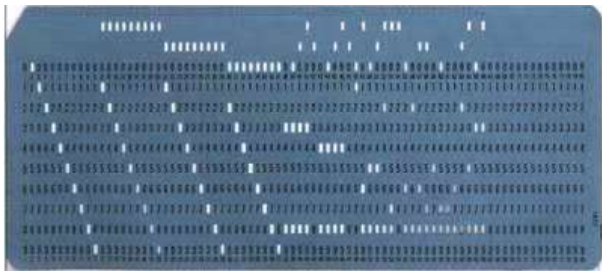
- 负责程序（指令序列）的执行
 - 指令序列也是存放在存储里面
 - 计算机加电启动后，**CPU**从一个固定的存储地址开始执行
- CPU指令集
 - 计算类：各类基本数学运算，如加减乘除、sin/cos 等等
 - I/O 类：从输入输出设备（存储）读数据、写数据
 - 指令跳转类：在满足特定条件下跳转到新的当前程序执行位置

机器语言：用二进制和编码方式提供的指令系统编程程序



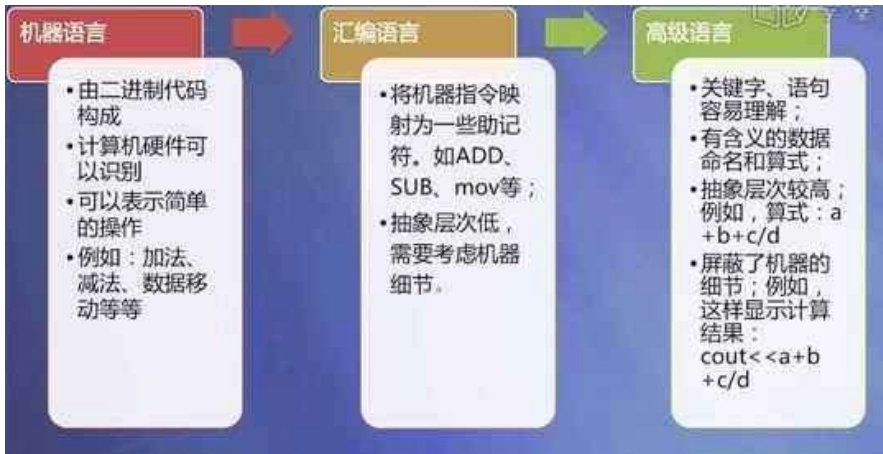
计算7+10并存储的程序

```
100001 10  
00000111  
  
100010 10  
00001010  
  
100101 11  
00000110  
  
111101 00
```

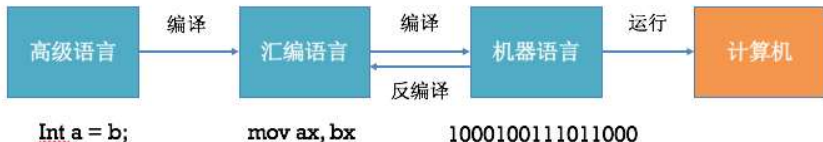


打孔卡

编程语言+编译器



编程语言+编译器



```
#include <stdio.h>
int main ()
{
    int i;

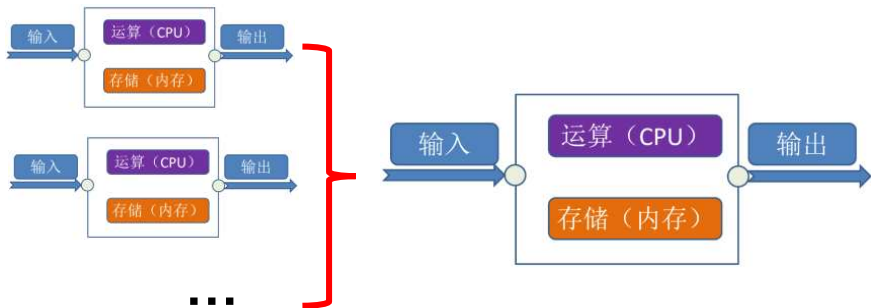
    for (i=1; i<=100; i=i+1)
        printf("%d\t", i);
    return(0);
}
```



```
0010011110111101111111111111100000
101011111011111110000000000010100
10101111101001000000000000100000
10101111101001010000000000100100
1010111110100000000000000011000
1010111110100000000000000011100
1000111110101110000000000011100
1000111110111000000000000011000
0000000111001110000000000011001
0010010111001000000000000000001
001010010000001000000001100101
101011111010100000000000001100
00000000000000001110000010010
00000011000111111001000010001
0001010001000011111111111011
101011111011100100000000011000
0011100000010000100000000000
100011111010101000000001000
```

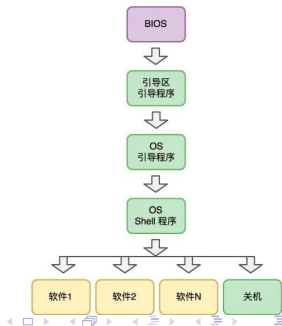
递归原则

- 1: 中央处理器 + 存储: 可以支持任意复杂的“计算”
-- $y=f(x)$
- 2: 输入输出设备: 是电脑无限可能的扩展能力

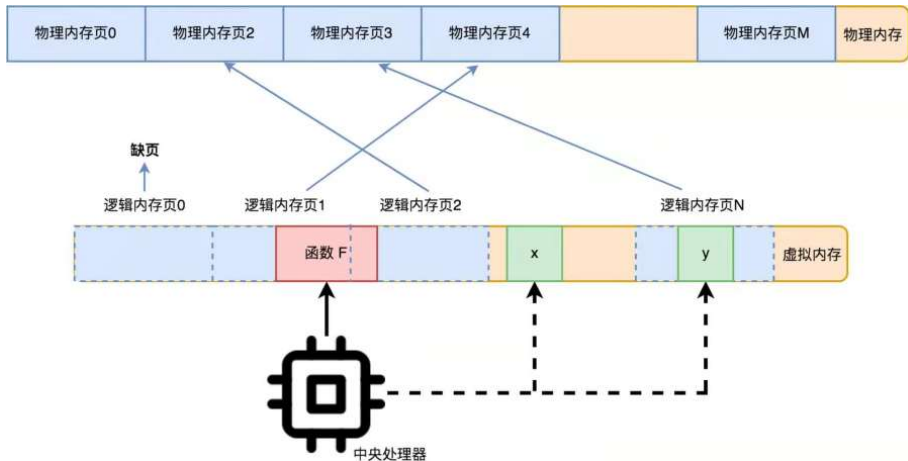


操作系统

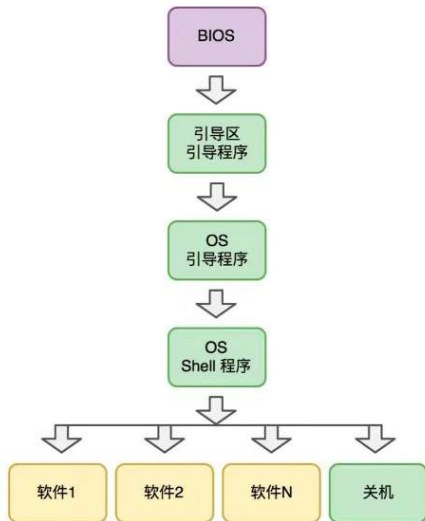
- 解决软件治理的问题
 - 进程与处理机管理、作业管理、存储管理、设备管理、文件管理
- 提供基础编程接口 (Windows API)
 - GDI: 文本、图像、画图
 - ...

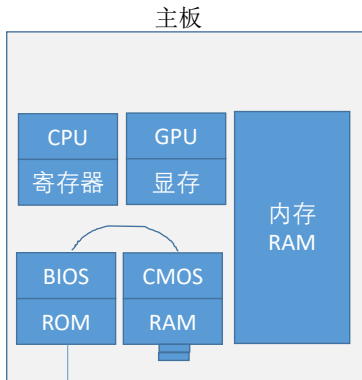


物理内存与虚拟内存

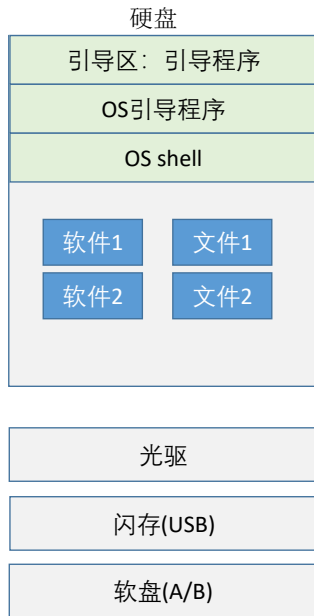


计算机运行全过程





硬件自检：各种驱动程序
更改的硬件配置存入CMOS
跳转到外部存储的引导区程序



“响应体”对象的抽象



例子：人、电视、空调、汽车...

(3) 面向对象编程

程序员的层次

- 架构师
 - 核心能力：抽象思维与抽象能力
 - 宏观的全局掌控能力
- 工程师
 - 软件工程
- 搬砖师（“码农”）
 - 编写代码

代码质量的评判维度

- 可阅读性：方便代码流转
- 可扩展性 / 可维护性：方便修改功能，添加新功能
- 可测试性：质量管理
- 可复用性：简化后续功能开发的难度

热身练习回顾

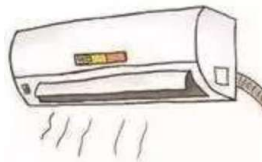
- 什么是数组？数组的抽象？
 - 需求、规格
- 面向对象的编程思想
 - 封装 (响应体)
 - 数据：存储
 - 代码：运算
 - 接口：I/O
- 扩充
 - vector, list
 - template



热身练习批改总结

关于编程

- 好习惯
 - 代码规范：程序员的“脸面”
 - 程序注释：边写代码边注释，日志...
 - Github
- 面向对象编程思维
 - 以“接口”（用户面板）为中心的版式
- 熟练C++指针、内存分配与释放、避免内存泄漏
- 调试程序：思路和逻辑清晰、多实践！
 - VS debug工具
 - F9, F5, F10, F11, ^F5, ^F10, Shift+F10, Shift+F11, ...
 - 各种查看窗口: stack, variable, watch...
 - 使用std::cout不断输出要检查的内容
- 程序员的终身修养
 - 泛型编程、模式设计、软件工程、架构师...

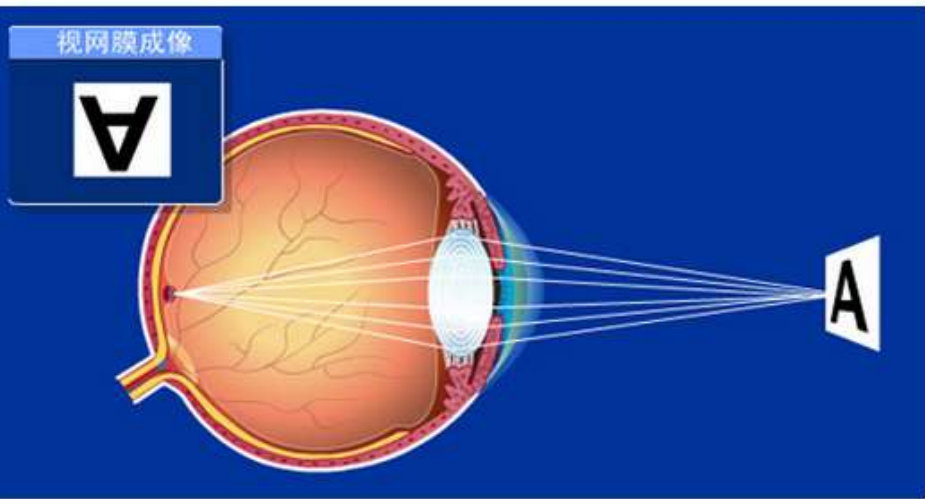


(4) 数字图像

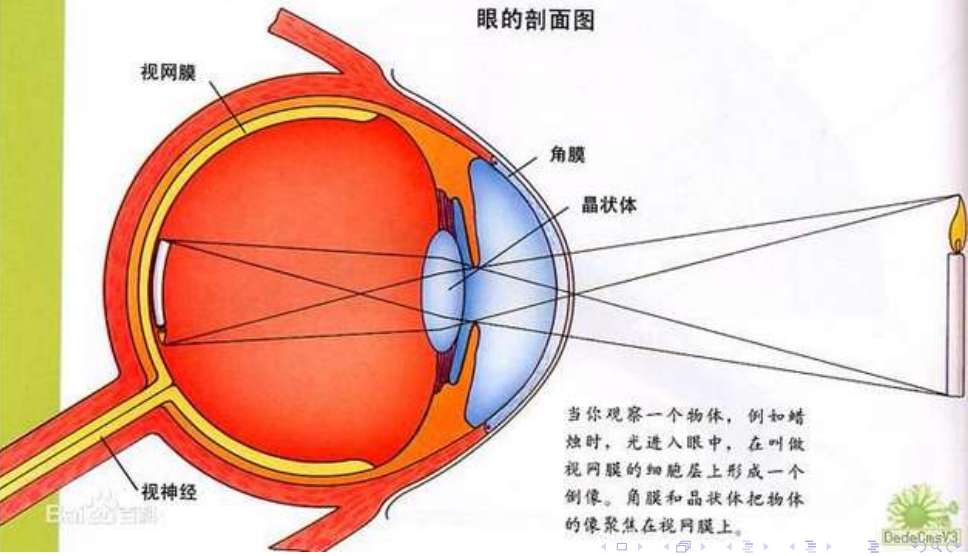
什么是图像？



视网膜成像：上亿个感光细胞

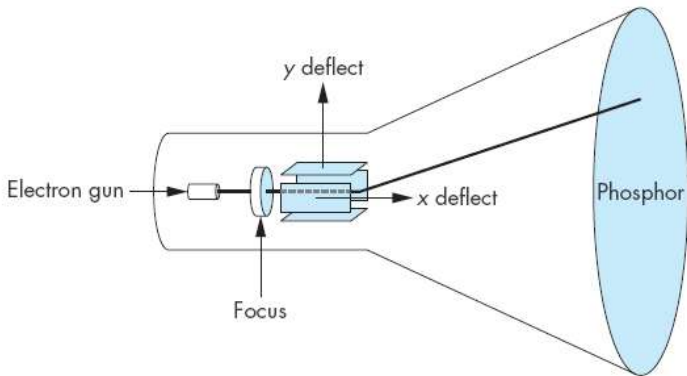


视网膜成像：上亿个感光细胞

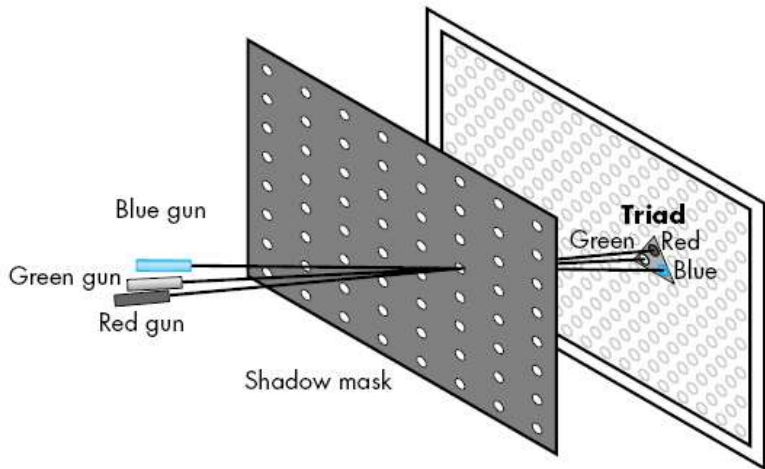


阴极射线显像管显示器 (CRT)

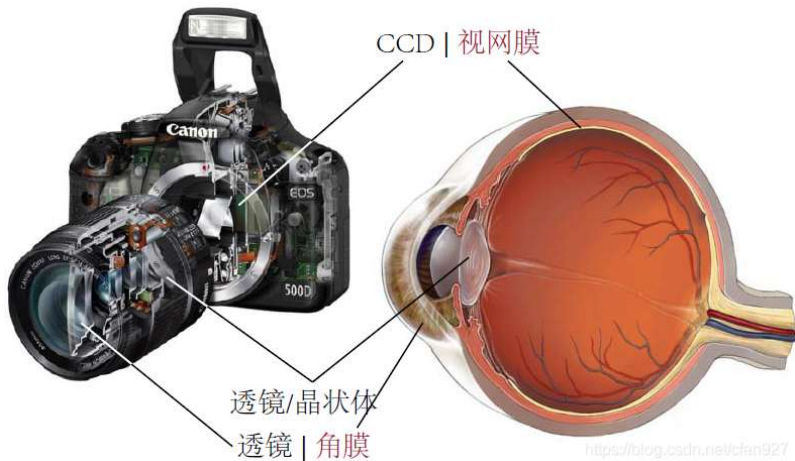
- 逐个将“点”打在屏幕上的相应位置
 - 逐行逐列扫描



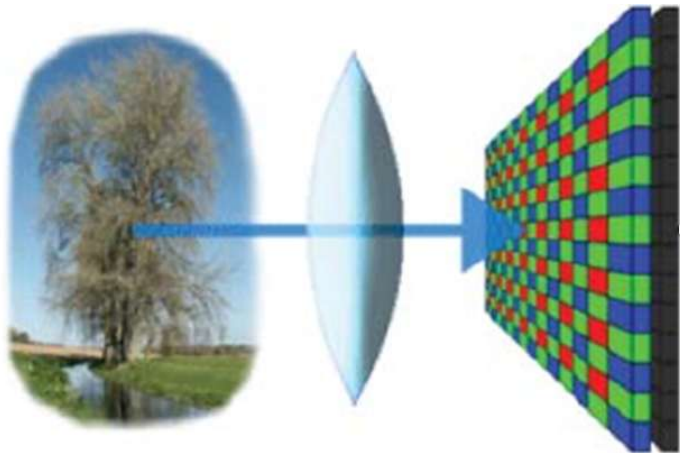
彩色CRT显示器（光栅显示器）



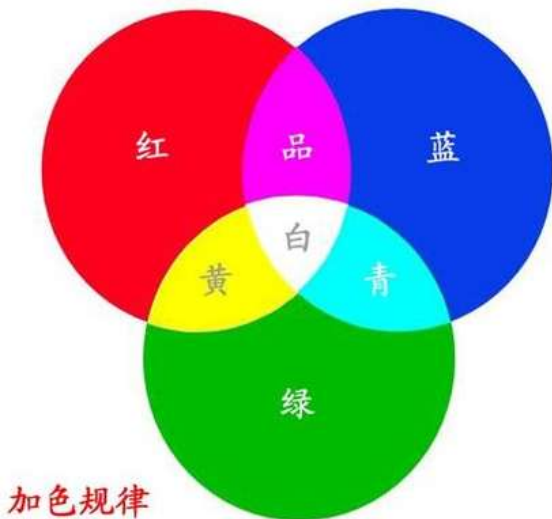
图像的抽象表达？



数字图像

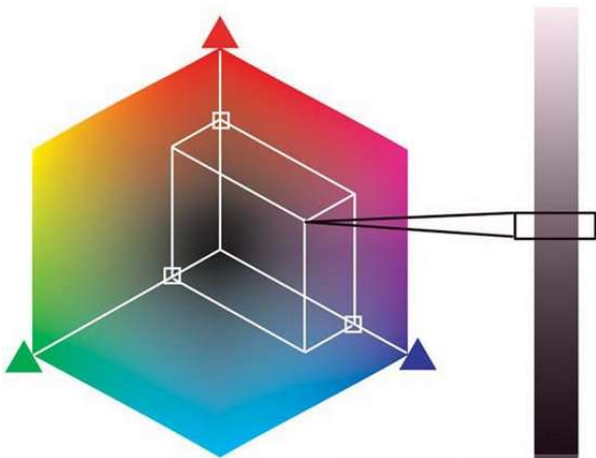


颜色三原色：红、绿、蓝



RGB颜色空间

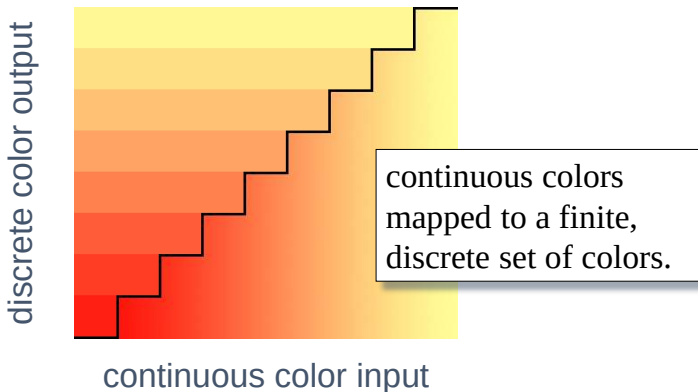
- $\text{Color} = a R + b G + c B$



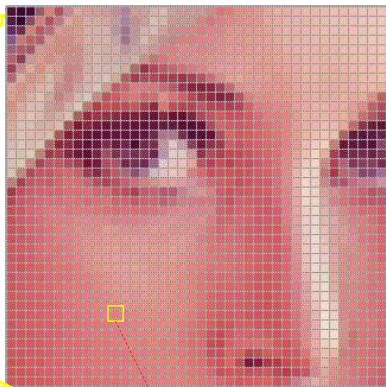
Color Spaces: Different Basis

- RGB
- CMY
- CIE XYZ
- $sl\alpha\beta$

数字图像：连续空间的离散采样



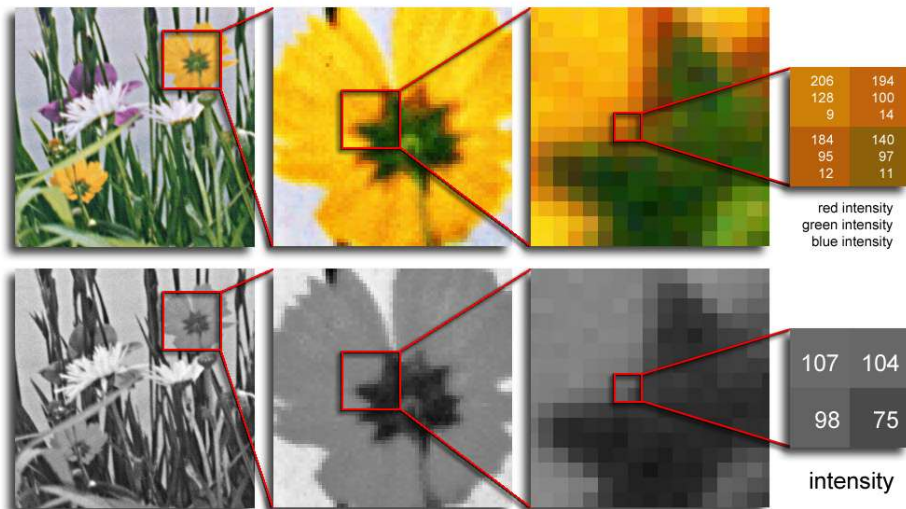
数字图像



$0.6 R + 0.3 G + 0.1 B$



Color Image and Gray Image

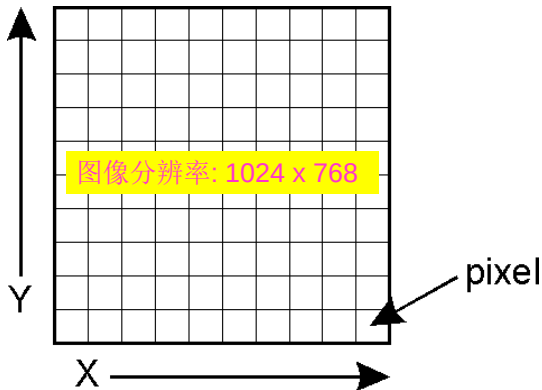


图像的抽象

- 我们需要哪些信息可以决定一幅图像？
 - 宽、高
 - 每个元素的颜色
- 所有像素的值！

图像的抽象

- 矩阵表达，分辨率



图像对象的接口？

- class CImage
- {
- public:
- int GetWidth();
- int Get Height();
- CColor GetPixelAt(int x, int y);
- void SetPixelAt(int x, int y, CColor c);
- };

图像编程

- C/C++
- QT
- OpenCV (Open Computer Vision)
- Matlab
- Matlab与C++混合编程

图像的种类

- Binary images (0 or 1)
- Gray images (0~255)
- Color images
 - indexed color images
 - full color images (24 bits per pixel, 8-red, 8-green, 8-blue))

A Binary Image (二值图像)



Gray Images (灰度图像)

- 8 bits per pixel



Full Color Images (全彩图像)

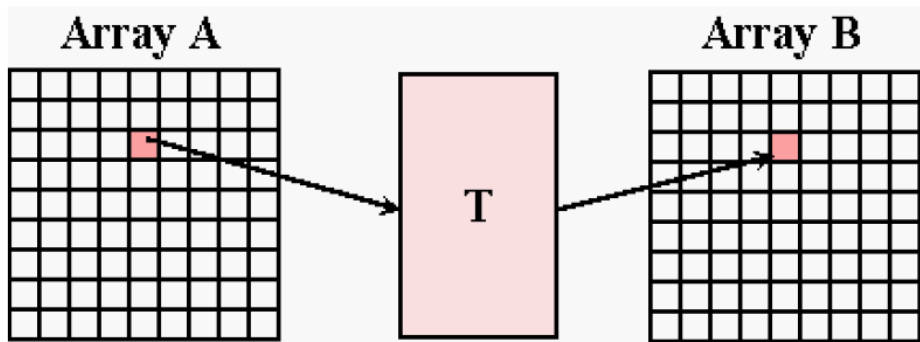
- 24 bits per pixel, and the three **channels** R G B are three gray images respectively
 - Each of the channels is encoded by 8 bits
- 32 bits
 - (R, G, B, a)

Color Components

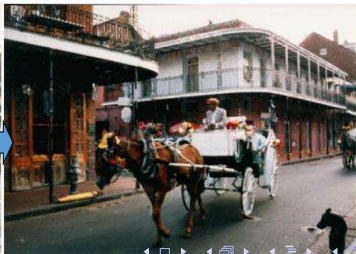


图像处理（图像变换）

$$B[x, y] = T[A[x, y]]$$



图像处理的例子



例子：彩色图像灰度化 (Color2Gray)

- 应用？



<https://www.rapidtables.com/convert/image/rgb-to-grayscale.html>

<https://ieeexplore.ieee.org/abstract/document/5445596>

<https://users.cs.northwestern.edu/~ago820/color2gray/>

问题：
彩色图像灰度化
哪个结果好？



输入图像



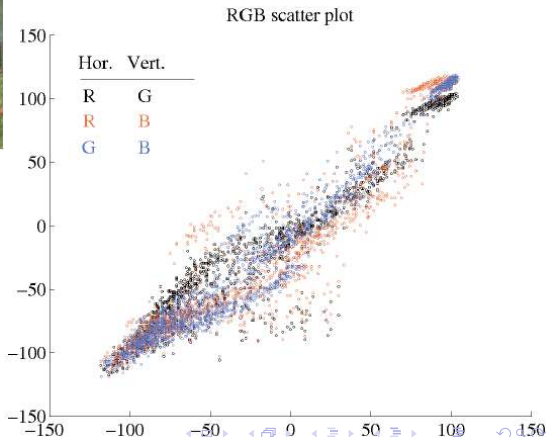
方法1结果



方法2结果



建模： 图像--3D空间的点集

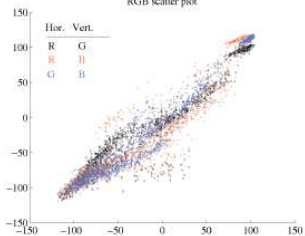


Color Spaces: Different Basis

- RGB
- CMY
- CIE XYZ
- $sl\alpha\beta$

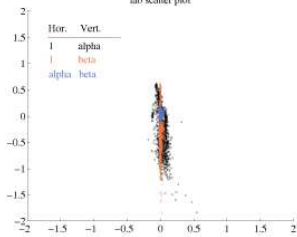


RGB scatter plot



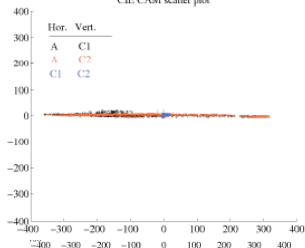
RGB

lab scatter plot



lab

CIE CAM scatter plot



CIE

Example: RGB $\rightarrow$ $l\alpha\beta$

$$\begin{bmatrix} L \\ M \\ S \end{bmatrix} = \begin{bmatrix} 0.3811 & 0.5783 & 0.0402 \\ 0.1967 & 0.7244 & 0.0782 \\ 0.0241 & 0.1288 & 0.8444 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

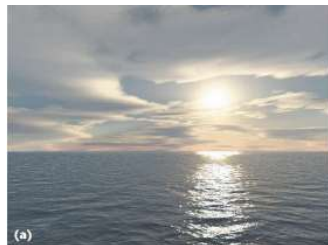
$$L = \log L$$

$$M = \log M$$

$$S = \log S$$

$$\begin{bmatrix} l \\ \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{3}} & 0 & 0 \\ 0 & \frac{1}{\sqrt{6}} & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & -2 \\ 1 & -1 & 0 \end{bmatrix} \begin{bmatrix} L \\ M \\ S \end{bmatrix}$$

例子2：图像颜色风格迁移（“学习”）



Reinhard et al. Color transfer between images. IEEE CG&A, 2001.
<https://ieeexplore.ieee.org/abstract/document/946629>

数学模型及算法

- 使点云位置、形状相似 (均值和方差变换)

$$l^* = l - \langle l \rangle$$

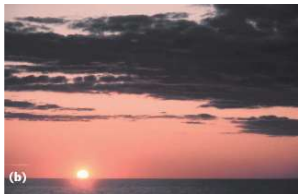
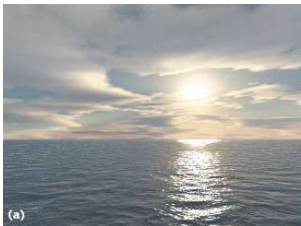
$$\alpha^* = \alpha - \langle \alpha \rangle$$

$$\beta^* = \beta - \langle \beta \rangle$$

$$l' = \frac{\sigma_l^l}{\sigma_s^l} l^*$$

$$\alpha' = \frac{\sigma_t^\alpha}{\sigma_s^\alpha} \alpha^*$$

$$\beta' = \frac{\sigma_t^\beta}{\sigma_s^\beta} \beta^*$$



例子3：灰度图像上色



+



=



+



=



作业1： 图像颜色变换

- 要求：
 - 任务：“彩色图像灰度化”、“图像颜色风格迁移”两个图像处理问题之一
 - 提交：作业报告+程序代码
 - 作业报告包括：数学模型、测试结果、实验分析等。
- 说明：
 - 关于两个问题网上有丰富的资源，可自行查找
 - 完成其中的1个问题，如有多种算法可进行比较
 - 鼓励有新想法或创新模型及解法
- 编程工具：C++，MATLAB，Python等均可
- 提交时间：2023年3月19日周日晚



中国科学技术大学
University of Science and Technology of China

谢谢！



中国科学技术大学
University of Science and Technology of China

数学建模 Mathematical Modeling

刘利刚

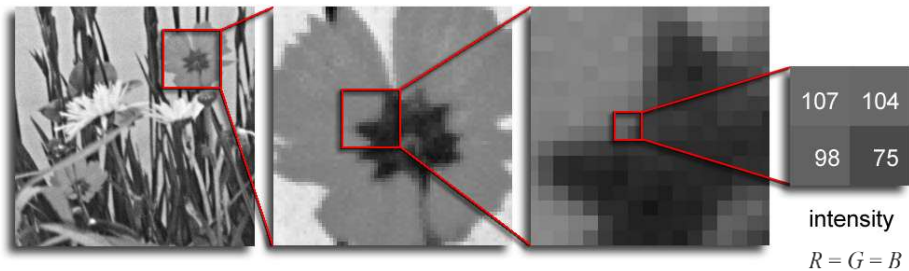
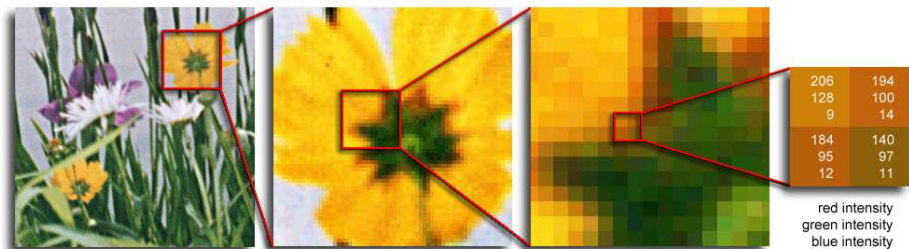
中国科学技术大学

高维数据分析模型

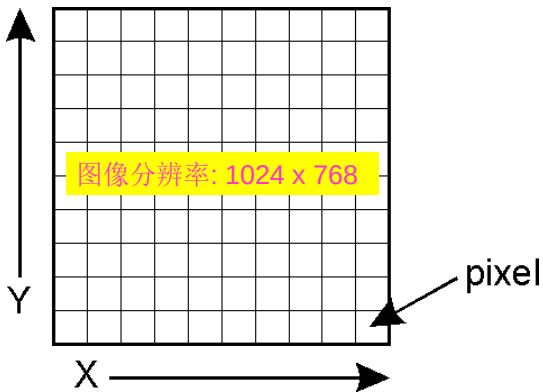
建模的第一步：抽象

- 抽象
 - 简化非本质的因素
 - 透过现象看本质
- 迭代
 - 将问题转化为已有问题
 - 组合、递归
- 方法论：20-80原则
 - Divide-and-conquer
 - Coarse-to-fine

数字图像的抽象



图像的数学表达

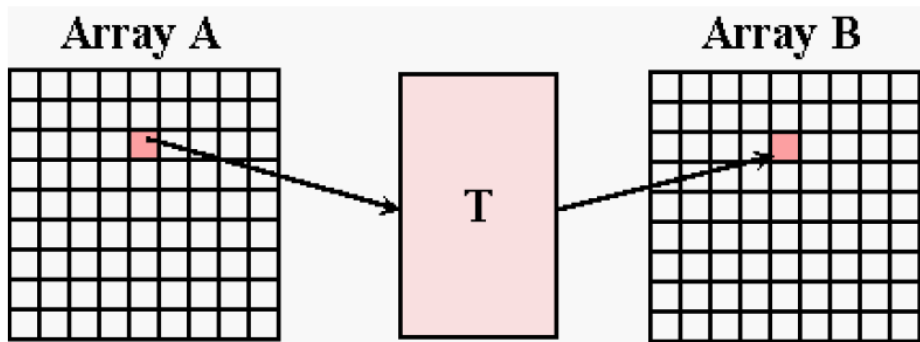


彩色图像的3个通道



图像的颜色变换

$$B[x, y] = T[A[x, y]]$$

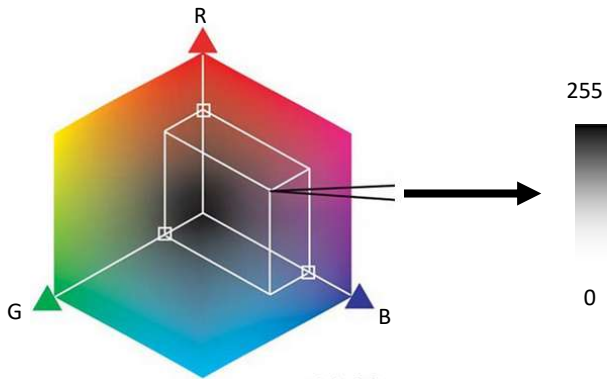


问题：彩色图像灰度化 (Color2Gray)

- 抽象：本质是个什么数学问题？



问题的描述



彩色图像灰度化：量化问题



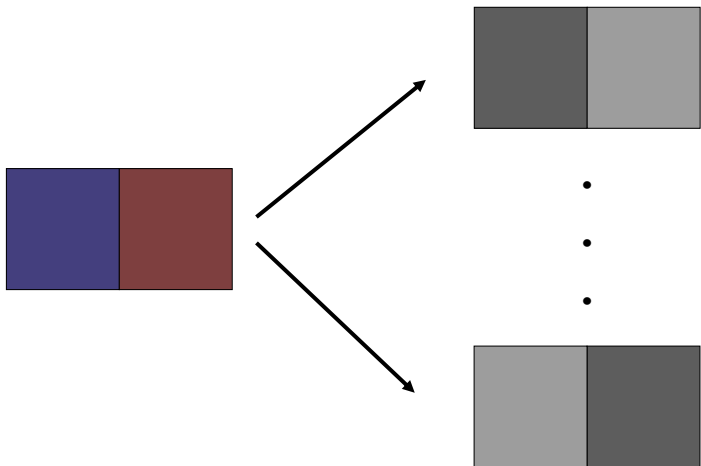
$$f: R^3 \rightarrow R^1$$

$$\begin{pmatrix} r \\ g \\ b \end{pmatrix} \rightarrow y$$

$$y = f(r, g, b)$$

实际生活中有哪些量化模型？

多解问题?



问题：彩色图像灰度化

是否有唯一解？



输入图像



方法1结果



方法2结果

1. 线性量化模型

pooling

$$f: R^n \rightarrow R^1$$
$$f(\mathbf{x}) = \sum a_i x_i = \boldsymbol{\alpha} \cdot \mathbf{x} \quad \mathbf{x} = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}$$

方法1:

$$R' = G' = B' = (R+G+B) / 3 = 0.333R + 0.333G + 0.333B$$

方法2:

$$R' = G' = B' = 0.2126R + 0.7152G + 0.0722B$$

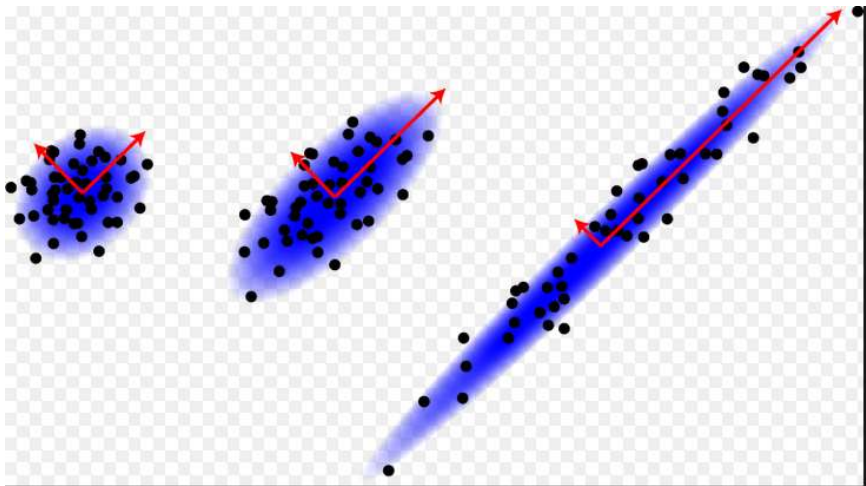
方法3:

$$R' = G' = B' = 0.299R + 0.587G + 0.114B$$

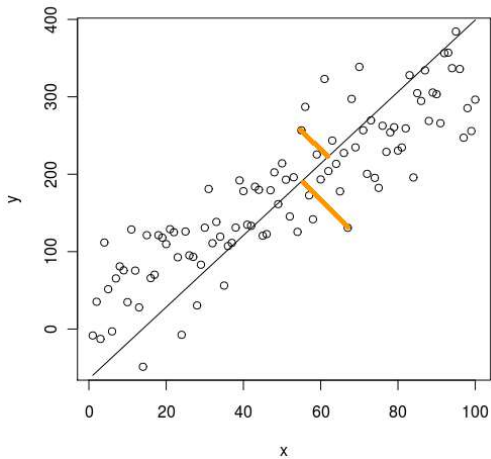
方法4:

$$y = \max\{R, G, B\}$$

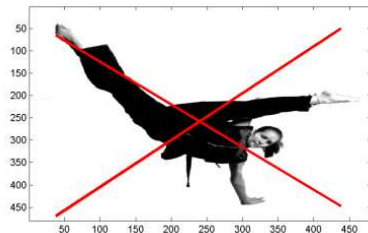
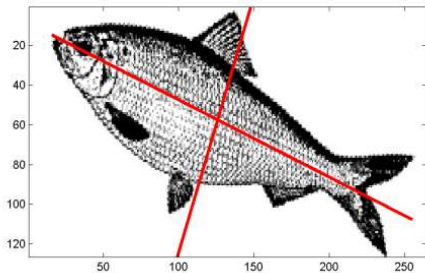
2. 线性降维模型（主成分分析PCA）



3D $\rightarrow$ 1D



Principal Component Analysis (PCA)



Principal Component Analysis

- PCA is the most commonly used dimension reduction technique.
- (Also called the Karhunen-Loeve transform).
- PCA – data samples $x_1, \dots, x_N$.
- Compute the mean $\mu = (1/N) \sum_{i=1}^N x_i$.
- Compute the covariance:

$$K = (1/N) \sum_{i=1}^N (x_i - \mu)(x_i - \mu)^T.$$

Principal Component Analysis

- Compute the eigenvalues λ and eigenvectors e of the matrix K .

- Solve $Ke = \lambda e$.

- Order them by magnitude:

$$\lambda_1 \geq \lambda_2 \geq \dots \lambda_N.$$

- PCA reduces the dimension by keeping direction e such that $\lambda > T$.

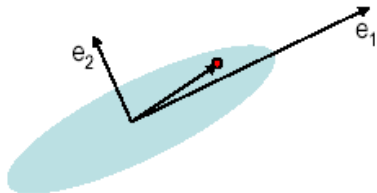
Principal Component Analysis

- For many datasets, most of the eigenvalues λ are negligible and can be discarded.

The eigenvalue λ measures the variation
In the direction e

Example:

$$\lambda_1 \neq 0, \lambda_2 = 0.$$



Principal Component Analysis

- Project the data onto the selected eigenvectors:

$$x \mapsto \mu + \sum_{i=1}^M a_i e_i$$

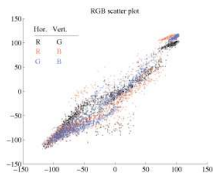
- Where

$$a_i = (x - \mu) \cdot e_i.$$

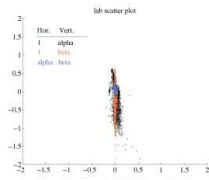
- $\frac{\sum_{i=1}^M \lambda_i}{\sum_{i=1}^N \lambda_i}$ is the proportion of data covered by the first M eigenvalues.

3. 空间变换模型：颜色空间

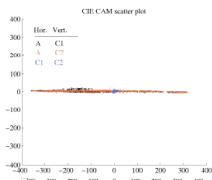
- RGB
- CMY
- CIE XYZ
- $l\alpha\beta$



RGB



$l\alpha\beta$



CIE

Example: RGB $\rightarrow$ $l\alpha\beta$

$$\begin{bmatrix} L \\ M \\ S \end{bmatrix} = \begin{bmatrix} 0.3811 & 0.5783 & 0.0402 \\ 0.1967 & 0.7244 & 0.0782 \\ 0.0241 & 0.1288 & 0.8444 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

$$L = \log L$$

$$M = \log M$$

$$S = \log S$$

$$\begin{bmatrix} l \\ \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{3}} & 0 & 0 \\ 0 & \frac{1}{\sqrt{6}} & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & -2 \\ 1 & -1 & 0 \end{bmatrix} \begin{bmatrix} L \\ M \\ S \end{bmatrix}$$

Example: RGB $\rightarrow$ HSV

The R, G, B values are divided by 255 to change the range from 0..255 to 0..1:

$$R' = R/255$$

$$G' = G/255$$

$$B' = B/255$$

$$C_{max} = \max(R', G', B')$$

$$C_{min} = \min(R', G', B')$$

$$\Delta = C_{max} - C_{min}$$

Hue calculation:

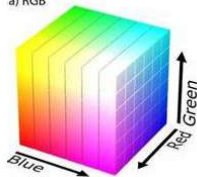
$$H = \begin{cases} 60^\circ \times \left(\frac{G' - B'}{\Delta} \bmod 6 \right) & , C_{max} = R' \\ 60^\circ \times \left(\frac{B' - R'}{\Delta} + 2 \right) & , C_{max} = G' \\ 60^\circ \times \left(\frac{R' - G'}{\Delta} + 4 \right) & , C_{max} = B' \end{cases}$$

Saturation calculation:

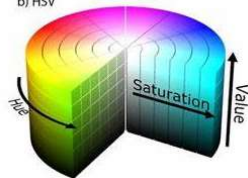
$$S = \begin{cases} 0 & , C_{max} = 0 \\ \frac{\Delta}{C_{max}} & , C_{max} \neq 0 \end{cases}$$

Value calculation: $V = C_{max}$

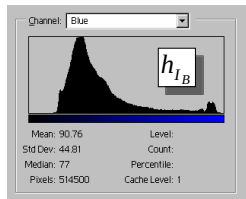
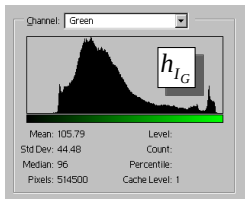
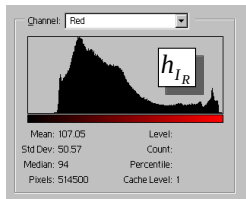
a) RGB



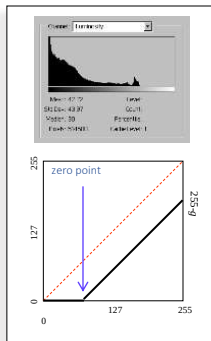
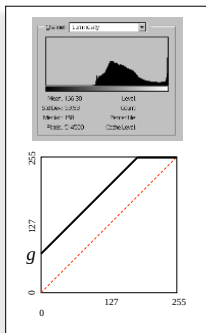
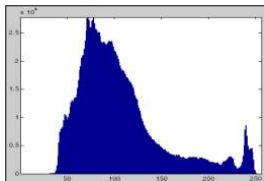
b) HSV



图像的颜色直方图



直方图变换





Color



CIE CAM 97



Photoshop LAB



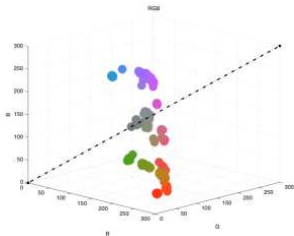
CIE XYZ



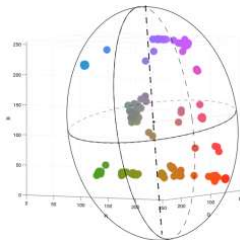
YCrCb

4. 非线性模型

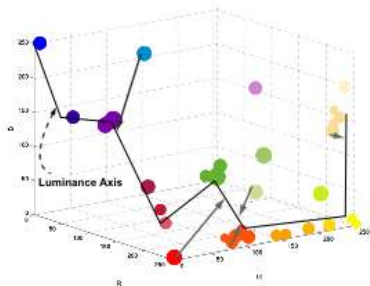
简单线性
投影变换



PCA线性
投影变换



非线性
投影变换



5. 非线性模型：保持视觉对比度



Color



New Algorithm



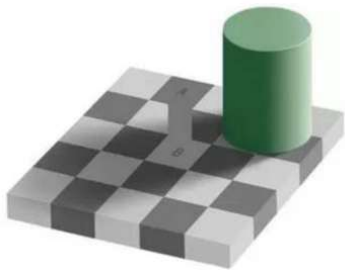
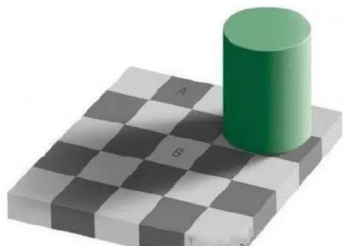
Grayscale

Gooch et al. Color2Gray: Saliency-Preserving Color Removal. Siggraph 2005.

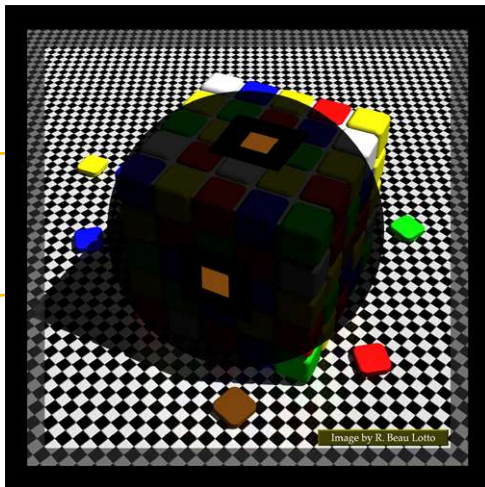
<https://users.cs.northwestern.edu/~ago820/color2gray/>

色彩产生的因素

- 客观物理因素
- 视觉感知因素



视觉感知的差异



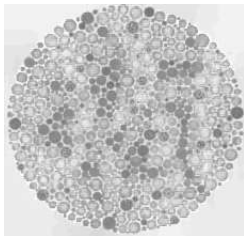
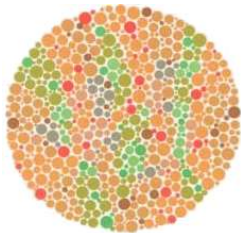
Color Illusion by
Lotto and Purves

<http://www.lottolab.org>

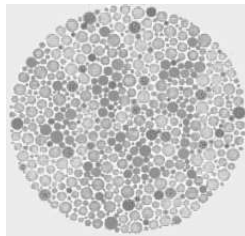
优化模型

$$\min \sum \sum \left((g_i - g_j) - \delta_{i,j} \right)^2$$

参数控制 μ : 邻域大小



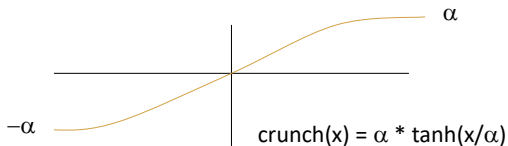
$\mu = 16$



$\mu = \text{entire image}$



参数控制 α : 色调差异



$\alpha = 5$



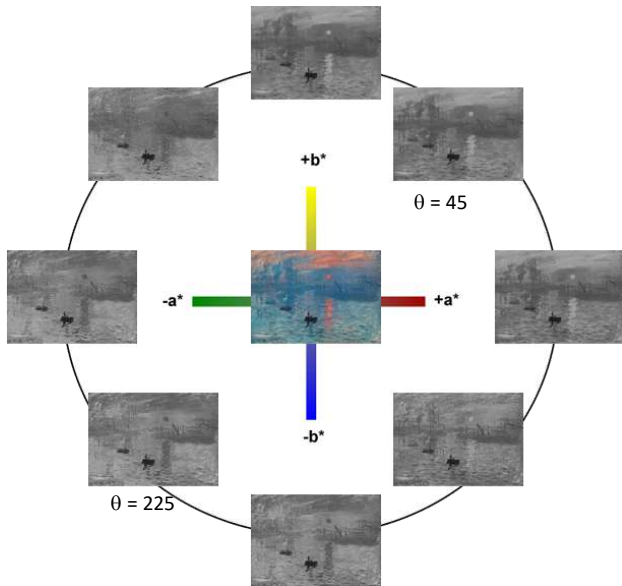
$\alpha = 10$

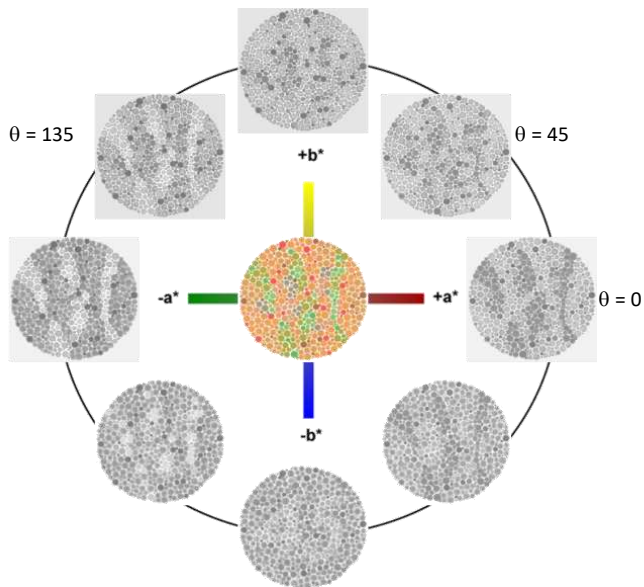
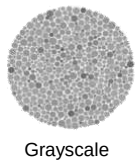


$\alpha = 25$



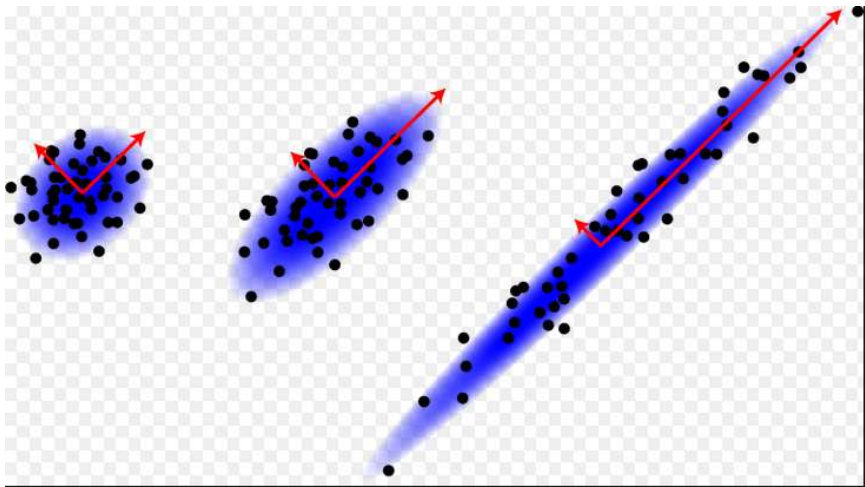
Photoshop Grayscale



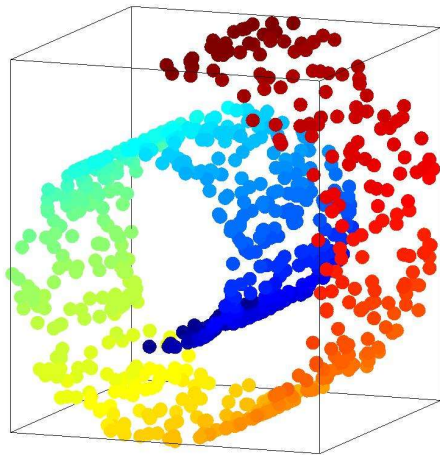


6. 高维数据分析模型： 非线性降维/流形学习

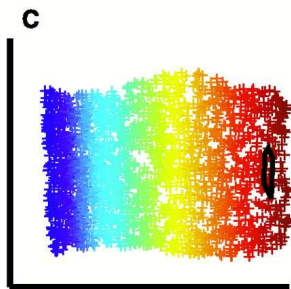
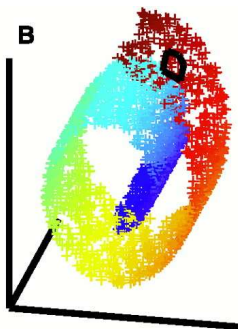
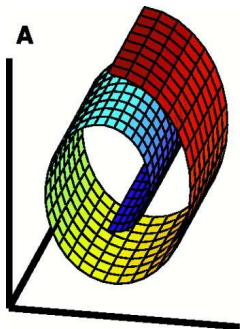
Ambient points always has low-dimensional structure!



How about these points?



2D manifold in 3D



数据的观察维数

- 任何数据都是某个高维空间的一个数（向量）



地震波、蛋白质结构
基因序列…

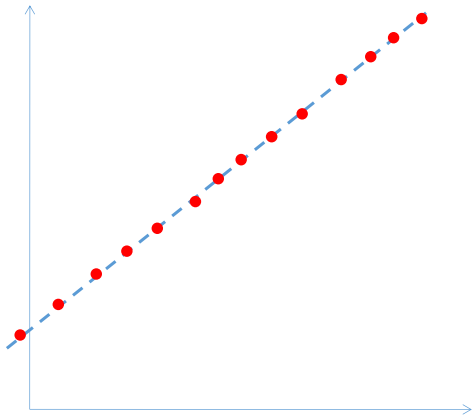
“特征”



$$X = (x_1, x_2, \dots, x_N) \in R^N$$

数据的维数真有那么高吗？

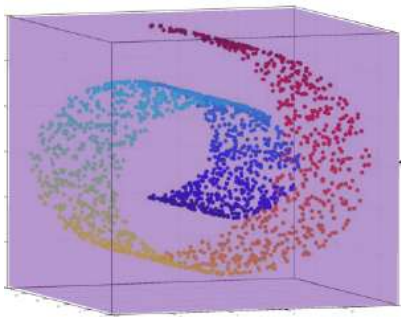
61.562198	21.702432
69.589896	23.116031
69.689998	30.561331
63.219798	29.909931
78.298497	24.213831
77.958698	31.596431
77.822498	39.146431
70.139797	38.301631
64.147498	37.712831
96.651898	26.682431
95.903298	34.240231
86.893197	32.779931
87.502498	25.348631
95.225397	41.707531
86.403297	40.250431
94.292697	56.312032
85.767398	55.206031
86.029197	47.734331
77.722297	46.807731



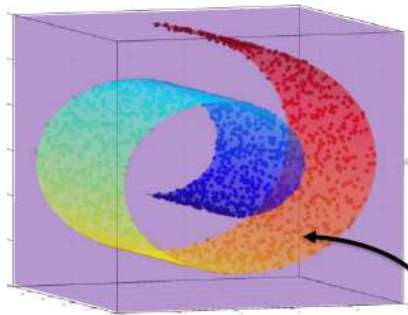
真实的维数

观察的维数

数据的真实维数可能很低



观察的维数



真实的维数

目标：高维数据中的低维结构

- 数据 >> 信息
- 数据 = 信息 + 不相关数据
- 目标：从不完全数据中“抽取”信息（结构）



“数据科学”

- 两个目标：
 - 数据表达 (representation)
 - 内在结构 (intrinsic structures)
- 两个任务：
 - 聚类 (clustering) : unsupervised
 - 分类 (classification) : supervised
 - Regression/fitting, deep learning...

数据的表达：特征/编码

- 量化
- 压缩
- 数据变换：从不同的空间来看数据

数据变换的例子: Wavelets

- Feature map

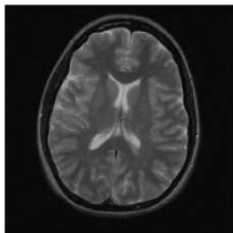
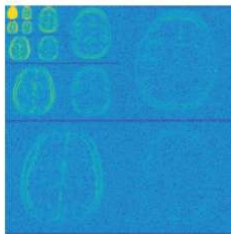
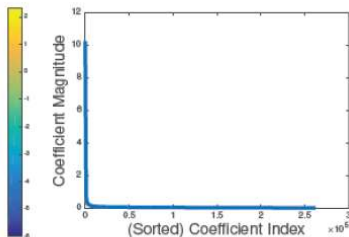


image $I(v)$

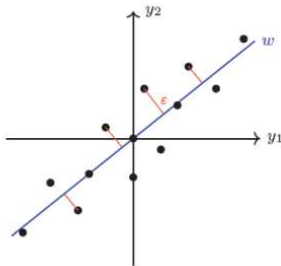
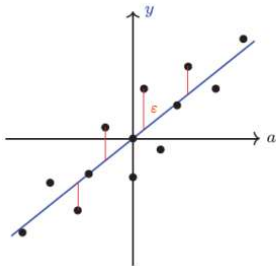
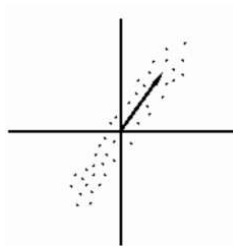


wavelet coefficients x : $I = \Psi[x]$.



Principal Component Analysis (PCA)

- Find linear subspace projection which preserves the data locations (under quadratic error)

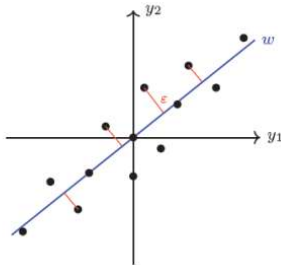
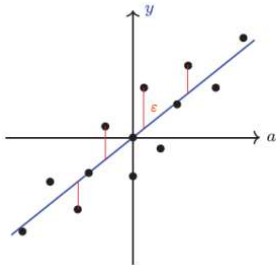
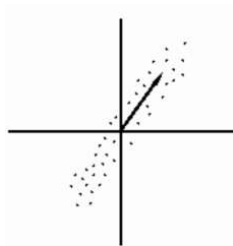


内在结构

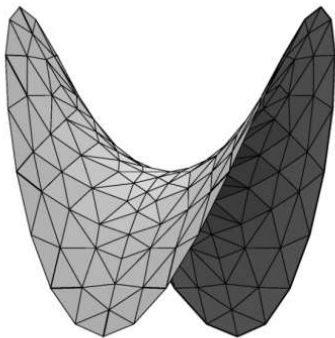
- 哪些数据是一组的（相似的）？
- 相似性的“度量”？
 - 距离
 - 线性空间（同一组基表达）
 - 非线性空间（流形空间）

Principal Component Analysis (PCA)

- Find linear subspace projection which preserves the data locations (under quadratic error)

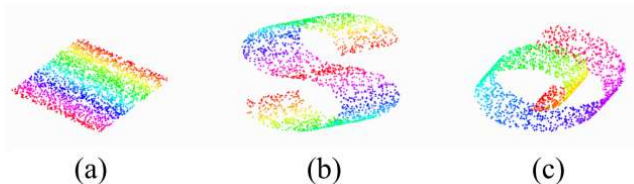


Manifolds



low-D surface
embedded in
high-D space

Manifolds



- PCA : works for (a)
- Doesn't do much good for (b) or (c)
 - Linear subspace doesn't explain it well
- What do we mean by “consistent locations”?
 - Preserve local relationships and structure
 - One possibility : preserve distances

Manifold learning

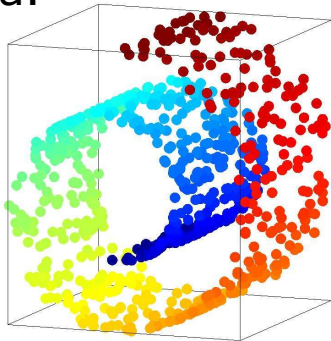
Find a low-D basis for describing high-D data.

$$X \approx X'$$

s.t.

$$\dim(X') \ll \dim(X)$$

uncovers the **intrinsic** dimensionality



Dimension Reduction

why do manifold learning?

1. data compression
2. "curse of dimensionality"
3. de-noising
4. visualization
5. reasonable distance metrics

Manifold Learning & Dimension Reduction

- Given a set of data points

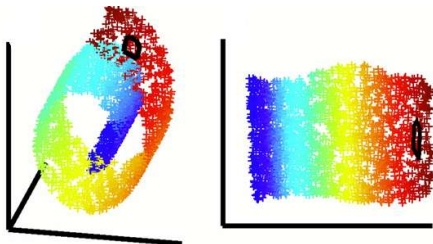
$$x_1, \dots, x_N \in \mathbb{R}^m$$

-

$$x_i = f(\tau_i) + \varepsilon_i, i = 1, \dots, N$$

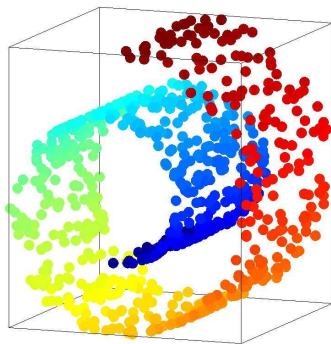
- Dimension Reduction: Find

- Manifold Learning: Find f τ_i



Key Difficulty

- data points **unorganized** (no adjacency relationship known)

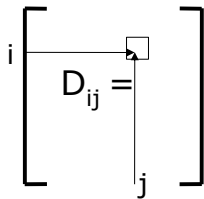


If we knew all pairwise distances...

	Chicago	Raleigh	Boston	Seattle	S.F.	Austin	Orlando
Chicago	0						
Raleigh	641	0					
Boston	851	608	0				
Seattle	1733	2363	2488	0			
S.F.	1855	2406	2696	684	0		
Austin	972	1167	1691	1764	1495	0	
Orlando	994	520	1105	2565	2458	1015	0

Multidimensional Scaling (MDS)

For n data points, and a distance matrix D ,



A diagram showing a distance matrix D enclosed in large square brackets. A horizontal line extends from the letter i on the left to a small square box inside the matrix. A vertical line extends from the letter j at the bottom to the same small square box. Inside this box, the text $D_{ij} =$ is written, indicating the distance between points i and j .

...we can construct a m -dimensional space to preserve inter-point distances by using the top eigenvectors of D scaled by their eigenvalues

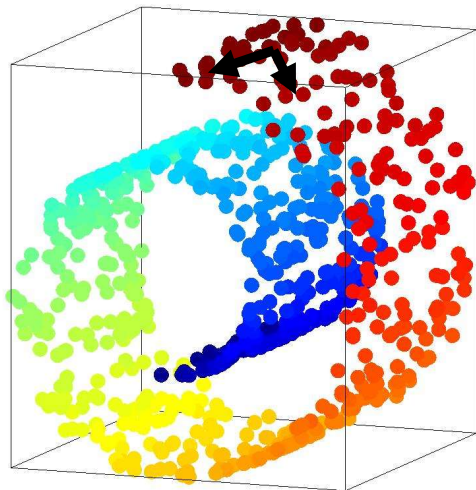
MDS result in 2D



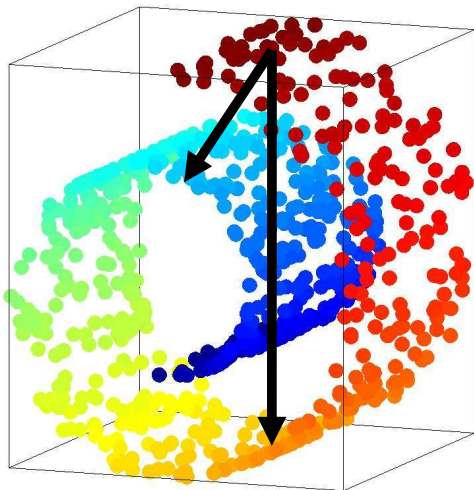
Actual plot of cities



Don't know distances



Don't know distnaces



Taxonomy of Dimension Reduction Techniques

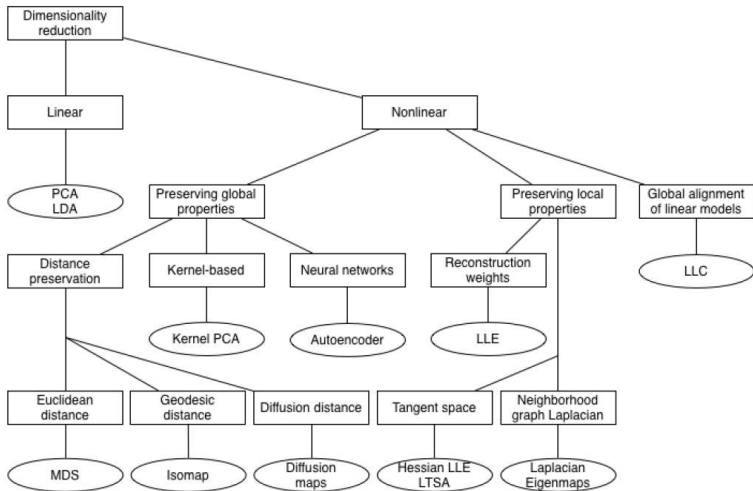


Figure From: L.J.P. van der Maaten et. al., *Dimensionality Reduction: A Comparative Review*

Methods of Manifold Learning

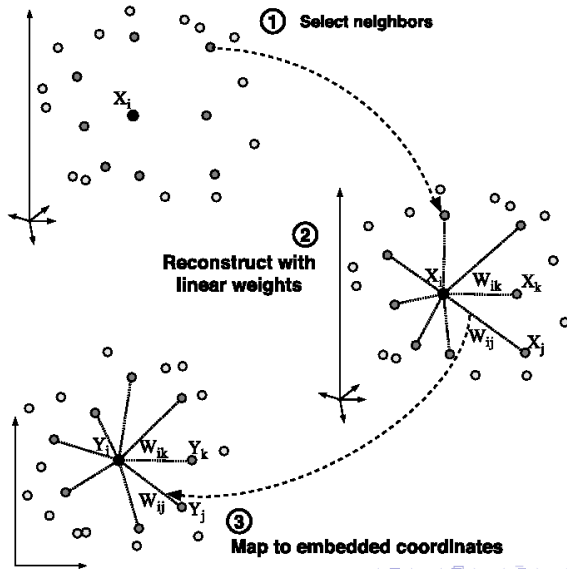
1. Locally Linear Embedding (LLE)

[Science, 2000]

Find a mapping to preserve
local linear relationships
between neighbors



Locally Linear Embedding



LLE: Two key steps

1. Find weight matrix W of linear coefficients:

$$\mathcal{E}(W) = \sum_i \left| \vec{X}_i - \sum_j W_{ij} \vec{X}_j \right|^2$$

Enforce sum-to-one constraint.

LLE: Two key steps

2. Find projected vectors Y to minimize reconstruction error

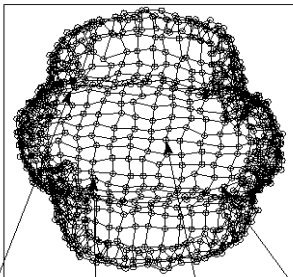
$$\Phi(Y) = \sum_i \left| \vec{Y}_i - \sum_j W_{ij} \vec{Y}_j \right|^2$$

must solve for whole dataset simultaneously

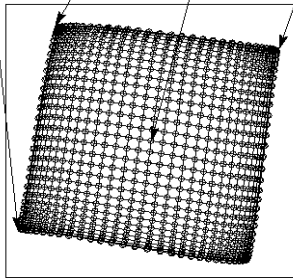
LLE: Result

preserves local
topology

PCA



LLE



LLE: pro and con

- no local minima, one free parameter
- incremental & fast
- simple linear algebra operations
- can distort global structure

2. Isomap

[Science, 2000]

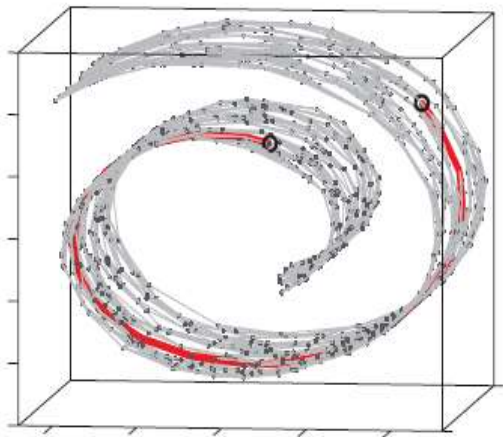
- Build a data graph G .
 - (u,v) is an edge iff $SSD(u,v)$ is small
- For any two points, we approximate the distance between them with the “shortest path” on G

Isomap

1. Build a sparse graph with K-nearest neighbors

$$D_g = \begin{bmatrix} \text{blue oval} \end{bmatrix}$$

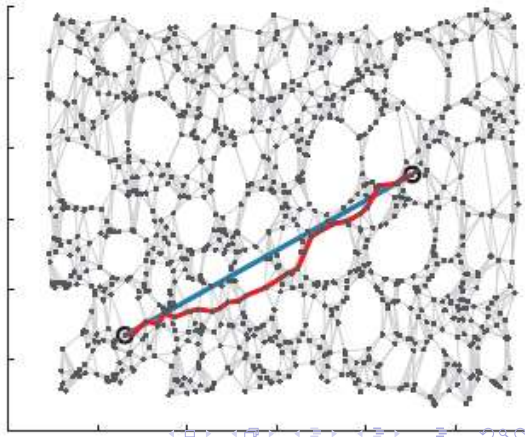
(distance matrix is sparse)



Isomap

2. Infer other interpoint distances by finding shortest paths on the graph (Dijkstra's algorithm).

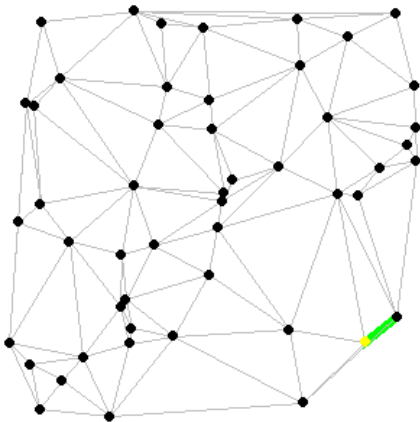
$$D_g = \begin{bmatrix} \text{blue oval} \end{bmatrix}$$



Isomap

shortest-distance on a graph is easy to compute

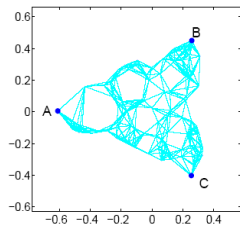
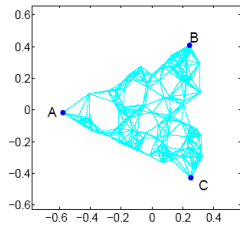
Dijkstra's algorithm



Isomap: pro and con

- preserves global structure
- few free parameters
- sensitive to noise, noise edges
- computationally expensive (dense matrix eigen-reduction)

Leakage problem



LLE和Isomap有效的原因

- 它们都是非参数的方法，不需要对流形的很多的参数假设.
- 它们是非线性的方法，都基于流形的内在几何结构，更能体现现实中数据的本质.
- 它们的求解简单，都转化为求解特征值问题，而不需要用迭代算法.

流形学习问题探讨

- 求解方法
 - 对嵌入映射或者低维流形作出某种特定的假设, 或者以保持高维数据的某种性质不变为目标.
 - 将问题转化为求解优化问题.
 - 提供有效的算法.
- 如何确定低维目标空间的维数?
- 当采样数据很稀疏时, 怎样进行有效的学习?
- 将统计学习理论引入流形学习对其泛化性能进行研究



中国科学技术大学
University of Science and Technology of China

谢谢！



中国科学技术大学
University of Science and Technology of China

数学建模 Mathematical Modeling

刘利刚

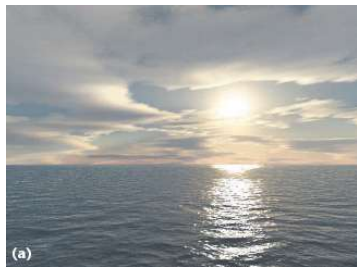
中国科学技术大学

高维数据变换模型

彩色图像颜色变换



彩色图像颜色变换



$$f: R^3 \rightarrow R^3$$

$$\begin{pmatrix} r \\ g \\ b \end{pmatrix} \rightarrow \begin{pmatrix} r' \\ g' \\ b' \end{pmatrix}$$

例子1: 色彩模式调节



- gamma



- brightness



original



+ brightness



+ gamma



histogram mod



- contrast



original

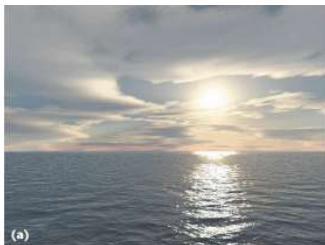


+ contrast



histogram EQ

例子2: Color Transfer



算法

- 使点云中心、形状相似（均值和方差变换）

$$l^* = l - \langle l \rangle$$

$$\alpha^* = \alpha - \langle \alpha \rangle$$

$$\beta^* = \beta - \langle \beta \rangle$$

$$l' = \frac{\sigma_t^l}{\sigma_s^l} l^*$$

$$\alpha' = \frac{\sigma_t^\alpha}{\sigma_s^\alpha} \alpha^*$$

$$\beta' = \frac{\sigma_t^\beta}{\sigma_s^\beta} \beta^*$$



例子3: Color Harmonization



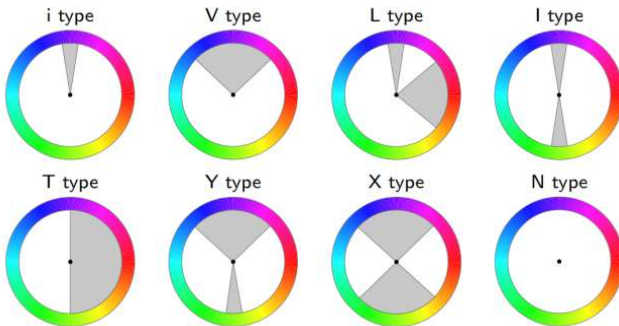
original image

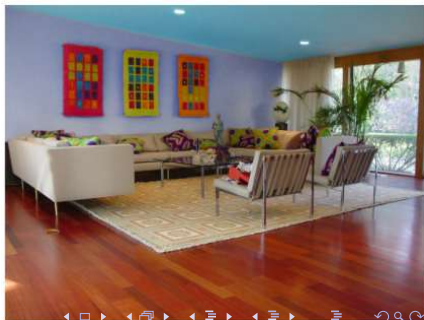


harmonized image

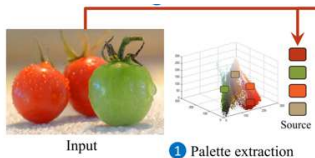
Daniel Cohen-Or et al. Color Harmonization. Siggraph 2006.

Harmonic templates on the hue wheel

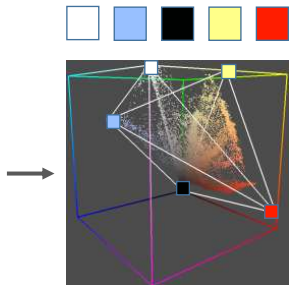




例子4：基于调色板的重着色



输入图像



简化的 RGB 凸包
(调色板)

Jianchao Tan et al. Palette-based image decomposition, harmonization, and color transfer. arXiv preprint arXiv:1804.01225 (2018).

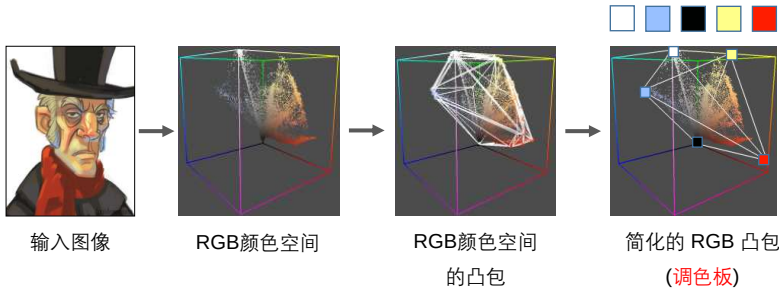
Jianchao Tan et al. Efficient palette-based decomposition and recoloring of images via RGBXY-space geometry. Siggraph Asia 2018.

Yili Wang et al. An Improved Geometric Approach for Palettebased Image Decomposition and Recoloring. Computer Graphics Forum, 2019.

基于几何凸包的调色板提取

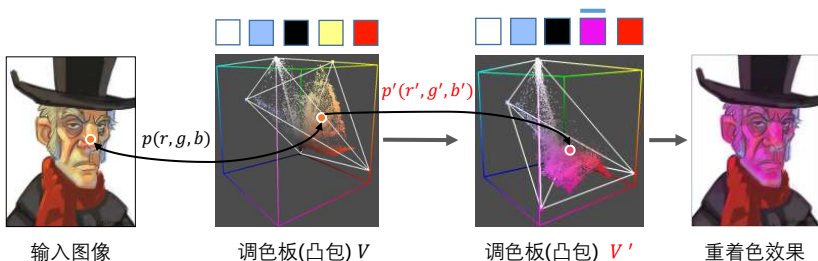
[Tan et al. 2017; Tan et al. 2018; Wang et al. 2019]

- 调色板提取方法
 - 求所有点的RGB 3D凸包
 - 凸包简化：减少顶点数



基于几何凸包的图像分解和重着色

- 图像分解和重着色
 - 类似于几何/动画中的插值系数和几何变形

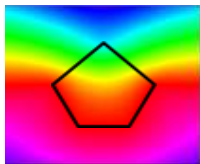
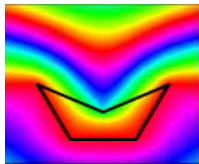
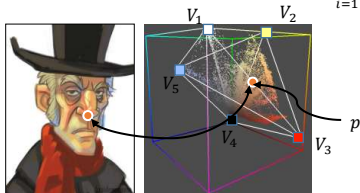


- Mixing weights: $p = \sum_{i=1}^{|V|} w_i V_i \quad 1 = \sum_{i=1}^{|V|} w_i$ Recoloring: $p' = \sum_{i=1}^{|V|} w_i V'_i$

基于几何凸包的图像分解和重着色

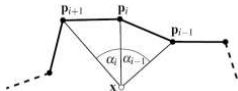
- Mixing weights 应满足

$$p = \sum_{i=1}^{|V|} w_i V_i \quad 1 = \sum_{i=1}^{|V|} w_i$$

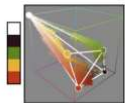
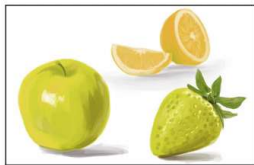


- 使用MVC (Mean Value Coordinates)

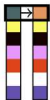
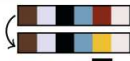
- 二维情况: $w_i = \frac{u_i}{\sum_{i=1}^m u_i}$, $u_i = \frac{\tan(\frac{\alpha_{i-1}}{2}) + \tan(\frac{\alpha_i}{2})}{\|p_i - x\|}$
- 非常光滑: 内部处处可导, 外部处处连续



实验结果



实验结果



更多实验结果



基于调色板的视频编辑重着色

- 挑战

- 视频颜色是随时间变化的，用户编辑也可以是随时间变化的

- 视频调色板的预期性质

- 具有较少数量的控制顶点、较低的重建误差、较好的紧凑性
 - 具有较好的帧间连续性，能处理复杂的颜色时间变化



Input video



Palette



Recolored video

例子5：保持流形结构的变换



User input

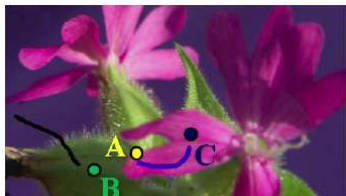


Photoshop

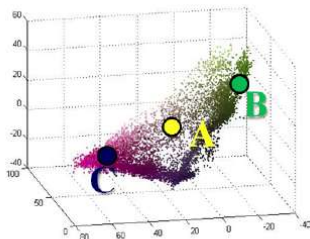


This method

Manifold Preserving Propagation



(a)



(b)



(c) [Xu et al. 2009]



(d) [Farbman et al. 2010]

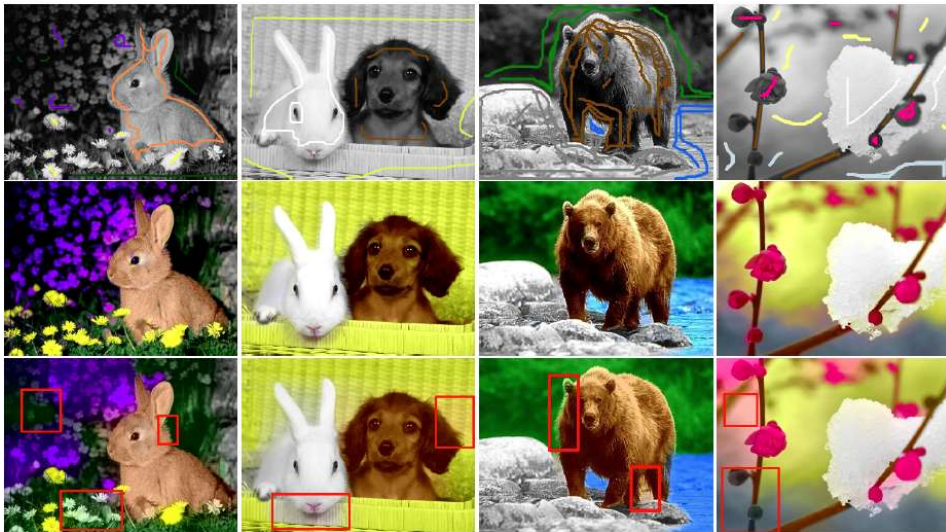


(e) [Li et al. 2010]



(f) Our result

More Results



灰度图像上色

Colorization

灰度图像上色

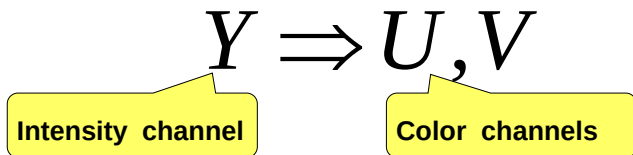


$$f: R^1 \rightarrow R^3$$

例1：交互式上色



Propagation using Optimization



“Neighboring pixels with similar intensities should have similar colors”

Propagation using Optimization

$$Y \Rightarrow U, V$$

Intensity channel

Color channels

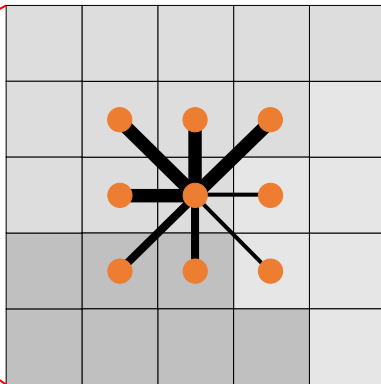
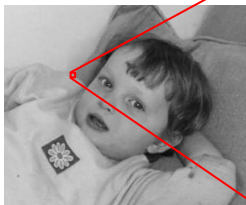
$$J(U) = \sum_r \left(U(r) - \sum_{s \in N(r)} w_{rs} U(s) \right)^2$$

- Minimize difference between color at a pixel and an *affinity-weighted average* of the neighbors

Affinity Functions

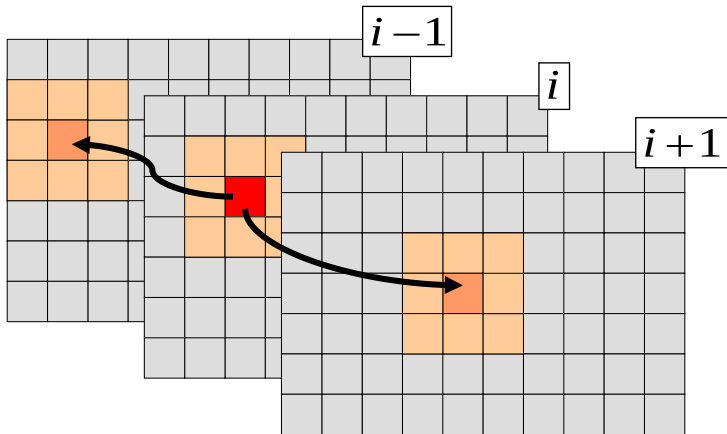
$$w_{rs} \propto e^{-(Y(r)-Y(s))^2 / \sigma_r^2}$$

σ_r proportional to local variance



Affinity Functions in Space-Time

$$w_{rs} \propto e^{-(Y(r)-Y(s))^2 / 2\sigma_r^2}$$



Minimizing cost function

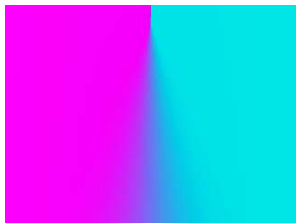
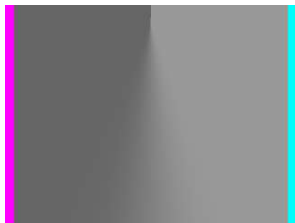
Minimize:

$$J(U) = \sum_r \left(U(r) - \sum_{s \in N(r)} w_{rs} U(s) \right)^2$$

Subject to *labeling constraints*

Since cost is quadratic, minimum can be found by solving sparse system of linear equations.

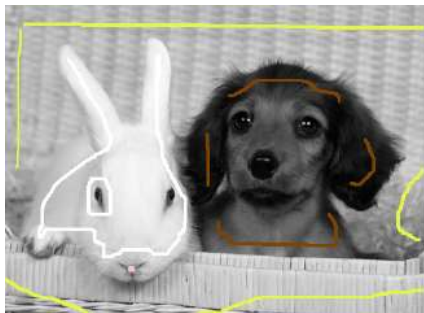
Color Interpolation



实验结果



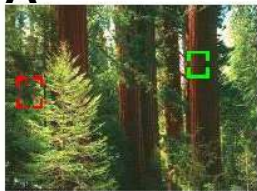
例子2：保持流形结构的方法



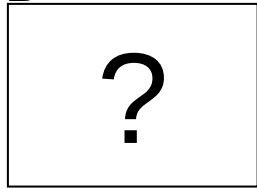
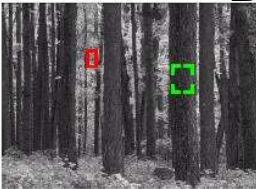
Colorization

例子3：基于类比的图像上色

A : A'



B : B'



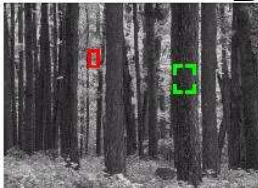
Hertzmann et al. 2001, Welsh et al. 2002

Colorization by Analogy

A : A'

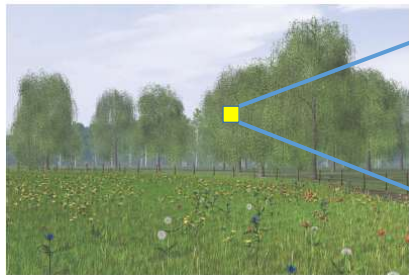


B : B'



Hertzmann et al. 2001, Welsh et al. 2002

Feature Space?



(r, g, b)

(r, g, b, u, v)



中国科学技术大学
University of Science and Technology of China

谢谢！



中国科学技术大学
University of Science and Technology of China

数学建模 Mathematical Modeling

刘利刚

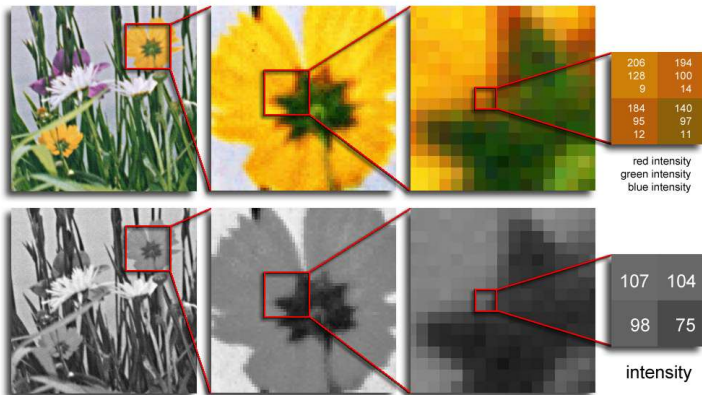
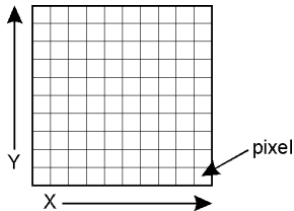
中国科学技术大学

矩阵分解模型

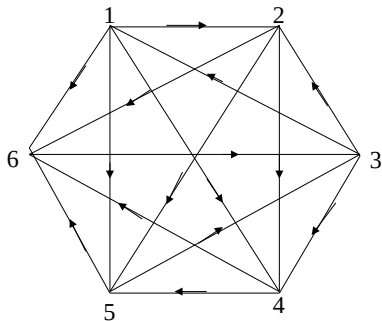
协方差矩阵

- 方差：单个随机变量的离散程度
- 协方差：两个随机变量的相似程度
- 协方差矩阵

光栅图像：矩阵



图的邻接矩阵



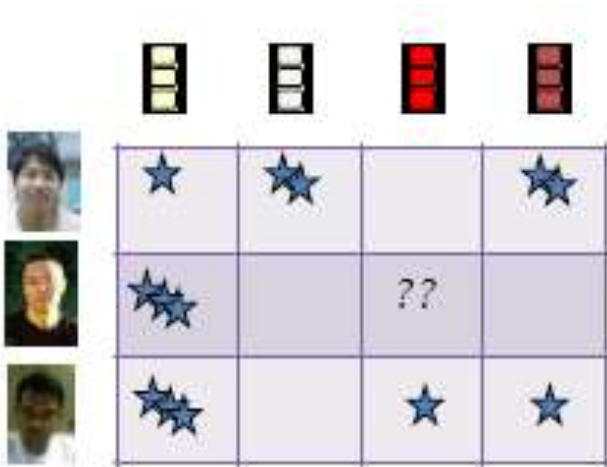
$$A = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

问题： 用户-物品评分

User /item	1	2	3	4	5	
1	5	4	4.5	?	3.9	
2	?	4.5	?	4.5	?	
3	4.5	?	4.4	4	4	
4	?	4.8	?	?	4.5	
5	4	?	4.5	5	?	
.....						

<http://blog.csdn.net/zhongkejijiang>

Netflix Challenge



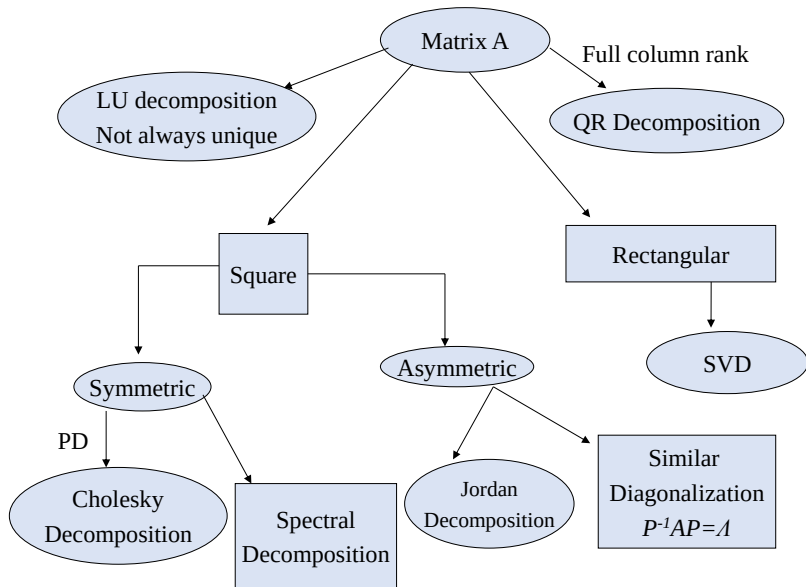
理解矩阵

- 变换
- 线性变换
- 基函数的转换

Matrix Decomposition (Factorization)

- LU decomposition
- QR decomposition
- Cholesky decomposition
- Jordan Decomposition
- Spectral decomposition (Eigendecomposition)
- Singular value decomposition (SVD)
- Low rank decomposition
- ...

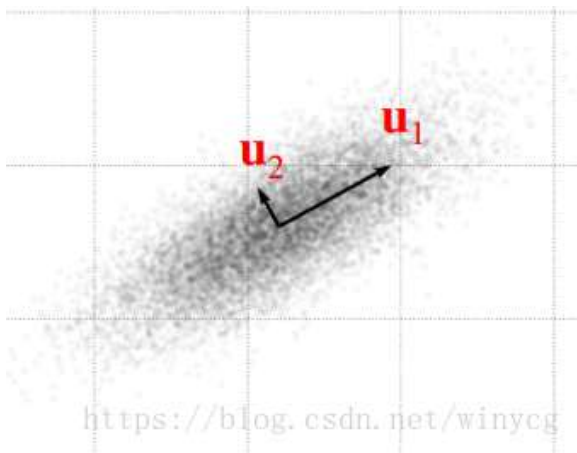
矩阵分解



SVD

图像压缩

SVD



$$\begin{matrix}
 & k \\
 \begin{matrix} m \times n \\ A_k \end{matrix} & = & \begin{matrix} k \\ m \times r \\ U \end{matrix} & \begin{matrix} k \\ r \times r \\ \Sigma \end{matrix} & \begin{matrix} r \times n \\ V^T \end{matrix} & k \\
 \end{matrix}$$

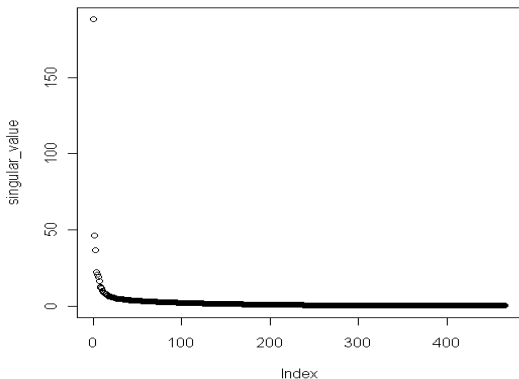
Approximation of A using k Factors

Image Compression Example



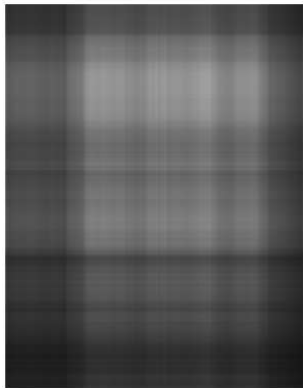
This image is 600×465 pixels

Singular values of flowers image

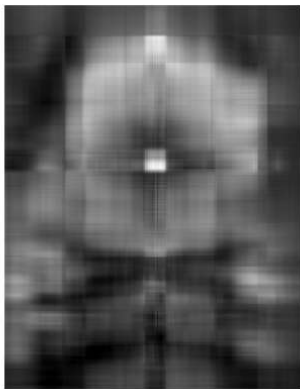


Plot of the singular values

Low rank Approximation to flowers image



Rank-1 approximation



Rank- 5
approximation

Low rank Approximation to flowers image



Rank-20 approximation



Rank-30 approximation

Low rank Approximation to flowers image



Rank-50 approximation



Rank-80 approximation

Low rank Approximation to flowers image



Rank-100 approximation



Rank-120 approximation

Low rank Approximation to flowers image



Rank-150 approximation



True Image

推荐系统

用户-物品评分

User /item	1	2	3	4	5	
1	5	4	4.5	?	3.9	
2	?	4.5	?	4.5	?	
3	4.5	?	4.4	4	4	
4	?	4.8	?	?	4.5	
5	4	?	4.5	5	?	
.....						

<http://blog.csdn.net/zhongkejijiang>

	item 1	item 2	item 3	item 4
user 1	R11	R12	R13	R14
user 2	R21	R22	R23	R24
user 3	R31	R32	R33	R34

R

=

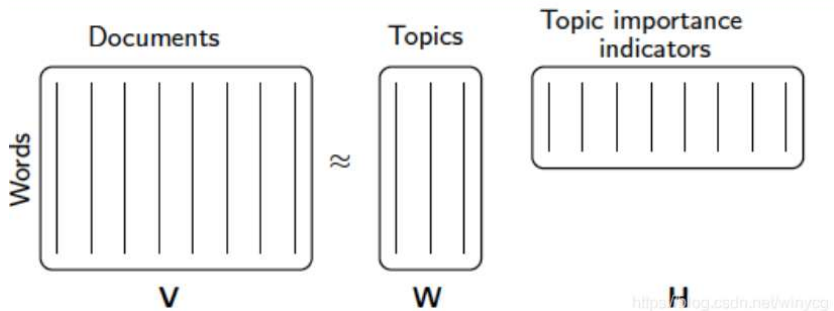
	class 1	class 2	class 3
user 1	P11	P12	P13
user 2	P21	P22	P23
user 3	P31	P32	P33

P

X

	item 1	item 2	item 3	item 4
class 1	Q11	Q12	Q13	Q14
class 2	Q21	Q22	Q23	Q24
class 3	Q31	Q32	Q33	Q34

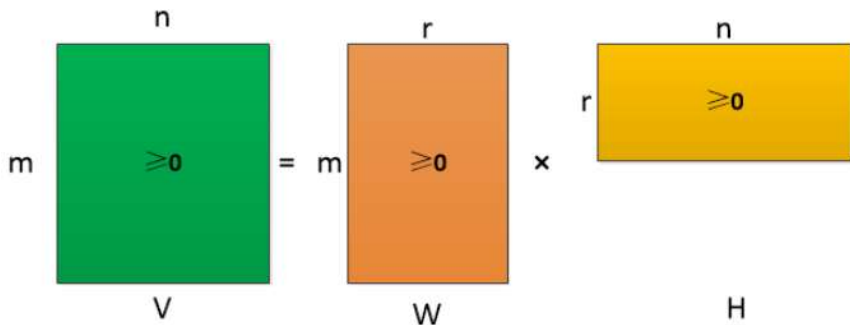
Q



<https://blog.csdn.net/winycg>

非负矩阵分解

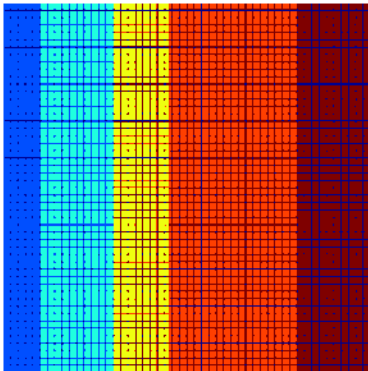
(Non-negative Matrix Factorization, NMF)



<https://www.biaodianfu.com/matrix-factorization.html>

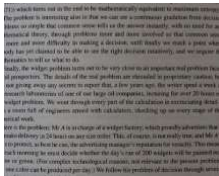
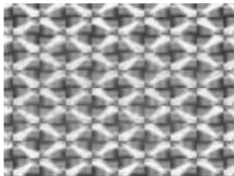
在图像处理中的应用

Low Rank Matrix



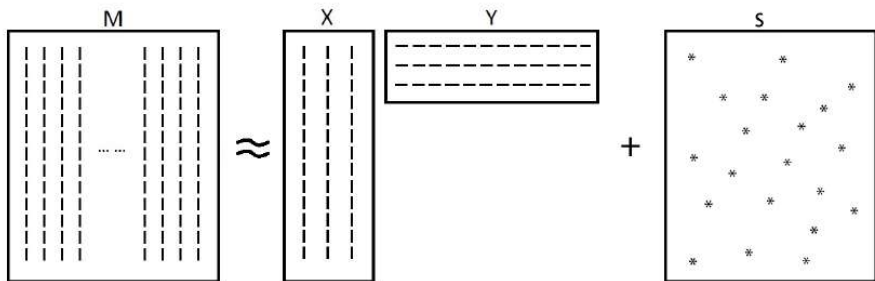
$$\mathbf{A} = [\mathbf{a}_1 \mid \cdots \mid \mathbf{a}_n] \in \mathbb{R}^{m \times n}, \quad \text{rank}(\mathbf{A}) \ll m.$$

高维数据往往具有低维结构



Visual data exhibit **low-dimensional structures** due to rich **local** regularities, **global** symmetries, **repetitive** patterns, or **redundant** sampling.

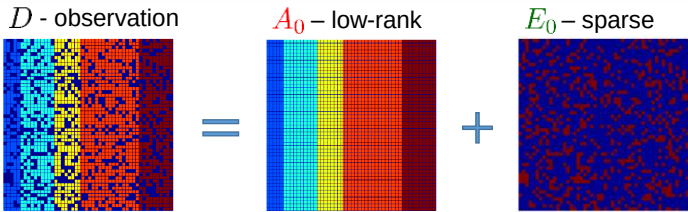
Low Rank + Sparse



$M = \text{low-rank matrix } XY + \text{sparse matrix } S$
(robust PCA)

ROBUST PCA – Problem Formulation

D - observation A_0 - low-rank E_0 - sparse



The diagram illustrates the Robust PCA problem formulation. It shows three heatmaps: D (observation), A_0 (low-rank), and E_0 (sparse). The equation $D = A_0 + E_0$ is represented by the heatmaps and the symbols $=$ and $+$.

Problem: Given $D = A_0 + E_0$, recover A_0 and E_0 .

Low-rank component Sparse component (gross errors)

Numerous approaches in the literature:

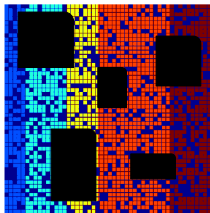
- Multivariate trimming [Gnanadesikan and Kettering '72]
- Power Factorization [Wieber'70s]
- Random sampling [Fischler and Bolles '81]
- Alternating minimization [Shum & Ikeuchi'96, Ke and Kanade '03]
- Influence functions [de la Torre and Black '03]

Key question: ***can guarantee correctness with an efficient algorithm?***

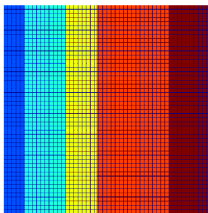
Implications: Highly Compressive Sensing of Structured Information!

Recover low-dimensional structures from a fraction of missing measurements with structured support.

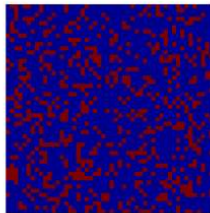
compressive samples



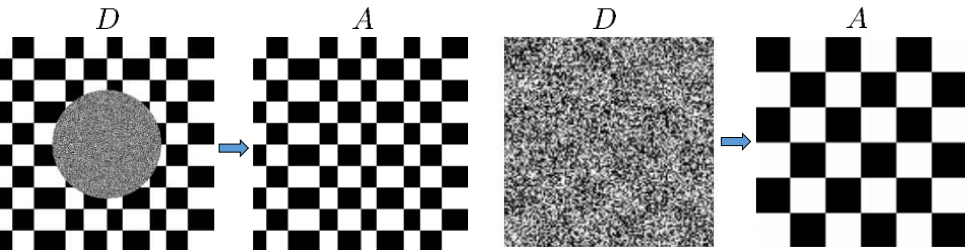
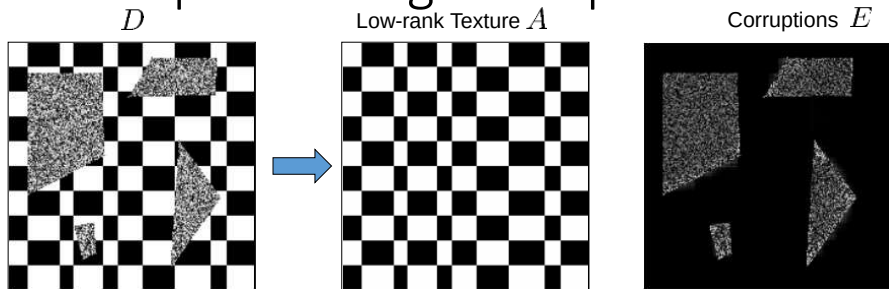
Low-rank Structures



Sparse Structures



Example 1: Image Completion



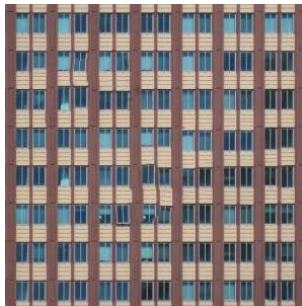
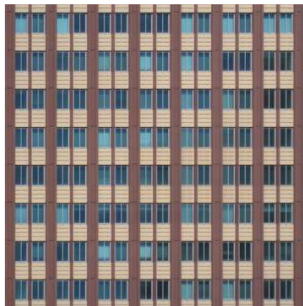
Low-rank Method

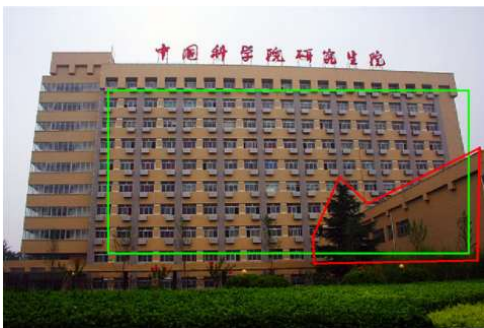
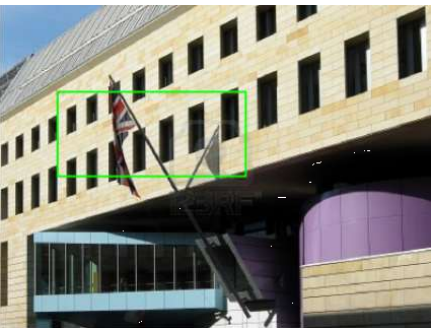
Photoshop

Input



Output



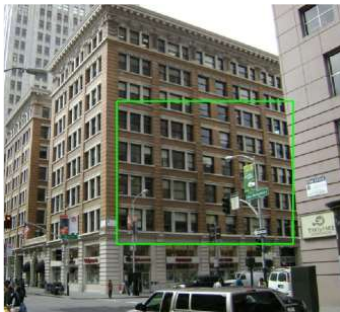


Repairing (Distorted) Low-rank Textures

Low-rank Method

Photoshop

Input



Output

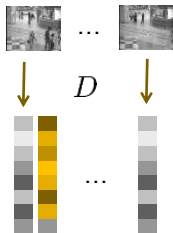


Repairing Video Frames: *background modeling from video*

Surveillance video

200 frames,
144 x 172 pixels,

Significant foreground
motion



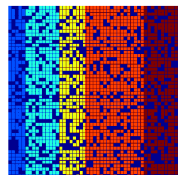
$$\text{Video } D = \text{Low-rank approx. } A + \text{Sparse error } E$$



Sensing or Imaging of Low-rank and Sparse Structures

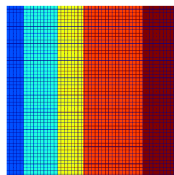
Fundamental Problem: *How to recover low-rank and sparse structures from*

corrupted data



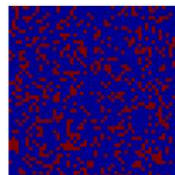
=

Low-rank Structures

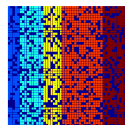


+

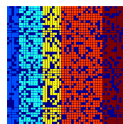
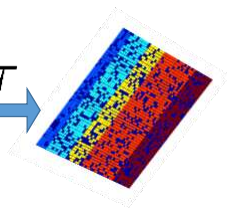
Sparse Structures



subject to either nonlinear deformation τ or linear compressive sampling $\mathcal{P}$?



τ

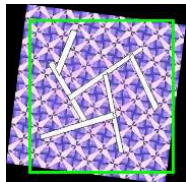


$\mathcal{P}$



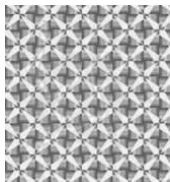
Reconstructing 3D Geometry and Structures

D – deformed observation



$\circ \tau =$

A – low-rank structures



+

E – sparse errors



Problem: Given $D \circ \tau = A_0 + E_0$, recover τ , A_0 and E_0 simultaneously.

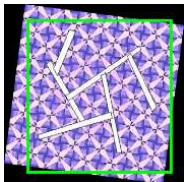
Low-rank component
(regular patterns...)

Sparse component
(occlusion, corruption, foreground...)

Parametric deformations
(affine, projective, radial distortion, 3D shape...)

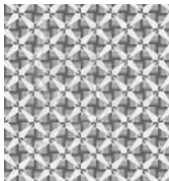
Transform Invariant Low-rank Textures (TILT)

D – deformed observation



$\circ \tau =$

A – low-rank structures



+

E – sparse errors



Objective:

Principal Component Pursuit::

$$\min \|A\|_* + \lambda \|E\|_1 \quad \text{subj} \quad A + E = D \circ \tau$$

Solution: Iteratively solving the linearized convex program::



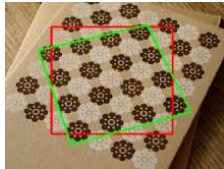
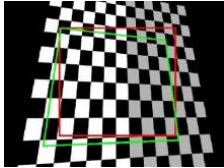
$$\min \|A\|_* + \lambda \|E\|_1 \quad \text{subj} \quad A + E = D \circ \tau_k + J \cdot \Delta \tau$$



Or reduced version: $\text{subj} \quad \mathcal{P}_Q[A + E] = \mathcal{P}_Q[D \circ \tau_k], \mathcal{P}_Q[J] = 0$

TILT: Shape from texture

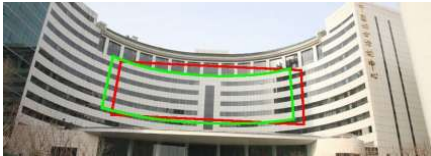
Input (red window D)



Output (rectified green window A)

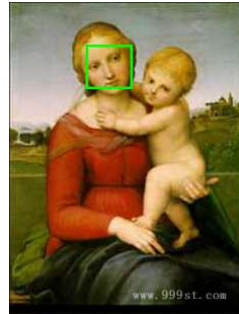
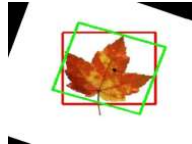
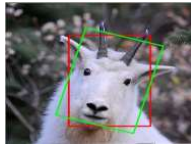


TILT: *Virtual reality*



Object Recognition: *Rectifying Pose of Objects*

Input (red window D)

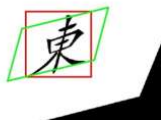


Output (rectified green window A)

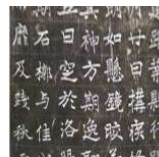


Object Recognition: *Regularity of Texts at All Scales!*

Input (red window D)



Output (rectified green window A)



Recognition: Street Sign Rectification



A_1 A_2 A_3 A_4

$$\min \sum_{i=1}^4 \|A_i\|_* + \lambda \|E_i\|_1$$

$$\text{subj } D \circ \tau = [A_1 \cdots A_4] + [E_1 \cdots E_4].$$

参考文献: SVD

- <https://zhuanlan.zhihu.com/p/36546367>
- <https://zhuanlan.zhihu.com/p/29846048>
- <https://www.zhihu.com/question/22237507>
- <https://zhuanlan.zhihu.com/p/360980054>
- https://math.mit.edu/classes/18.095/2016IAP/lec2/SVD_Notes.pdf

参考文献: Low rank

- Robust PCA
 - Emmanuel J. Candès, Xiaodong Li, Yi Ma, John Wright. Robust principal component analysis? Journal of the ACM, 58 (3), 2011.
 - <https://arxiv.org/pdf/0912.3599.pdf>
 - <https://dl.acm.org/doi/10.1145/1970392.1970395>
 - <https://proceedings.neurips.cc/paper/2009/file/c45147dee729311ef5b5c3003946c48f-Paper.pdf>
- Repairing Sparse Low-rank Texture
 - Xiao Liang, Xiang Ren, Zhengdong Zhang, and Yi Ma, European Conference on Computer Vision (ECCV), October 2012.
 - https://people.csail.mit.edu/zhangzd/papers/recover_low-rank_texture_final.pdf
- TILT: Transform-Invariant Low-rank Textures,
 - Zhengdong Zhang, Arvind Ganesh, Xiao Liang, and Yi Ma, Volume 99, Number 1, page 1-24, the International Journal of Computer Vision (IJCV), August 2012.
 - <https://arxiv.org/pdf/1012.3216.pdf>

作业2：基于矩阵分解的图像处理

- 要求：如下任务2选1
 - 任务1：基于矩阵SVD分解的图像压缩
 - 任务2：基于矩阵低秩分解的图像修复
- 提交
 - 作业报告，包括：数学模型、测试结果、实验分析等
- 提交时间：2023年4月9日周日晚



中国科学技术大学
University of Science and Technology of China

谢谢！



中国科学技术大学

University of Science and Technology of China

数学建模

Mathematical Modeling

刘利刚

中国科学技术大学

最优化模型及求解

最优化模型

- 许多数学模型都是最优化模型
- 也称为数学规划模型
 - 线性规划、非线性规划、整数规划（0~1规划）
- 比如：课本第五章、第七章等

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

目标函数 or 能量函数

$$\text{s.t. } g(x) = 0$$

等式约束

$$h(x) \geq 0$$

不等式约束

x ~决策变量

$f(x)$ ~目标函数

例1：（课本 § 5.1）投资效益问题。

一个公司有22亿元资金可用来投资，现有6个项目可供选择，各项目所需投资金额和预计年收益如表所示。应选择哪几个项目投资收益最大？

项目 i	1	2	3	4	5	6
投资 p_i	5	2	6	4	6	8
收益 q_i	0.5	0.4	0.6	0.5	0.9	1

【解答】

- 设 $x_i = 1$ 表示投资项目 i ，否则 $x_i = 0$ 。
- 约束： $\sum_{i=1}^6 p_i x_i \leq 22$ 。
- 目标：最大化 $S = \sum_{i=1}^6 q_i x_i$ 。0-1规划。
- 解得 $x = (0, 1, 1, 0, 1, 1)$ 。

例2：（课本 § 5. 2）加工次序问题。

有10个工件在一台机床上加工。这10个工件中，有些工件必须在另一些工件加工完毕之后才能加工。按规定，在工件运抵后266小时内应加工完毕，否则要支付一定的赔款，赔款数正比于延误的时间。

工件号	1	2	3	4	5	6	7	8	9	10
加工时间	20	28	25	45	16	12	60	10	20	30
前期工件号	3	8	7	/	1, 2, 6	8	4	3	5	9
每小时赔款	12	14	15	10	10	11	12	8	6	7

设10个工件运抵后 T 小时才开始加工。要求安排一个加工次序，使得支付的赔款最少。

【模型建立】

- 设 $n = 10$, 工件 i 的加工时间 t_i , 每小时赔款 q_i 。
- 设 $x_{ij} = 1$ 表示工件 i 是第 j 个加工, 否则 $x_{ij} = 0$ 。
- 约束: $\sum_{j=1}^n x_{ij} = 1, \forall i, \sum_{i=1}^n x_{ij} = 1, \forall j$ 。
- “工件 i 必须在工件 k 加工完毕之后才能加工”等价于

$$\sum_{j=1}^m x_{ij} \leq \sum_{j=1}^m x_{kj}, \forall m。$$

- 目标: 最小化赔款总额 S , 其中

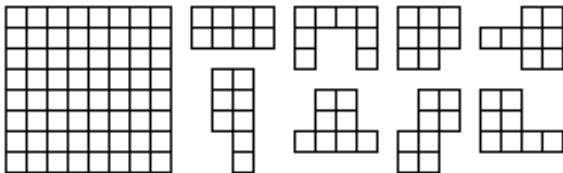
$$S = \sum_{j=1}^n \left(\sum_{i=1}^n q_i x_{ij} \right) \max \left(0, T - 266 + \sum_{i=1}^n t_i (x_{i1} + \cdots + x_{ij}) \right)$$

是 $\{x_{ij}\}$ 的非线性函数。这种问题称为非线性规划。

- 可遍历所有满足约束条件的加工次序, 找出 S 的最小值。

例3：（课本 § 5.4）拼板问题。

有一种智力玩具，由一块面积为 8×8 的正方形底板和8块形状不同、颜色各异的小拼板构成。每块小拼板的两面着相同的颜色，皆由8个单位面积的小正方形构成。有多少种不同的方法能将这8块小拼板正好拼成 8×8 的正方形后放置在底板上？



【模型建立】

- 每个拼板可以旋转90,180,270度，也可以翻转之后再旋转。8块拼板共有38种形态。
- 把底板上的64个小块（小单位正方形）按照第1行从左到右,...,第8行从左到右的顺序编号。每种形态上的小块也如此编号。
- 设可把拼板按照 $x_1, \dots, x_8$ 的顺序依次放在底板上，使得拼板 x_1 的第1块为底板的第1块，拼板 x_2 的第1块为未被拼板 x_1 覆盖的最小编号块，...，拼板 x_8 的第1块为未被拼板 x_7 覆盖的最小编号块。
- 所求问题转化为求向量 $x = (x_1, \dots, x_8)$ ，其中 $1 \leq x_i \leq 38$ ，对应8块不同拼板，并且能够覆盖所有64个小块。
- 可用深度优先搜索算法求出所有满足条件的 x 。

优化问题的一般形式

高维实值函数: $f: \mathbb{R}^n \rightarrow \mathbb{R}$

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

目标函数 or 能量函数

$$\text{s.t. } g(x) = 0$$

等式约束

$$h(x) \geq 0$$

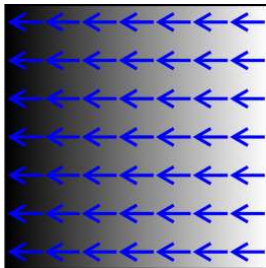
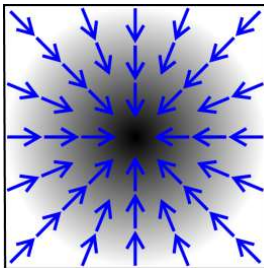
不等式约束

- Two roles
 - Client: Which optimization tool is relevant?
 - 不同的优化问题须用不同的优化方法
 - Designer: Can I design an algorithm for this problem?
 - 特定的优化问题需要设计特定的优化方法达到最佳性能
- Optimization is a **huge** field.

梯度 (Gradient): 一阶导数

$$f : \mathbb{R}^n \rightarrow \mathbb{R}$$

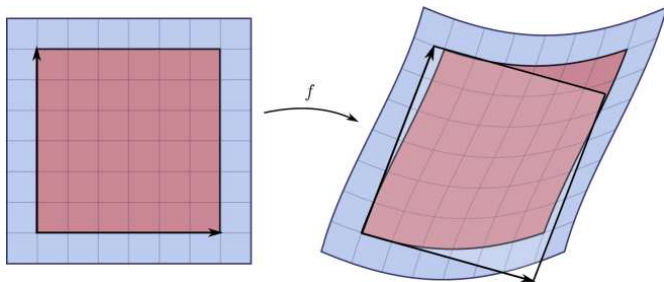
$$\rightarrow \nabla f = \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \dots, \frac{\partial f}{\partial x_n} \right)$$



Jacobian: 一阶“导数”矩阵

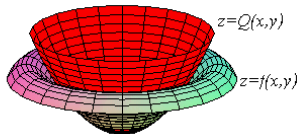
$$f : \mathbb{R}^n \rightarrow \mathbb{R}^m$$

$$\rightarrow (Df)_{ij} = \frac{\partial f_i}{\partial x_j}$$



Hessian : 二阶“导数”矩阵

$$f : \mathbb{R}^n \rightarrow \mathbb{R} \rightarrow H_{ij} = \frac{\partial^2 f}{\partial x_i \partial x_j}$$



$$f(x) \approx f(x_0) + \nabla f(x_0)^\top (x - x_0) + (x - x_0)^\top H f(x_0) (x - x_0)$$

驻点 (Critical point)

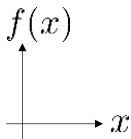
$$\nabla f(x) = 0$$

(unconstrained)

Saddle point

Local max

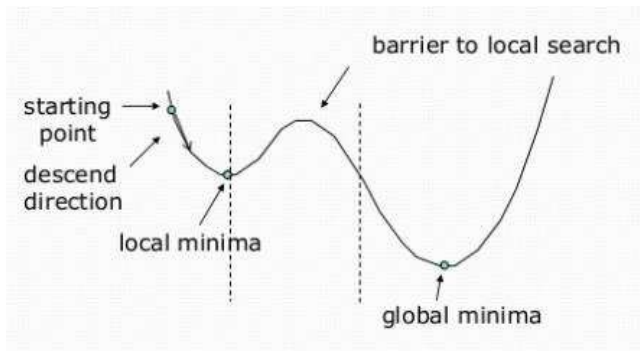
Local min



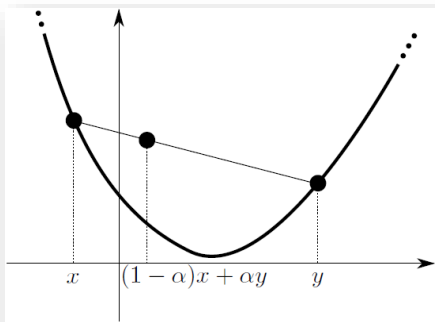
Critical points may not be minima.

一般非线性函数的最小值

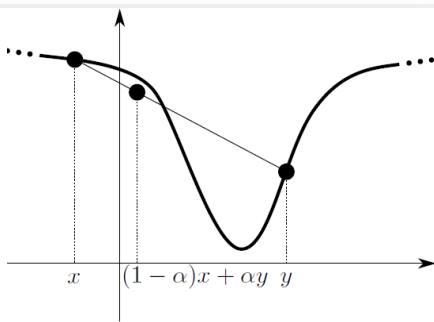
- 仍无法求解!
- 数值求解
 - 从某初值开始, 逐步找其附近的极小值



凸函数的驻点就是最小值



(a) Convex



(b) Quasiconvex

优化问题的类型

- Constrained / Unconstrained
- Linear / Nonlinear
- Global / Local
- Convex / Nonconvex
- Continuous / Discrete
- Stochastic / Deterministic
- Single objective / Multiple objectives

$$\text{minimize } (E_1(x), E_2(x), \dots, E_k(x))$$

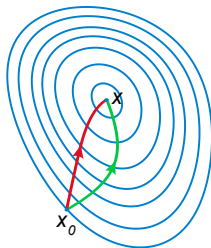
$$E = \lambda_1 E_1 + \lambda_2 E_2 + \dots + \lambda_k E_k$$

无约束的优化问题

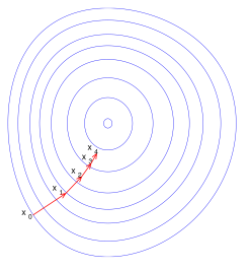
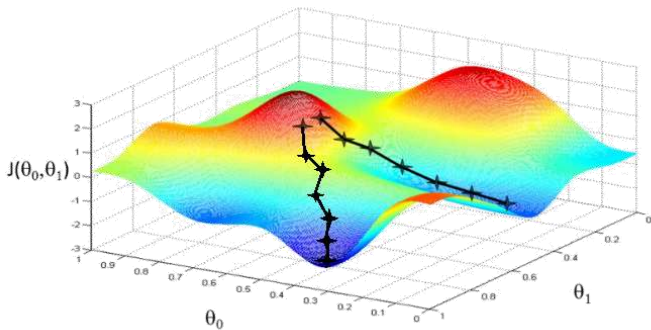
Unconstrained Optimization

$$\min_x f(x)$$

- Gradient descent
- Newton
- Quasi-Newton
- Coordinate descent

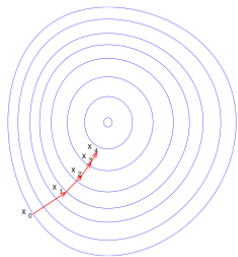
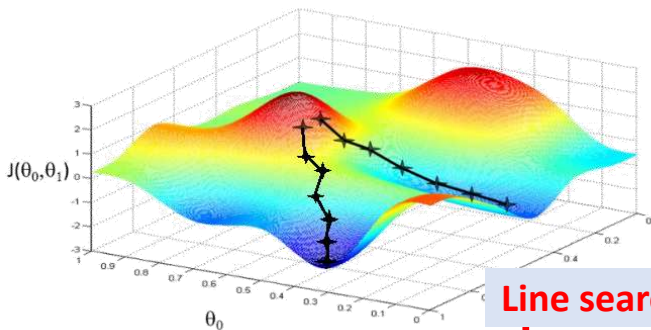


梯度下降法 (Gradient descent)



$$x_{k+1} = x_k - \alpha_k \nabla f(x_k)$$

梯度下降法 (Gradient descent)



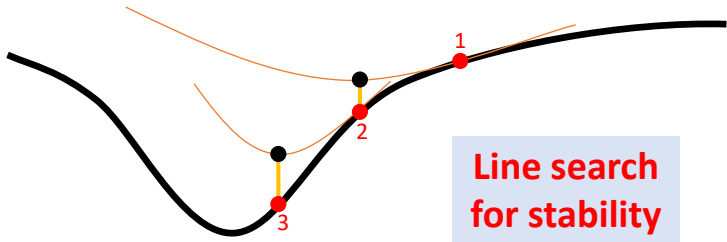
Line search

$$x_{k+1} = x_k - \alpha_k \nabla f(x_k)$$

Gradient descent

牛顿法 (Newton's method)

$$x_{k+1} = x_k - [Hf(x_k)]^{-1} \nabla f(x_k)$$



拟牛顿法 (Quasi-Newton)

$$x_{k+1} = x_k - M_k^{-1} \nabla f(x_k)$$

**Hessian
approximation**

- **Estimate the Hessian based on previous gradients**
- **Recursively inverse Hessian**

- BFGS (Broyden–Fletcher–Goldfarb–Shanno algorithm)
- L-BFGS

坐标下降法 (Coordinate descent)

Obj: minimize $_{x,y}$ $E(x, y)$

- Alternating variables

Repeat

$$1. y_{k+1} = \min_y E(x_k, y)$$

$$2. x_{k+1} = \min_x E(x, y_{k+1})$$

Software

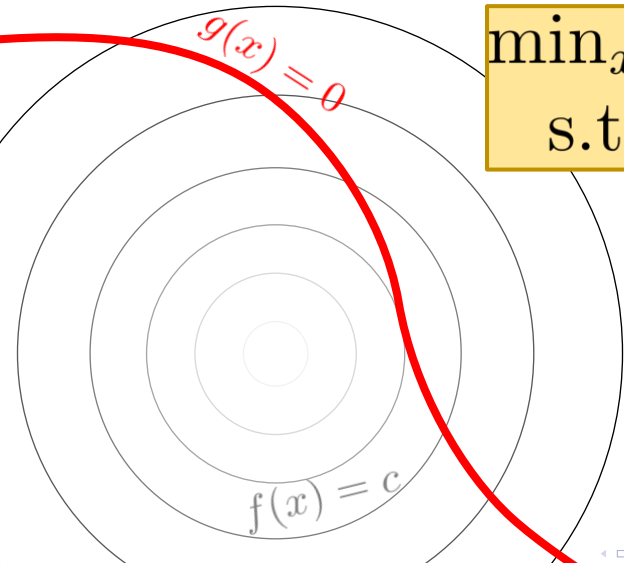
- **Matlab**: `fminunc` or `minfunc`
- **C++**: `libLBFGS`, `dlib`, others

Typically provide functions for **function** and **gradient**
(and optionally, **Hessian**).

等式约束的优化问题

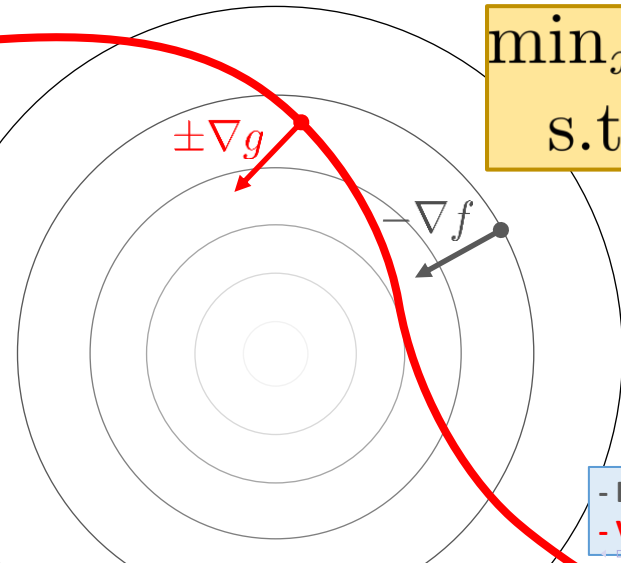
Lagrange Multipliers: Idea

$$\begin{array}{ll} \min_x & f(x) \\ \text{s.t.} & g(x) = 0 \end{array}$$



Lagrange Multipliers: Idea

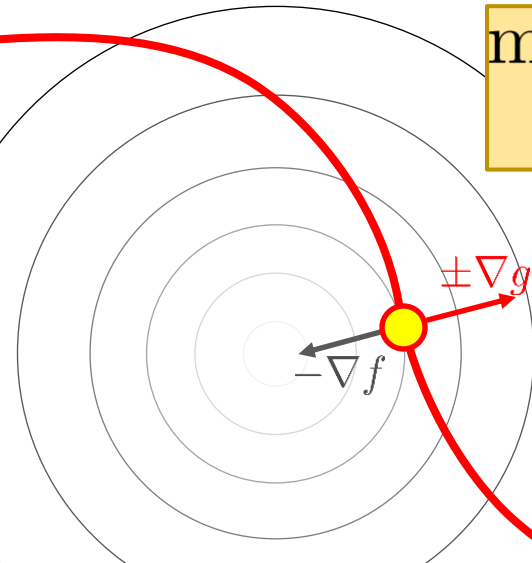
$$\begin{array}{ll} \min_x & f(x) \\ \text{s.t.} & g(x) = 0 \end{array}$$



- Decrease f : $-\nabla f$
- Violate constraint: $\pm \nabla g$

Lagrange Multipliers: Idea

$$\begin{aligned} \min_x \quad & f(x) \\ \text{s.t.} \quad & g(x) = 0 \end{aligned}$$



Want: $\nabla f \parallel \nabla g$
 $\implies \nabla f = \lambda \nabla g$

Use of Lagrange Multipliers

Turns constrained optimization into
unconstrained root-finding.

$$\begin{aligned}\nabla f(x) &= \lambda \nabla g(x) \\ g(x) &= 0\end{aligned}$$

Many Options

- **Reparameterization**

Eliminate constraints to reduce to unconstrained case

- **Newton's method**

Approximation: quadratic function with linear constraint

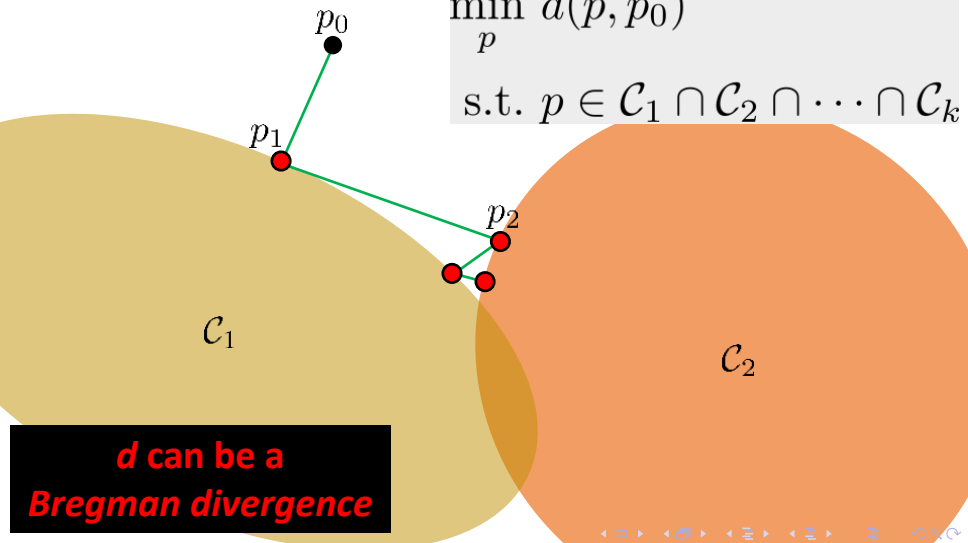
- **Penalty method**

Augment objective with barrier term, e.g. $f(x) + \rho|g(x)|$

Alternating Projection

$$\min_p d(p, p_0)$$

$$\text{s.t. } p \in \mathcal{C}_1 \cap \mathcal{C}_2 \cap \dots \cap \mathcal{C}_k$$



***d can be a
Bregman divergence***

Augmented Lagrangians

$$\begin{array}{ll}\min_x & f(x) \\ \text{s.t.} & g(x) = 0\end{array}$$

↓

$$\begin{array}{ll}\min_x & f(x) + \frac{\rho}{2} \|g(x)\|_2^2 \\ \text{s.t.} & g(x) = 0\end{array}$$

**Does nothing when
constraint is
satisfied**

Add constraint to objective

Alternating Direction Method of Multipliers (ADMM)

$$\begin{array}{ll}\min_{x,z} & f(x) + g(z) \\ \text{s.t.} & Ax + Bz = c\end{array}$$

$$\Lambda_\rho(x, z; \lambda) = f(x) + g(z) + \lambda^\top (Ax + Bz - c) + \frac{\rho}{2} \|Ax + Bz - c\|_2^2$$

$$x \leftarrow \arg \min_x \Lambda_\rho(x, z, \lambda)$$

$$z \leftarrow \arg \min_z \Lambda_\rho(x, z, \lambda)$$

$$\lambda \leftarrow \lambda + \rho(Ax + Bz - c)$$

The Art of ADMM “Splitting”

$$\left\{ \begin{array}{ll} \min_J & \sum_i \|J_i\|_2 \\ \text{s.t.} & MJ = b \end{array} \right\} \longrightarrow \left\{ \begin{array}{ll} \min_{J, \bar{J}} & \sum_i (\|J_i\|_2 + \frac{\rho}{2} \|J_i - \bar{J}_i\|_2^2) \\ \text{s.t.} & M\bar{J} = b \\ & J = \bar{J} \end{array} \right\}$$

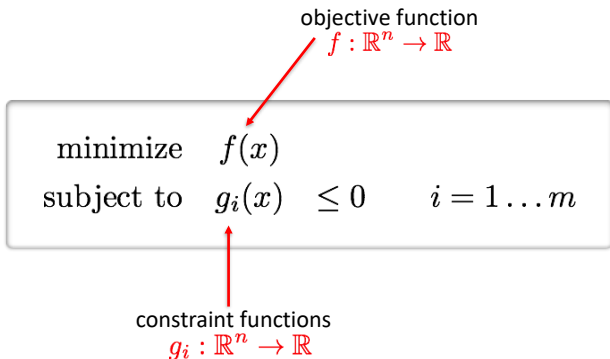

**Augmented
part**

Want two *easy* subproblems

不等式约束的优化问题

一般形式

objective function
 $f : \mathbb{R}^n \rightarrow \mathbb{R}$

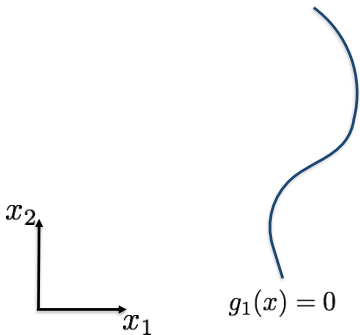


minimize $f(x)$
subject to $g_i(x) \leq 0 \quad i = 1 \dots m$

constraint functions
 $g_i : \mathbb{R}^n \rightarrow \mathbb{R}$

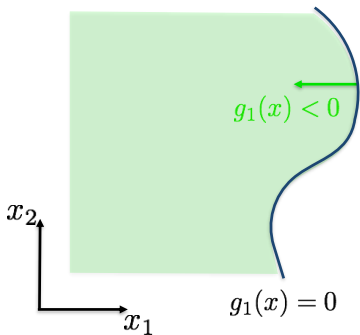
几何解释

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



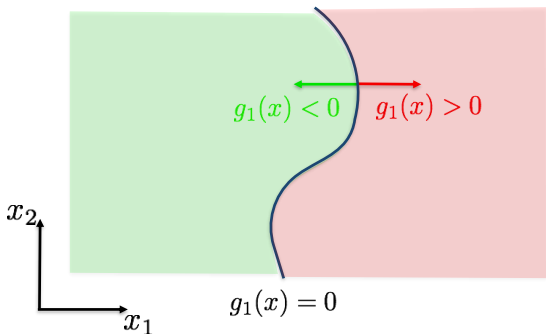
几何解释

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



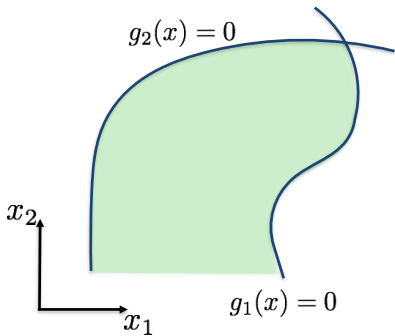
几何解释

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



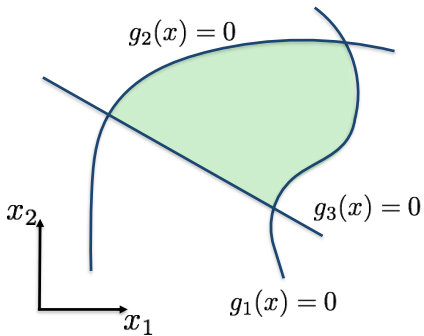
几何解释

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



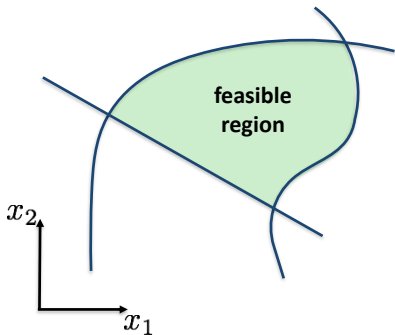
几何解释

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

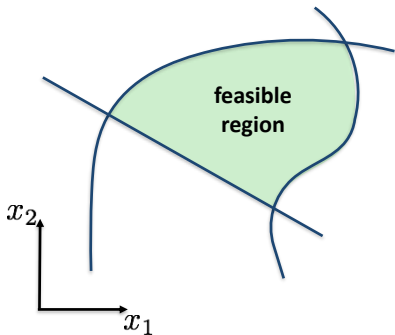


几何解释

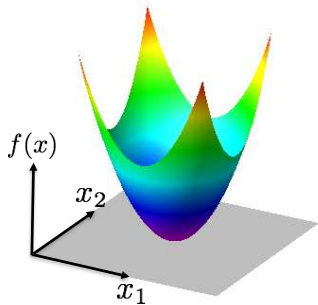
$$\begin{array}{ll}\text{minimize} & f(x) \\ \text{subject to} & g_i(x) \leq 0 \quad i = 1 \dots m\end{array}$$



几何解释

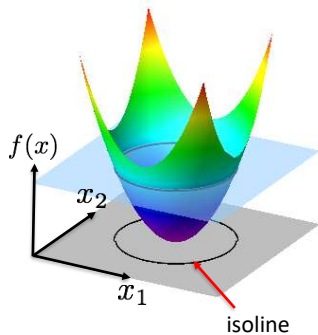
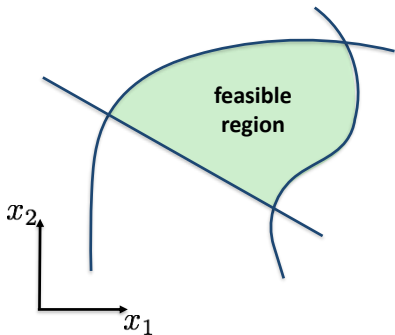


$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & g_i(x) \leq 0 \quad i = 1 \dots m \end{array}$$

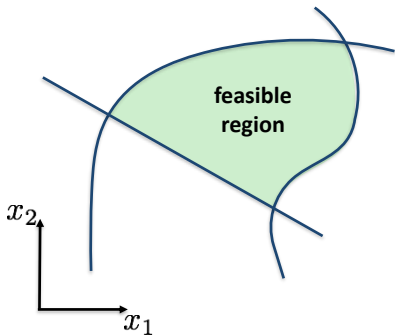


几何解释

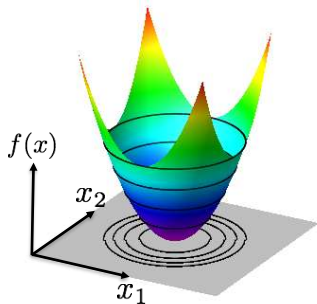
$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & g_i(x) \leq 0 \quad i = 1 \dots m \end{array}$$



几何解释

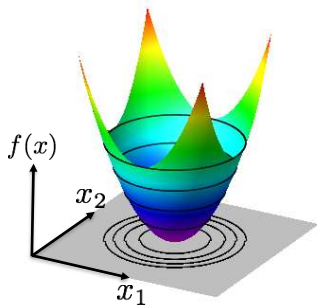
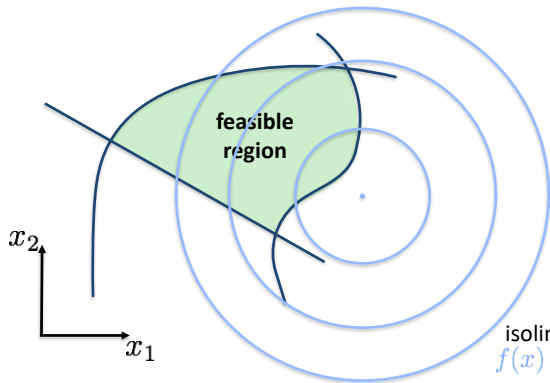


$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & g_i(x) \leq 0 \quad i = 1 \dots m \end{array}$$



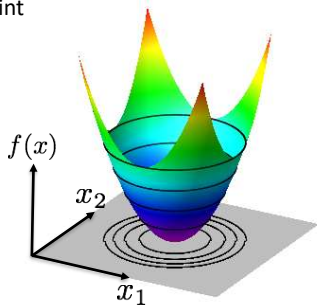
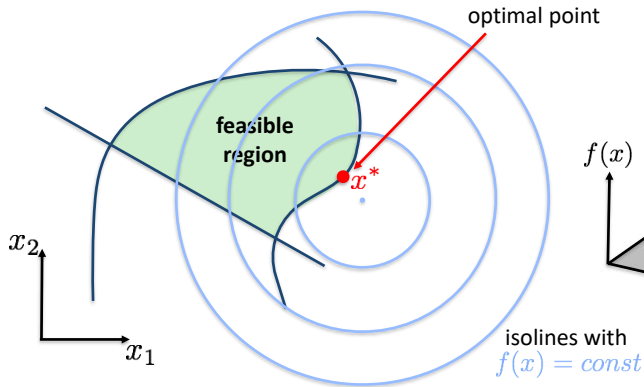
几何解释

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



几何解释

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



First-Order Optimality Conditions

- **Necessary condition for minimum of**

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

- **Langrangian:** $L(x, \lambda) = f(x) + \sum_{i=1}^m \lambda_i g_i(x)$

- **Karush-Kuhn-Tucker (KKT)**

conditions for **minimum** x^*

1. Stationarity: $\nabla f(x^*) + \sum_{i=1}^m \lambda_i \nabla g_i(x^*) = 0$
2. Primal feasibility: $g_i(x^*) \leq 0$
3. Dual feasibility: $\lambda_i \geq 0$
4. Complementary slackness: $\lambda_i g_i(x^*) = 0$

without constraints just

$$\nabla f(x) = 0$$

First-Order Optimality Conditions

- **Necessary condition for minimum of**

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

- **Langrangian:** $L(x, \lambda) = f(x) + \sum_{i=1}^m \lambda_i g_i(x)$

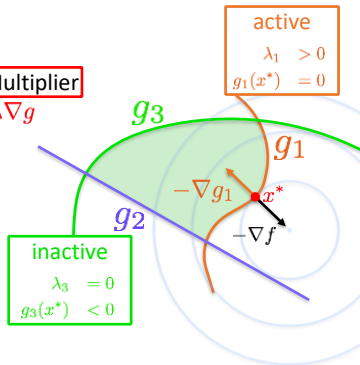
Recall Lagrange Multiplier

- **Karush-Kuhn-Tucker (KKT)**

conditions for **minimum** x^*

1. Stationarity: $\nabla f(x^*) + \sum_{i=1}^m \lambda_i \nabla g_i(x^*) = 0$
2. Primal feasibility: $g_i(x^*) \leq 0$
3. Dual feasibility: $\lambda_i \geq 0$
4. Complementary slackness: $\lambda_i g_i(x^*) = 0$

$$\nabla f = -\lambda \nabla g$$



优化方法

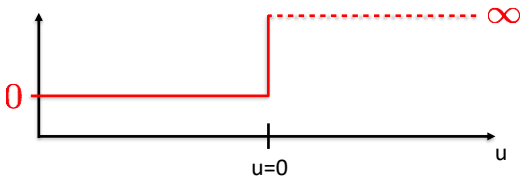
- Active set method

Repeated update active set

1. active set define **equality** constraints
2. *compute* the **Lagrange multipliers** of the active set
3. *remove* constraints with **negative** Lagrange multipliers
4. *search* for infeasible constraints

- Barrier method (Interior Point Method)

$$I_-(u) = \begin{cases} 0 & \text{if } u \leq 0 \\ \infty & \text{if } u > 0 \end{cases}$$



minimize $f(x)$ constrained
subject to $g_i(x) \leq 0 \quad i = 1 \dots m$



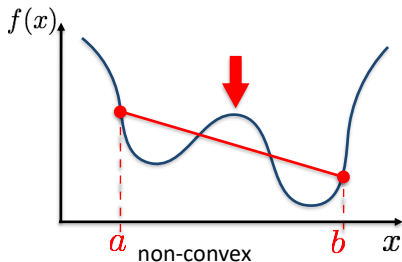
equivalent

unconstrained
minimize $f(x) + \sum_{i=1}^m I_-(g_i(x))$

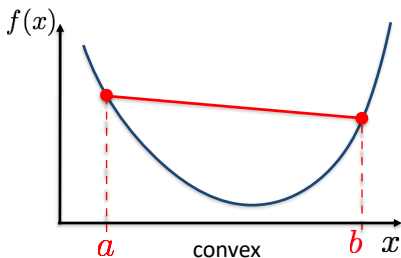
Convex Optimization

凸函数能保证找到全局最小值

- Searching globally optimal solutions usually requires **convexity**!
- f convex if: $f((1-t)a + tb) \leq (1-t)f(a) + tf(b)$ $t \in [0, 1]$



vs.



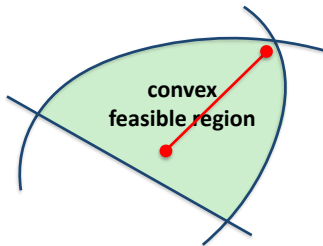
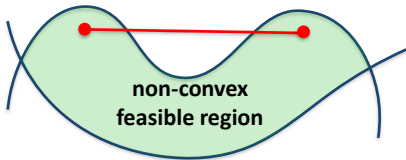
凸优化问题

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

is **convex optimization problem** if $f(x)$ and all $g_i(x)$ are convex functions

consequences

- feasible region is convex set



凸优化问题

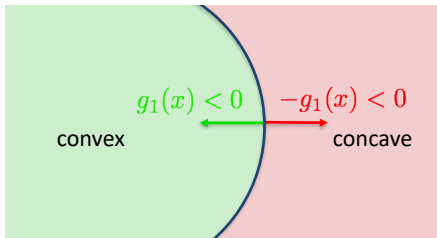
$$\begin{array}{ll}\text{minimize} & f(x) \\ \text{subject to} & g_i(x) \leq 0 \quad i = 1 \dots m\end{array}$$

is **convex optimization problem** if $f(x)$ and all $g_i(x)$ are convex functions

consequences

- feasible region is convex set
- equality constraints can only be affine, i.e. $g_i(x) = a^T x + b$ since

$$g_i(x) = 0 \iff \begin{cases} g_i(x) & \leq 0 \\ -g_i(x) & \leq 0 \end{cases}$$



凸优化的主要方法

- Linear Programming

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad q^T x \quad \text{subject to} \quad \begin{array}{l} Ax = a \\ Bx < b \end{array}$$

- Quadratic Programming

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad \frac{1}{2} x^T Q x + q^T x \quad \text{subject to} \quad \begin{array}{l} Ax = a \\ Bx < b \end{array}$$

- Conic Programming

$$\sqrt{\sum_{i=1}^n x_i^2} \leq x_0$$

- Semidefinite Programming (SDP)

$$\underset{A \in \mathbb{R}^{n \times n}}{\text{minimize}} \quad E(A) \quad \text{subject to} \quad A \geq 0$$

A is SPD
 $\lambda_i(A) \geq 0, \forall i$

其他优化问题

线性规划问题

- 线性函数 $f(\mathbf{x}) = \mathbf{c}^T \mathbf{x}$ 在线性约束 $A\mathbf{x} \geq \mathbf{b}$ 下的最小值，其中 A 是矩阵， $\mathbf{b}, \mathbf{c}, \mathbf{x}$ 是列向量。
- 若要求 $\mathbf{x}$ 的分量是整数，则称为整数规划问题。
- 若要求 $\mathbf{x}$ 的分量是0或1，则称为0-1规划问题。
- 线性规划问题有些特殊高效的解法：比如图解法、单纯形法等

Nonlinear Least Squares

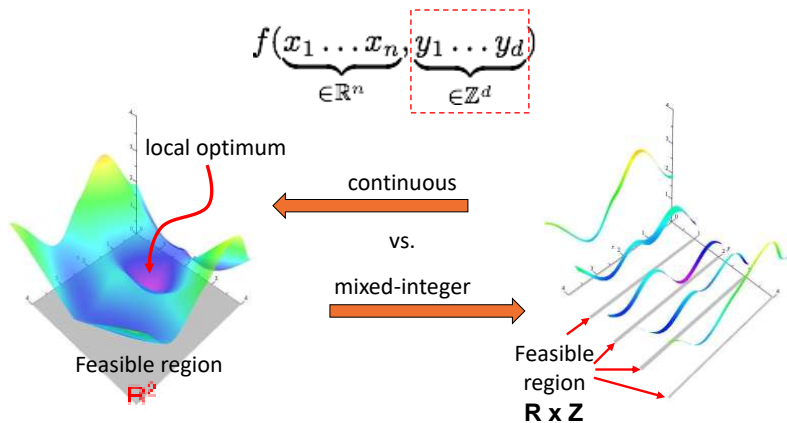
Obj: minimize $\sum_i e_i^2(x)$

- Gauss-Newton
- Levenberg-Marquardt

$$\nabla^2 e_i^2 \approx 2(\nabla e_i)^T \nabla e_i$$

$$\nabla^2 \approx J^T J$$

Mixed-Integer Optimization



优化相关的软件

- **Eigen** — linear algebra
- **IPOPT** — fast opensource C++ interior point method
- **Mosek** — commercial (convex) optimization in C, Java, Python...
- **Gurobi** — commercial mixed-integer optimization
- **CPLEX** — commercial mixed-integer optimization
- **Matlab** — many algorithms, good for prototyping
- **CVX** — prototyping for convex optimization
- **CoMISo** — unified interface to above algorithms

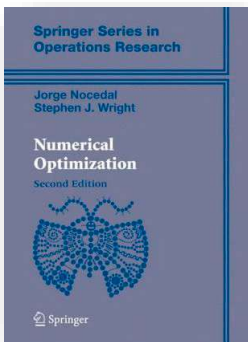
参考书目

Optimization is a **huge** field!

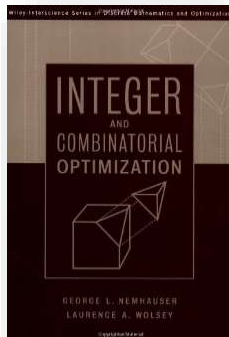


S. Boyd and L. Vandenberghe
Convex Optimization
Cambridge University Press, 2004.

Get PDF online:
<http://stanford.edu/~boyd/cvxbook/>



J. Nocedal and S. J. Wright
Numerical Optimization
Springer, 2006.



G. L. Nemhauser and L. A. Wolsey
Integer and Combinatorial Optimization
John Wiley & Sons, 1999.



中国科学技术大学
University of Science and Technology of China

谢谢！



中国科学技术大学
University of Science and Technology of China

数学建模 Mathematical Modeling

刘利刚

中国科学技术大学

数据建模方法

聚类分析模型

Clustering Analysis

**物以类聚
人以群分**

蠼螋分类问题



(一) 背景知识

两种蠼螋 Af 和 Apf 已由生物学家罗纳 (w.L.Grogn) 和维尔恩 (W.W.Wirth) 于 1981 年根据它们的触角长 (mm) 和翅长 (mm) 加以区分, 6 只 Apf 和 9 只 Af 蠼螋的触长, 翅长数据如下:

Apf: (1.14, 1.78), (1.18, 1.96), (1.20, 1.86), (1.26, 2.00), (1.28, 2.00), (1.30, 1.96) ;

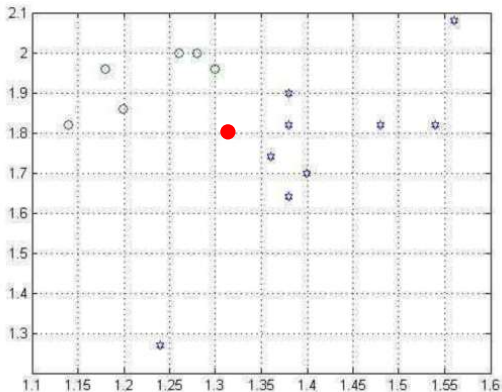
Af: (1.24, 1.72), (1.36, 1.74), (1.38, 1.64), (1.38, 1.82), (1.38, 1.90), (1.40, 1.70), (1.48, 1.82), (1.54, 1.82), (1.56, 2.08)。

在生物学中, 根据触角长和翅长来识别一只蠼螋标本是 Af 还是 Apf 是很重要的。

(二) 要解决的问题

- 1、根据给定的数据, 制定一种方法, 正确区分两类蠼螋;
- 2、用我们的方法对触长、翅长分别为 (1.24, 1.80)、(1.28, 1.84)、(1.40, 2.04) 的三个样本进行识别;
- 3、假设 Af 是宝贵的传粉益虫, Apf 是某种疾病的载体, 在这种情况下我们是否应该修改所用的分类方法, 且如何修改。

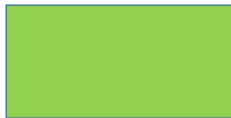
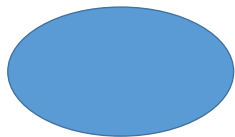
蠓虫分类问题：有监督分类



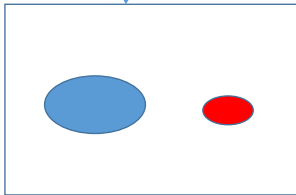
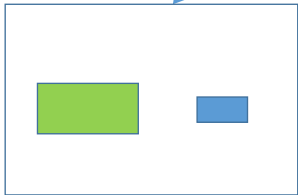
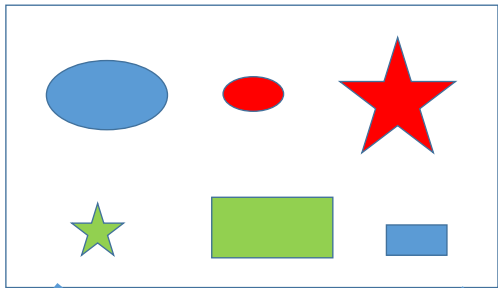
Af	1	2	3	4	5	6	7	8	9
触角长度	1.24	1.36	1.38	1.38	1.38	1.4	1.48	1.54	1.56
翼长	1.72	1.74	1.64	1.82	1.9	1.7	1.82	1.82	2.08

Apf	1	2	3	4	5	6
触角长度	1.14	1.18	1.20	1.26	1.28	1.30
翼长	1.78	1.96	1.86	2.00	2.00	1.96

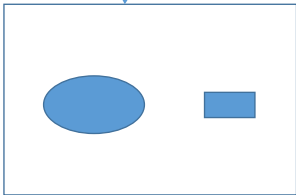
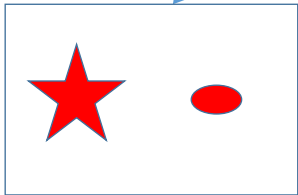
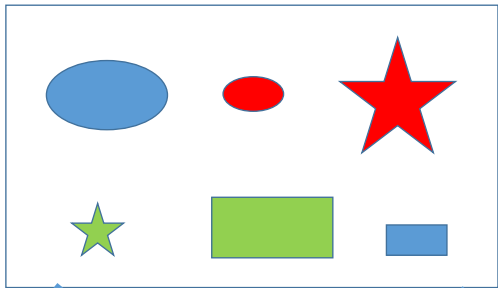
图形分类：无监督分类
下列图形能分成哪几类？



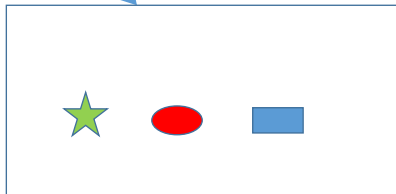
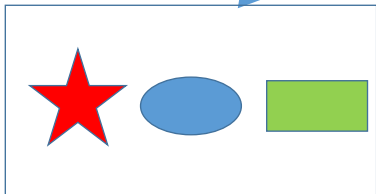
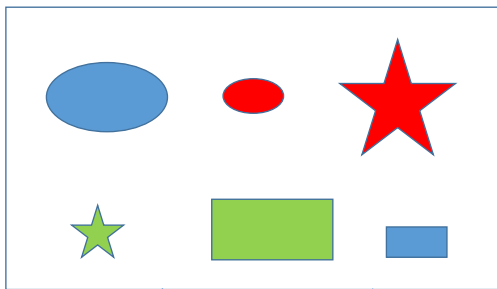
分类-1：按形状



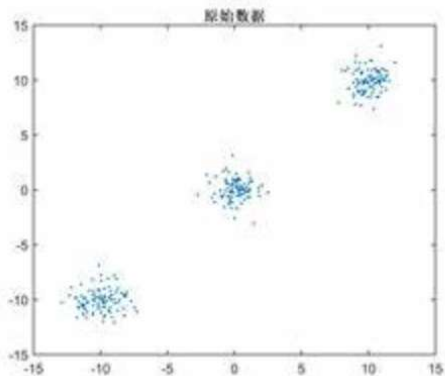
分类-2：按颜色



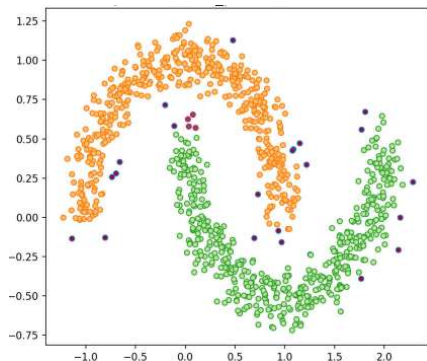
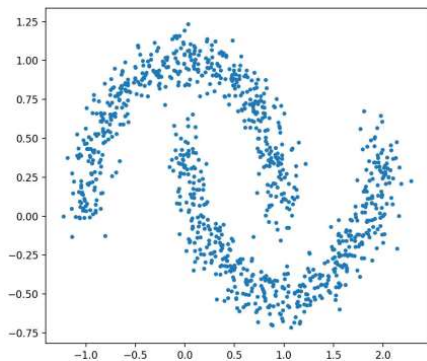
分类-3：按大小



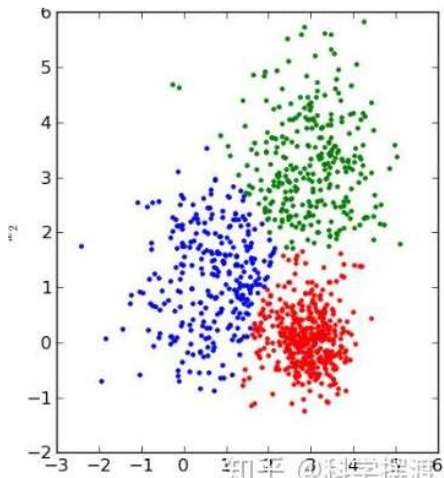
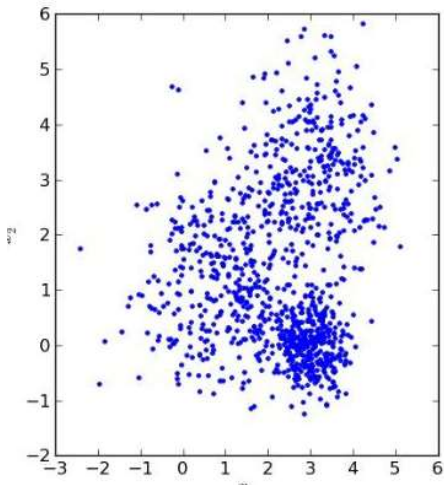
例子：分几类？



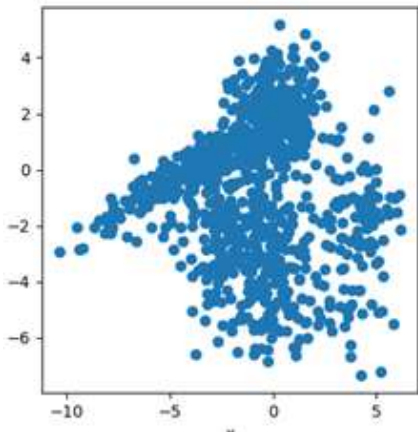
例子：分几类？



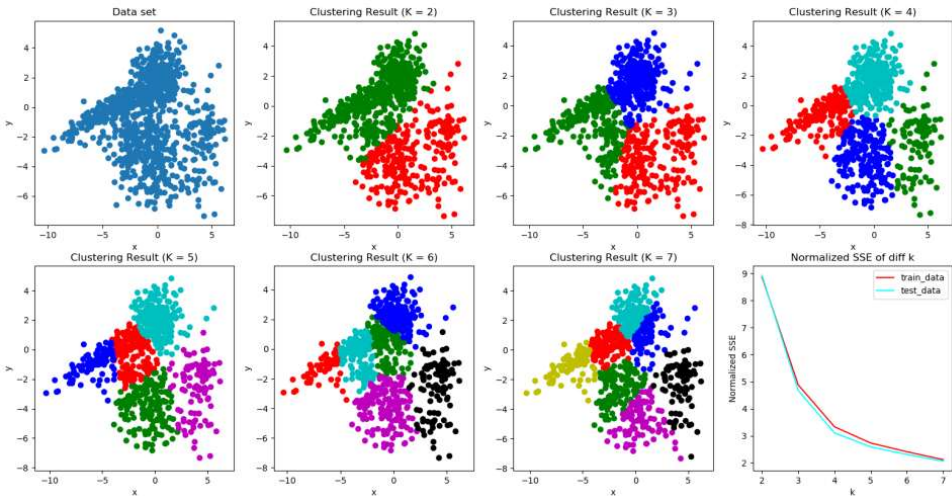
例子：分几类？



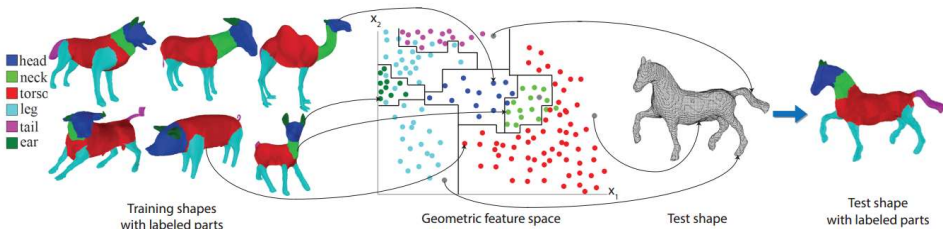
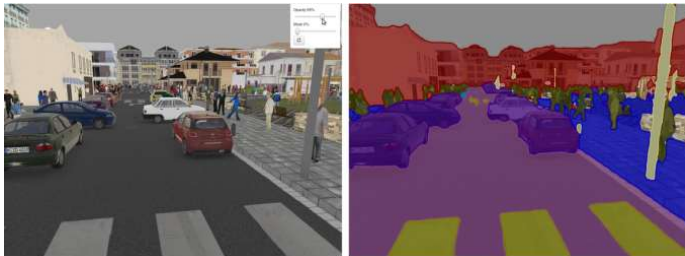
例子：分几类？



例子：分几类？



应用：图像分割与形状分割

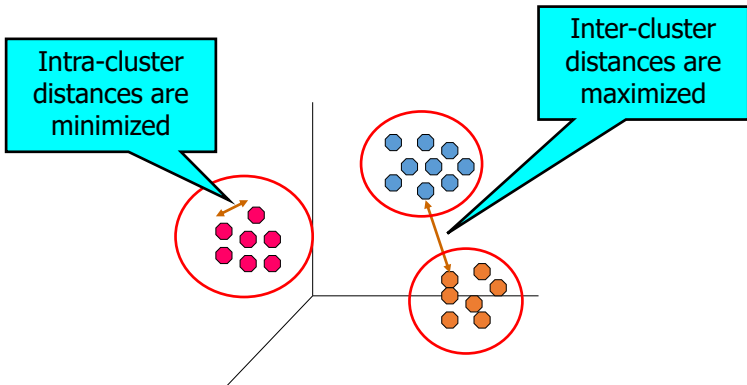


聚类分析 (cluster analysis)

- 聚类分析指将物理或抽象对象的集合分组为由类似的对象组成的多个类的分析过程。
- 定义：
- 给定一个数据点集，聚类算法将每个数据点分类到一个特定的组中。
- 同一组数据点具有相似的性质或（和）特征，不同组数据点具有高度不同的性质或（和）特征。

What is Cluster Analysis?

- Finding groups of objects such that the objects in a group will be similar (or related) to one another and different from (or unrelated to) the objects in other groups



What is Cluster Analysis?

- Cluster: a collection of data objects
 - Similar to one another within the same cluster
 - Dissimilar to the objects in other clusters
- Cluster analysis
 - Grouping a set of data objects into clusters
- Clustering is **unsupervised classification**: no predefined classes
- Typical applications
 - As a **stand-alone tool** to get insight into data distribution
 - As a **preprocessing step** for other algorithms

General Applications of Clustering

- Pattern Recognition
- Spatial Data Analysis
 - create thematic maps in GIS by clustering feature spaces
 - detect spatial clusters and explain them in spatial data mining
- Image Processing
- Economic Science (especially market research)
- WWW
 - Document classification
 - Cluster Weblog data to discover groups of similar access patterns

Examples of Clustering Applications

- Marketing: Help marketers discover distinct groups in their customer bases, and then use this knowledge to develop targeted marketing programs
- Land use: Identification of areas of similar land use in an earth observation database
- Insurance: Identifying groups of motor insurance policy holders with a high average claim cost
- City-planning: Identifying groups of houses according to their house type, value, and geographical location
- Earth-quake studies: Observed earth quake epicenters should be clustered along continent faults

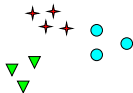
聚类分析的应用

- 数据预处理过程中，对于复杂结构的多维数据可以通过聚类分析的方法对数据进行聚集，使复杂结构数据标准化。
- 用来发现数据项之间的依赖关系，从而去除或合并有密切依赖关系的数据项。
- 为某些数据挖掘方法（如关联规则、粗糙集方法），提供预处理功能。
- 在商业类问题上，聚类分析是细分市场的有效工具，被用来发现不同的客户群，并且它通过对不同的客户群的特征的刻画，被用于研究消费者行为，寻找新的潜在市场等等。

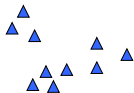
Notion of a Cluster can be Ambiguous



How many clusters?



Six Clusters



Two Clusters

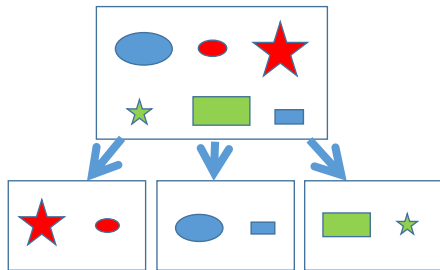
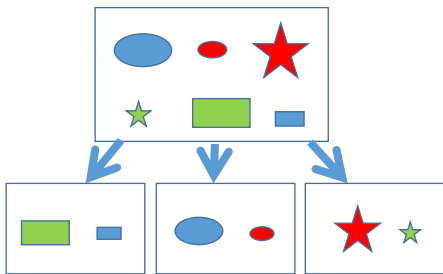


Four Clusters

What Is Good Clustering?

- A good clustering method will produce high quality clusters with
 - high intra-class similarity
 - low inter-class similarity
- The quality of a clustering result depends on both the similarity measure used by the method and its implementation.
- The quality of a clustering method is also measured by its ability to discover some or all of the hidden patterns.

不同的特征导致不同的分类结果



特征

特征

量化

特征向量

颜色



$$\begin{pmatrix} r \\ g \\ b \end{pmatrix}$$

形状



n



$$\begin{pmatrix} r \\ g \\ b \\ n \\ A \\ D \end{pmatrix}$$

大小



$$\begin{pmatrix} A \\ D \end{pmatrix}$$

聚类分析的过程

- 特征选择和提取
- 度量：距离和相似系数
- 聚类算法设计
- 聚类验证

距离

- 绝对值距离 $d_{ij}(1) = \sum_{k=1}^p |x_{ik} - x_{jk}|$
- 欧氏(Euclidean)距离 $d_{ij}(2) = \left[\sum_{k=1}^p (x_{ik} - x_{jk})^2 \right]^{1/2}$
- 切比雪夫(Chebychev)距离 $d_{ij}(\infty) = \max_{1 \leq k \leq p} |x_{ik} - x_{jk}|$
- 明氏(Minkowski)距离 $d_{ij}(q) = \left(\sum_{k=1}^p |x_{ik} - x_{jk}|^q \right)^{1/q}$

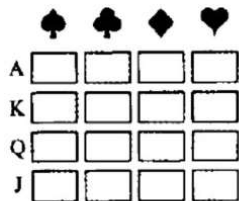
相似系数

- 夹角余弦

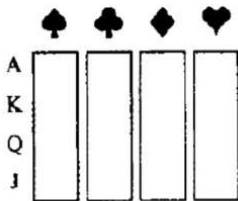
$$\cos \theta_{st} = \frac{\sum_{i=1}^n x_{is} x_{it}}{\sqrt{\sum_{i=1}^n x_{is}^2 \cdot \sum_{i=1}^n x_{it}^2}}$$

- Pearson 相关系数 $r_{st} = \frac{\sum_{i=1}^n (x_{is} - \bar{x}_s)(x_{it} - \bar{x}_t)}{\sqrt{\sum_{i=1}^n (x_{is} - \bar{x}_s)^2 \cdot \sum_{i=1}^n (x_{it} - \bar{x}_t)^2}}$

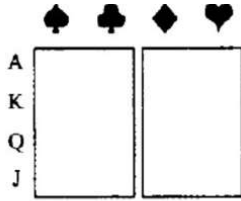
相似性的不同定义



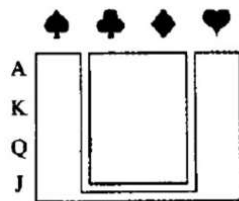
(a) 单张套



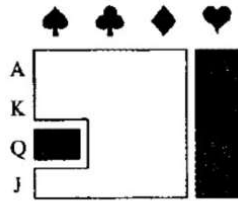
(b) 同花套



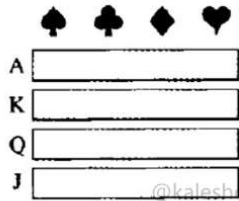
(c) 黑红套



(d) 大小套（桥牌）



(e) 红心加黑桃 Q 与其他套



(f) 同字套

两种主要的聚类方法

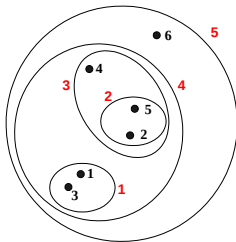
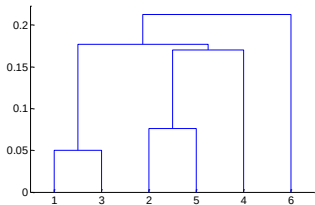
- **层次聚类方法： Hierarchical Clustering** – final clusters are built following a distinct set of sequential steps
- **非层次聚类方法： Non-Hierarchical Clustering** – Clusters are built in such a way that if M clusters are built there is no guarantee that putting together two of the clusters would give rise to the same $(M-1)$ clusters built separately by the method.

两种主要的聚类方法

- **层次聚类方法： Hierarchical Clustering** – final clusters are built following a distinct set of sequential steps
- **非层次聚类方法： Non-Hierarchical Clustering** – Clusters are built in such a way that if M clusters are built there is no guarantee that putting together two of the clusters would give rise to the same $(M-1)$ clusters built separately by the method.

Hierarchical Clustering

- Produces a set of ***nested clusters*** organized as a hierarchical tree
- Can be visualized as a **dendrogram**
 - A tree-like diagram that records the sequences of merges or splits



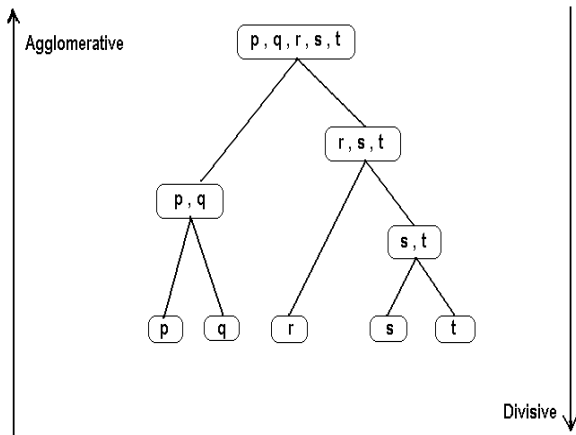
Strengths of Hierarchical Clustering

- No assumptions on the number of clusters
 - Any desired number of clusters can be obtained by 'cutting' the dendrogram at the proper level
- Hierarchical clusterings may correspond to meaningful taxonomies
 - Example in biological sciences (e.g., phylogeny reconstruction, etc), web (e.g., product catalogs) etc

Hierarchical Clustering

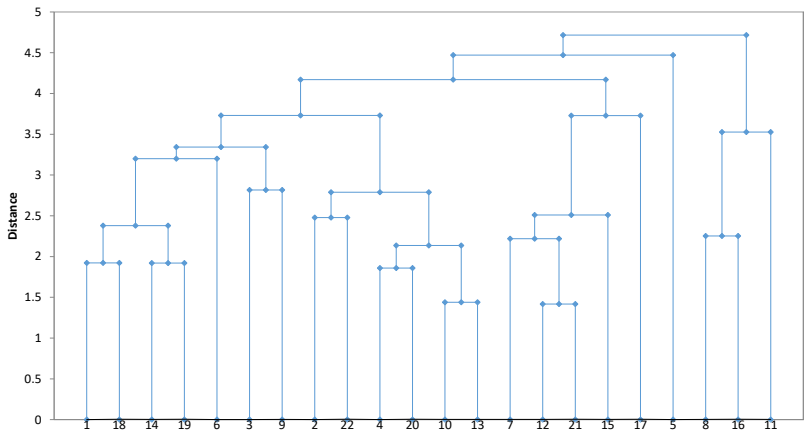
- Two main types of hierarchical clustering
 - **Agglomerative:**
 - Start with the points as individual clusters
 - At each step, merge the closest pair of clusters until only one cluster (or k clusters) left
 - **Divisive:**
 - Start with one, all-inclusive cluster
 - At each step, split a cluster until each cluster contains a point (or there are k clusters)
- Traditional hierarchical algorithms use a similarity or distance matrix
 - Merge or split one cluster at a time

Agglomerative versus Divisive Approaches



The Dendrogram of Hierarchical Clustering

Dendrogram(Average linkage)



Complexity of hierarchical clustering

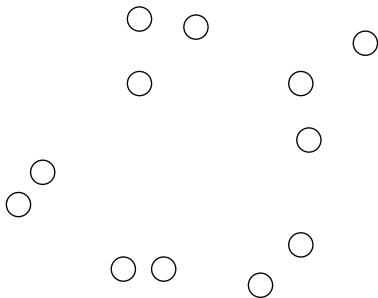
- Distance matrix is used for deciding which clusters to merge/split
- At least quadratic in the number of data points
- Not usable for large datasets

Agglomerative clustering algorithm

- Most popular hierarchical clustering technique
- Basic algorithm
 1. Compute the distance matrix between the input data points
 2. Let each data point be a cluster
 3. **Repeat**
 4. Merge the two closest clusters
 5. Update the distance matrix
 6. **Until** only a single cluster remains
- Key operation is the computation of the distance between two clusters
 - Different definitions of the distance between clusters lead to different algorithms

Input/ Initial setting

- Start with clusters of individual points and a distance/proximity matrix



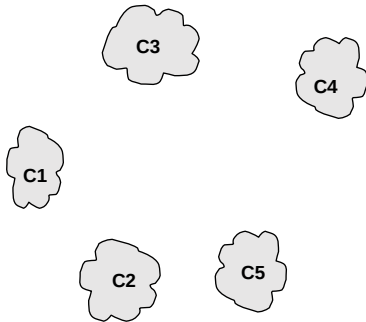
	p1	p2	p3	p4	p5	...
p1						
p2						
p3						
p4						
p5						
⋮						

Distance/Proximity Matrix



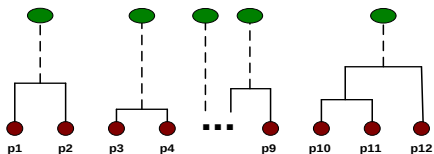
Intermediate State

- After some merging steps, we have some clusters



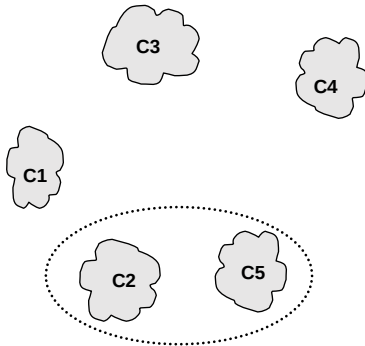
	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

Distance/Proximity Matrix



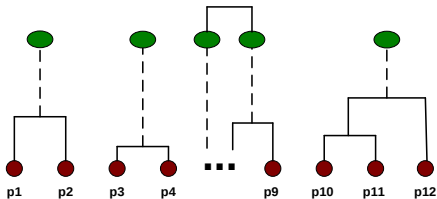
Intermediate State

- Merge the two closest clusters (C2 and C5) and update the distance matrix.



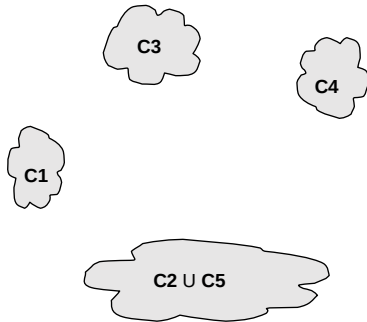
	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

Distance/Proximity Matrix

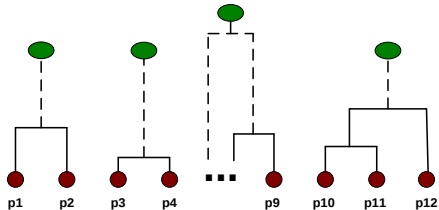


After Merging

- “How do we update the distance matrix?”

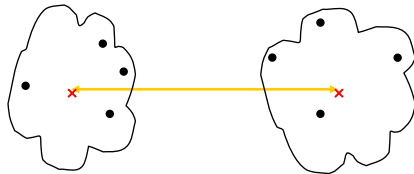
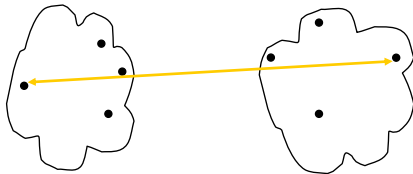
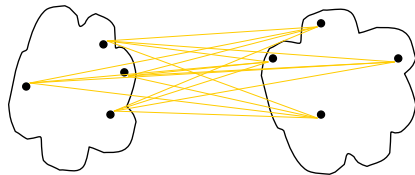
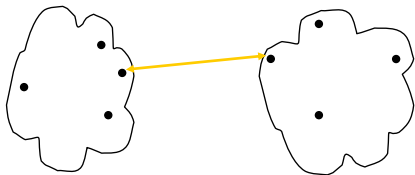
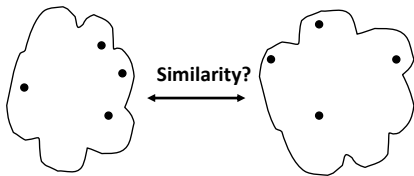


		C2 $\cup$ C5	C3	C4
C1		?		
C2 $\cup$ C5	?	?	?	?
C3		?		
C4		?		



Distance between two clusters

- Each cluster is a set of points
- How do we define distance between two sets of points
 - Lots of alternatives
 - Not an easy task



Distance between two clusters

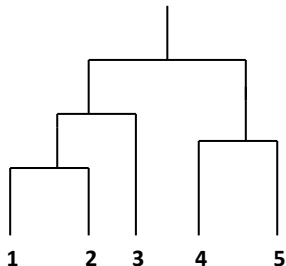
- **Single-link distance** between clusters C_i and C_j is the ***minimum distance*** between any object in C_i and any object in C_j
- The distance is **defined by the two most similar objects**

$$D_{sl}(C_i, C_j) = \min_{x,y} \{d(x,y) \mid x \in C_i, y \in C_j\}$$

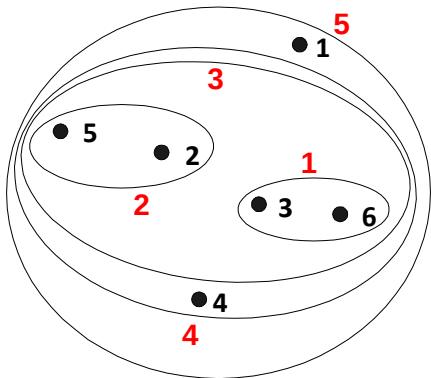
Single-link clustering: example

- Determined by one pair of points, i.e., by one link in the proximity graph.

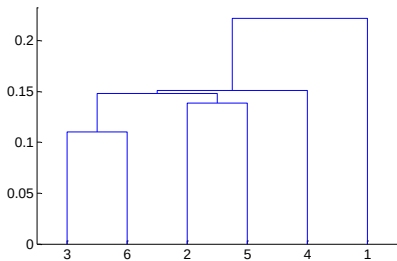
	I1	I2	I3	I4	I5
I1	1.00	0.90	0.10	0.65	0.20
I2	0.90	1.00	0.70	0.60	0.50
I3	0.10	0.70	1.00	0.40	0.30
I4	0.65	0.60	0.40	1.00	0.80
I5	0.20	0.50	0.30	0.80	1.00



Single-link clustering: example

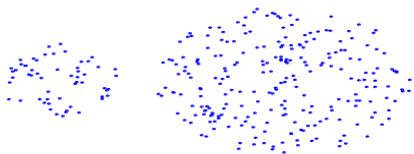


Nested Clusters

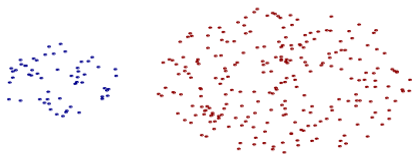


Dendrogram

Strengths of single-link clustering



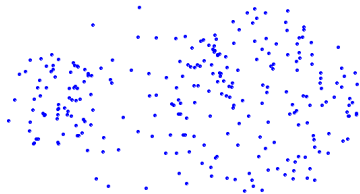
Original Points



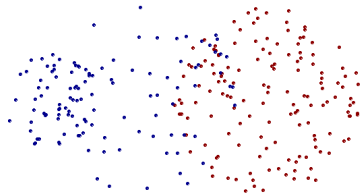
Two Clusters

- Can handle non-elliptical shapes

Limitations of single-link clustering



Original Points



Two Clusters

- Sensitive to noise and outliers
- It produces long, elongated clusters

Distance between two clusters

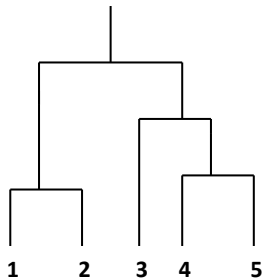
- **Complete-link distance** between clusters C_i and C_j is the *maximum distance* between any object in C_i and any object in C_j
- The distance is **defined by the two most dissimilar objects**

$$D_{cl}(C_i, C_j) = \max_{x,y} \{d(x, y) \mid x \in C_i, y \in C_j\}$$

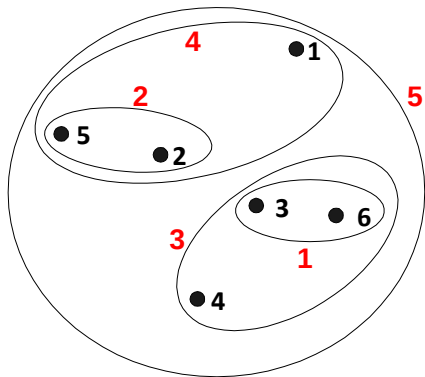
Complete-link clustering: example

- Distance between clusters is determined by the two most distant points in the different clusters

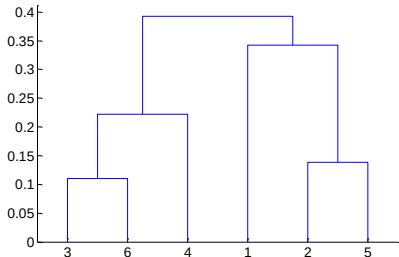
	11	12	13	14	15
11	1.00	0.90	0.10	0.65	0.20
12	0.90	1.00	0.70	0.60	0.50
13	0.10	0.70	1.00	0.40	0.30
14	0.65	0.60	0.40	1.00	0.80
15	0.20	0.50	0.30	0.80	1.00



Complete-link clustering: example

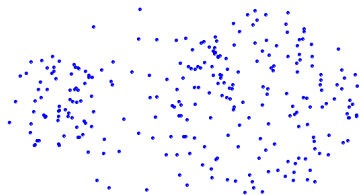


Nested Clusters

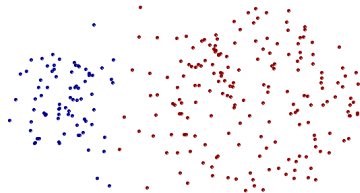


Dendrogram

Strengths of complete-link clustering



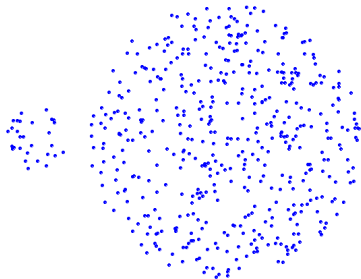
Original Points



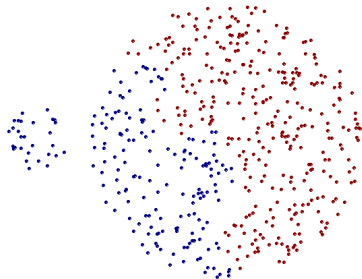
Two Clusters

- **More balanced clusters (with equal diameter)**
- **Less susceptible to noise**

Limitations of complete-link clustering



Original Points



Two Clusters

- Tends to break large clusters
- All clusters tend to have the same diameter – small clusters are merged with larger ones

Distance between two clusters

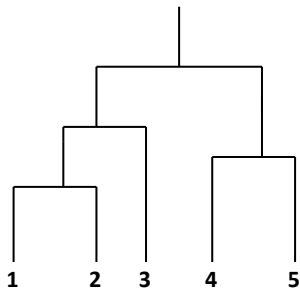
- **Group average distance** between clusters C_i and C_j is the ***average distance*** between any object in C_i and any object in C_j

$$D_{avg}(C_i, C_j) = \frac{1}{|C_i| \times |C_j|} \sum_{x \in C_i, y \in C_j} d(x, y)$$

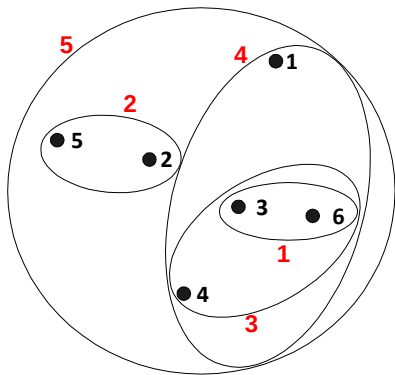
Average-link clustering: example

- Proximity of two clusters is the average of pairwise proximity between points in the two clusters.

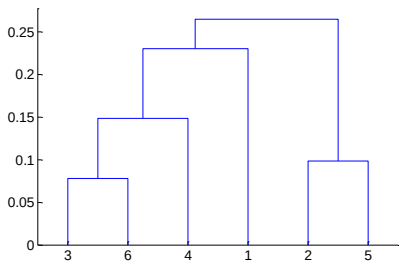
	l1	l2	l3	l4	l5
l1	1.00	0.90	0.10	0.65	0.20
l2	0.90	1.00	0.70	0.60	0.50
l3	0.10	0.70	1.00	0.40	0.30
l4	0.65	0.60	0.40	1.00	0.80
l5	0.20	0.50	0.30	0.80	1.00



Average-link clustering: example



Nested Clusters



Dendrogram

Average-link clustering: discussion

- Compromise between Single and Complete Link
- Strengths
 - Less susceptible to noise and outliers
- Limitations
 - Biased towards globular clusters

Distance between two clusters

- **Centroid distance** between clusters C_i and C_j is the distance between the centroid r_i of C_i and the centroid r_j of C_j

$$D_{centroids}(C_i, C_j) = d(r_i, r_j)$$

Distance between two clusters

- **Ward's distance** between clusters C_i and C_j is the *difference* between the **total within cluster sum of squares for the two clusters separately**, and the **within cluster sum of squares resulting from merging the two clusters** in cluster C_{ij}

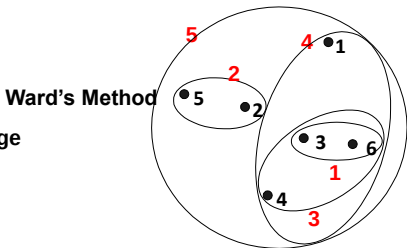
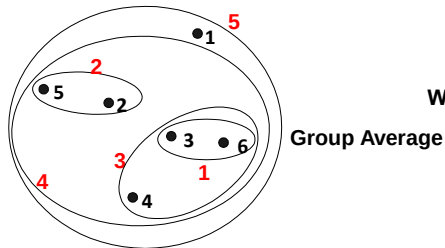
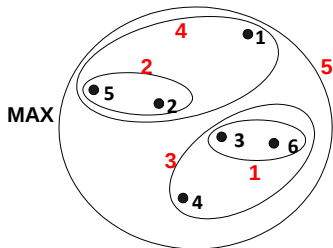
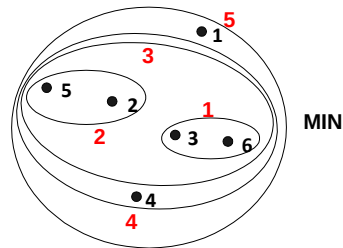
$$D_w(C_i, C_j) = \sum_{x \in C_i} (x - r_i)^2 + \sum_{x \in C_j} (x - r_j)^2 - \sum_{x \in C_{ij}} (x - r_{ij})^2$$

- r_i : centroid of C_i
- r_j : centroid of C_j
- r_{ij} : centroid of C_{ij}

Ward's distance for clusters

- Similar to group average and centroid distance
- Less susceptible to noise and outliers
- Biased towards globular clusters
- Hierarchical analogue of k-means
 - Can be used to initialize k-means

Hierarchical Clustering: Comparison



Hierarchical Clustering: Time and Space requirements

- For a dataset X consisting of n points
- $O(n^2)$ **space**; it requires storing the distance matrix
- $O(n^3)$ **time** in most of the cases
 - There are n steps and at each step the size n^2 distance matrix must be updated and searched
 - Complexity can be reduced to $O(n^2 \log(n))$ time for some approaches by using appropriate data structures

Divisive hierarchical clustering

- Start with a single cluster composed of all data points
- Split this into components
- Continue recursively
- *Monothetic* divisive methods split clusters using one variable/dimension at a time
- *Polythetic* divisive methods make splits on the basis of all variables together
- Any intercluster distance measure can be used
- Computationally intensive, less widely used than agglomerative methods

两种主要的聚类方法

- **层次聚类方法： Hierarchical Clustering** – final clusters are built following a distinct set of sequential steps
- **非层次聚类方法： Non-Hierarchical Clustering** – Clusters are built in such a way that if M clusters are built there is no guarantee that putting together two of the clusters would give rise to the same $(M-1)$ clusters built separately by the method.

K-Means Clustering

- K-Means Clustering is a **non-hierarchical method** in the sense that if one has 2 clusters, say, generated by pre-specifying 2 means (centroids) in the K-means algorithm and 3 clusters generated by pre-specifying 3 means in the K-means algorithm, then it may be the case that no combination of any two clusters of the 3 cluster group can give rise to the 2 cluster grouping.

The *K-Means* Clustering Method

$K=2$

Arbitrarily choose K
object as initial
cluster center

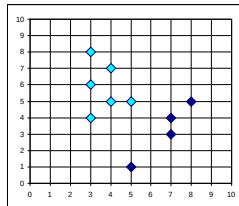
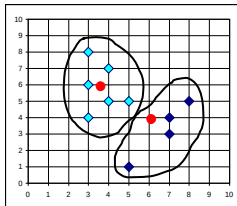
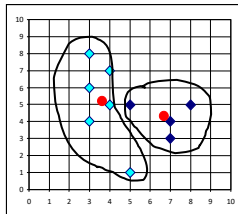
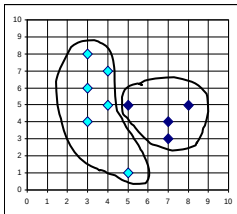
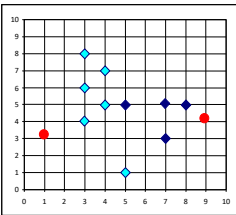
Assign
each
objects
to most
similar
center

reassign

Update
the
cluster
means

reassign

Update
the
cluster
means



Steps in the K-Means Clustering Approach

- Given a set of observations $(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n)$ where each observation is a d-dimensional real vector, then K-means clustering aims to partition the n observations into K sets ($K < n$), $\mathcal{S} = \{S_1, S_2, \dots, S_K\}$ so as to minimize the within-cluster sum of squares (WCSS):

$$\arg \min_{\mathcal{S}} \sum_{i=1}^K \sum_{x_j \in S_i} \|\mathbf{x}_j - \boldsymbol{\mu}_i\|^2 \quad (1)$$

where $\boldsymbol{\mu}_i$ is the mean of the points in S_i . Now minimizing (1) can, in theory, be done by the **integer programming method** but this can be extremely time-consuming. Instead the **Lloyd algorithm** is more often used.

Lloyd Algorithm

- The steps of the **Lloyd algorithm** are as follows. Given the initial set of K-means $\mathbf{m}_1^{(1)}, \dots, \mathbf{m}_K^{(1)}$ which can be specified randomly or by some heuristic, the algorithm proceeds by alternating between two steps:

- Assignment Step: Assign each observation to the cluster with the closest mean

$$S_i^{(t)} = \{x_j: \|x_j - \mathbf{m}_i^{(t)}\| \leq \|x_j - \mathbf{m}_{i^*}^{(t)}\|\} \text{ for all } i^* = 1, 2, \dots, K. \quad (2)$$

- Update Step: Calculate the **new means** to be the centroids of the observations in the clusters, i.e.

$$\mathbf{m}_i^{(t+1)} = \frac{1}{|S_i^{(t)}|} \sum_{x_j \in S_i^{(t)}} x_j \text{ for } i = 1, 2, \dots, K. \quad (3)$$

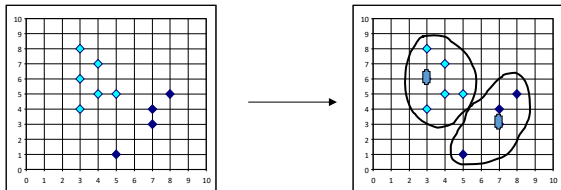
- Repeat the Assignment and Update steps until WCSS (equation (1)) no longer changes. Then the centroids and members of the K clusters are determined.
- Note**: When using random assignment of the K-means to start the algorithm, one might try several starting point K-means and then choose the “best” starting point to be the random K-means that produces the smallest WCSS among all of the random starting points tried in the K-means procedure.
- Regardless of the clustering technique used, one should strive to choose clusters that are interpretable and make sense given the domain-specific knowledge that we have about the problem at hand.

Variations of the *K-Means* Method

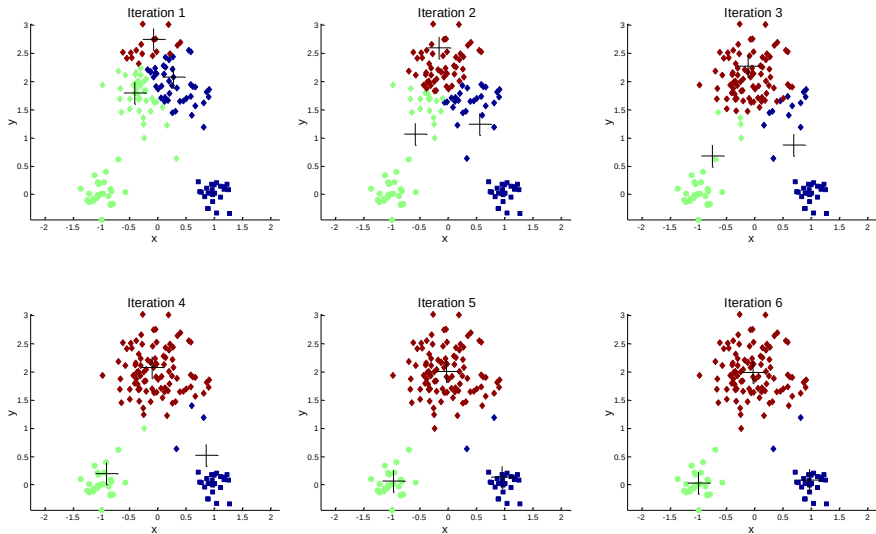
- A few variants of the *k-means* which differ in
 - Selection of the initial *k* means
 - Dissimilarity calculations
 - Strategies to calculate cluster means
- Handling categorical data: *k-modes* (Huang'98)
 - Replacing means of clusters with modes
 - Using new dissimilarity measures to deal with categorical objects
 - Using a frequency-based method to update modes of clusters
 - A mixture of categorical and numerical data: *k-prototype* method

What is the problem of k-Means Method?

- The k-means algorithm is sensitive to outliers !
 - Since an object with an extremely large value may substantially distort the distribution of the data.
- K-Medoids: Instead of taking the **mean** value of the object in a cluster as a reference point, **medoids** can be used, which is the **most centrally located** object in a cluster.



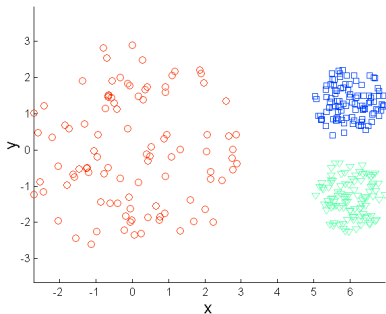
Importance of Choosing Initial Centroids



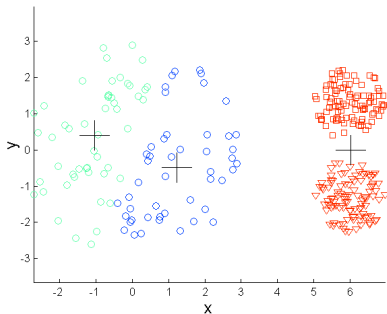
Solutions to Initial Centroids Problem

- Multiple runs
 - Helps, but probability is not on your side
- Sample and use hierarchical clustering to determine initial centroids
- Select more than k initial centroids and then select among these initial centroids
 - Select most widely separated
- Postprocessing
- Bisecting K-means
 - Not as susceptible to initialization issues

Limitations of K-means: Differing Density

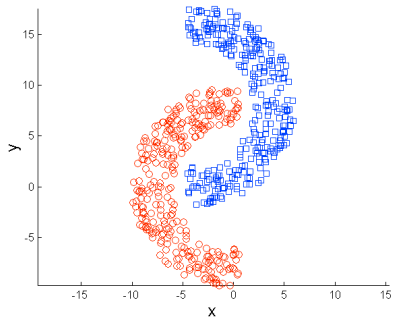


Original Points

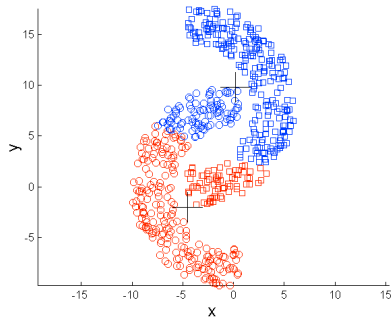


K-means (3 Clusters)

Limitations of K-means: Non-globular Shapes



Original Points



K-means (2 Clusters)

Model-based clustering

- Assume data generated from **k** probability distributions
- **Goal:** find the distribution parameters
- **Algorithm:** Expectation Maximization (EM)
- **Output:** Distribution parameters and a **soft** assignment of points to clusters

Model-based clustering

- Assume k probability distributions with parameters: $(\theta_1, \dots, \theta_k)$
- Given data X , compute $(\theta_1, \dots, \theta_k)$ such that $\Pr(X|\theta_1, \dots, \theta_k)$ [likelihood] or $\ln(\Pr(X|\theta_1, \dots, \theta_k))$ [loglikelihood] is maximized.
- Every point $x \in X$ need not be generated by a single distribution but it can be generated by multiple distributions with some probability [soft clustering]

EM Algorithm

- Initialize k distribution parameters $(\theta_1, \dots, \theta_k)$; Each distribution parameter corresponds to a cluster center
- Iterate between two steps
 - **E**xpectation step: (probabilistically) assign points to clusters
 - **M**aximization step: estimate model parameters that maximize the likelihood for the given assignment of points

EM Algorithm

- Initialize **k** cluster centers
- Iterate between two steps
 - **E**xpectation step: assign points to clusters

$$\Pr(x_i \in C_k) = \Pr(x_i | C_k) / \sum_j \Pr(x_i | C_j)$$
$$w_k = \frac{\sum_i \Pr(x_i \in C_k)}{n}$$

- **M**aximization step: estimate model parameters

$$r_k = \frac{1}{n} \sum_{i=1}^n \frac{\Pr(x_i \in C_k)}{\sum_k \Pr(x_i \in C_k)}$$

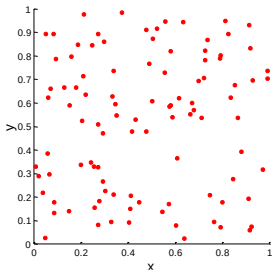
Cluster Evaluation

Cluster Validity

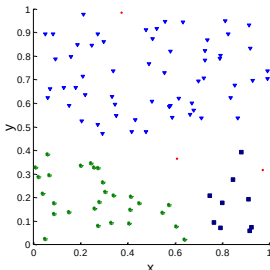
- For supervised classification we have a variety of measures to evaluate how good our model is
 - Accuracy, precision, recall
- For cluster analysis, the analogous question is how to evaluate the “goodness” of the resulting clusters?
- But “clusters are in the eye of the beholder”!
- Then why do we want to evaluate them?
 - To avoid finding patterns in noise
 - To compare clustering algorithms
 - To compare two sets of clusters
 - To compare two clusters

Clusters found in Random Data

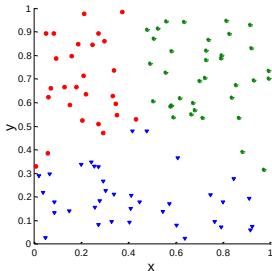
Random Points



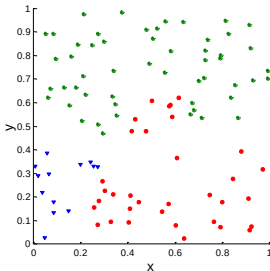
DBSCAN



K-means



Complete Link



Different Aspects of Cluster Validation

1. Determining the **clustering tendency** of a set of data, i.e., distinguishing whether non-random structure actually exists in the data.
2. Comparing the results of a cluster analysis to externally known results, e.g., to externally given class labels.
3. Evaluating how well the results of a cluster analysis fit the data *without* reference to external information.
 - Use only the data
4. Comparing the results of two different sets of cluster analyses to determine which is better.
5. Determining the 'correct' number of clusters.

For 2, 3, and 4, we can further distinguish whether we want to evaluate the entire clustering or just individual clusters.

Measures of Cluster Validity

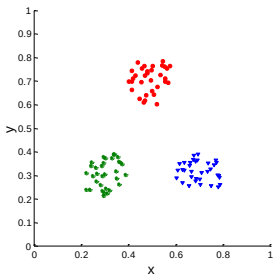
- Numerical measures that are applied to judge various aspects of cluster validity, are classified into the following three types.
 - **External Index:** Used to measure the extent to which cluster labels match externally supplied class labels.
 - Entropy
 - **Internal Index:** Used to measure the goodness of a clustering structure *without* respect to external information.
 - Sum of Squared Error (SSE)
 - **Relative Index:** Used to compare two different clusterings or clusters.
 - Often an external or internal index is used for this function, e.g., SSE or entropy
- Sometimes these are referred to as **criteria** instead of **indices**
 - However, sometimes criterion is the general strategy and index is the numerical measure that implements the criterion.

Measuring Cluster Validity Via Correlation

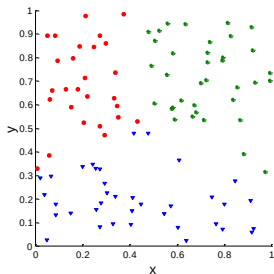
- Two matrices
 - Proximity Matrix
 - “Incidence” Matrix
 - One row and one column for each data point
 - An entry is 1 if the associated pair of points belong to the same cluster
 - An entry is 0 if the associated pair of points belongs to different clusters
- Compute the correlation between the two matrices
 - Since the matrices are symmetric, only the correlation between $n(n-1) / 2$ entries needs to be calculated.
- High correlation indicates that points that belong to the same cluster are close to each other.
- Not a good measure for some density or contiguity based clusters.

Measuring Cluster Validity Via Correlation

- Correlation of incidence and proximity matrices for the K-means clusterings of the following two data sets.



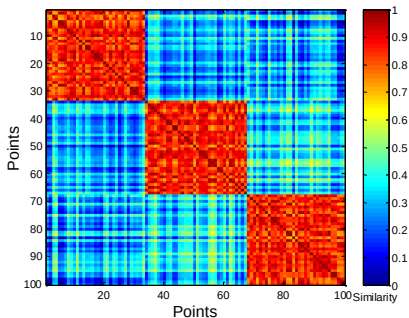
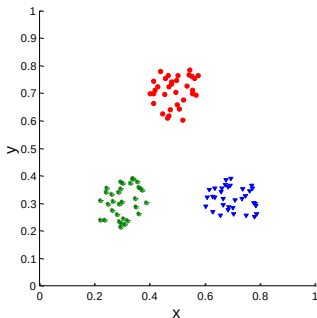
Corr = -0.9235



Corr = -0.5810

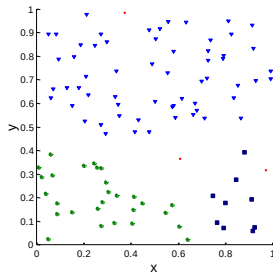
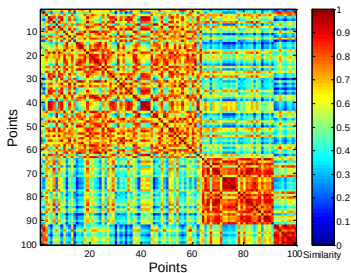
Using Similarity Matrix for Cluster Validation

- Order the similarity matrix with respect to cluster labels and inspect visually.



Using Similarity Matrix for Cluster Validation

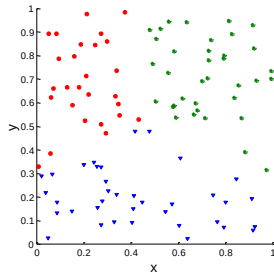
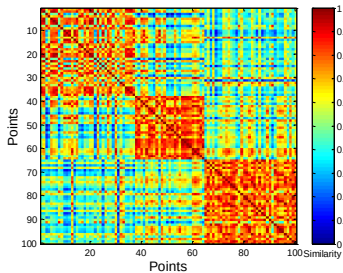
- Clusters in random data are not so crisp



DBSCAN

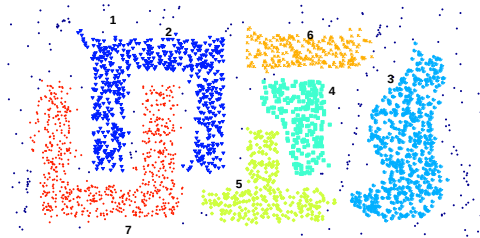
Using Similarity Matrix for Cluster Validation

- Clusters in random data are not so crisp

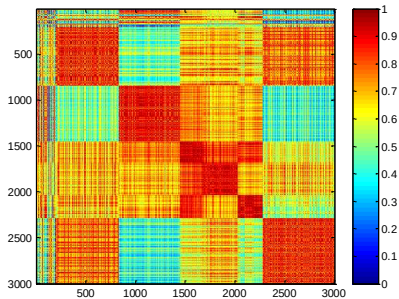


K-means

Using Similarity Matrix for Cluster Validation

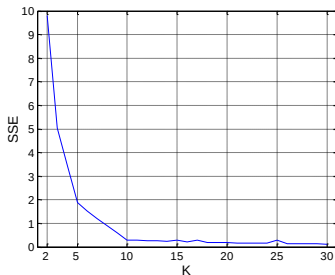
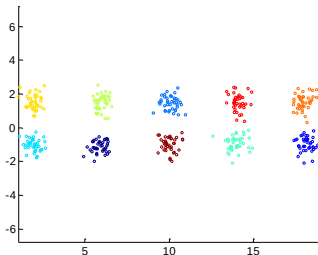


DBSCAN



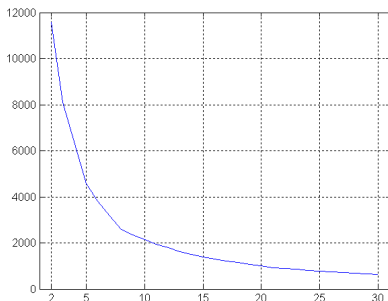
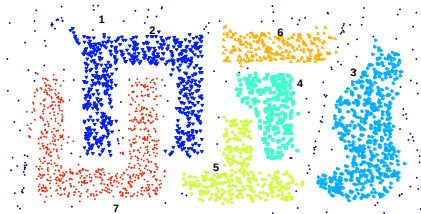
Internal Measures: SSE

- Clusters in more complicated figures aren't well separated
- Internal Index: Used to measure the goodness of a clustering structure without respect to external information
 - SSE
- SSE is good for comparing two clusterings or two clusters (average SSE).
- Can also be used to estimate the number of clusters



Internal Measures: SSE

- SSE curve for a more complicated data set



SSE of clusters found using K-means

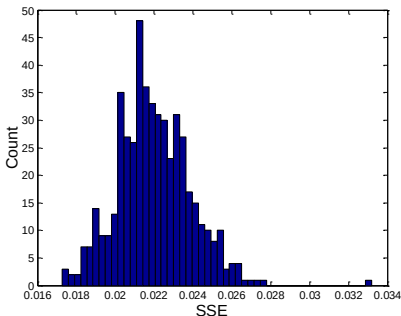
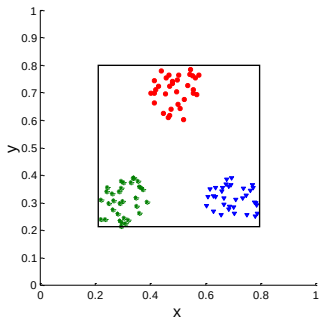
Framework for Cluster Validity

- Need a framework to interpret any measure.
 - For example, if our measure of evaluation has the value, 10, is that good, fair, or poor?
- Statistics provide a framework for cluster validity
 - The more “atypical” a clustering result is, the more likely it represents valid structure in the data
 - Can compare the values of an index that result from random data or clusterings to those of a clustering result.
 - If the value of the index is unlikely, then the cluster results are valid
 - These approaches are more complicated and harder to understand.
- For comparing the results of two different sets of cluster analyses, a framework is less necessary.
 - However, there is the question of whether the difference between two index values is significant

Statistical Framework for SSE

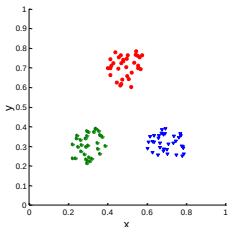
- Example

- Compare SSE of 0.005 against three clusters in random data
- Histogram shows SSE of three clusters in 500 sets of random data points of size 100 distributed over the range 0.2 – 0.8 for x and y values

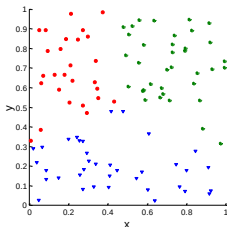


Statistical Framework for Correlation

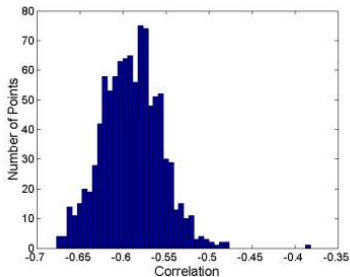
- Correlation of incidence and proximity matrices for the K-means clusterings of the following two data sets.



Corr = -0.9235



Corr = -0.5810



Internal Measures: Cohesion and Separation

- **Cluster Cohesion:** Measures how closely related are objects in a cluster
 - Example: SSE
- **Cluster Separation:** Measure how distinct or well-separated a cluster is from other clusters
- Example: Squared Error

- Cohesion is measured by the within cluster sum of squares (SSE)

$$WSS = \sum_i \sum_{x \in C_i} (x - m_i)^2$$

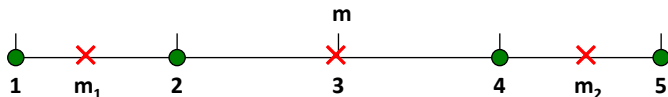
- Separation is measured by the between cluster sum of squares

$$BSS = \sum_i |C_i| (m - m_i)^2$$

- Where $|C_i|$ is the size of cluster i

Internal Measures: Cohesion and Separation

- Example: SSE
 - $BSS + WSS = \text{constant}$



K=1 cluster:

$$WSS = (1-3)^2 + (2-3)^2 + (4-3)^2 + (5-3)^2 = 10$$

$$BSS = 4 \times (3-3)^2 = 0$$

$$Total = 10 + 0 = 10$$

K=2 clusters:

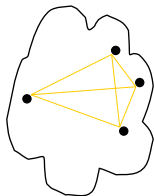
$$WSS = (1-1.5)^2 + (2-1.5)^2 + (4-4.5)^2 + (5-4.5)^2 = 1$$

$$BSS = 2 \times (3-1.5)^2 + 2 \times (4.5-3)^2 = 9$$

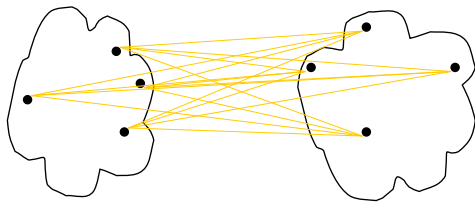
$$Total = 1 + 9 = 10$$

Internal Measures: Cohesion and Separation

- A proximity graph based approach can also be used for cohesion and separation.
 - Cluster cohesion is the sum of the weight of all links within a cluster.
 - Cluster separation is the sum of the weights between nodes in the cluster and nodes outside the cluster.



cohesion



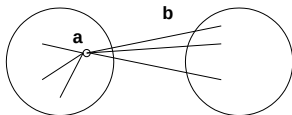
separation

Internal Measures: Silhouette Coefficient

- Silhouette Coefficient combine ideas of both cohesion and separation, but for individual points, as well as clusters and clusterings
- For an individual point, i
 - Calculate a = average distance of i to the points in its cluster
 - Calculate b = min (average distance of i to points in another cluster)
 - The silhouette coefficient for a point is then given by

$$s = 1 - a/b \text{ if } a < b, \text{ (or } s = b/a - 1 \text{ if } a \geq b, \text{ not the usual case)}$$

- Typically between 0 and 1.
- The closer to 1 the better.



- Can calculate the Average Silhouette width for a cluster or a clustering

External Measures of Cluster Validity: Entropy and Purity

Table 5.9. K-means Clustering Results for LA Document Data Set

Cluster	Entertainment	Financial	Foreign	Metro	National	Sports	Entropy	Purity
1	3	5	40	506	96	27	1.2270	0.7474
2	4	7	280	29	39	2	1.1472	0.7756
3	1	1	1	7	4	671	0.1813	0.9796
4	10	162	3	119	73	2	1.7487	0.4390
5	331	22	5	70	13	23	1.3976	0.7134
6	5	358	12	212	48	13	1.5523	0.5525
Total	354	555	341	943	273	738	1.1450	0.7203

entropy For each cluster, the class distribution of the data is calculated first, i.e., for cluster j we compute p_{ij} , the ‘probability’ that a member of cluster j belongs to class i as follows: $p_{ij} = m_{ij}/m_j$, where m_j is the number of values in cluster j and m_{ij} is the number of values of class i in cluster j . Then using this class distribution, the entropy of each cluster j is calculated using the standard formula $e_j = \sum_{i=1}^L p_{ij} \log_2 p_{ij}$, where the L is the number of classes. The total entropy for a set of clusters is calculated as the sum of the entropies of each cluster weighted by the size of each cluster, i.e., $e = \sum_{i=1}^K \frac{m_i}{m} e_j$, where m_j is the size of cluster j , K is the number of clusters, and m is the total number of data points.

purity Using the terminology derived for entropy, the purity of cluster j , is given by $\text{purity}_j = \max_i p_{ij}$ and the overall purity of a clustering by $\text{purity} = \sum_{i=1}^K \frac{m_i}{m} \text{purity}_j$.

Final Comment on Cluster Validity

“The validation of clustering structures is the most difficult and frustrating part of cluster analysis.

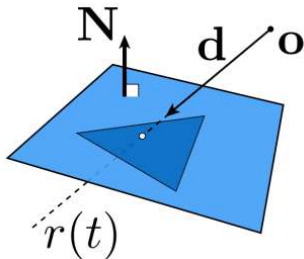
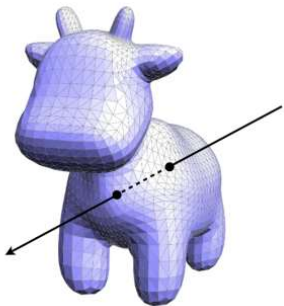
Without a strong effort in this direction, cluster analysis will remain a black art accessible only to those true believers who have experience and great courage.”

Algorithms for Clustering Data, Jain and Dubes

应用：搜索加速结构

大量搜索计算

- 点与三角网格的求交
- 点与三角形的求交

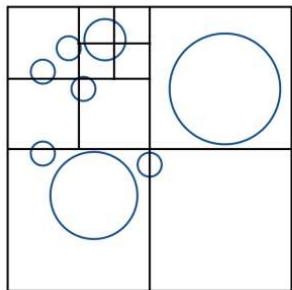


加速策略: Bounding Volumes

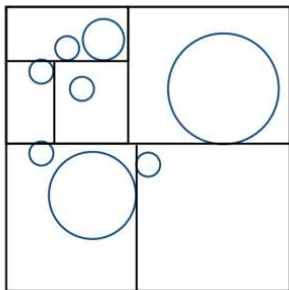
- Axis-Aligned Bounding Box (AABB) (轴对齐包围盒)
- 包围球



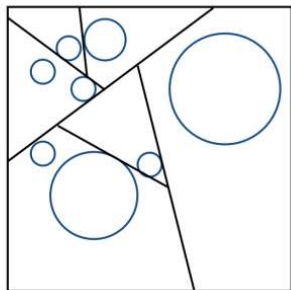
加速策略：空间组织



Oct-Tree

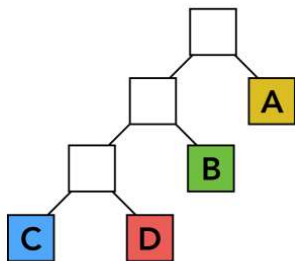
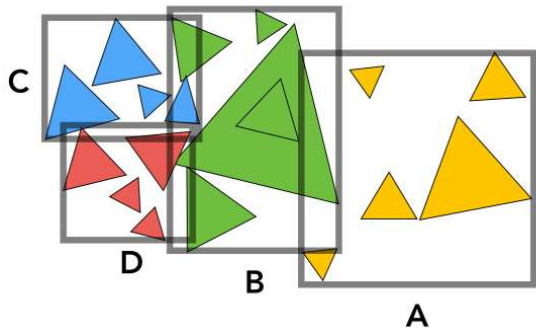


KD-Tree



BSP-Tree

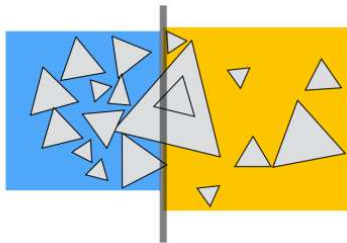
加速结构: Bounding Volume Hierarchy (BVH)



Spatial vs Object Partitions

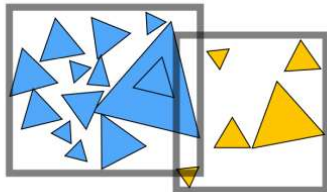
Spatial partition (e.g. KD-tree)

- Partition space into non-overlapping regions
- An object can be contained in multiple regions



Object partition (e.g. BVH)

- Partition set of objects into disjoint subsets
- Bounding boxes for each set may overlap in space



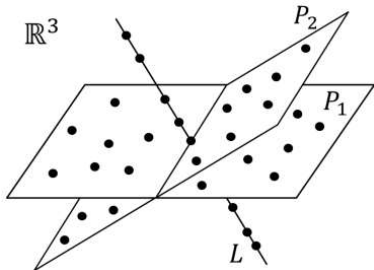
Clustering-II

复杂高维数据

子空间聚类 (Subspace clustering)

- Input:

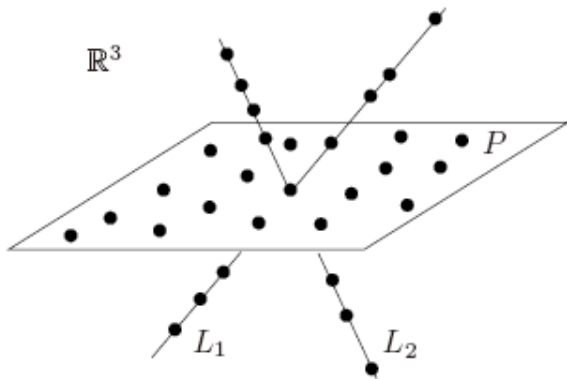
- high dimensional datasets having low intrinsic dimensions
- $\{x_j\}_{j=1,\dots,N}, x_t \in \mathbb{R}^D$



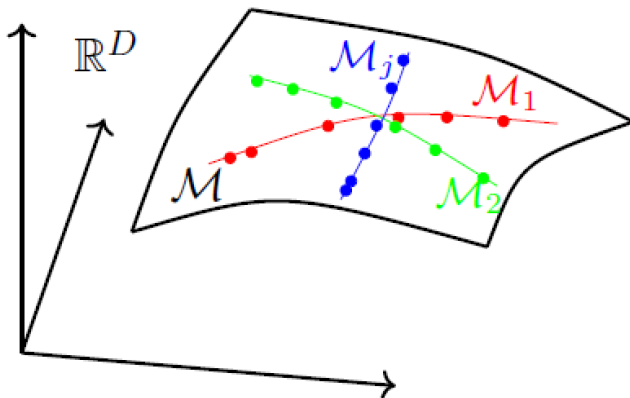
- Output:

- multiple low-dimensional linear subspaces
- L, P_1, P_2

Sub-manifolds



High-Dim Data with Mixed (multiple) Low-Dim Structures



Sparse subspace clustering(SSC)

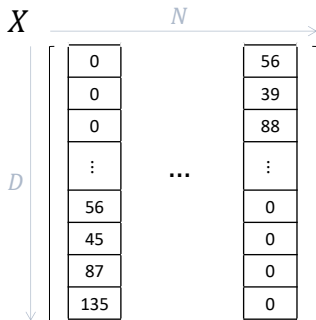
[Elhamifar and Vidal 2009]

- Based on the observation:
 - each point can always be represented as a linear combination of the points belonging to the same subspace

$$x_j = \sum_{i=1}^N w_{ij} x_i, j = 1, \dots, N$$

$X = XW$, where

$$X = (x_1, \dots, x_N) \in \mathbb{R}^{D \times N}$$



Sparse subspace clustering(SSC)

[Elhamifar and Vidal 2009]

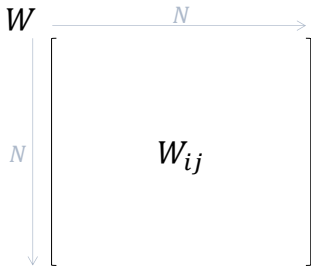
- Based on the observation:
 - each point can always be represented as a linear combination of the points belonging to the same subspace

$$x_j = \sum_{i=1}^N w_{ij} x_i, j = 1, \dots, N$$

$X = XW$, where

$$X = (x_1, \dots, x_N) \in \mathbb{R}^{D \times N}$$

$$W = (w_{ij}) \in \mathbb{R}^{N \times N}$$



Sparse subspace clustering(SSC)

[Elhamifar and Vidal 2009]

- Based on the observation:
 - each point can always be represented as a linear combination of the points belonging to the same subspace

$$\begin{aligned} \min_W \quad & \|W\|_{1,1} \\ \text{s.t.} \quad & X = XW, \text{diag}(W) = 0 \end{aligned}$$



$$\begin{aligned} \min_W \quad & \|XW - X\|_F^2 + \lambda \|W\|_{1,1} \\ \text{s.t.} \quad & \text{diag}(W) = 0 \end{aligned}$$

where $\|W\|_{1,1} = \sum \|w_j\|_1 = \sum |w_{ij}|$

SSQP

[Wang et al. 2011]

$$\begin{aligned} \min_W \quad & \|XW - X\|_F^2 + \lambda \|W\|_{1,1} \\ \text{s. t.} \quad & \text{diag}(W) = 0 \end{aligned}$$



$$\begin{aligned} \min_W \quad & \|XW - X\|_F^2 + \lambda \|W^T W\|_{1,1} \\ \text{s. t.} \quad & W \geq 0, \text{diag}(W) = 0 \end{aligned}$$

- $W \geq 0$: provides better interpretations
- $\|W^T W\|_{1,1}$: more efficient than SSC
- Block diagonal property:

$$\bar{W}^* = \Gamma^{-1} \bar{W} \Gamma = \begin{pmatrix} \bar{W}^*_1 & & & 0 \\ & \bar{W}^*_2 & & \\ & & \ddots & \\ 0 & & & \bar{W}^*_K \end{pmatrix}_{N \times N}$$

**Desired property
for affinity matrix!**

where Γ is a permutation matrix, submatrix $\bar{W}^*_k \in \mathbb{R}^{N_k \times N_k}$

Supervised learning (DL)

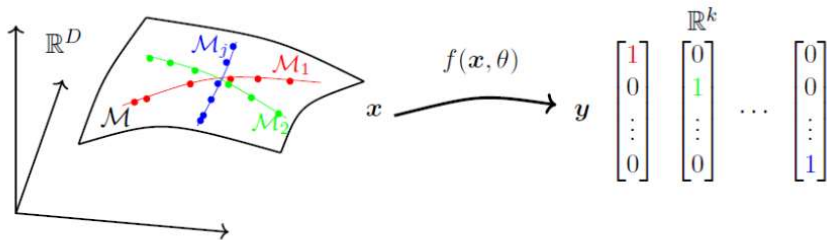


Figure: Black Box Classification: y is the class label of x represented as a “one-hot” vector in $\mathbb{R}^k$. To learn a nonlinear mapping $f(\cdot, \theta) : x \mapsto y$, say modeled by a deep network.

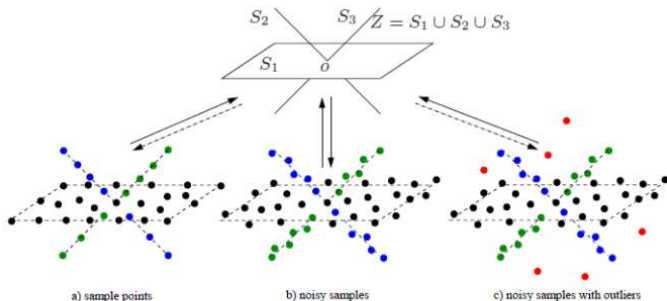
In a supervised setting, using cross-entropy (CE) loss:

$$\min_{\theta \in \Theta} \text{CE}(\theta, x, y) \doteq -\mathbb{E}[\langle y, \log[f(x, \theta)] \rangle] \approx -\frac{1}{m} \sum_{i=1}^m \langle y_i, \log[f(x_i, \theta)] \rangle$$

A Principled Computational Approach

For high-dim data with mixed **low-dim** structures:

learn to compress, and compress to learn!



Generalized PCA for mixture of subspaces [Vidal, Ma, and Sastry, 2005]

Clustering via Compression

[Yi Ma, Harm Derksen, Wei Hong, and John Wright, TPAMI'07]

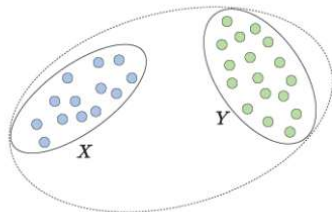
A Fundamental Idea:

Data belong to mixed low-dim structures should be compressible.

Cluster Criterion:

Whether the number of binary bits required to store the data:

$$\#bits(\mathbf{X} \cup \mathbf{Y}) \geq \#bits(\mathbf{X}) + \#bits(\mathbf{Y})?$$



*"The whole is greater than the sum of the parts."
– Aristotle, 320 BC*

Coding Length Function for Subspace-Like Data

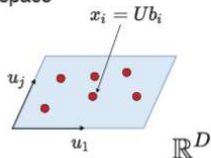
Theorem (Ma, TPAMI'07)

The number of bits needed to encode data $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m] \in \mathbb{R}^{D \times m}$ up to a precision $\|\mathbf{x} - \hat{\mathbf{x}}\|_2 \leq \epsilon$ is bounded by:

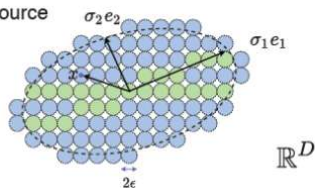
$$L(\mathbf{X}, \epsilon) \doteq \left(\frac{m + D}{2} \right) \log \det \left(\mathbf{I} + \frac{D}{m\epsilon^2} \mathbf{X} \mathbf{X}^\top \right).$$

This can be derived from constructively quantifying SVD of $\mathbf{X}$ or by sphere packing $\text{vol}(\mathbf{X})$ as samples of a noisy Gaussian source.

Linear subspace



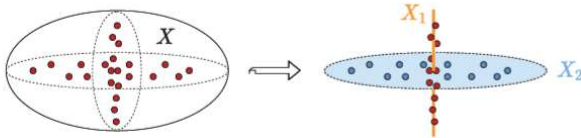
Gaussian source



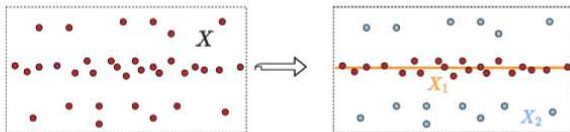
Cluster to Compress

$$L(\mathbf{X}) \geq L^c(\mathbf{X}) \doteq L(\mathbf{X}_1) + L(\mathbf{X}_2) + H(|\mathbf{X}_1|, |\mathbf{X}_2|)?$$

partitioning:



sifting:



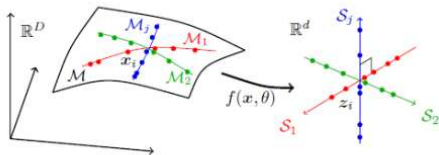
Represent Mixed Data

Given

samples $x \in \mathbb{R}^D$ drawn from a mixture of k submanifolds $\mathcal{M} = \{\mathcal{M}_j\}_{j=1}^k$, we

seek a good representation

$z \in \mathbb{R}^d$ through a continuous mapping: $f(x, \theta) : \mathbb{R}^D \rightarrow \mathbb{R}^d$.



Goals of “**re-present**” the data $f(x, \theta) : x \mapsto z$:

- from non-parametric (samples) to more compact (models).
- from nonlinear structures in x to linear in z .
- from separable x to maximally discriminative z .

What is a good representation? (Why a deep neural network?)

Seeking a Linear Discriminative Representation (LDR)

Desiderata: Representation $z = f(x, \theta)$ have the following properties:

- ① *Within-Class Compressible:* Features of the same class/cluster should be highly compressed in a **low-dimensional** linear subspace.
- ② *Between-Class Discriminative:* Features of different classes/clusters should be in highly **incoherent** linear subspaces.
- ③ *Maximally Informative Representation:* Dimension (or variance) of features for each class/cluster should be **as large as possible**.

Is there a principled measure for all such properties, together?

Measure of Compactness for a Linear Representation

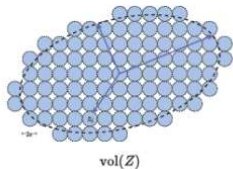
Consider a feature mapping:

$$\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_m] \in \mathbb{R}^{D \times m} \xrightarrow{f(\mathbf{x}, \theta)} \mathbf{Z}(\theta) = [\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_m] \in \mathbb{R}^{d \times m}.$$

The average coding length per sample (rate) subject to a distortion ϵ :

$$R(\mathbf{Z}, \epsilon) \doteq \frac{1}{2} \log \det \left(\mathbf{I} + \frac{d}{m\epsilon^2} \mathbf{Z} \mathbf{Z}^\top \right). \quad (3)$$

Rate distortion is an intrinsic measure for the volume of all features.



Learning Representation to Cluster and Classify

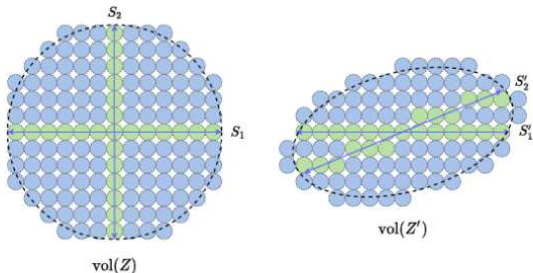
A Fundamental Idea: maximize the **difference** between the coding rate of all features and the average rate of features in the classes:

$$\Delta R(Z, \Pi, \epsilon) = \underbrace{\frac{1}{2} \log \det \left(I + \frac{d}{m\epsilon^2} Z Z^\top \right)}_R - \underbrace{\sum_{j=1}^k \frac{\text{tr}(\Pi_j)}{2m} \log \det \left(I + \frac{d}{\text{tr}(\Pi_j)\epsilon^2} Z \Pi_j Z^\top \right)}_{R^c}.$$

- R : **expand** all features Z as **large** as possible.
- R^c : **compress** each class Z_j as **small** as possible.

Slogan: similarity contracts and dissimilarity contrasts!

Sphere Packing and Counting



Example: two subspaces S_1 and S_2 in $\mathbb{R}^2$.

- $\log \#(\text{green spheres} + \text{blue spheres}) = \text{rate of span of all samples } R.$
- $\log \#(\text{green spheres}) = \text{rate of the two subspaces } R^c.$
- $\log \#(\text{blue spheres}) = \text{rate reduction gain } \Delta R.$

Maximal Coding Rate Reduction (MCR²)

[Yu, Chan, You, Song, Ma, NeurIPS2020]

Learn a mapping $f(x, \theta)$ (for a given partition Π):

$$X \xrightarrow{f(x, \theta)} Z(\theta) \xrightarrow{\Pi, \epsilon} \Delta R(Z(\theta), \Pi, \epsilon) \quad (5)$$

so as to **Maximize the Coding Rate Reduction (MCR²)**:

$$\begin{aligned} \max_{\theta} \quad & \Delta R(Z(\theta), \Pi, \epsilon) = R(Z(\theta), \epsilon) - R^c(Z(\theta), \epsilon \mid \Pi), \\ \text{subject to} \quad & \|Z_j(\theta)\|_F^2 = m_j, \Pi \in \Omega. \end{aligned} \quad (6)$$

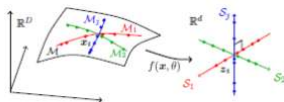
Since ΔR is *monotonic* in the scale of Z , one needs to:

normalize the features $z = f(x, \theta)$ so as to compare $Z(\theta)$ and $Z(\theta')$!

Batch normalization, Sergey Ioffe and Christian Szegedy, 2015.

Layer normalization'16, instance normalization'16; group normalization'18...

Compared to conventional deep neural networks:



	Conventional DNNs	Compression ReduNets
Objectives	label fitting	rate reduction
Deep architectures	trial & error	iterative optimization
Layer operators	empirical	projected gradient
Shift invariance	CNNs+augmentation	invariant ReduNets
Initializations	random/pre-design	forward computed
Training/fine-tuning	back prop	forward/back prop
Interpretability	black box	white box
Representations	unknown	incoherent subspaces

See more...

- 马毅（加利福尼亚大学伯克利分校）：
 - 基于第一原理的深度（卷积）神经网络
- <https://mp.weixin.qq.com/s/JopCQqoFq5CDl6nikrQ7yw>

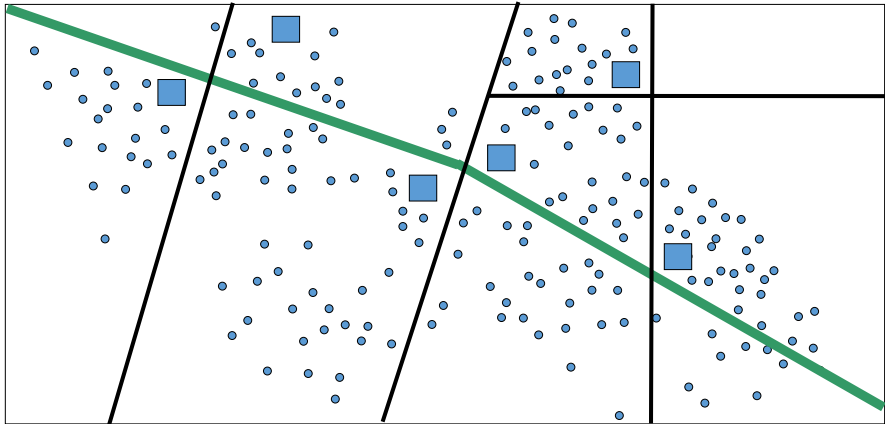
Summary

Problems and Challenges

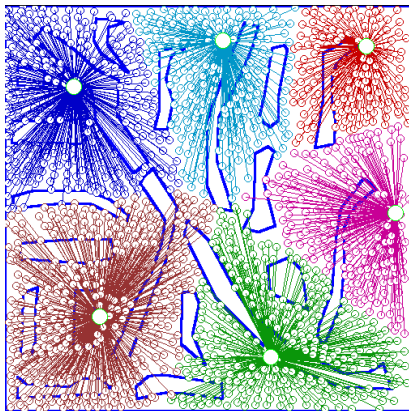
- Considerable progress has been made in scalable clustering methods
 - Partitioning: k-means, k-medoids, CLARANS
 - Hierarchical: BIRCH, CURE
 - Density-based: DBSCAN, CLIQUE, OPTICS
 - Grid-based: STING, WaveCluster
 - Model-based: Autoclass, Denclue, Cobweb
- Current clustering techniques do not address all the requirements adequately
- Constraint-based clustering analysis: Constraints exist in data space (bridges and highways) or in user queries

Constraint-Based Clustering Analysis

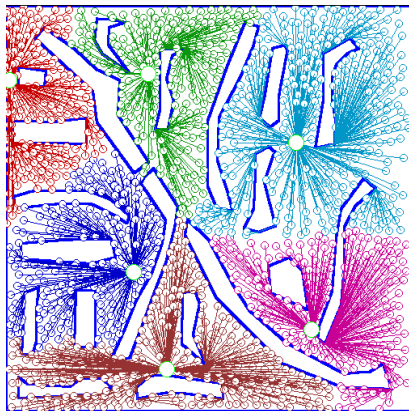
- Clustering analysis: less parameters but more user-desired constraints, e.g., an ATM allocation problem



Clustering With Obstacle Objects



Not Taking obstacles into account



Taking obstacles into account

Summary

- **Cluster analysis** groups objects based on their **similarity** and has wide applications
- Measure of similarity can be computed for **various types of data**
- Clustering algorithms can be **categorized** into partitioning methods, hierarchical methods, density-based methods, grid-based methods, and model-based methods
- **Outlier detection** and analysis are very useful for fraud detection, etc. and can be performed by statistical, distance-based or deviation-based approaches
- There are still lots of research issues on cluster analysis, such as **constraint-based clustering**

有监督的分类？

（一）背景知识

两种蠼螋 Af 和 Apf 已由生物学家罗纳 (w.L.Grogn) 和维尔恩 (W.W.Wirth) 于 1981 年根据它们的触角长 (mm) 和翅长 (mm) 加以区分, 6 只 Apf 和 9 只 Af 蠼螋的触长, 翅长数据如下:

Apf: (1.14, 1.78), (1.18, 1.96), (1.20, 1.86), (1.26, 2.00), (1.28, 2.00), (1.30, 1.96) ;

Af: (1.24, 1.72), (1.36, 1.74), (1.38, 1.64), (1.38, 1.82), (1.38, 1.90), (1.40, 1.70), (1.48, 1.82), (1.54, 1.82), (1.56, 2.08)。

在生物学中, 根据触角长和翅长来识别一只蠼螋标本是 Af 还是 Apf 是很重要的。

（二）要解决的问题

- 1、根据给定的数据, 制定一种方法, 正确区分两类蠼螋;
- 2、用我们的方法对触长、翅长分别为 (1.24, 1.80)、(1.28, 1.84)、(1.40, 2.04) 的三个样本进行识别;
- 3、假设 Af 是宝贵的传粉益虫, Apf 是某种疾病的载体, 在这种情况下我们是否应该修改所用的分类方法, 且如何修改。

References (1)

- R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. SIGMOD'98
- M. R. Anderberg. Cluster Analysis for Applications. Academic Press, 1973.
- M. Ankerst, M. Breunig, H.-P. Kriegel, and J. Sander. Optics: Ordering points to identify the clustering structure, SIGMOD'99.
- P. Arabie, L. J. Hubert, and G. De Soete. Clustering and Classification. World Scientific, 1996
- M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases. KDD'96.
- M. Ester, H.-P. Kriegel, and X. Xu. Knowledge discovery in large spatial databases: Focusing techniques for efficient class identification. SSD'95.
- D. Fisher. Knowledge acquisition via incremental conceptual clustering. Machine Learning, 2:139-172, 1987.
- D. Gibson, J. Kleinberg, and P. Raghavan. Clustering categorical data: An approach based on dynamic systems. In Proc. VLDB'98.
- S. Guha, R. Rastogi, and K. Shim. Cure: An efficient clustering algorithm for large databases. SIGMOD'98.
- A. K. Jain and R. C. Dubes. Algorithms for Clustering Data. Printice Hall, 1988.

References (2)

- L. Kaufman and P. J. Rousseeuw. Finding Groups in Data: an Introduction to Cluster Analysis. John Wiley & Sons, 1990.
- E. Knorr and R. Ng. Algorithms for mining distance-based outliers in large datasets. VLDB'98.
- G. J. McLachlan and K.E. Bksford. Mixture Models: Inference and Applications to Clustering. John Wiley and Sons, 1988.
- P. Michaud. Clustering techniques. Future Generation Computer systems, 13, 1997.
- R. Ng and J. Han. Efficient and effective clustering method for spatial data mining. VLDB'94.
- E. Schikuta. Grid clustering: An efficient hierarchical clustering method for very large data sets. Proc. 1996 Int. Conf. on Pattern Recognition, 101-105.
- G. Sheikholeslami, S. Chatterjee, and A. Zhang. WaveCluster: A multi-resolution clustering approach for very large spatial databases. VLDB'98.
- W. Wang, Yang, R. Muntz, STING: A Statistical Information grid Approach to Spatial Data Mining, VLDB'97.
- T. Zhang, R. Ramakrishnan, and M. Livny. BIRCH : an efficient data clustering method for very large databases. SIGMOD'96.



中国科学技术大学
University of Science and Technology of China

谢谢！



中国科学技术大学
University of Science and Technology of China

数学建模 Mathematical Modeling

刘利刚

中国科学技术大学

数据拟合模型

问题1：农业生产实验模型

- 在研究农业生产的试验中，为分析某地区土豆产量与化肥的关系，得到了每公顷地的氮肥的施肥量与土豆产量的对应关系如下

氮肥量(kg)	0	34	67	101	135	202	259	336	404	471
土豆产量(kg)	15.18	21.36	25.72	32.29	34.03	39.45	43.15	43.46	40.83	30.75

问题2：血药浓度模型

- 通过实验测得一次性快速静脉注射300mg药物后随时间变化的血药浓度数据

t(h)	0.25	0.5	1	1.5	2	3	4	6	8
y(ug/ml)	19.21	18.15	15.36	14.10	12.89	9.32	7.45	5.24	3.01

问题3： 牙膏的销售量

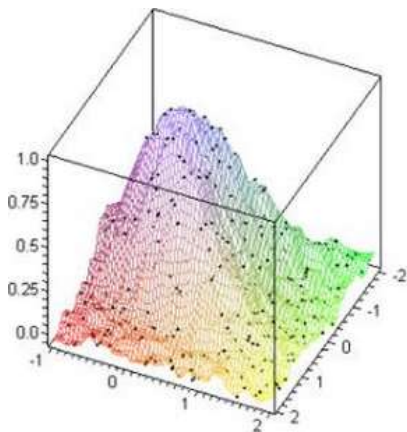
问题 建立牙膏销售量与价格、广告投入之间的模型
预测在不同价格和广告费用下的牙膏销售量
收集了30个销售周期本公司牙膏销售量、价格、
广告费用，及同期其它厂家同类牙膏的平均售价

销售周期	本公司价格(元)	其它厂家价格(元)	广告费用(百万元)	价格差(元)	销售量(百万支)
1	3.85	3.80	5.50	-0.05	7.38
2	3.75	4.00	6.75	0.25	8.51
...	...	...	...	...	...
29	3.80	3.85	5.80	0.05	7.93
30	3.70	4.25	6.80	0.55	9.26

问题4：地形高度测量



数学模型：散乱点插值



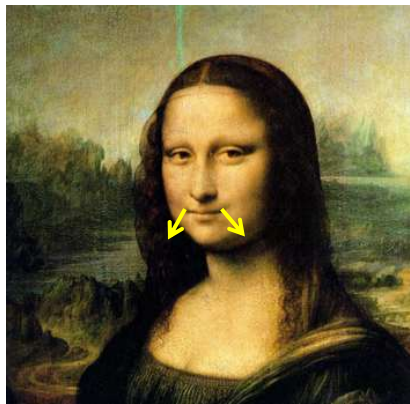
问题5： 图像变形

- 用户交互

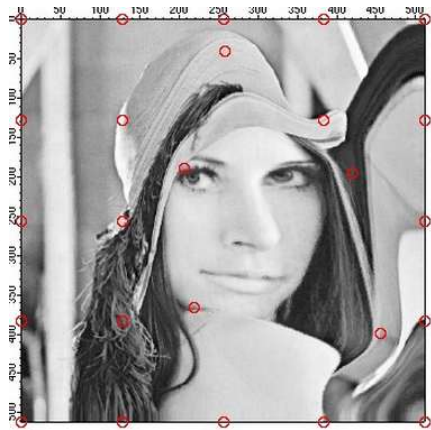
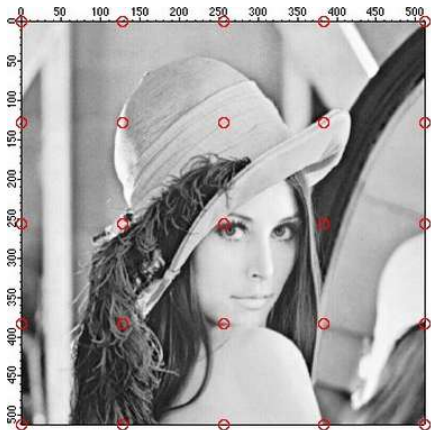


Demo

图像变形：用户交互



数学模型?



问题6:

图像识别: 判断图片中是否有“猫”?



判断图片中是否有“猫”？

- 获得观察数据：标注的图片



判断图片中是否有“猫”？

- 求一个函数，使得

$$f(\text{img1}) = \text{"1"}$$

$$f(\text{img2}) = \text{"0"}$$

$$f(\text{img3}) = \text{"1"}$$

$$f(\text{img4}) = \text{"0"}$$

如果该函数在观察数据上都成立，则可相信它近似反映（逼近）了规律

$$f(\text{img1}) = \text{"1"}$$

$$f(\text{img2}) = \text{"0"}$$

$$f(\text{img3}) = \text{"1"}$$

$$f(\text{img4}) = \text{"0"}$$

将输入表达为同一维度的数据



1000x1000的矩阵

$$\begin{pmatrix} p_{1,1} \\ p_{1,2} \\ \vdots \\ p_{i,j} \\ \vdots \\ p_{n,n} \end{pmatrix}$$

$$\xrightarrow{f} 0/1$$

维数太高!

Find a function:



1000x1000的矩阵

$$\begin{pmatrix} p_{1,1} \\ p_{1,2} \\ \vdots \\ p_{i,j} \\ \vdots \\ p_{n,n} \end{pmatrix}$$



特征向量：
降维、抽象

$$\begin{pmatrix} r \\ g \\ b \\ n \\ A \\ D \end{pmatrix}$$

f

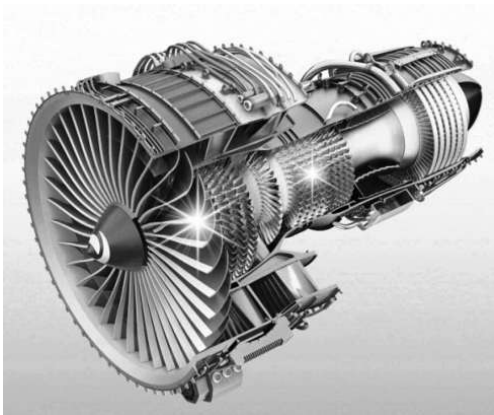
0/1

问题7：求船型曲线

- 已知：某条船的侧面投影曲线图
- 求：该投影轮廓线的表达函数？
- 方法：
 - 从投影曲线上描（采样）一系列点
 - 找一个函数拟合这些采样点



工业产品



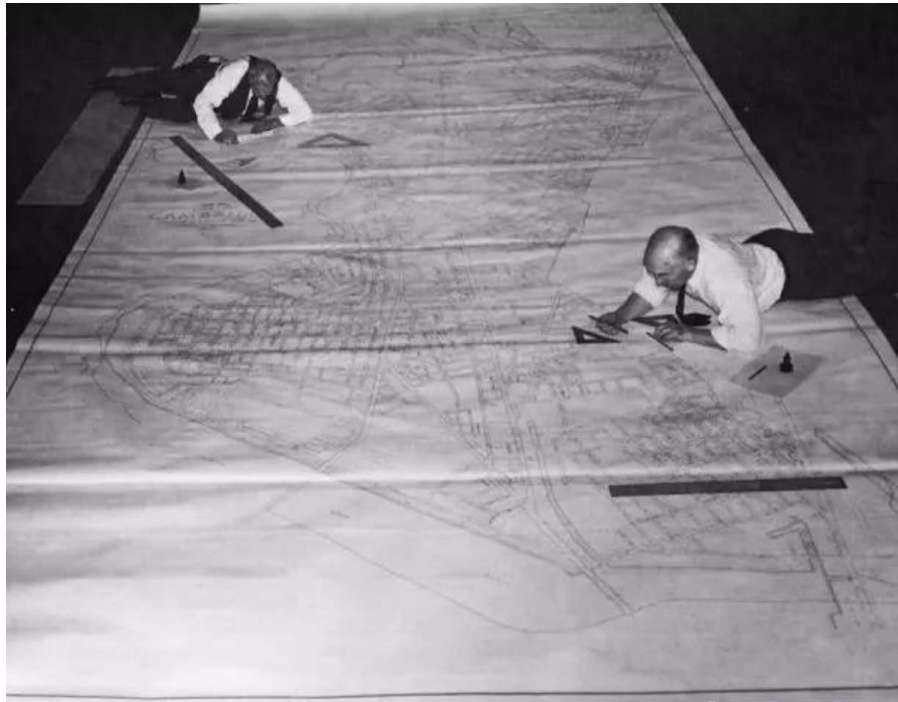
没有电脑之前，
设计师是如何画图设计的？





放样方法

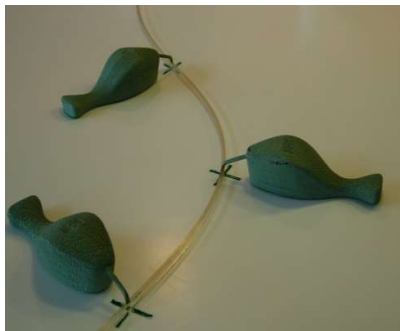
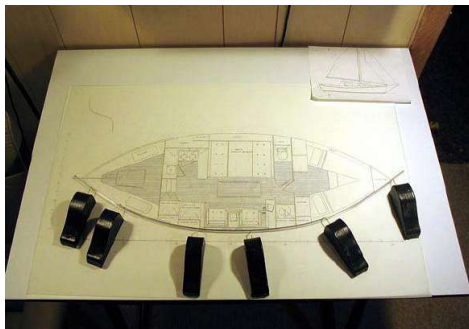






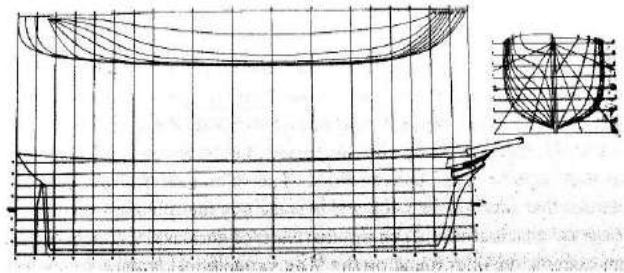
样条与压铁

- 样条：具有一定弹性的软木条
- 压铁：较重的铁块，用来固定样条所经过的点

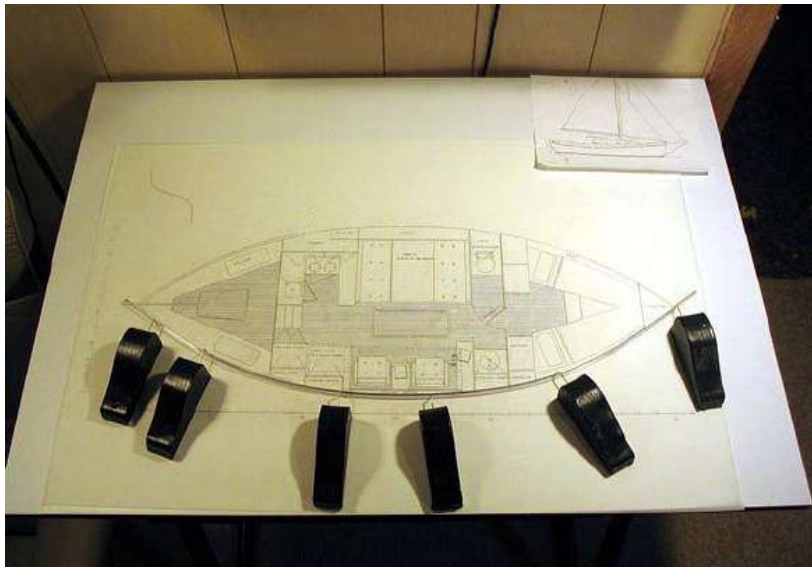


自由曲线

- 在产品初始设计阶段，描述其外形的曲线或曲面常常只有大致形状或只知道它通过一些空间点列，称为**型值点**。



样条所形成的曲线的数学表达？

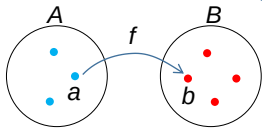


映射与函数

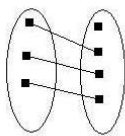
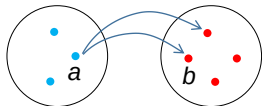
映射 (mapping)

- 两个非空集合 A 和 B 的映射 $f: A \rightarrow B$ ：对 A 中的任何一个元素 a ，有唯一的一个 B 中的元素 b 与之对应，记为 $f(a) = b$
 - b 称为 a 的象， a 称为 b 的原象
 - A 称为定义域， B 称为值域

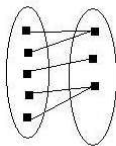
映射：



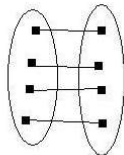
非映射：



单射



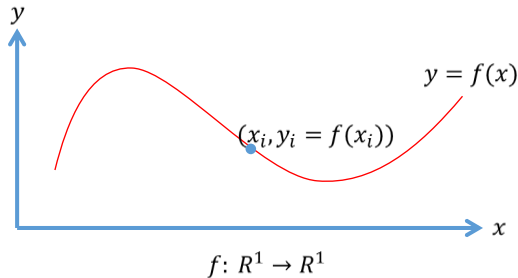
满射



双射

函数 (Function)

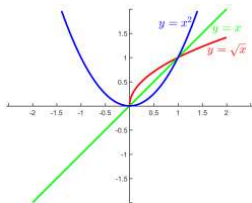
- 非空**实数集**之间的映射称为（一元）函数 $y = f(x)$ ，或变换
- 函数的图像（函数的可视化）：所有有序对 **$(x, f(x))$** 组成的集合



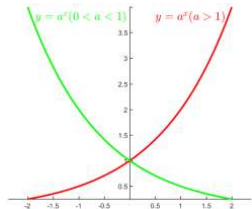
一元函数

• 有哪些函数？

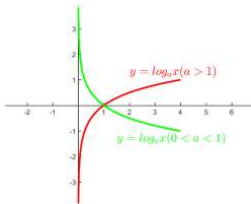
- 幂函数
- 三角函数
- 对数函数
- 指数函数
- 三角函数
- 反三角函数
- ...



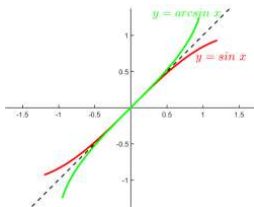
幂函数



指数函数



对数函数



(反)三角函数

函数的集合（函数空间）

- 用若干简单函数（“基函数”）线性组合张成一个函数空间
 - $L = \text{span}\{f_1, f_2, \dots, f_n\} = \{\sum_{i=1}^n a_i f_i(x) \mid a_i \in R\}$
 - 每个函数就表达（对应）为 n 个实数，即系数向量 $(a_1, a_2, \dots, a_n)$

幂基 $\{x^k, k = 0, 1, \dots, n\}$



$$f(x) = \sum_{k=0}^n w_k x^k$$

多项式函数空间

三角函数基



$$f(x) = a_0 + \sum_{k=1}^n (a_k \cos kx + b_k \sin kx)$$

三角函数空间

空间的完备性：这个函数空间是否可以表示（逼近）任意函数？

赋范空间

- 内积诱导范数、距离
 - $\langle f, g \rangle = \int_a^b f(x)g(x)dx$
- 度量空间：可度量函数之间的距离
 - L_p 范数
- 赋范空间+完备性=巴拿赫空间
- 内积空间（无限维）+完备性=希尔伯特空间

万能逼近定理： Weierstrass逼近定理

- 定理1：闭区间上的连续函数可用多项式级数一致逼近
- 定理2：闭区间上周期为 2π 的连续函数可用三角函数级数一致逼近

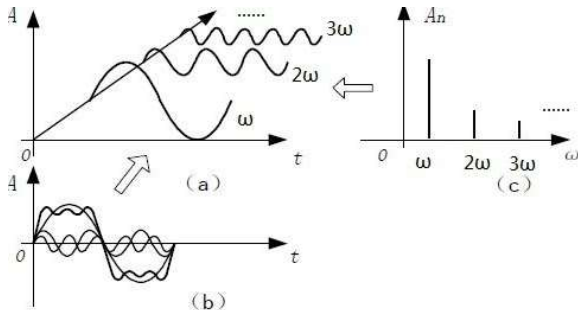
对 $[a, b]$ 上的任意连续函数 g ，及任意给定的 $\varepsilon > 0$ ，必存在 n 次代数多项式 $f(x) = \sum_{k=0}^n w_k x^k$ ，使得

$$\min_{x \in [a, b]} |f(x) - g(x)| < \varepsilon.$$

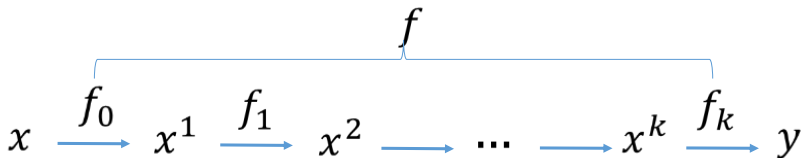
傅里叶级数

$$f(t) = A_0 + \sum_{n=1}^{\infty} [a_n \cos(n\omega t) + b_n \sin(n\omega t)]$$

$$f(t) = A_0 + \sum_{n=1}^{\infty} A_n \sin(n\omega t + \psi_n)$$



更复杂的函数：函数复合



$$f = f_k \circ f_{k-1} \circ \dots \circ f_0$$

$$\frac{1}{1 + \left(\frac{2x}{1+x^2}\right)^2}, \quad f(x) = \frac{e^x}{x^2 + 1}$$

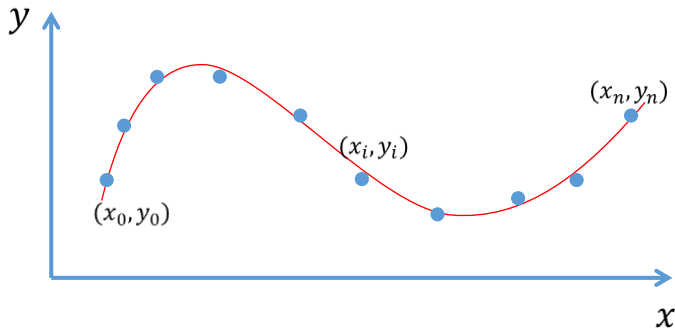
问题：如何求满足要求的函数？

- 大部分的实际应用问题
 - 可建模为：找一个映射/变换/函数
 - 输入不一样、变量不一样、维数不一样
- 如何找函数的三步曲：
 - 到哪找？
 - 确定某个函数集合/空间
 - 找哪个？
 - 度量哪个函数是好的/“最好”的
 - 怎么找？
 - 求解或优化：不同的优化方法与技巧，既要快、又要好...
- 【注】这里先暂时刻定为单变量的函数形式

数据拟合

拟合(Fitting)问题

- 输入：一些观察的数据点
- 输出：反映这些数据规律的函数 $y = f(x)$



$$x \in R^1, y \in R^1$$

1. 到哪找?

- 选择一个函数空间

- 线性函数空间 $A = \text{span}\{B_0(x), \dots, B_n(x)\}$

- 多项式函数 $\text{span}\{1, x, x^2, \dots, x^n\}$

- RBF函数

- 三角函数

- 函数表达为

- $f(x) = \sum_{k=0}^n a_k B_k(x)$

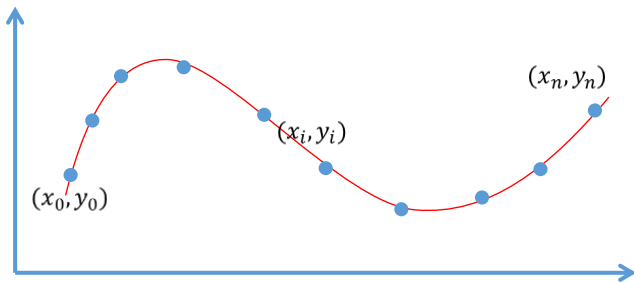
- 求 $n + 1$ 个系数 $(a_0, \dots, a_n)$

待定系数

2. 找哪个?

- 目标1: 函数经过每个数据点 (插值)

$$y_i = f(x_i), i = 0, 1, \dots, n$$



3. 怎么找?

- 目标1: 每个数据点都要**插值** (零误差)
 - $y_i = f(x_i), i = 0, 1, \dots, n$
- 联立, 求解线性方程组:
 - $\sum_{k=0}^n a_k B_k(x_i) = y_i, i = 0, 1, \dots, n$
 - 求解 $(n+1) \times (n+1)$ 线性方程组
 - n 次 Lagrange 插值多项式
 - 病态问题: 系数矩阵条件数高时, 求解不稳定

Lagrange插值函数

- 插值 $n+1$ 个点、次数不超过 n 的多项式是存在而且是唯一的
 - ($n+1$ 个变量, $n+1$ 个方程)

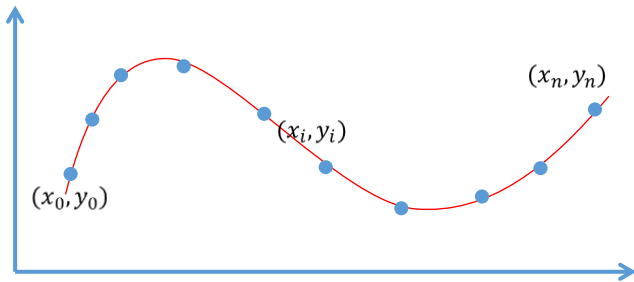
$$p_k(x) = \prod_{i \in B_k} \frac{x - x_i}{x_k - x_i}$$

插值函数的自由度 = 未知量个数 - 已知量个数

2-2. 找哪个?

- 目标2：函数尽量靠近数据点（逼近）

$$- \min \sum_{i=0}^n (y_i - f(x_i))^2$$



3-2. 怎么找?

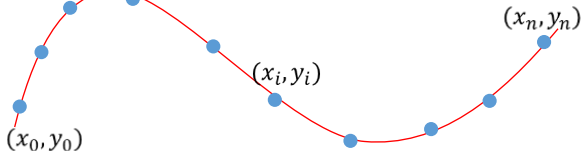
- 目标2: 函数尽量靠近数据点 (逼近)
 - $\min \sum_{i=0}^n (y_i - f(x_i))^2$
- 对各系数求导, 得法方程 (线性方程组)
 - $AX = b$

最小二乘法

- 问题:
 - 点多, 系数少?
 - 点少, 系数多?

Recap: 拟合--插值或逼近

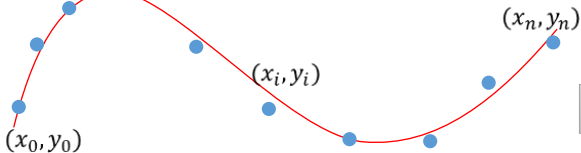
零误差



插值

$$y_i = f(x_i)$$

最小平方误差
(最小二乘拟合)

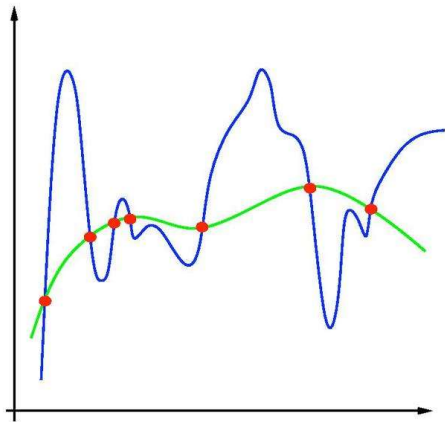


逼近

$$\min \sum_{i=0}^n (y_i - f(x_i))^2$$

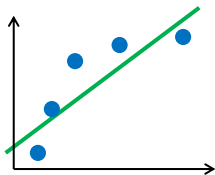
Overfitting (过拟合)

- 误差为0，但是拟合的函数并无使用价值！



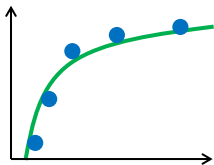
欠拟合或过拟合

- 如何选择合适的基函数？



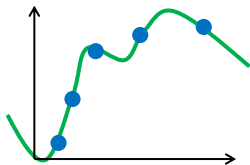
$$w_0 + w_1 x$$

High bias
(underfitting)



$$w_0 + w_1 x + w_2 x^2$$

“Just right”



$$w_0 + w_1 x + w_2 x^2 + w_3 x^3 +$$

High variance
(overfitting)

- 需要根据不同的应用与需求，不断尝试
(不断 “调参”)

避免过拟合的常用方法

- 数据去噪
 - 剔除训练样本中噪声
- 数据增广
 - 增加样本数，或者增加样本的代表性和多样性
- 模型简化
 - 预测模型过于复杂，拟合了训练样本中的噪声
 - 选用更简单的模型，或者对模型进行裁剪
- 正则约束
 - 适当的正则项，比如方差正则项、稀疏正则项

$$y = \sum_{i=0}^n w_i B_i(x)$$

岭回归正则项

- 选择一个函数空间
 - 基函数的线性表达

$$W = (w_0, w_1, \dots, w_n)$$

$$y = f(x) = \sum_{i=0}^n w_i B_i(x)$$

- 最小二乘拟合

$$\min_W \|Y - XW\|^2$$

- Ridge regression (岭回归)

$$\min_W \|Y - XW\|^2 + \mu \|W\|_2^2$$

稀疏学习：稀疏正则化

- 冗余基函数（过完备）
- 通过优化来选择合适的基函数
 - 系数向量的 L_0 模（非0元素个数）尽量小
 - 挑选（“学习”）出合适的基函数

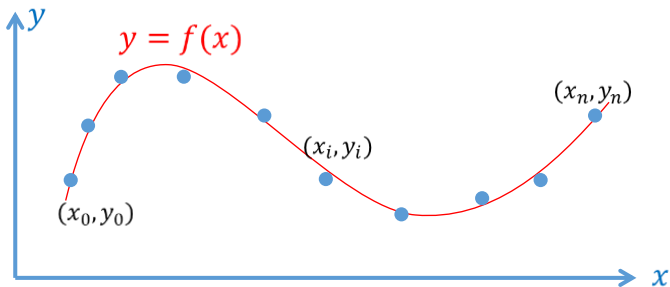
$$\min_{\alpha} \|Y - XW\|^2 + \mu \|W\|_0$$

$$\min_{\alpha} \|Y - XW\|^2, \quad \text{s.t. } \|W\|_0 \leq \beta$$

数据拟合模型求解

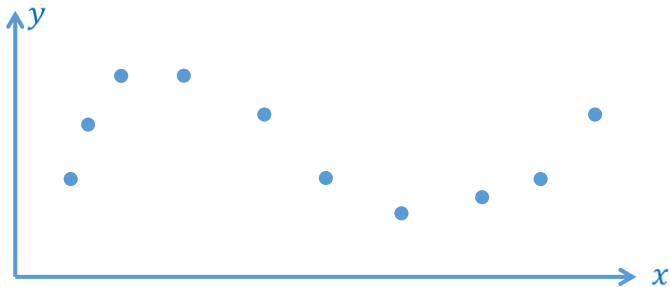
函数拟合问题

- 输入：一些观察（采样）的数据点 $\{x_i, y_i\}_{i=0}^n$
- 输出：拟合数据点的函数 $y = f(x)$ ，并用于预测



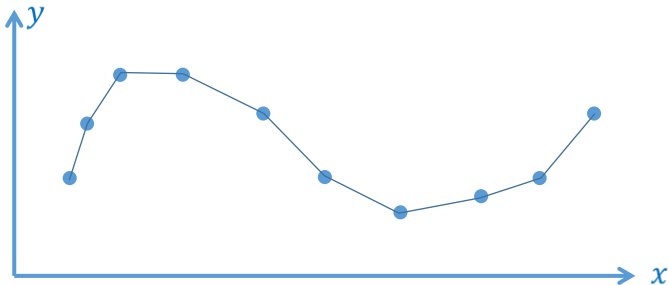
这种拟合函数有多少个？

拟合函数的“好坏”



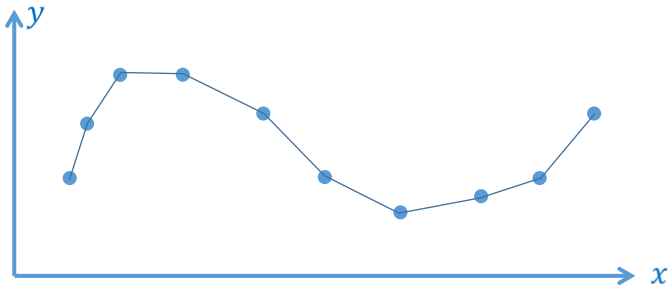
拟合函数的“好坏”

- 分段线性插值函数 $y = f_1(x)$
 - 数据误差为0



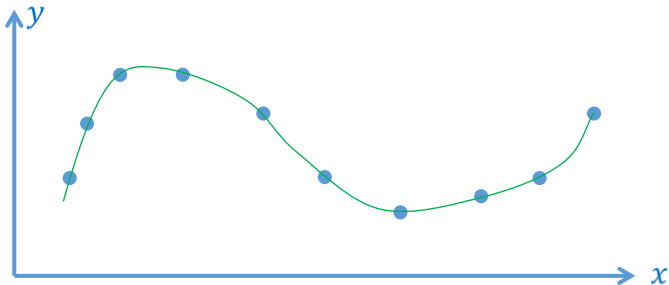
拟合函数的“好坏”

- 分段线性插值函数 $y = f_1(x)$
 - 数据误差为0
 - 函数性质不够好：只有 C^0 连续，不光滑（数值计算）



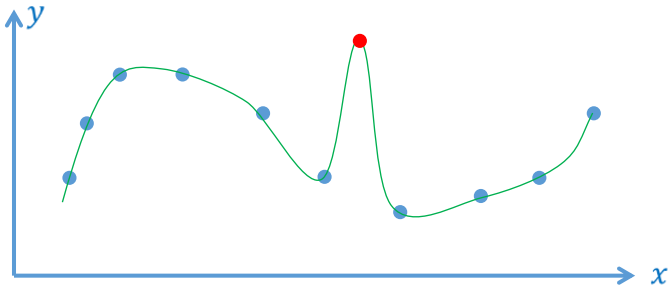
拟合函数的“好坏”

- 光滑插值函数 $y = f_2(x)$
 - 数据误差为0



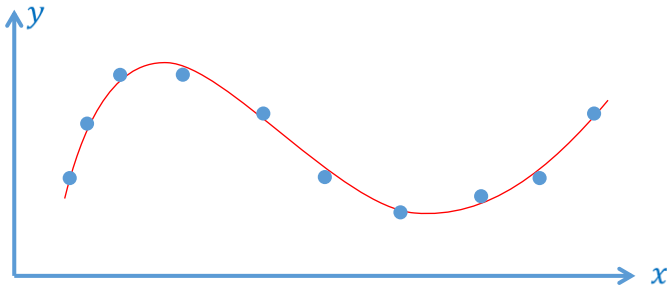
拟合函数的“好坏”

- 光滑插值函数 $y = f_2(x)$
 - 数据误差为0
 - 可能被“差数据”（噪声、outliers）带歪，导致函数性质不好、预测不可靠



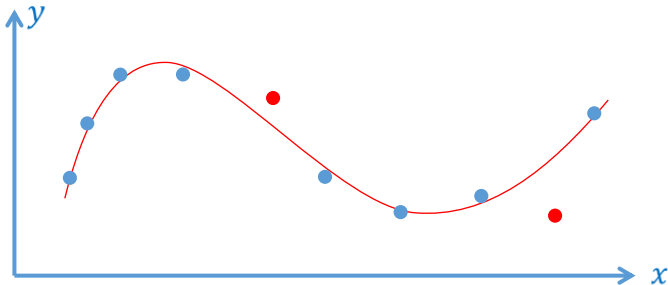
拟合函数的“好坏”

- 逼近拟合函数 $y = f_3(x)$
 - 数据误差不为0，但足够小



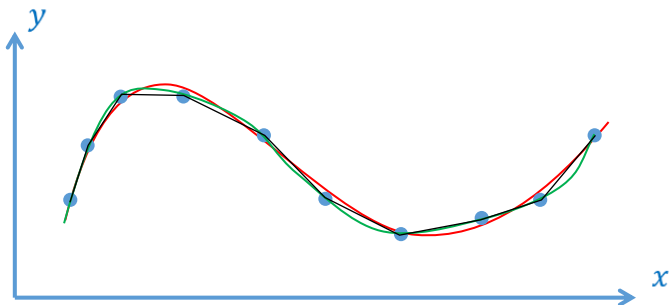
拟合函数的“好坏”

- 逼近拟合函数 $y = f_3(x)$
 - 数据误差不为0，但足够小



拟合函数的“好坏”

- 分段线性函数 $f = f_1(x)$
- 光滑插值函数 $f = f_2(x)$
- 逼近拟合函数 $f = f_3(x)$



求拟合函数：应用驱动

- 大部分的实际应用问题
 - 可建模为：找一个映射/变换/函数
 - 输入不一样、变量不一样、维数不一样
- 三步曲方法论：
 - 到哪找？
 - 确定某个函数集合/空间
 - 找哪个？
 - 度量哪个函数是好的/“最好”的
 - 怎么找？
 - 求解或优化：不同的优化方法与技巧，既要快、又要好...

1. 到哪找?

- 选择一个函数空间

- 线性函数空间 $A = \text{span}\{B_0(x), \dots, B_n(x)\}$

- 多项式函数 $\text{span}\{1, x, x^2, \dots, x^n\}$

- RBF函数

- 三角函数

- 函数表达为

- $f(x) = \sum_{k=0}^n a_k B_k(x)$

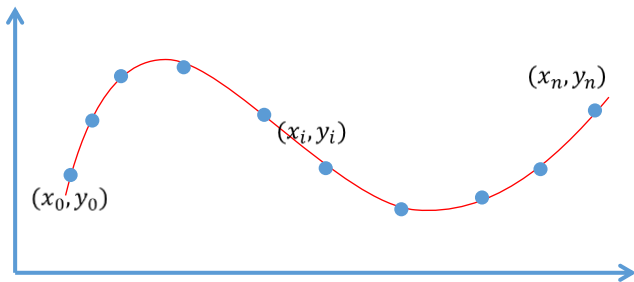
- 求 $n + 1$ 个系数 $(a_0, \dots, a_n)$

待定系数

2. 找哪个?

- 目标1: 函数经过每个数据点 (插值)

$$y_i = f(x_i), i = 0, 1, \dots, n$$



3. 怎么找?

- 目标1: 每个数据点都要**插值** (零误差)
 - $y_i = f(x_i), i = 0, 1, \dots, n$
- 联立, 求解线性方程组:
 - $\sum_{k=0}^n a_k B_k(x_i) = y_i, i = 0, 1, \dots, n$
 - 求解 $(n+1) \times (n+1)$ 线性方程组
 - n 次 Lagrange 插值多项式
 - 病态问题: 系数矩阵条件数高时, 求解不稳定

Lagrange插值函数

- 插值 $n+1$ 个点、次数不超过 n 的多项式是存在而且是唯一的
 - ($n+1$ 个变量, $n+1$ 个方程)

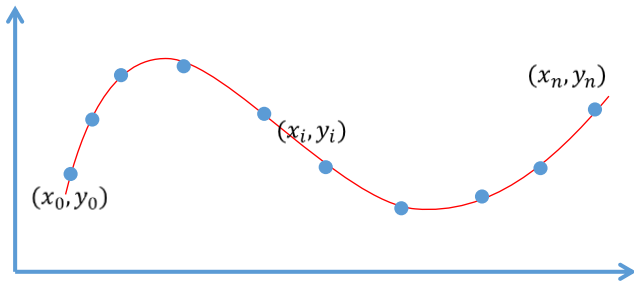
$$p_k(x) = \prod_{i \in B_k} \frac{x - x_i}{x_k - x_i}$$

插值函数的自由度 = 未知量个数 - 已知量个数

2-2. 找哪个?

- 目标2：函数尽量靠近数据点（逼近）

$$- \min \sum_{i=0}^n (y_i - f(x_i))^2$$



3-2. 怎么找?

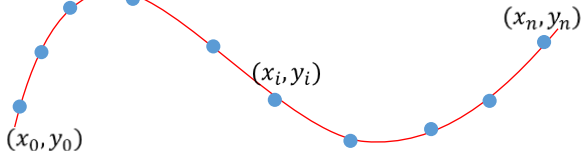
- 目标2: 函数尽量靠近数据点 (逼近)
 - $\min \sum_{i=0}^n (y_i - f(x_i))^2$
- 对各系数求导, 得法方程 (线性方程组)
 - $AX = b$

最小二乘法

- 问题:
 - 点多, 系数少?
 - 点少, 系数多?

Recap: 拟合--插值或逼近

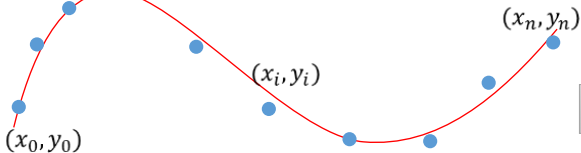
零误差



插值

$$y_i = f(x_i)$$

最小平方误差
(最小二乘拟合)

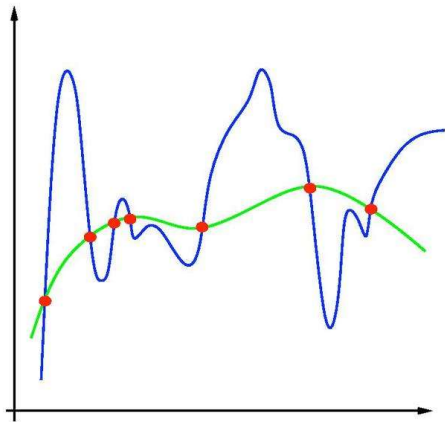


逼近

$$\min \sum_{i=0}^n (y_i - f(x_i))^2$$

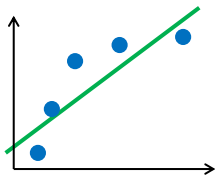
Overfitting (过拟合)

- 误差为0，但是拟合的函数并无使用价值！



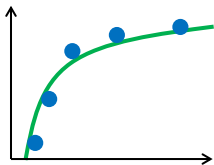
欠拟合或过拟合

- 如何选择合适的基函数？



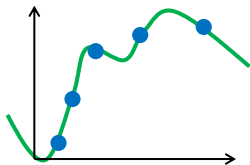
$$w_0 + w_1 x$$

High bias
(underfitting)



$$w_0 + w_1 x + w_2 x^2$$

“Just right”



$$w_0 + w_1 x + w_2 x^2 + w_3 x^3 +$$

High variance
(overfitting)

- 需要根据不同的应用与需求，不断尝试
(不断 “调参”)

Recap: 数据拟合的方法论

- 到哪找?
 - 确定函数的表达形式（函数集、空间） $L = \text{span}\{b_0(x), \dots, b_n(x)\}$
 - 待定基函数的组合系数（求解变量） $f_\lambda(x) = \sum_{k=0}^n \lambda_k b_k(x)$
- 找哪个?
 - 优化模型（最小化问题）
 - 能量项 = 误差项 + 正则项
 - 统计模型、规划模型...
- 怎么找?
 - 求解误差函数的驻点（导数为0之处）
 - 转化为系数的方程组
 - 如果是欠定的（有无穷多解），则修正模型
 - 改进/增加各种正则项：Lasso、岭回归、稀疏正则项...
 - 返回修改模型

f 由待定系数 $\lambda = \begin{pmatrix} \lambda_1 \\ \dots \\ \lambda_n \end{pmatrix}$ 确定

1. 多项式插值

多项式插值定理

定理：若 x_i 两两不同，则对任意给定的 y_i ，存在唯一的次数至多是 n 次的多项式 p_n ，使得 $p_n(x_i) = y_i$ $i = 0, \dots, n$ 。

证明：在幂基 $\{1, x, \dots, x^n\}$ 下待定多项式 p 的形式为：

$$p(x) = a_0 + a_1x + a_2x^2 + \dots + a_nx^n$$

由插值条件 $p(x_i) = y_i, i = 0, \dots, n$ ，得到如下方程组：

$$\begin{pmatrix} 1 & x_0 & x_0^2 & \dots & x_0^n \\ 1 & x_1 & x_1^2 & \dots & x_1^n \\ 1 & x_2 & x_2^2 & \dots & x_2^n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \dots & x_n^n \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}$$

如果基函数选取不一样，方程组的系数矩阵不同

系数矩阵为 Vandermonde 矩阵，其行列式非零，因此方程组有唯一解。

技巧1：构造插值问题的通用解

- 构造插值问题的通用解

- 给定 $n + 1$ 个点 $\{(x_0, y_0), \dots, (x_n, y_n)\}$, 寻找一组次数为 n 的多项式基函数 l_i 使得

$$l_i(x_j) = \begin{cases} 1, & \text{若 } i = j \\ 0, & \text{若 } i \neq j \end{cases}$$

- 插值问题的解为:

$$P(x) = y_0 l_0(x) + y_1 l_1(x) + \dots + y_n l_n(x) = \sum_{i=0}^n y_i l_i(x)$$

一般形式

- 怎么计算多项式 $l_i(x)$?

- n 阶多项式, 且有以下 n 个根

$$x_0, x_1, x_2, \dots, x_{i-1}, x_{i+1}, \dots, x_n$$

- 故可表示为

$$l_i(x)$$

$$= C_i(x - x_0)(x - x_1) \dots (x - x_{i-1})(x - x_{i+1}) \dots (x - x_n)$$

$$= C_i \prod_{j \neq i} (x - x_j)$$

- 由 $l_i(x_i) = 1$ 可得

$$1 = C_i \prod_{j \neq i} (x_i - x_j) \Rightarrow C_i = \frac{1}{\prod_{j \neq i} (x_i - x_j)}$$

技巧1: 构造插值问题的通用解

- 最终多项式基函数为

$$l_i(x) = \frac{\prod_{j \neq i} (x - x_j)}{\prod_{j \neq i} (x_i - x_j)}$$

- 多项式 $l_i(x)$ 被称为拉格朗日多项式

技巧2：更方便的求解表达

- Newton插值：具有相同“导数”（差商）的多项式构造（ n 阶Taylor展开）

定义：

一阶差商：

$$f[x_0, x_1] = \frac{f(x_1) - f(x_0)}{x_1 - x_0}$$

k 阶差商：

设 $\{x_0, x_1, \dots, x_k\}$ 互不相同， $f(x)$ 关于 $\{x_0, x_1, \dots, x_k\}$ 的 k 阶差商为：

$$f[x_0, x_1, \dots, x_k] = \frac{f[x_1, \dots, x_k] - f[x_0, x_1, \dots, x_{k-1}]}{x_k - x_0}$$

所以 Newton 插值多项式表示为：

$$N_n(x) = f(x_0) + f[x_0, x_1](x - x_0) + \dots + f[x_0, x_1, \dots, x_n](x - x_0) \cdots (x - x_{n-1})$$

多项式插值存在的问题

- 系统矩阵稠密
- 依赖于基函数选取，矩阵可能病态，导致难于求解（求逆）

病态矩阵示例

- 考虑二元方程组
 - 解为(1,1)

$$\begin{aligned}x_1 + 0.5x_2 &= 1.5 \\ 0.667x_1 + 0.333x_2 &= 1\end{aligned}$$

- 对第二个方程右边项扰动0.001
 - 解为(0,3)

$$\begin{aligned}x_1 + 0.5x_2 &= 1.5 \\ 0.667x_1 + 0.333x_2 &= 0.999\end{aligned}$$


- 对矩阵系数进行扰动
 - 解为(2,-1)

$$\begin{aligned}x_1 + 0.5x_2 &= 1.5 \\ 0.667x_1 + 0.334x_2 &= 1\end{aligned}$$


病态问题

- 输入数据的细微变化导致输出(解)的剧烈变化
- 将线性方程看成直线（超平面）
 - 当系统病态时，直线变为近似平行
 - 求解（即直线求交）变得困难、不精确

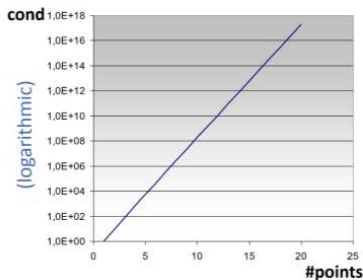
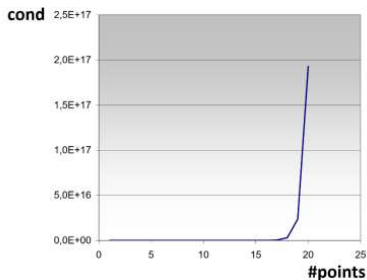
矩阵条件数

$$\kappa_2(A) = \frac{\max_{x \neq 0} \frac{\|Ax\|}{\|x\|}}{\min_{x \neq 0} \frac{\|Ax\|}{\|x\|}}$$

- 等于最大特征值和最小特征值之间比例
- 条件数大意味着基元之间有太多相关性

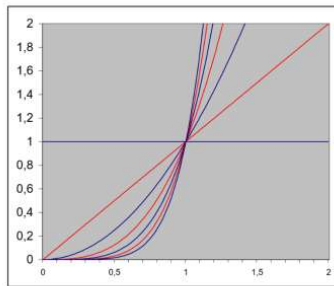
矩阵条件数

- 多项式插值问题是病态的
 - 对于等距分布的数据点 x_i ，范德蒙矩阵的条件数随着数据点数 n 呈指数级增长（多项式的最高次数为 $n - 1$ ）



为什么？

- 幂（单项式）函数基
 - 幂函数之间差别随着次数增加而减小
 - 不同幂函数之间唯一差别为增长速度(x^i 比 x^{i-1} 增长快)



幂(单项式) 函数

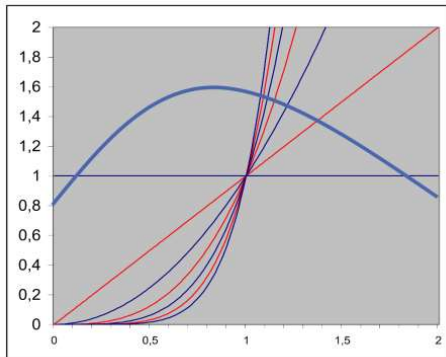
函数互相抵消

- 单项式：

- 从左往右
- 首先常函数1主宰
- 接着 x 增长最快
- 接着 x^2 增长最快
- 接着 x^3 增长最快
- ...

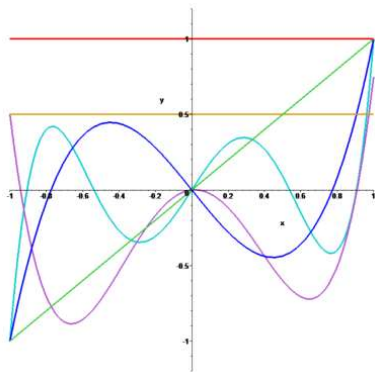
- 趋势

- 好的基函数一般需要系数交替
- 互相抵消问题

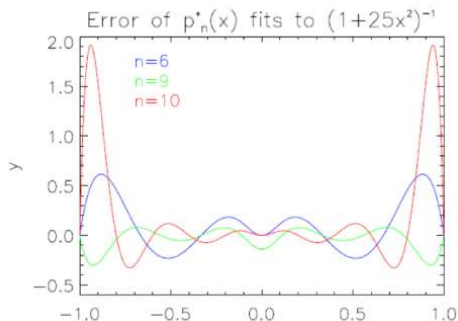
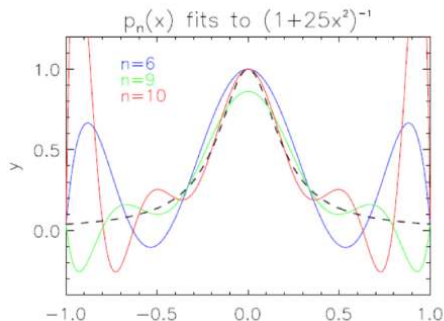


解决方法

- 使用正交多项式基
- 如何获得？
 - Gram-Schmidt正交化



多项式插值结果好吗？



振荡(龙格Runge)现象和对插值点数的高度敏感性
观察 $n = 9$ (10个数据点) 和 $n = 10$ (11个数据点) 的差别

结论

- 多项式插值不稳定
 - 控制点的微小变化可导致完全不同的结果
 - 振荡(Runge)现象
 - 多项式随着插值点数(可以是细微)增加而摆动
- ➡ 需要更好的基函数来做插值
- Bernstein基函数?
 - 分片多项式?

2. 多项式逼近

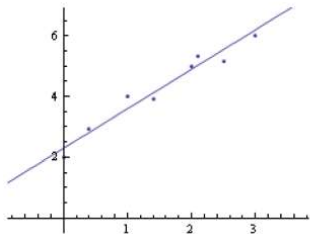
为什么逼近?

- 数据点含噪声、outliers等
- 更紧凑的表达
- 计算简单、更稳定

最小二乘逼近

- 逼近问题

- 给定一组线性无关的连续函数集合 $B = \{b_1, \dots, b_n\}$ 和一组结点 $\{(x_1, y_1), \dots, (x_m, y_m)\}$, 其中 $m > n$
- 在 B 张成空间中哪个函数 $f \in \text{span}(B)$ 对结点逼近最好?
- 示例: 给定一组点, 找到最佳逼近的线性函数
- 怎么定义“最佳逼近”?

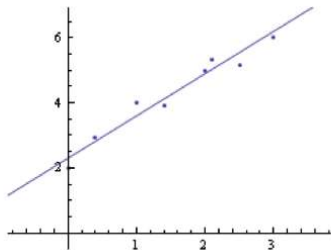


最佳逼近的定义

- 最小二乘逼近

$$\operatorname{argmin}_{f \in \operatorname{span}(B)} \sum_{j=1}^m (f(x_j) - y_j)^2$$

$$\begin{aligned} \sum_{j=1}^m (f(x_j) - y_j)^2 &= \sum_{j=1}^m \left(\sum_{i=1}^n \lambda_i b_i(x_j) - y_j \right)^2 \\ &= (M\lambda - y)^T (M\lambda - y) \\ &= \lambda^T M^T M \lambda - y^T M \lambda - \lambda^T M^T y + y^T y \\ &= \lambda^T M^T M \lambda - 2y^T M \lambda + y^T y \end{aligned}$$



$$M = \begin{pmatrix} b_1(x_1) & \dots & b_n(x_1) \\ \vdots & \ddots & \vdots \\ b_1(x_m) & \dots & b_n(x_m) \end{pmatrix}$$

求解

- 关于 λ 的二次多项式

$$\lambda^T M^T M \lambda - 2y^T M \lambda + y^T y$$

- 法方程

- 最小解满足

$$M^T M \lambda = M^T y$$

- 提示

- 最小化二次目标函数 $x^T A x + b^T x + c$
 - 充分必要条件: $2Ax = -b$

3. 函数空间及基函数

为什么用多项式？

- 易于计算，表现良好，光滑， ...
- 稠密性与完备性：**表达能力足够！**
 - 魏尔斯特拉斯Weierstrass定理：令 f 为闭区间 $[a, b]$ 上任意连续函数，则对任意给定 ε ，存在 n 和**多项式** P_n 使得
$$|f(x) - P_n(x)| < \varepsilon, \forall x \in [a, b]$$
 - Weierstrass只证明了存在性，未给出多项式

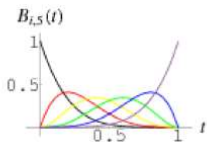
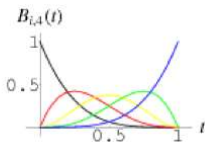
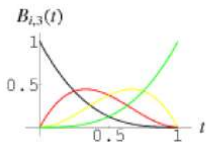
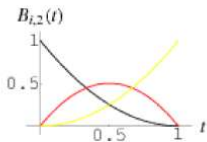
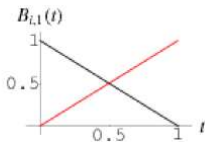
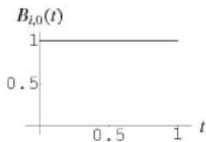
用Bernstein多项式做逼近

- 伯恩斯坦Bernstein给出了构造性证明（强大！）
 - 对 $[0, 1]$ 区间上任意连续函数 $f(x)$ 和任意正整数 n ，以下不等式对所有 $x \in [0, 1]$ 成立

$$|f(x) - B_n(f, x)| < \frac{9}{4} m_{f,n}$$

- $m_{f,n} = \max_{y_1, y_2 \in [0, 1] \text{ 且 } |y_1 - y_2| < \frac{1}{\sqrt{n}}} |f(y_1) - f(y_2)|$
- $B_n(f, x) = \sum_{j=0}^n f(x_j) b_{n,j}(x)$ ，其中 x_j 为 $[0, 1]$ 上等距采样点
- $b_{n,j} = \binom{n}{j} x^j (1-x)^{n-j}$ 为Bernstein多项式

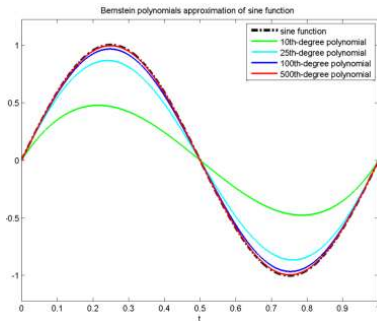
Bernstein多项式



- $b_{0,0}(x) = 1$
- $b_{0,1}(x) = 1 - x$, $b_{1,1} = x$
- $b_{0,2}(x) = (1 - x)^2$, $b_{1,2} = 2x(1 - x)$, $b_{2,2} = x^2$
- $b_{0,3}(x) = (1 - x)^3$, $b_{1,3} = 3x(1 - x)^2$, $b_{2,3} = 3x^2(1 - x)$, $b_{3,3} = x^3$
- $b_{0,4}(x) = (1 - x)^4$, $b_{1,4} = 4x(1 - x)^3$, $b_{2,4} = 6x^2(1 - x)^2$, $b_{3,4} = 4x^3(1 - x)$, $b_{4,4} = x^4$

用Bernstein多项式做逼近

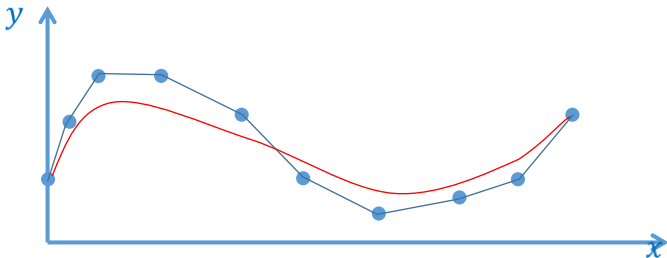
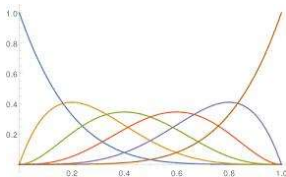
- Bernstein基函数的良好性质：非常好的几何意义！
 - 正性、权性（和为1） $\rightarrow$ 凸包性
 - 变差缩减性
 - 递归线性求解方法
 - 细分性
 - ...
- Bernstein多项式逼近示例
 - 逼近结果优秀
 - 需要高阶



丰富的理论：CAGD课程

关于Bernstein函数...

- $B_n(f, x) = \sum_{j=0}^n f\left(\frac{j}{n}\right) b_{n,j}(x)$
- 两种观点：
 - 几何观点、代数观点



丰富的理论：CAGD课程

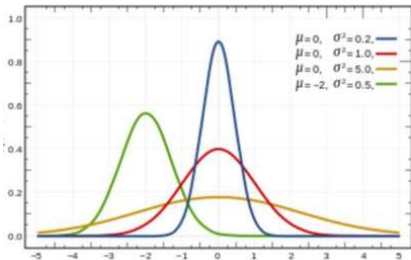
4. RBF函数插值/逼近

Gauss函数

- 两个参数：均值 μ ，方差 σ

$$g_{\mu,\sigma}(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

- 几何意义：
 - 均值 μ ：位置
 - 方差 σ ：支集宽度



- 不同均值和方差的Gauss函数都线性无关
 - 有什么启发？

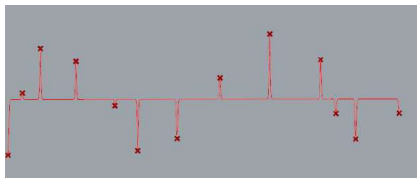
RBF函数拟合

- RBF函数

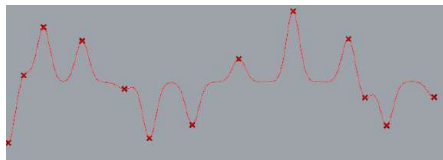
$$f(x) = b_0 + \sum_{i=1}^n b_i g_i(x)$$

- 方法
- 原因

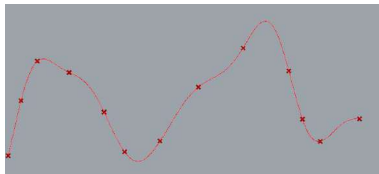
讨论：现象



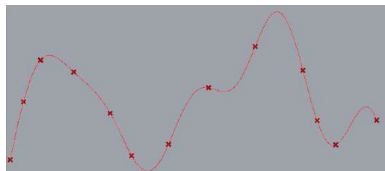
$\sigma = 0.1$



$\sigma = 1$

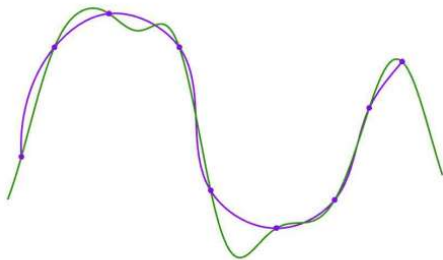
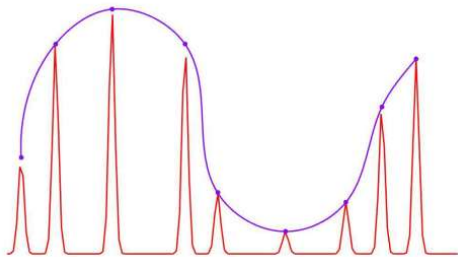


$\sigma = 5$



$\sigma = 10$

讨论：现象



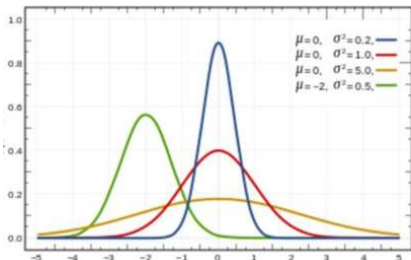
思考：

- 均值 μ 和方差 σ 是否可以一起来优化？

$$g_{\mu,\sigma}(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

$$f(x) = b_0 + \sum_{i=1}^n b_i g_i(x)$$

- 问题？



5. 从另一个角度来看拟合函数

Gauss拟合函数

- 一般Gauss函数表达为标准Gauss函数的形式

$$g_{\mu,\sigma}(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} = g_{0,1}(ax+b)$$

$$a = \frac{x}{\sigma}, b = -\frac{\mu}{\sigma}$$

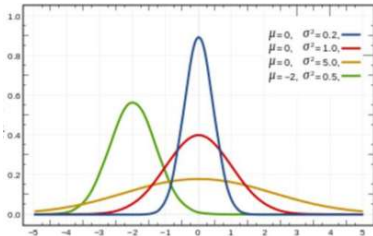
$$f(x) = b_0 + \sum_{i=1}^n b_i g_i(x)$$



$$f(x) = \omega_0 + \sum_{i=1}^n \omega_i \underbrace{g_{0,1}(a_i x + b_i)}$$

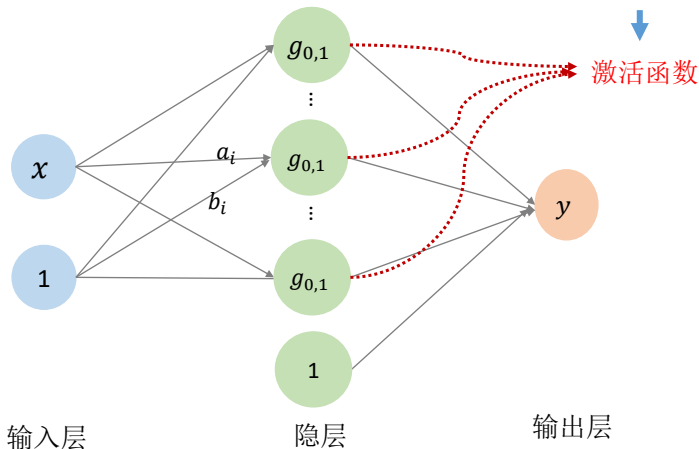


基函数是由一个基本函数通过平移和伸缩变换而来的

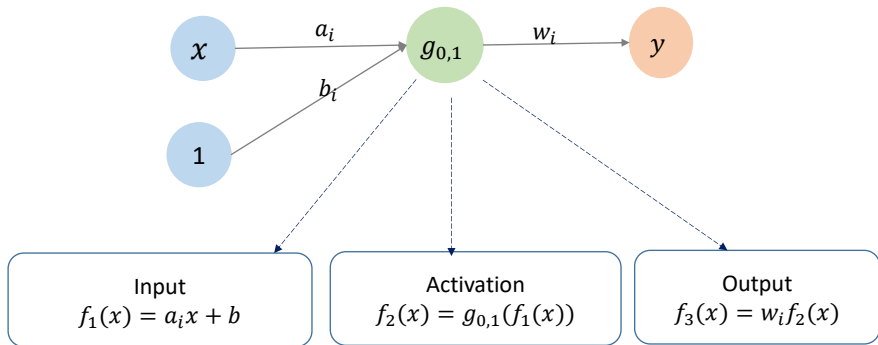


换个方式看函数：神经网络

- 将Gauss函数看成网络 $f(x) = \omega_0 + \sum_{i=1}^n \omega_i g_{0,1}(a_i x + b_i)$



抽象：神经元



RBF 神经网络

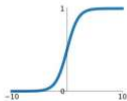
- 高维情形: RBF (Radial Basis Function), 径向基函数
- 一种特殊的BP网络
 - 优化: BP算法
- 核函数思想
- Gauss函数的特性: 拟局部性

思考：激活函数的选择？

- 启发：由一个简单的函数通过（仿射）变换构造出一组基函数，张成一个函数空间
- 表达能力是否足够强：是否完备/稠密的？

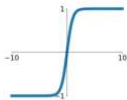
Sigmoid

$$\sigma(x) = \frac{1}{1+e^{-x}}$$



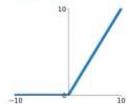
tanh

$$\tanh(x)$$



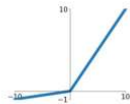
ReLU

$$\max(0, x)$$



Leaky ReLU

$$\max(0.1x, x)$$



Maxout

$$\max(w_1^T x + b_1, w_2^T x + b_2)$$

ELU

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



高维情形：多元函数

(后面的课程再展开解释)

- 变量的多个分量的线性组合

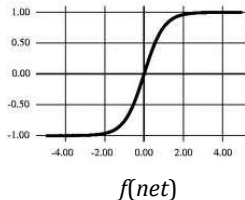
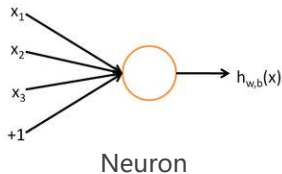
$$(x_1, x_2, \dots, x_n) \longrightarrow g_{0,1}(a_1^i x_1 + a_2^i x_2 + \dots + a_n^i x_n + b_i)$$

- 单隐层神经网络函数：

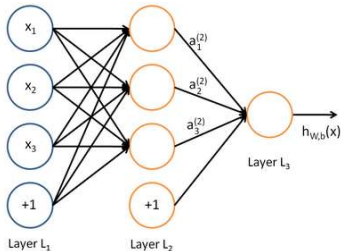
$$f(x_1, x_2, \dots, x_n) = \omega_0 + \sum_{i=1}^n \omega_i g_{0,1}(a_1^i x_1 + a_2^i x_2 + \dots + a_n^i x_n + b_i)$$

多层神经网络：多重复合的函数

- 线性函数和非线性函数的多重复合



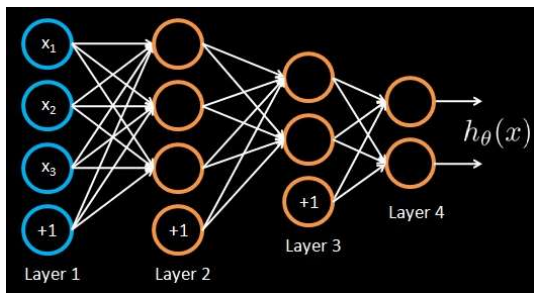
$$h_{W,b}(x) = f(W^T x) = f(\sum_{i=1}^3 W_i x_i + b)$$



Neural network

$$\begin{aligned} a_1^{(2)} &= f(W_{11}^{(1)} x_1 + W_{12}^{(1)} x_2 + W_{13}^{(1)} x_3 + b_1^{(1)}) \\ a_2^{(2)} &= f(W_{21}^{(1)} x_1 + W_{22}^{(1)} x_2 + W_{23}^{(1)} x_3 + b_2^{(1)}) \\ a_3^{(2)} &= f(W_{31}^{(1)} x_1 + W_{32}^{(1)} x_2 + W_{33}^{(1)} x_3 + b_3^{(1)}) \\ h_{W,b}(x) &= a_1^{(3)} = f(W_{11}^{(2)} a_1^{(2)} + W_{12}^{(2)} a_2^{(2)} + W_{13}^{(2)} a_3^{(2)} + b_1^{(2)}) \end{aligned}$$

用神经网络函数来拟合数据



Regression problem:

Input: Given training set $(x_1, y_1), (x_2, y_2), (x_3, y_3), \dots$

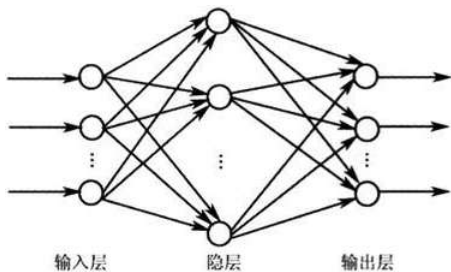
Output: Adjust **parameters θ** (for every node) to make:

$$h(x_i) \approx y_i$$

Why it works?

- 万能逼近定理：自由度足够多！

$$F(\mathbf{x}) = \sum_{i=1}^N v_i \varphi(\mathbf{w}_i^T \mathbf{x} + b_i)$$



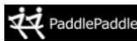
与传统拟合一样存在
同样的问题：
函数个数如何选择？！

调参！

Deep Learning Frameworks



DL4J Deep Learning for Java



theano



dmlc
mxnet

Caffe

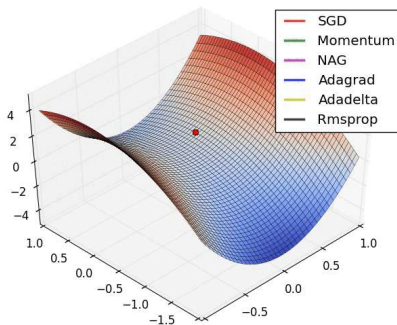


NVIDIA DIGITS

使用深度学习的方法

- 问题建模
 - 理解问题、问题分解（多个映射级联） ...
- 找哪个？
 - 损失函数、各种Penalty、正则项...
- 到哪找？
 - 神经网络函数、网络简化...
- 怎么找？
 - 优化方法（BP方法）
 - 初始值、参数...

调参：有耐心、有直觉...



作业3：数据拟合模型

- 任务：以下2选1
 - 1. 平面点列的插值曲线
 - 2. 图像变形
- 目标
 - 学习和实现数据拟合的数学模型并进行可视化
- 任务
 - 可尝试使用不同函数实现拟合模型，并进行比较
 - 常用函数类型：多项式函数、RBF函数、神经网络函数...
 - 使用棋盘格数据可视化2D拟合函数的性能



中国科学技术大学
University of Science and Technology of China

谢谢！



中国科学技术大学
University of Science and Technology of China

数学建模 Mathematical Modeling

刘利刚

中国科学技术大学

4. 神经网络模型

思考： 图像内容识别

Panorama (全景图: 图像拼图)



图像像素对应



图像像素对应



判断图片中是否有“猫”？

- 如何建立数学模型来设计算法？



判断图片中是否有“猫”？

- 获得观察数据：标注的图片



判断图片中是否有“猫”？

- 求一个函数，使得

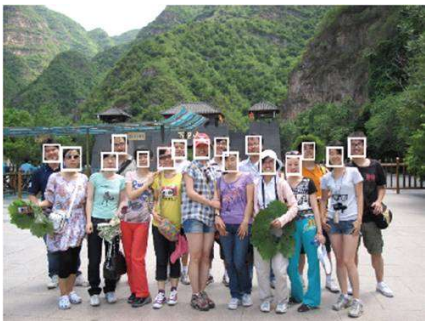
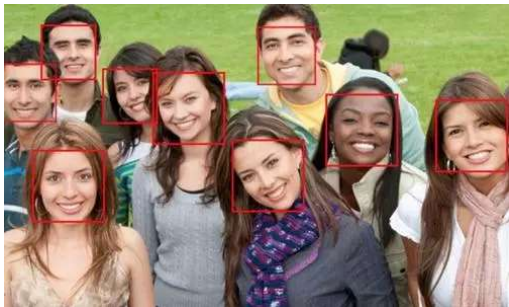
$$f(\text{) = "1"$$

$$f(\text{) = "0"$$

$$f(\text{) = "1"$$

$$f(\text{) = "0"$$

人脸检测 (多人)



人脸识别



如果该函数在观察数据上都成立，则可相信它近似反映（逼近）了规律

$$f(\text{) = \text{"1"}$$

$$f(\text{) = \text{"0"}$$

$$f(\text{) = \text{"1"}$$

$$f(\text{) = \text{"0"}$$

将输入表达为同一维度的数据



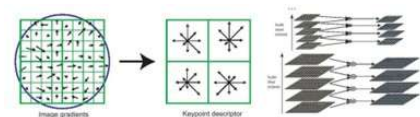
1000x1000的矩阵

$$\begin{pmatrix} p_{1,1} \\ p_{1,2} \\ \vdots \\ p_{i,j} \\ \vdots \\ p_{n,n} \end{pmatrix}$$

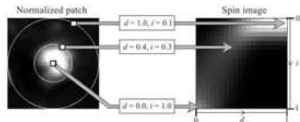
$$\xrightarrow{f} 0/1$$

维数太高!

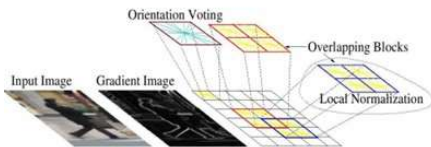
众多的图像人工特征(Descriptors) (Hand-crafted features)



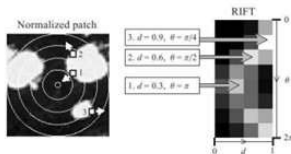
SIFT



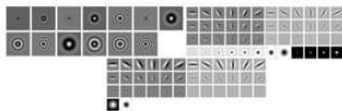
Spin image



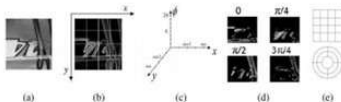
HoG



RIFT



Textons



GLOH

Find a function:



1000x1000的矩阵

$$\begin{pmatrix} p_{1,1} \\ p_{1,2} \\ \vdots \\ p_{i,j} \\ \vdots \\ p_{n,n} \end{pmatrix}$$



特征向量：
降维、抽象

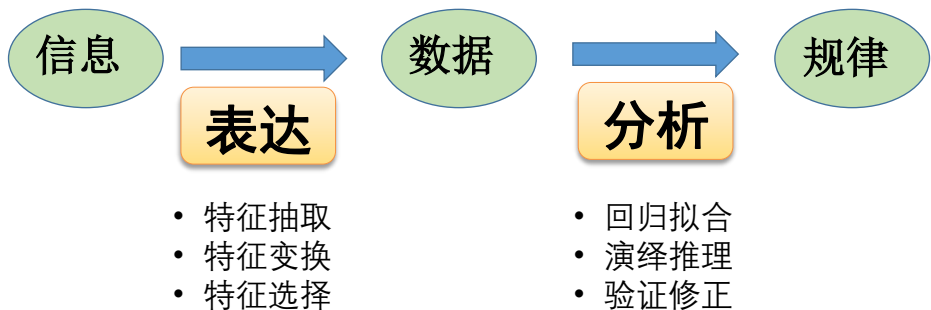
$$\begin{pmatrix} r \\ g \\ b \\ n \\ A \\ D \end{pmatrix}$$

f

0/1

数据空间的变换

科学研究的范式



步骤1：将数据**变换**到其他表达形式（空间）

步骤2：再在该空间中进行分析与处理

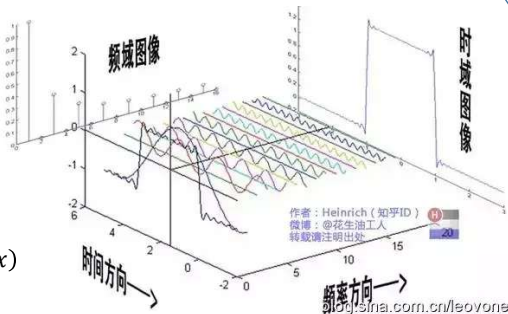
思考问题的范式： 变换到不同空间来考虑问题

- 将信息（数据）变换到不同的空间
- 再在该空间解决问题：建模、算法、实现

一个熟知的例子：

Fourier分析：
从时域变换到频域

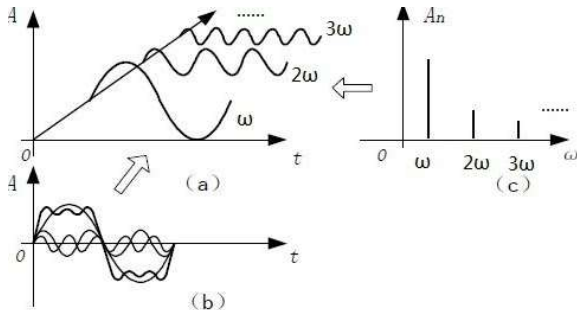
$$f(x) = a_0 + \sum_{k=1}^n (a_k \cos kx + b_k \sin kx)$$



傅里叶级数

$$f(t) = A_0 + \sum_{n=1}^{\infty} [a_n \cos(n\omega t) + b_n \sin(n\omega t)]$$

$$f(t) = A_0 + \sum_{n=1}^{\infty} A_n \sin(n\omega t + \psi_n)$$



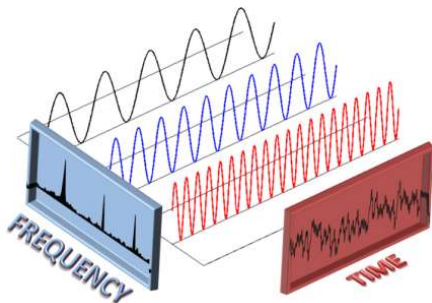
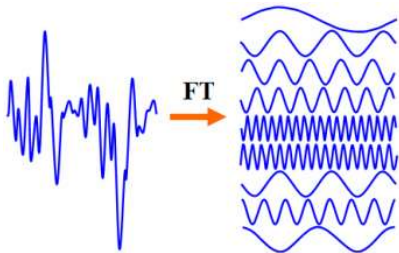
什么是“变换”： 抽取特征

- 变换就是函数： 参数化
- 变换后的空间称为隐空间（Latent space）
- 变换后的数据称为 “特征”（Feature）
- 变换的目的： 将问题简化、结构简单化
 - 例子： Fourier分析、极坐标

$$f(x) = a_0 + \sum_{k=1}^n (a_k \cos kx + b_k \sin kx)$$

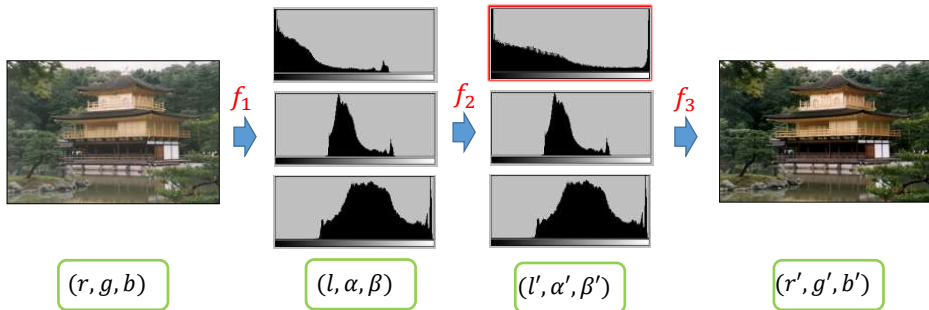
例1: Fourier变换: 从时域变换到频域

$$f(t) = A_0 + \sum_{n=1}^{\infty} A_n \cos\left(\frac{2\pi n}{T} t\right) + B_n \sin\left(\frac{2\pi n}{T} t\right)$$



例2：图像的色彩调节

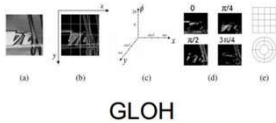
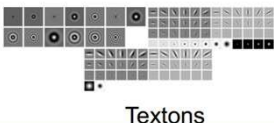
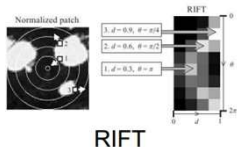
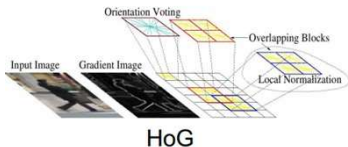
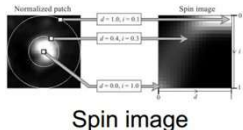
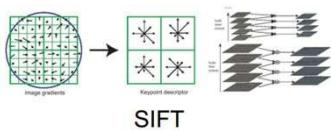
X $\xrightarrow{f}$ Y



$$Y = f(X) = f_3(f_2(f_1(X)))$$

$$f = f_3 \circ f_2 \circ f_1$$

例3：图像的各种变换



将像素的RGB表达变换到各种其他形式的表达

初始空间称为观察空间(ambient space),
变换后的空间称为隐空间(latent space)

变换后的数据称为“特征”

寻求合适的变换是解决问题的关键

初始空间称为**观察空间**(ambient space),
变换后的空间称为**隐空间**(latent space)
变换后的数据称为**特征**(feature)

如何将数据变换到合适的空间,
是研究不同问题的关键因素!
注意: 可以进行**多次**变换(Deep)

如何将数据变换到合适的空间，
是研究该问题的关键因素！
(注意：可以进行多次变换→“Deep”)

**然后在这个合适的空间来建立
数学模型、进行求解...**

如何寻求合适的变换（因果关系）？

- 有些关系可以**精确定量（方程）**描述
 - **显式**方程（解析解）：
 - 比如： $F = ma$, $v = \frac{1}{2}gt^2$, $E = mc^2$
 - **隐式**方程（无解析解）：数值解
 - 比如：函数方程、微分方程、积分方程
- 有些关系是经验性的常识，**无法精确**描述
 - 概率描述（经验、直觉）
- 有些关系**无法定量**描述，只有**观察数据**(x_i, y_i)

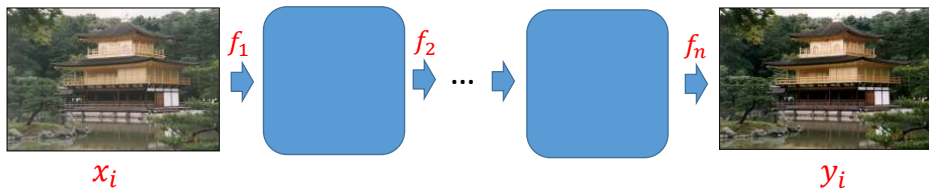
数据驱动的方法

- 根据观察数据来“学习”规律
- 本质是拟合（回归）问题

学习：由样本数据来“拟合”变换

- 一次变换往往难以达到目的，多次变换得到合理规律的可能性更大

X $\xrightarrow{f}$ Y



$$Y = f(X) = f_n(\cdots (f_2(f_1(X)))$$

$$f = f_n \circ \cdots \circ f_2 \circ f_1$$

“学习”模型

Machine Learning

≈ Looking for a Function

- Speech Recognition

$$f(\text{audio waveform}) = \text{"How are you"}$$

- Image Recognition

$$f(\text{cat image}) = \text{"Cat"}$$

- Playing Go

$$f(\text{Go board state}) = \text{"5-5"} \quad (\text{next move})$$

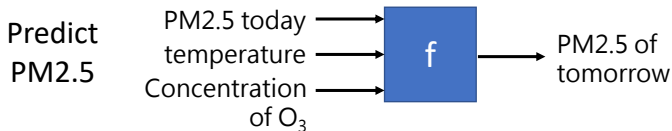
- Dialogue System

$$f(\text{"Hi"}) = \text{"Hello"}$$

(what the user said) (system response)

Different types of Functions

Regression: The function outputs a scalar.



Classification: Given options (**classes**), the function outputs the correct one.

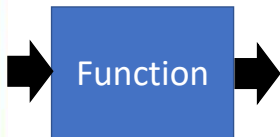


Different types of Functions

Classification: Given options (**classes**), the function outputs the correct one.



Playing GO



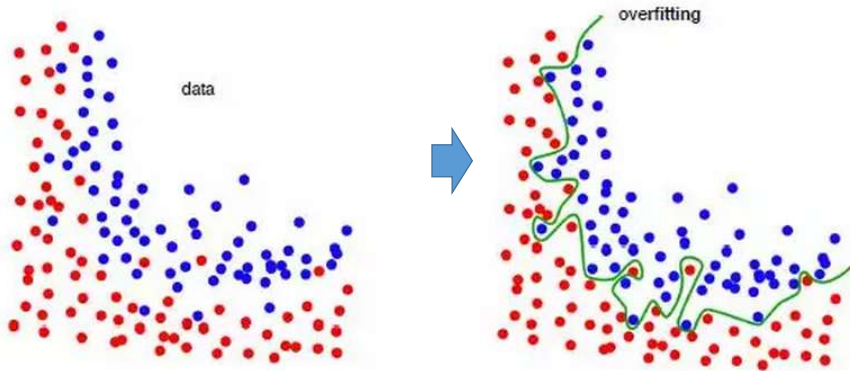
Each position
is a class
(19 x 19 classes)

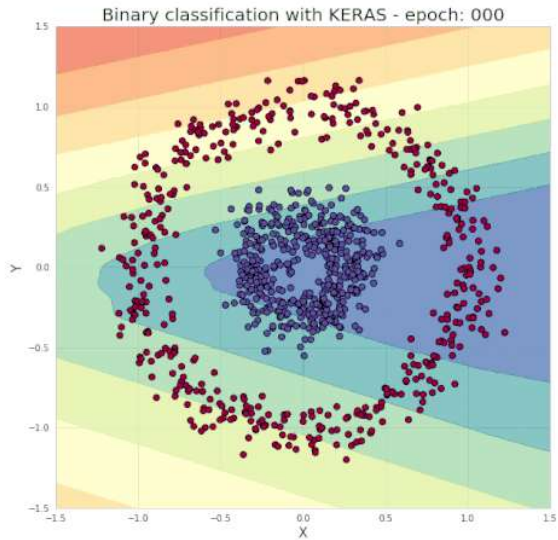


a position on
the board

Next move

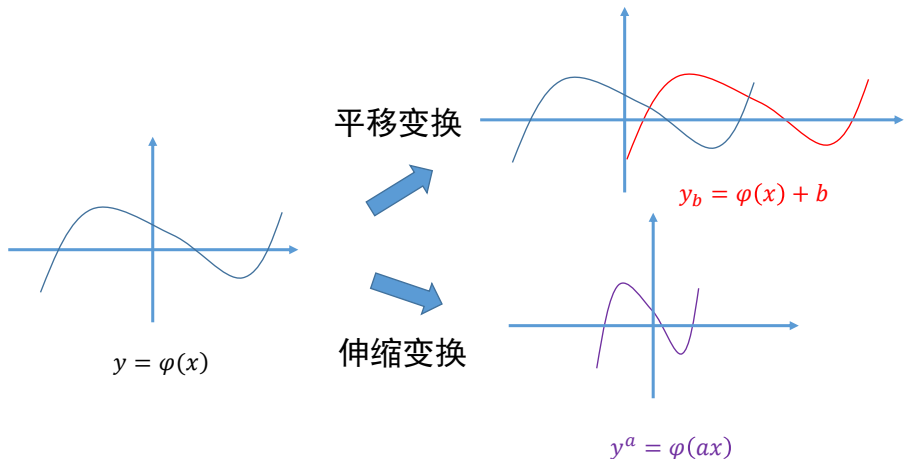
例子：分类问题





“通用”函数变换？

一个想法：用元函数及其变换



万有逼近定理

【定义 7.1】称函数 $\varphi(x)$ 为压缩函数，如果 $\varphi(x)$ 单调不减，且满足

$$\lim_{x \rightarrow -\infty} \varphi(x) = 0, \quad \lim_{x \rightarrow +\infty} \varphi(x) = 1.$$

【定理 7.1】若 φ 是压缩函数，则 $S(\varphi)$ 在 C^m 中一致稠密，在 M^m 中按如下距离 ρ_μ 下稠密：

$$\rho_\mu = \inf\{\varepsilon > 0: \mu\{x: |f(x) - g(x)| > \varepsilon\} < \varepsilon\}.$$

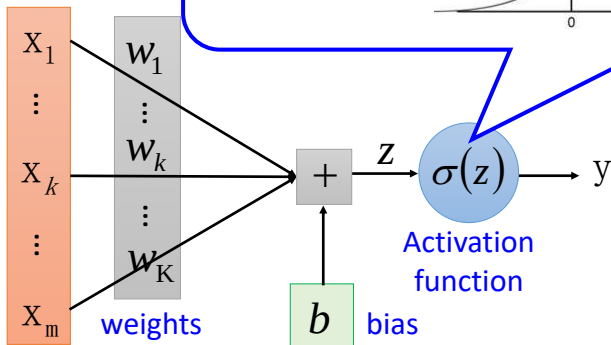
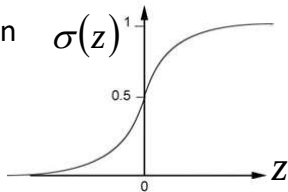
【定理 7.3】若 φ 是连续有界的非常值函数，则 $S(\varphi)$ 在 C^m 中稠密。

高维Input: 将线性函数复合一个非线性函数
作为一个“**神经元**”

A simple function

Sigmoid Function

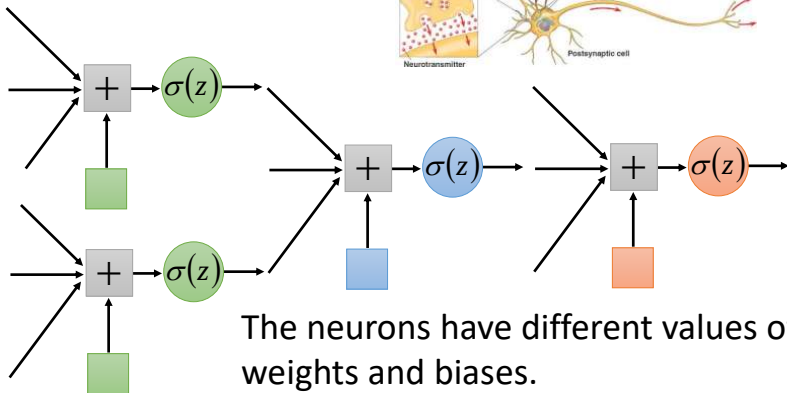
$$\sigma(z) = \frac{1}{1 + e^{-z}}$$



$$z = w_1 x_1 + \dots + w_k x_k + \dots + w_m x_m + b$$

Neural Network

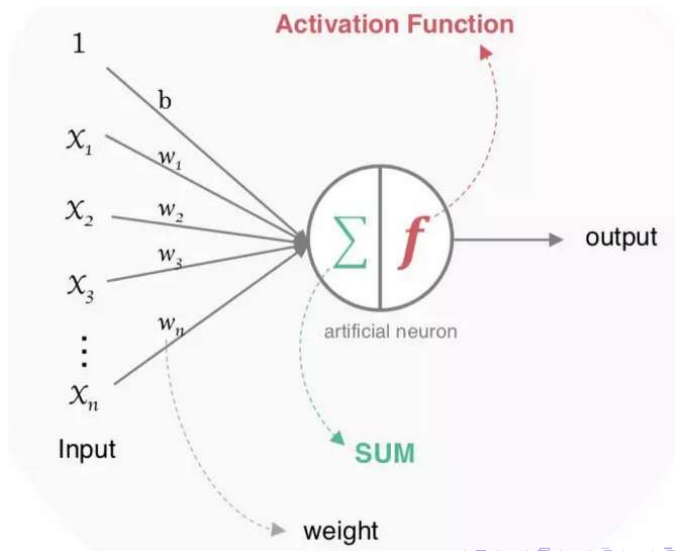
Different connections lead to different network structures



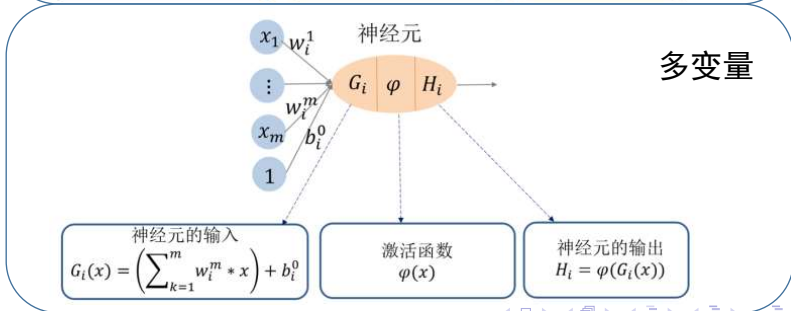
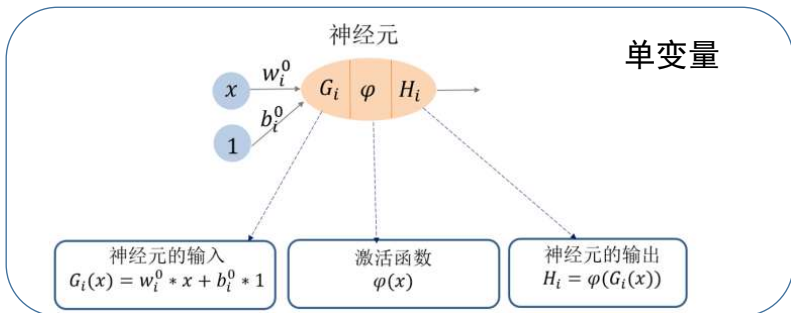
The neurons have different values of weights and biases.

Weights and biases are network parameters θ

高维Input: 将线性函数复合一个非线性函数
作为一个“**神经元**”

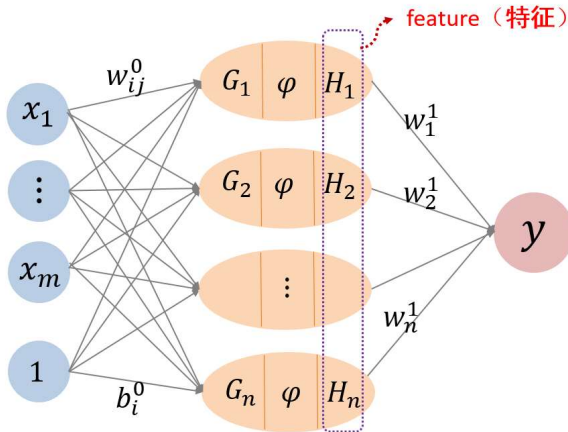


神经元



单隐层神经网络： $y(x) = \sum_{k=1}^n w_k^1 H_k(x)$

一个复杂的函数：由很多小函数组合而成



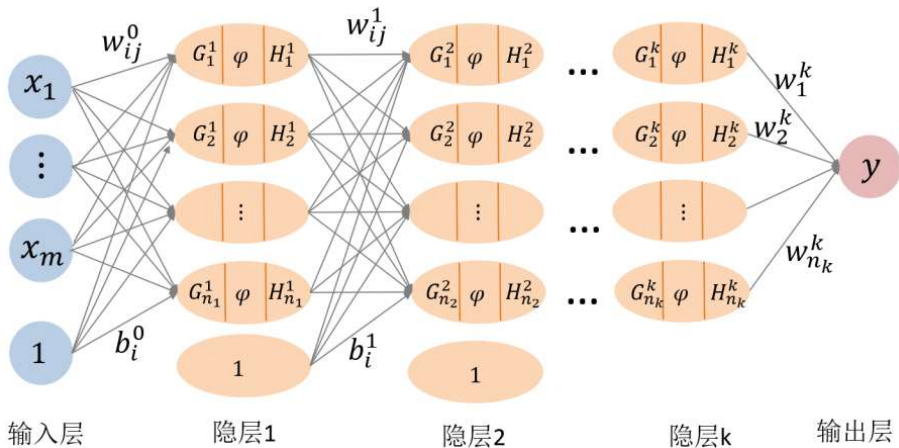
输入层

隐层

输出层

多隐层神经网络：多重复合函数

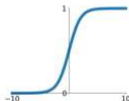
一个复杂的函数：由很多小函数复合而成



Activation Functions (激活函数)

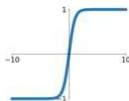
Sigmoid

$$\sigma(x) = \frac{1}{1+e^{-x}}$$



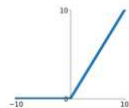
tanh

$$\tanh(x)$$



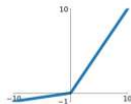
ReLU

$$\max(0, x)$$



Leaky ReLU

$$\max(0.1x, x)$$



Maxout

$$\max(w_1^T x + b_1, w_2^T x + b_2)$$

ELU

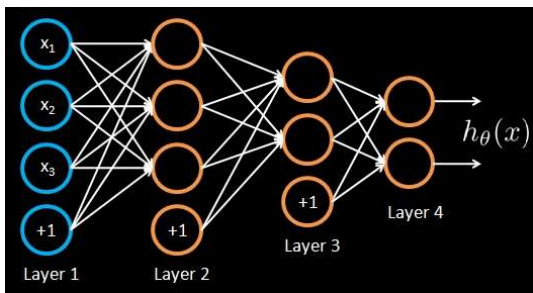
$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



使用简单元函数及其平移构成“基函数”（归一化）

【类比：径向基函数(RBF)】

用神经网络函数来拟合数据



Regression problem:

Input: Given training set $(x_1, y_1), (x_2, y_2), (x_3, y_3), \dots$

Output: Adjust **parameters θ** (for every node) to make:

$$h(x_i) \approx y_i$$

优化算法（BP算法）

给定了一批训练数据，计算神经网络的问题就是通过极小化损失函数来找到网络中的所有权值参数（包括偏置参数）。这个过程本质上就是利用这个神经网络函数来拟合这些数据，这个过程也称为“训练”或“学习”。

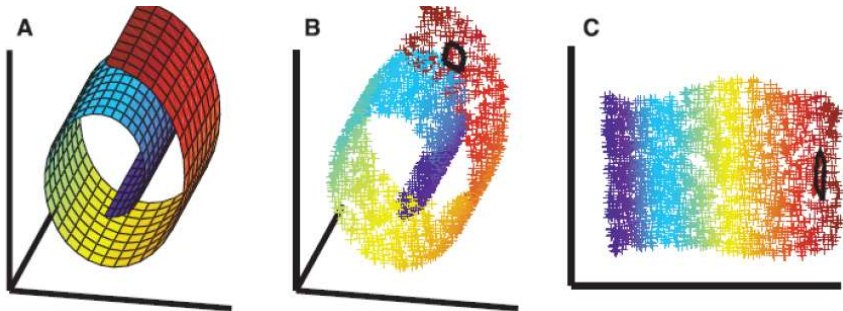
误差反向传播算法（Error Back Propagation, BP）是神经网络训练时采用的一种通用方法。本质是随机梯度下降法。最早见于如下论文：

[8-1] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning internal representations by back-propagating errors. Nature, 323(99): 533-536, 1986.

由于整个网络是一个复合函数，因此，训练的过程就是不断地应用“复合函数求导”（链式法则）及“梯度下降法”。反向传播算法从神经网络的输出层开始，利用递推公式根据后一层的误差计算本层的误差，通过误差计算本层参数的梯度值，然后将差项传播到前一层。然后根据这些误差来更新这些系数参数。

数据的真实维度？

- 图形学：三维曲面已知是二维流形
- 一般问题：并不知道数据的真实维数
 - 流形学习



神经网络函数的表达能力

- 神经网络的表达能力取决于其参数个数
 - 网络足够大，参数就足够多，表达能力就足够强
 - 任何一个特定的网络不可能表达任意复杂的函数

深度学习三步骤

在哪找？

定义函数空间：合适的神经网络

哪个好？

定义损失函数：评价函数的好坏

怎么找？

优化求解方法：找到想要的函数

上述步骤不断尝试

深度神经网络不是新发明， 为什么近年来爆发？

大数据

高效算法

高性能机器

机器学习

非端到端的学习

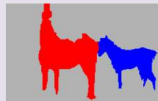


特征
工程

人工设计
特征

预测
模型

分割识别



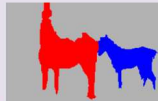
端到端的学习



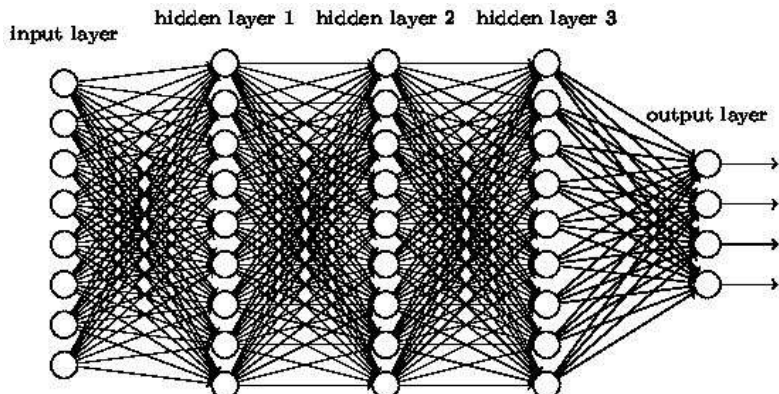
特征学习

深度神经网络

分割识别



深度学习： 基于深度神经网络的数据拟合



优势：自动抽象特征、无须预测模型！
整合了两个过程（端对端）！

Key: Extracting Feature Hierarchies

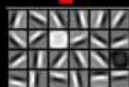
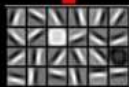
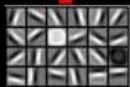
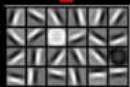
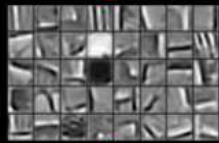
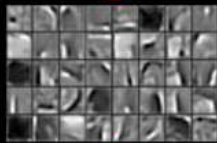
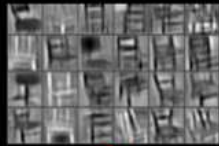
Features learned from training on different object classes.

Faces

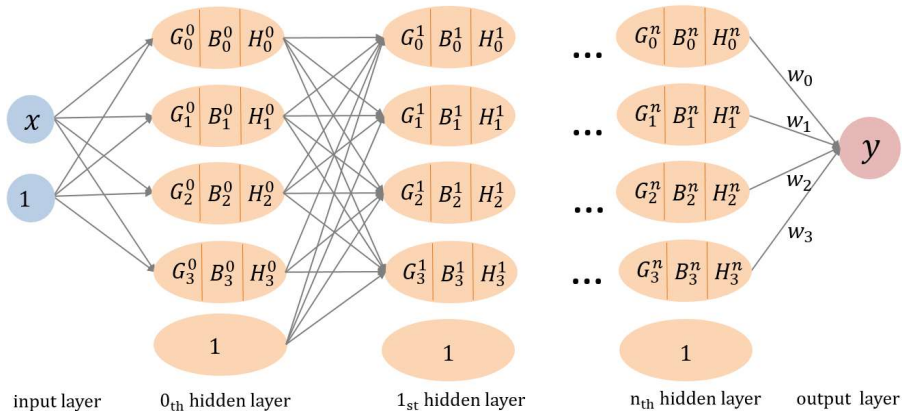
Cars

Elephants

Chairs



! 统一的深度网络框架!



**Traditional
function**

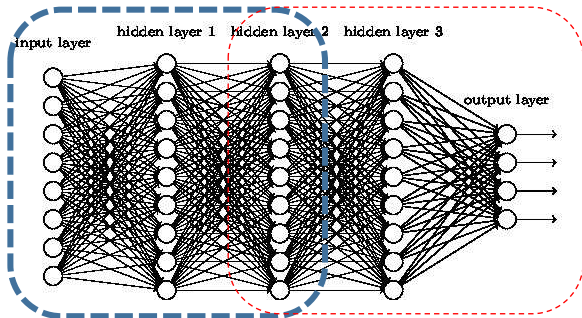
DNN

**RBF
network**

ELM

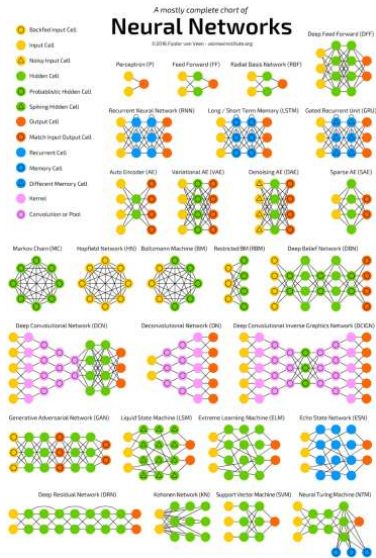
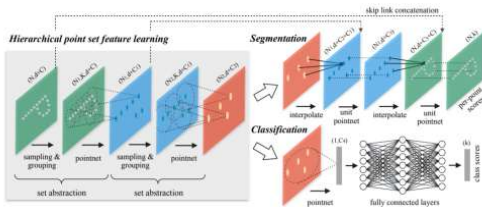
...

Recap: 深度神经网络函数

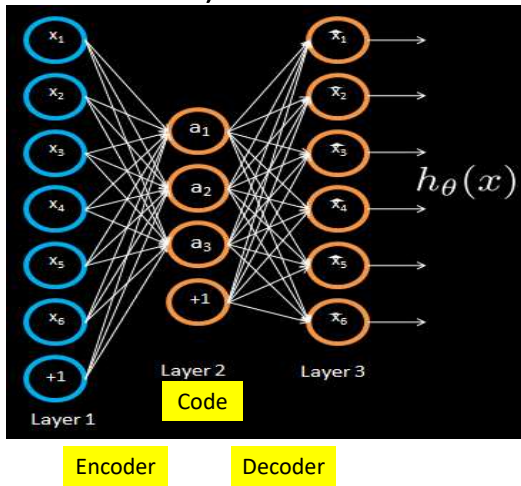


- 中间的任何一个隐层都可以作为**特征**
- 中间的任何连续多层之间都可作为一个函数 (function blocks), 可以任意**组合与拼接**

轻松看懂所有网络结构！

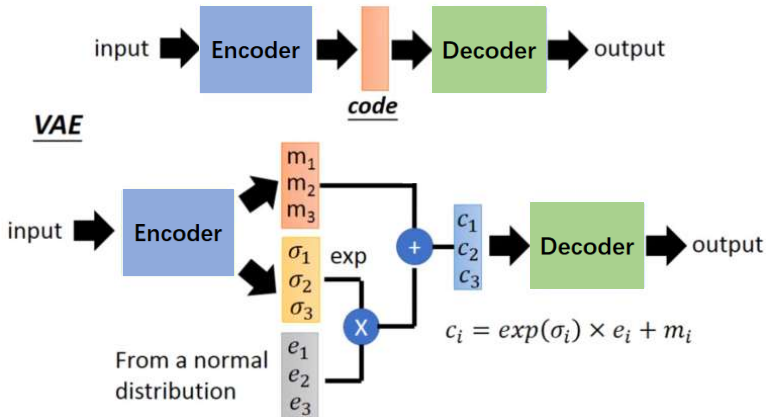


Unsupervised Feature Learning (Auto-Encoder)



$$h(x) = x$$

Variational Auto-Encoder (VAE)



变种的神经元

- 卷积神经元 (Convolutional cells)
- 池化神经元和插值神经元 (Pooling and interpolating cells)
- 概率神经元：均值神经元和标准方差神经元 (Mean and standard deviation cells)
- 循环神经元 (Recurrent cells)
- 长短期记忆神经元 (Long short term memory cells)
- 门控循环神经元 (Gated recurrent cells)
- ...

不断涌现出的各种神经网络及学习方法

- Convolutional Neural Network (CNN)
- Recurrent Neural Network (RNN)
- Generative Adversarial Network (GAN)
- Reinforcement Learning
- Transfer learning
- Metric learning
- ...

层出不穷的新名字

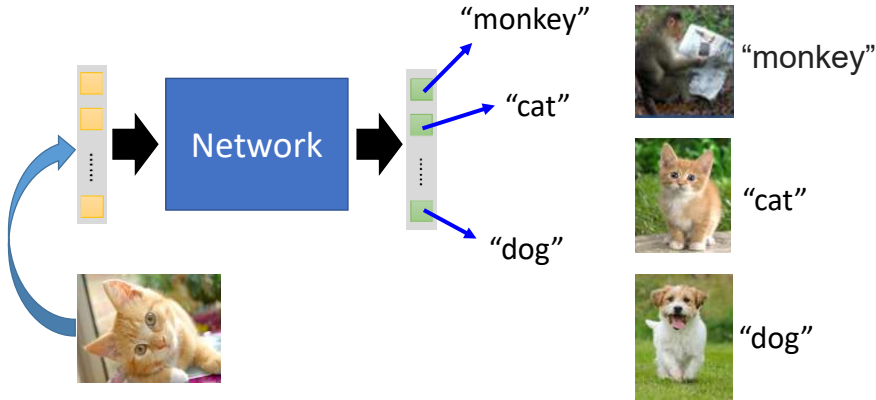
- -machine “机”
- -net “网”

使用深度学习的方法

- 理解问题
- 数学建模
- 设计合适的神经网络
- 调试和修改网络

For example, you can do

- Image Recognition



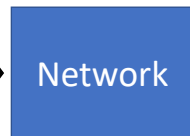
For example, you can do

Spam
filtering

"Talk" in e-mail



"free" in e-mail

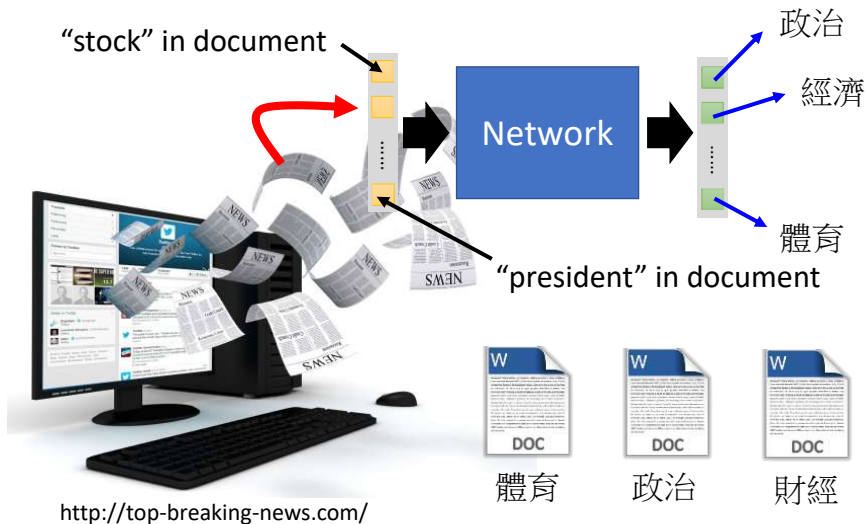


1/0
(Yes/No)



(<http://spam-filter-review.toptenreviews.com/>)

For example, you can do



机器学习：不仅仅是深度神经网络

- 逻辑回归
- SVM
- KNN
- 决策树
- 随机森林
- Clustering（聚类分析）
- Boosting
- 贝叶斯网络
- 非监督学习
- ...

学习资料

- 李宏毅（台湾大学），《机器学习》
 - <https://speech.ee.ntu.edu.tw/~hylee/ml/2021-spring.html>
- Andrew Ng（Coursera，神经网络与深度学习）
 - <https://www.coursera.org/learn/neural-networks-deep-learning>



中国科学技术大学
University of Science and Technology of China

谢谢！



中国科学技术大学
University of Science and Technology of China

数学建模 Mathematical Modeling

刘利刚

中国科学技术大学

4. 各种网络模型

Machine Learning

Machine learning is to find a function f

$$f : X \rightarrow Y$$

Regression: output a scalar

Classification: output a “class” (one-hot vector)

1	0	0
---	---	---

Class 1

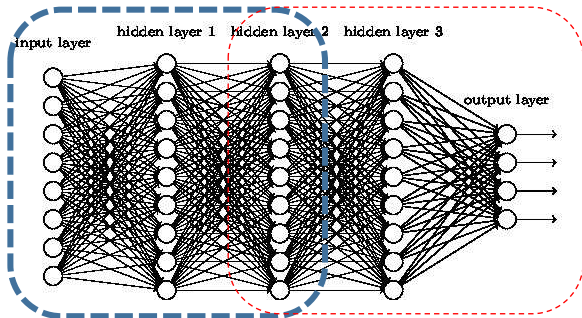
0	1	0
---	---	---

Class 2

0	0	1
---	---	---

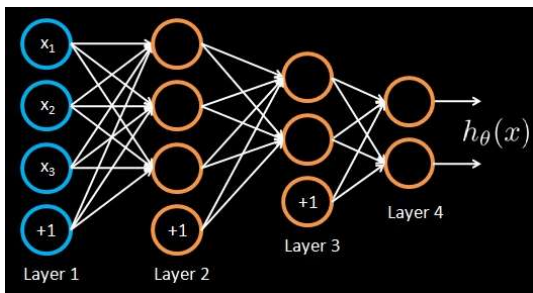
Class 3

Recap: 深度神经网络函数



- 中间的任何一个隐层都可以作为**特征**
- 中间的任何连续多层之间都可作为一个函数 (function blocks), 可以任意**组合与拼接**

用神经网络函数来拟合数据



Regression problem:

Input: Given training set $(x_1, y_1), (x_2, y_2), (x_3, y_3), \dots$

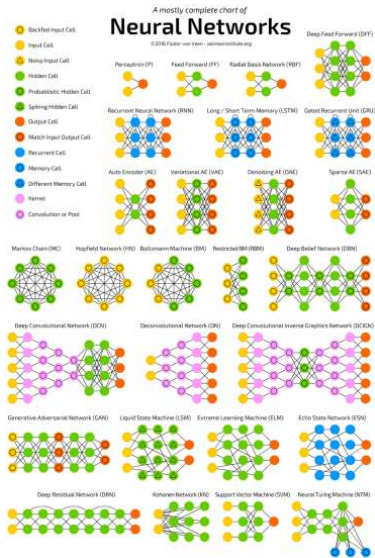
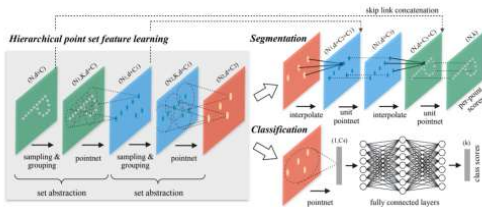
Output: Adjust **parameters θ** (for every node) to make:

$$h(x_i) \approx y_i$$

神经网络模型

- 从数学上看，本质是函数拟合模型
 - 数据驱动的方法
 - “学习”→“拟合”
- 优势：任何维数的大型函数空间、超大规模数据拟合、可优化求解
 - “可微优化”(differential rendering/simulation...)
- 并不是真正意义上的“智能”
 - 能替代人们做不少事情
- 更确切：神经网络计算

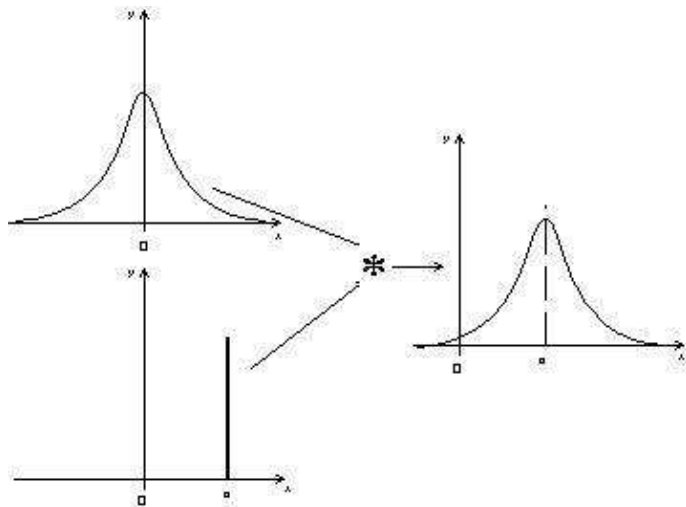
轻松看懂所有网络结构！

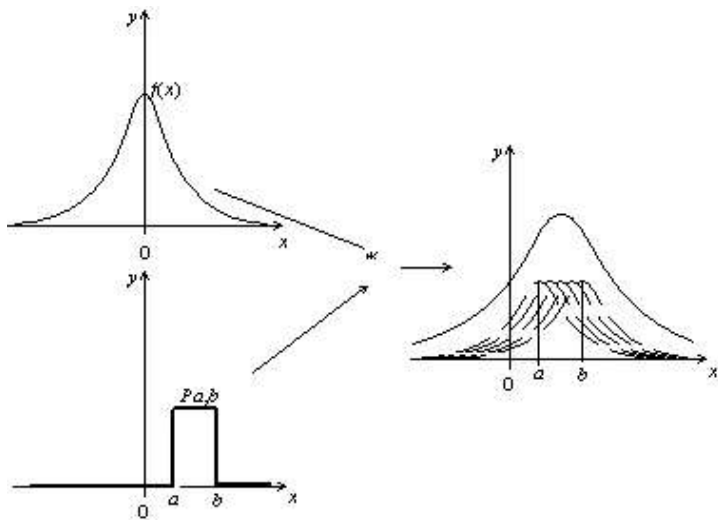


卷积 (Convolution)

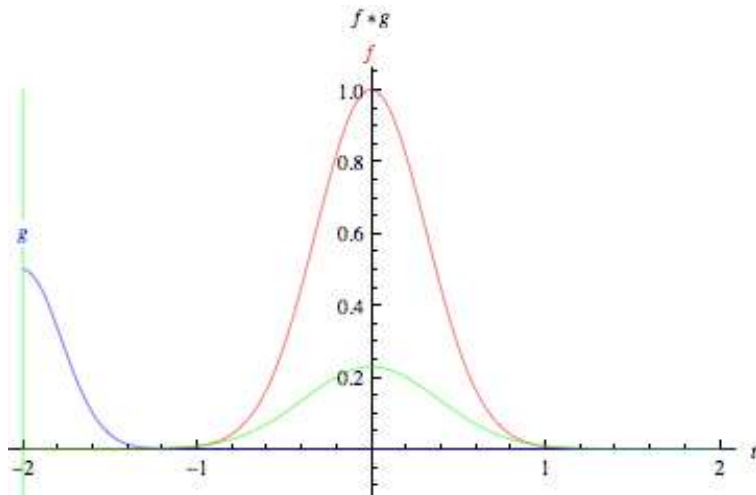
卷积：数学定义

$$f(x) * g(x) = \int_{-\infty}^{\infty} f(\tau) \cdot g(x - \tau) d\tau$$

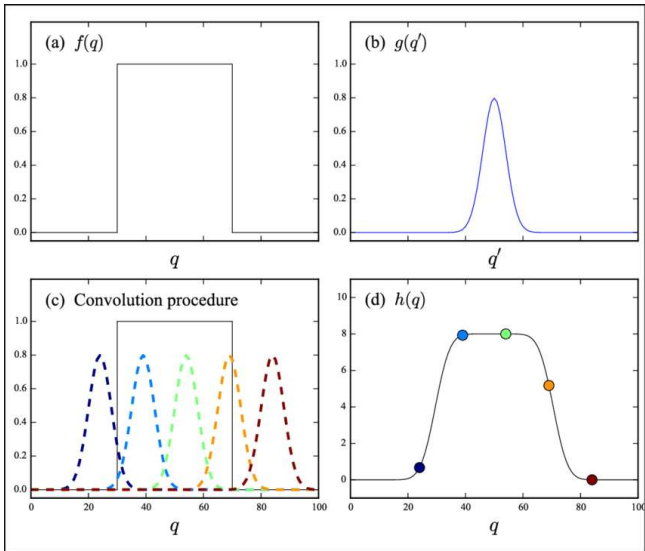




Gaussian Convolution (Filtering)



Gaussian Convolution (Filtering)



$$\begin{aligned}
 (f * g)(x) &\stackrel{\text{def.}}{=} \int_{-\infty}^{+\infty} f(x-t) \cdot g(t) \, dt \\
 &= \int_{+\infty}^{-\infty} f(T) \cdot g(x-T) \, d(-T) \\
 &= \int_{-\infty}^{+\infty} f(T) \cdot g(x-T) \, dT \\
 &\stackrel{\text{def.}}{=} (g * f)(x)
 \end{aligned}$$

2D Convolution

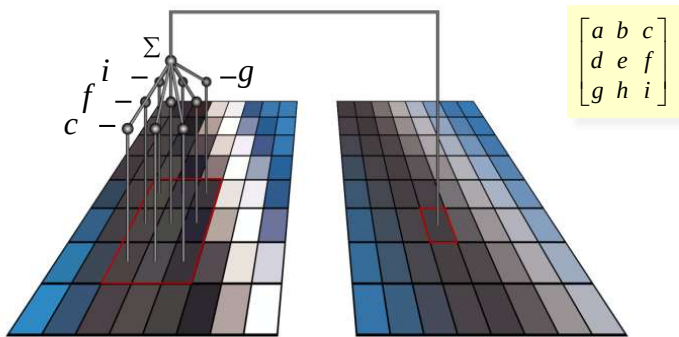
If a MW transformation is *linear* then it is a *convolution*:

$$\mathbf{J}(r,c) = \mathbf{I} * \mathbf{h}(r,c) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \mathbf{I}(r-r', c-c') \mathbf{h}(r', c') dr' dc',$$

for a real image ($\mathbf{I}: R \times R \rightarrow R$), or for a digital image ($\mathbf{I}: Z \times Z \rightarrow Z$):

$$\mathbf{J}(r,c) = \mathbf{I} * \mathbf{h}(r,c) = \sum_{r'=-s}^s \sum_{c'=-d}^d \mathbf{I}(r-r', c-c') \mathbf{h}(r', c')$$

Convolution by Moving Window

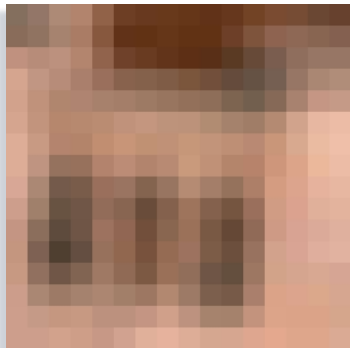


Another example

Moving Window Transform: Example

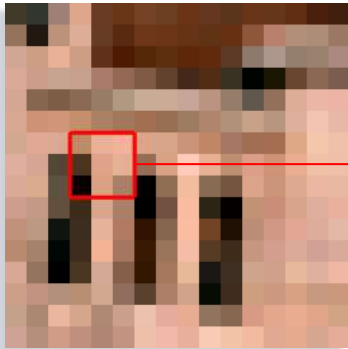


original

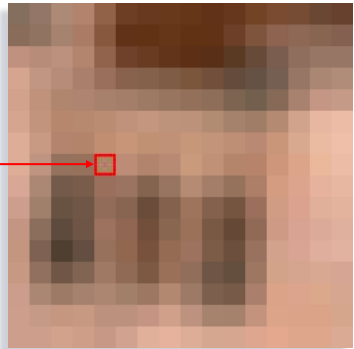


3x3 average

Moving Window Transform: Example

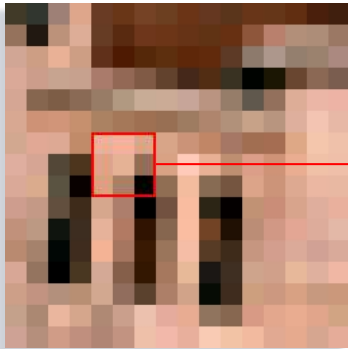


original

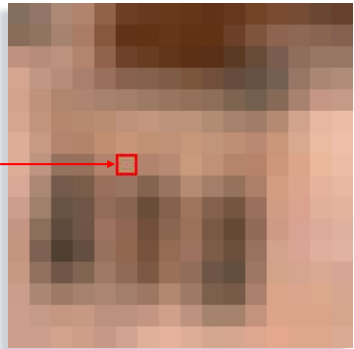


3x3 average

Moving Window Transform: Example

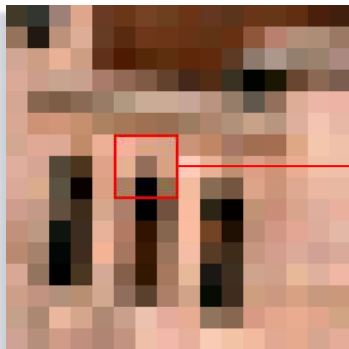


original

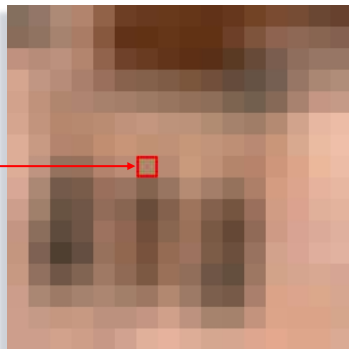


3x3 average

Moving Window Transform: Example

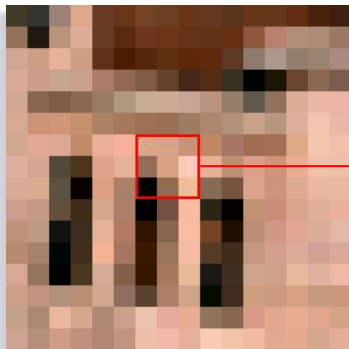


original

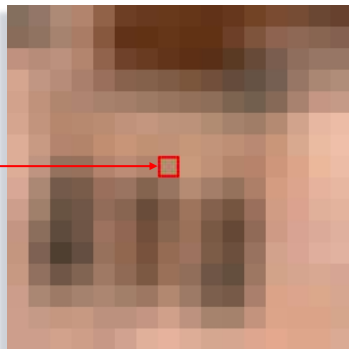


3x3 average

Moving Window Transform: Example

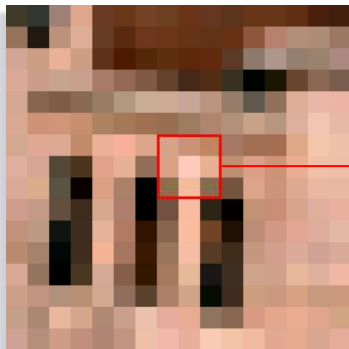


original

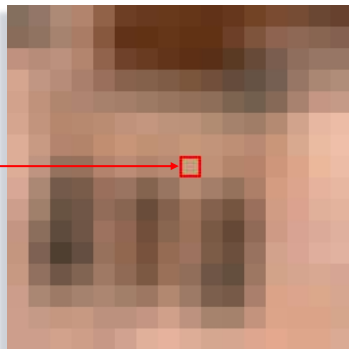


3x3 average

Moving Window Transform: Example

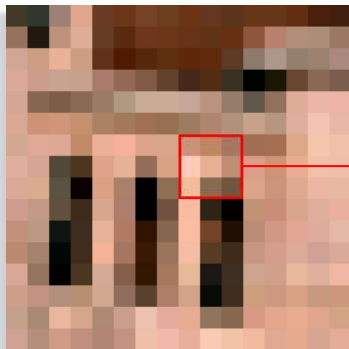


original

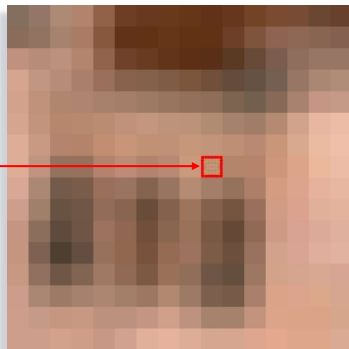


3x3 average

Moving Window Transform: Example

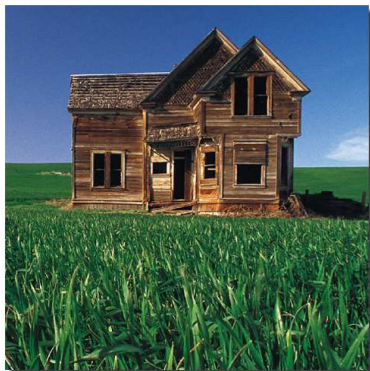
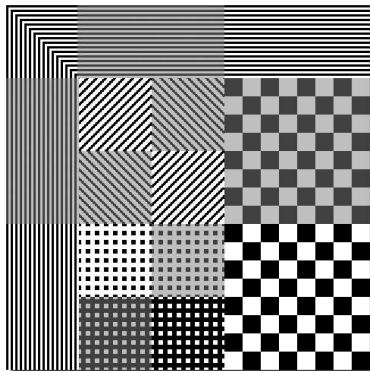


original



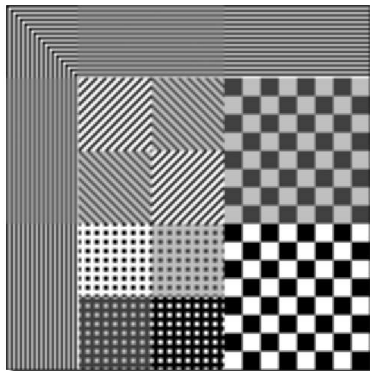
3x3 average

Convolution Examples: Original Images



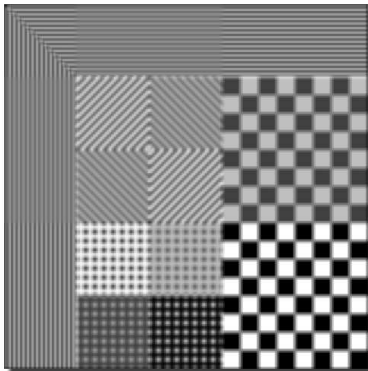
$$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Convolution Examples: 3×3 Blur



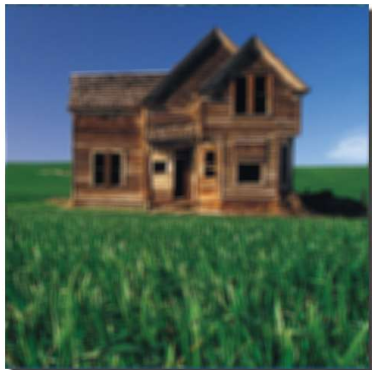
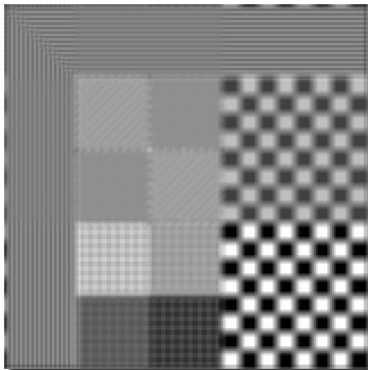
$$\frac{1}{25} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Convolution Examples: 5x5 Blur

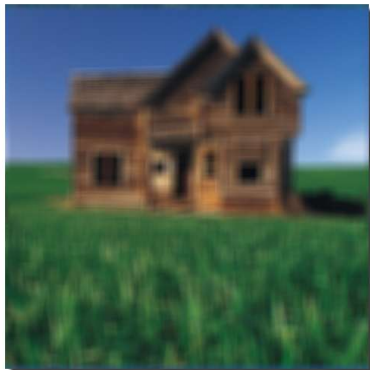
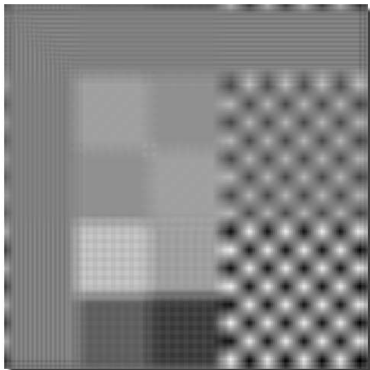


$$\frac{1}{81} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Convolution Examples: 9×9 Blur

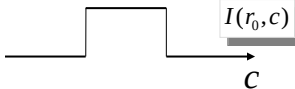
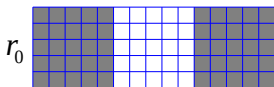


Convolution Examples: 17x17 Blur



Vertical Edge Detection

Image

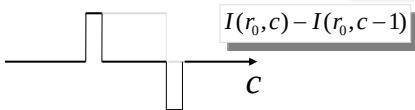
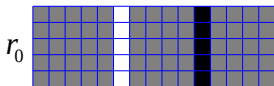


255

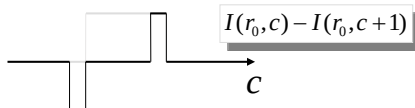
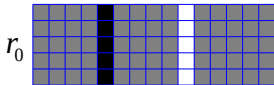
0

-255

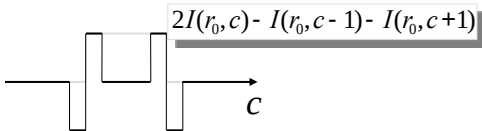
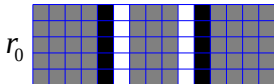
Backward
Difference



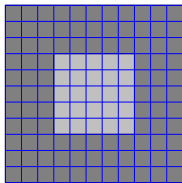
Forward
Difference



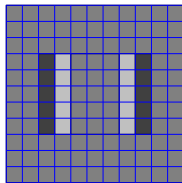
Sum of
Differences



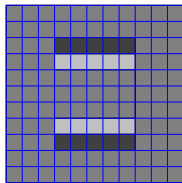
Symmetric Edge Detection



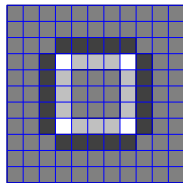
$I(r,c)$



$$2I(r,c) - I(r,c-1) - I(r,c+1)$$



$$2I(r,c) - I(r-1,c) - I(r+1,c)$$



$$4I(r,c) - I(r-1,c) - I(r+1,c) - I(r,c-1) - I(r,c+1)$$

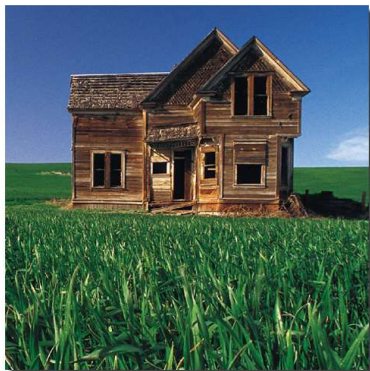
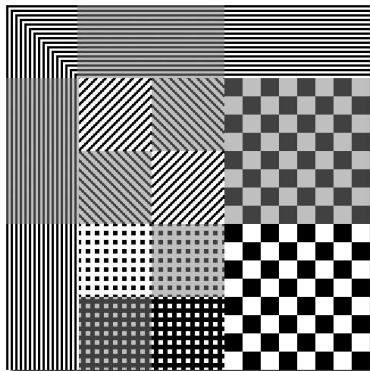
□ 510
 ■ 255
 ■ 0
 ■ -255

-1	2	-1

	-1	
	2	
	-1	

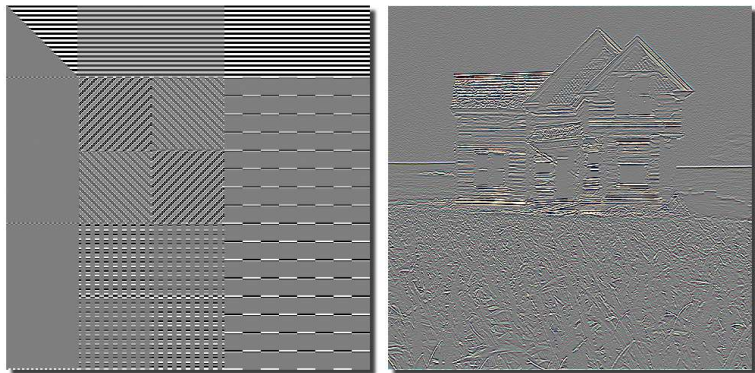
	-1	
-1	4	-1
	-1	

Convolution Examples: Original Images



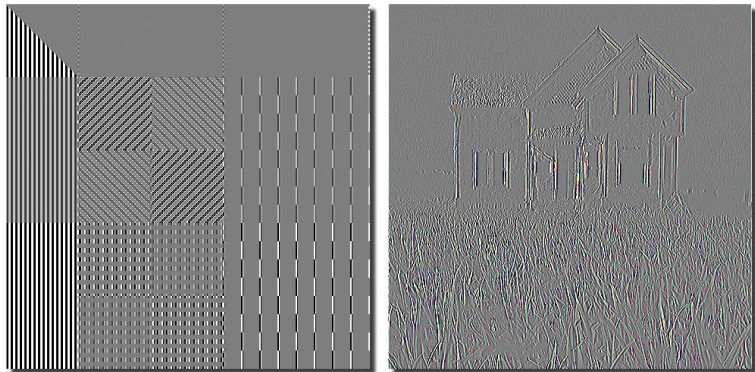
$$\begin{bmatrix} -1 \\ 2 \\ -1 \end{bmatrix}$$

Convolution Examples: Vertical Difference



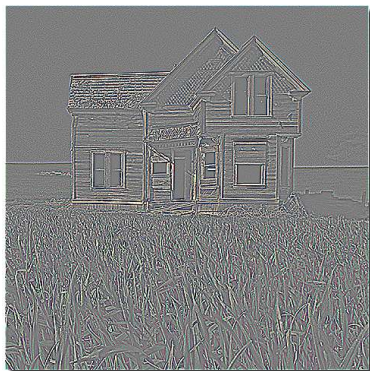
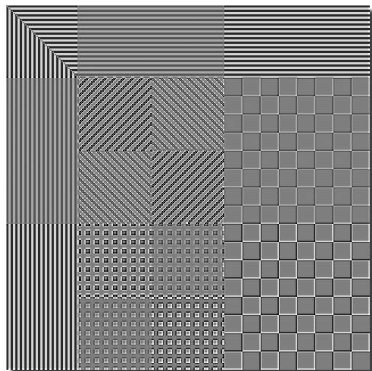
$$\begin{bmatrix} -1 & 2 & -1 \end{bmatrix}$$

Convolution Examples: Horizontal Difference



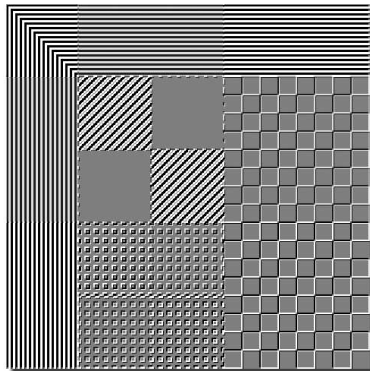
$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

Convolution Examples: H + V Diff.



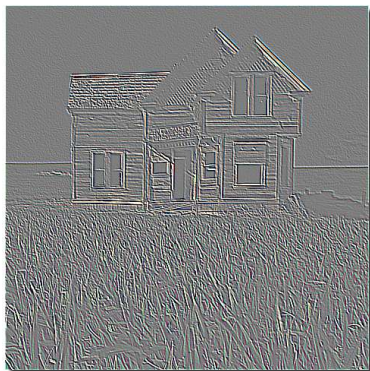
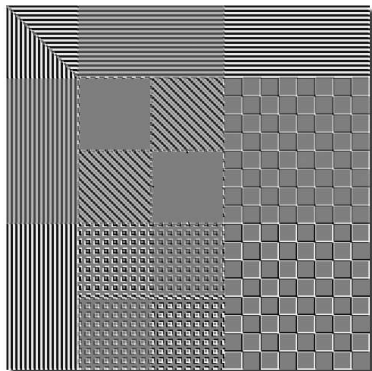
$$\begin{bmatrix} -1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & -1 \end{bmatrix}$$

Convolution Examples: Diagonal Difference



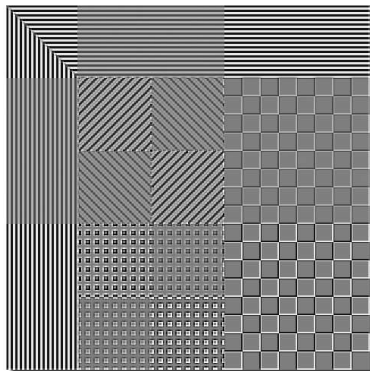
$$\begin{bmatrix} 0 & 0 & -1 \\ 0 & 2 & 0 \\ -1 & 0 & 0 \end{bmatrix}$$

Convolution Examples: Diagonal Difference



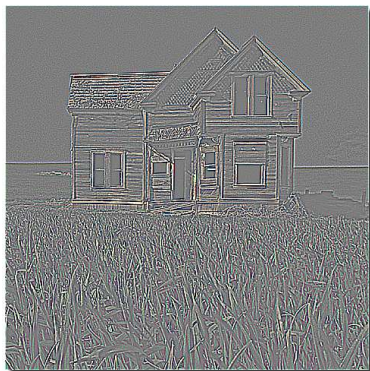
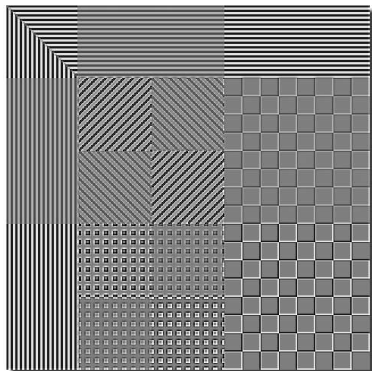
$$\begin{bmatrix} -1 & 0 & -1 \\ 0 & 4 & 0 \\ -1 & 0 & -1 \end{bmatrix}$$

Convolution Examples: D + D Difference



$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

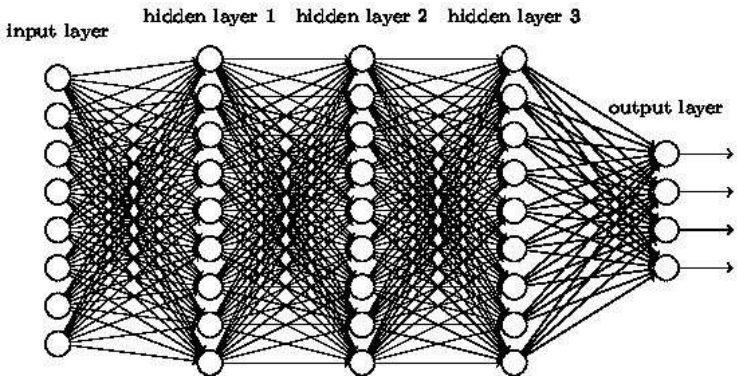
Convolution Examples: H + V + D Diff.



卷积神经网络

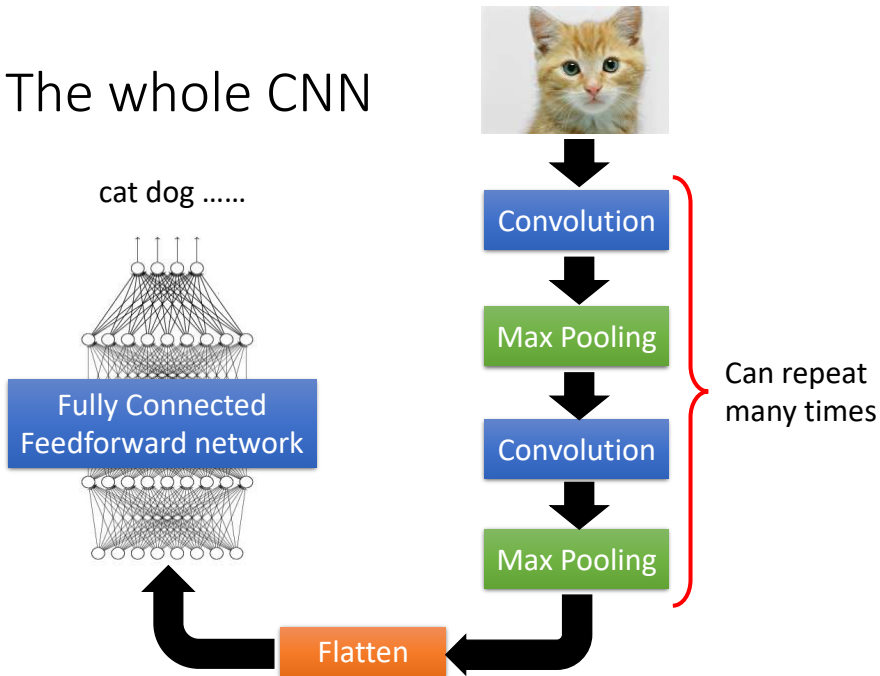
Convolutional Neural Network (CNN)

深度神经网络：是否都要全连接？



局部关联（局部视野）？

The whole CNN



The whole CNN

Property 1

- Some patterns are much smaller than the whole image

Property 2

- The same patterns appear in different regions.

Property 3

- Subsampling the pixels will not change the object



Convolution

Max Pooling

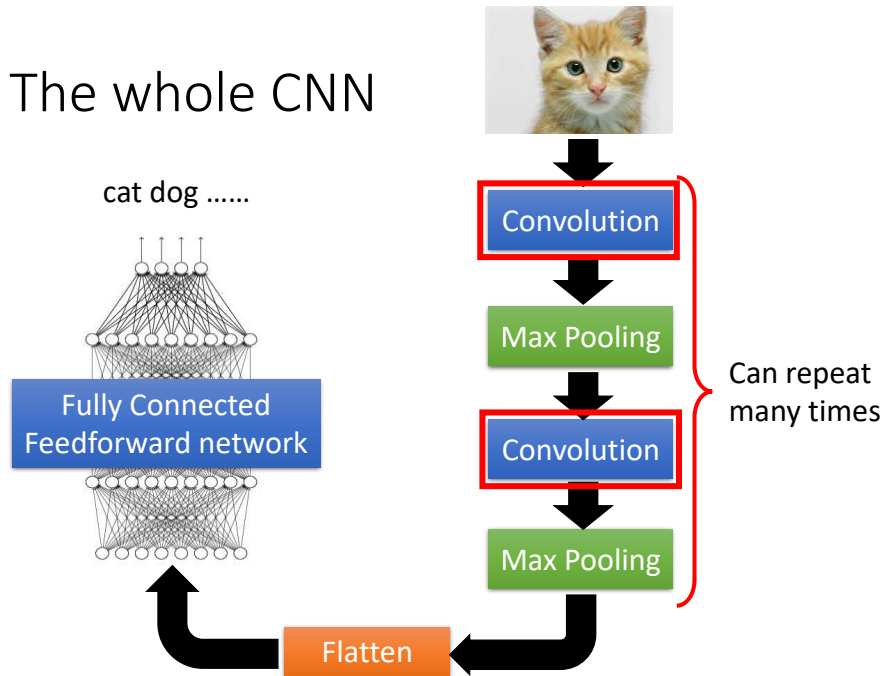
Convolution

Max Pooling

Flatten

Can repeat many times

The whole CNN



CNN – Convolution

Those are the network parameters to be learned.

1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

6 x 6 image

1	-1	-1
-1	1	-1
-1	-1	1

Filter 1

Matrix

-1	1	-1
-1	1	-1
-1	1	-1

Filter 2

Matrix

⋮

Property 1

Each filter detects a small pattern (3 x 3).

CNN – Convolution

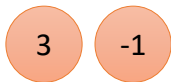
stride=1

1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

6 x 6 image

1	-1	-1
-1	1	-1
-1	-1	1

Filter 1



CNN – Convolution

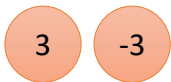
1	-1	-1
-1	1	-1
-1	-1	1

Filter 1

If stride=2

1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

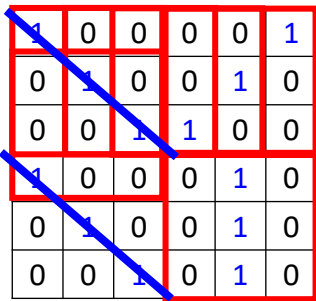
6 x 6 image



We set stride=1 below

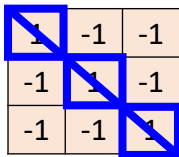
CNN – Convolution

stride=1



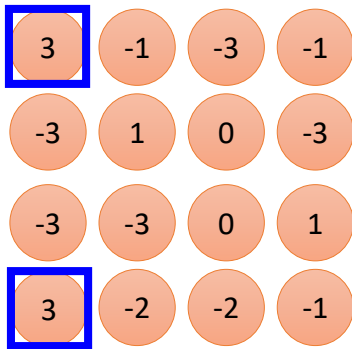
1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

6 x 6 image



1	-1	-1
-1	1	-1
-1	-1	1

Filter 1



3	-1	-3	-1
-3	1	0	-3
-3	-3	0	1
3	-2	-2	-1

Property 2

CNN – Convolution

-1	1	-1
-1	1	-1
-1	1	-1

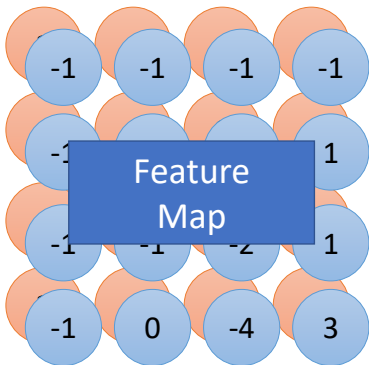
Filter 2

stride=1

1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

6 x 6 image

Do the same process for every filter



4 x 4 image

CNN – Zero Padding

1	-1	-1
-1	1	-1
-1	-1	1

Filter 1

0	0	0					
0	1	0	0	0	0	1	
0	0	1	0	0	1	0	
	0	0	1	1	0	0	
	1	0	0	0	1	0	
	0	1	0	0	1	0	0
	0	0	1	0	1	0	0
							0
							0
							0

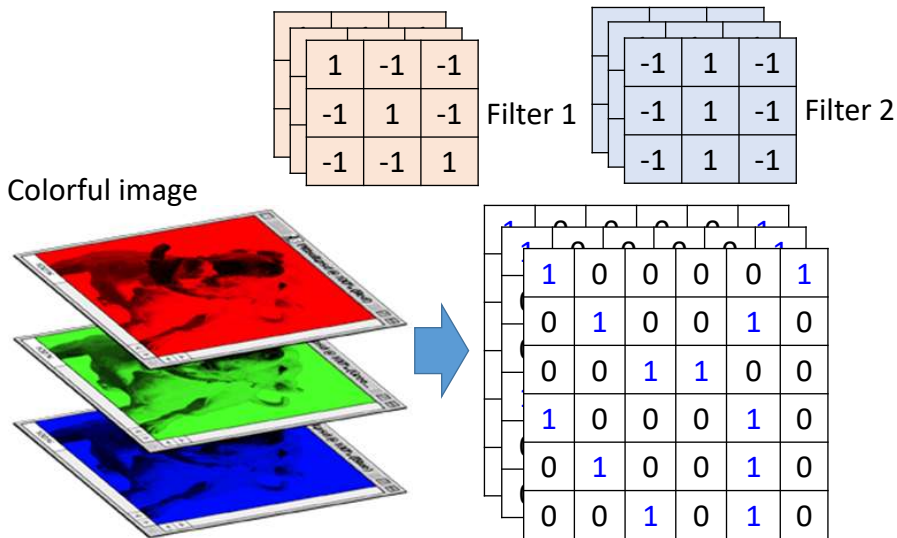
6 x 6 image

You will get another 6 x 6 images in this way



Zero padding

CNN – Colorful image



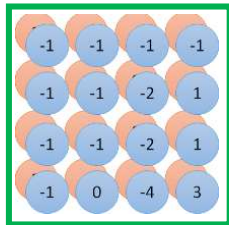
Convolution v.s. Fully Connected

1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

image

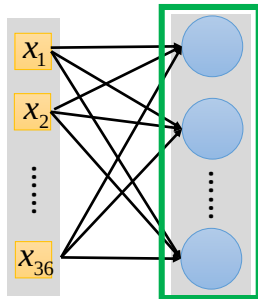
1	-1	-1
-1	1	-1
-1	-1	1

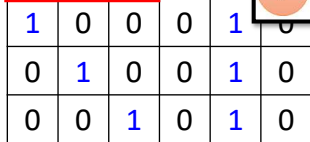
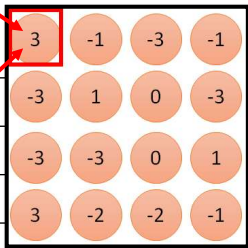
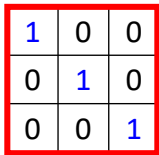
-1	1	-1
-1	1	-1
-1	1	-1



Fully-
connected

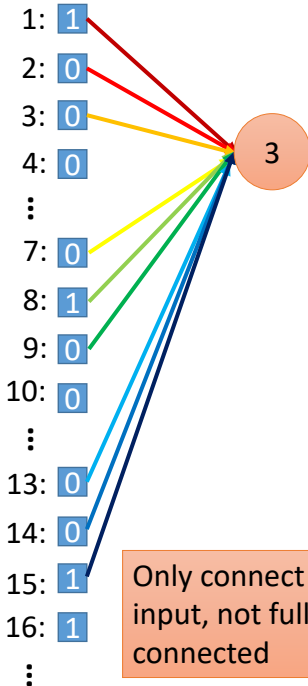
1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0



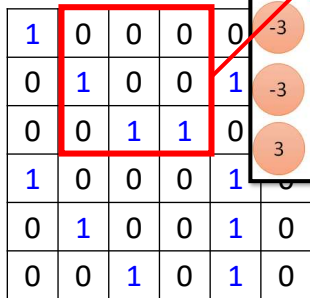


6 x 6 image

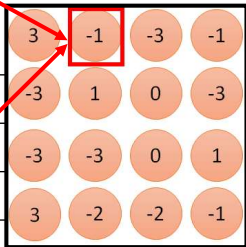
Less parameters!



Only connect to 9 input, not fully connected

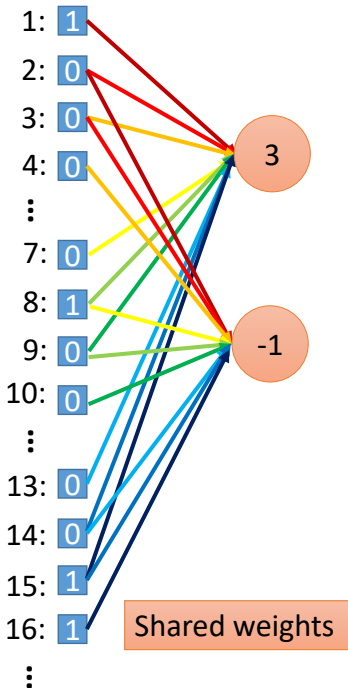


6 x 6 image

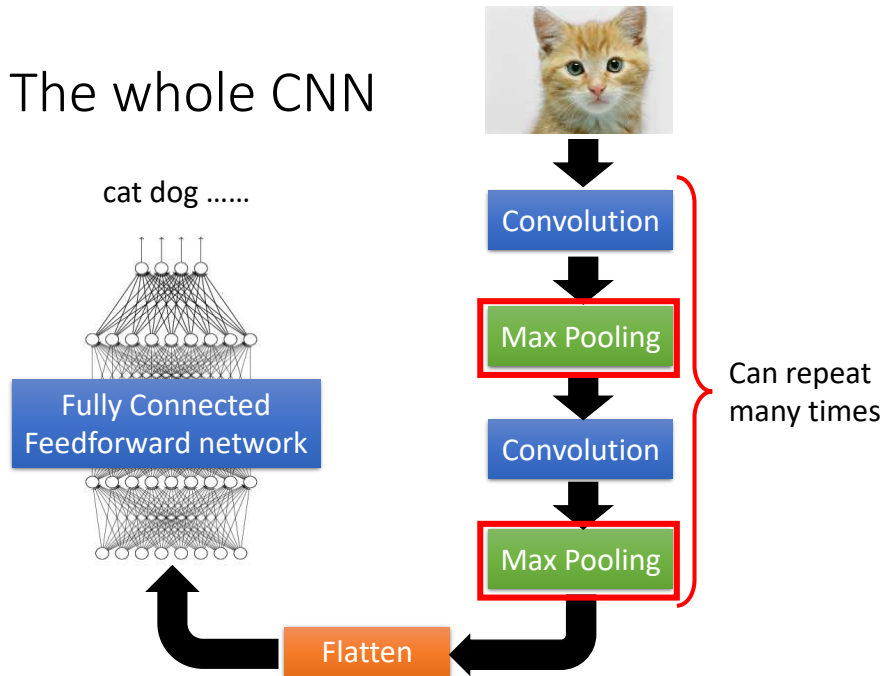


Less parameters!

Even less parameters!



The whole CNN



CNN – Max Pooling

1	-1	-1
-1	1	-1
-1	-1	1

Filter 1

-1	1	-1
-1	1	-1
-1	1	-1

Filter 2

3	-1	-3	-1
-3	1	0	-3
-3	-3	0	1
3	-2	-2	-1

-1	-1	-1	-1
-1	-1	-2	1
-1	-1	-2	1
-1	0	-4	3

CNN – Max Pooling

1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

6 x 6 image



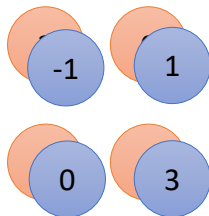
Conv



Max
Pooling



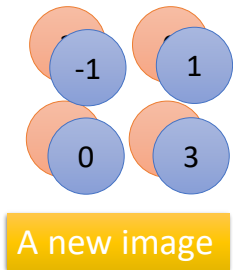
New image
but smaller



2 x 2 image

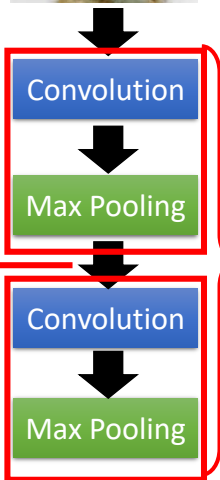
Each filter
is a channel

The whole CNN



Smaller than the original image

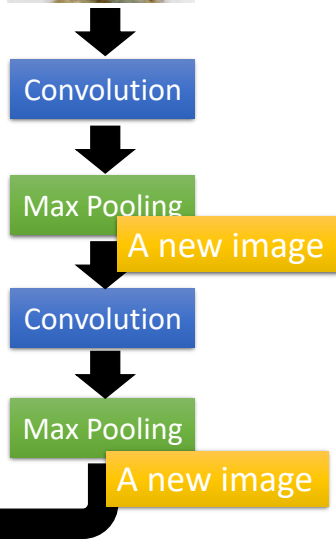
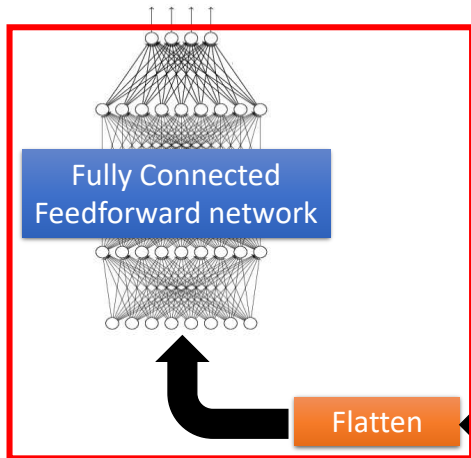
The number of the channel is the number of filters



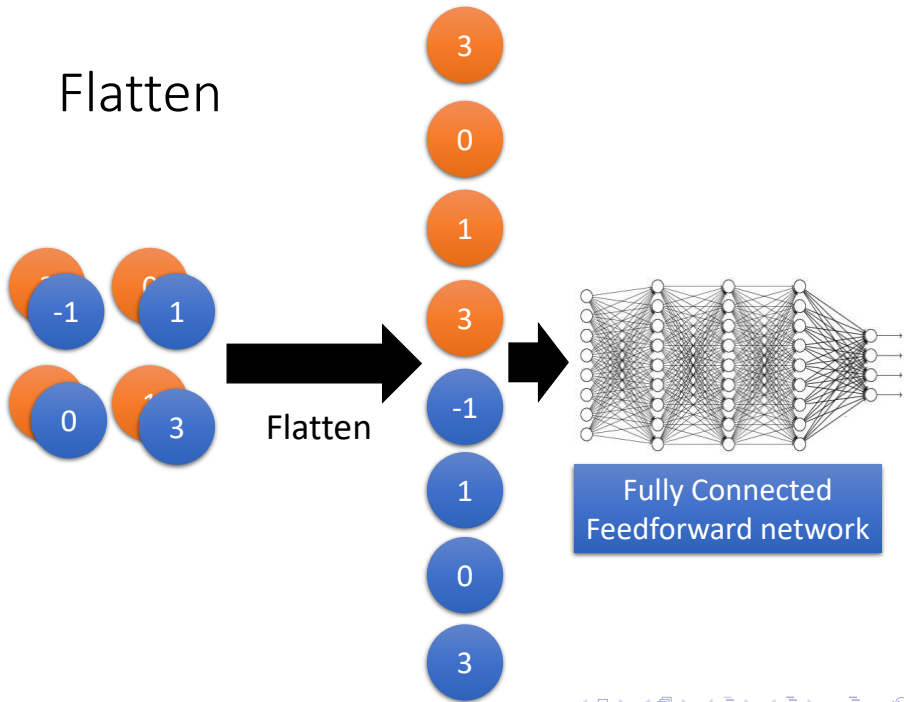
Can repeat many times

The whole CNN

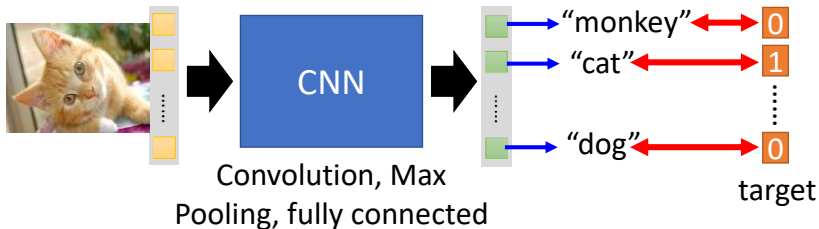
cat dog



Flatten



Convolutional Neural Network



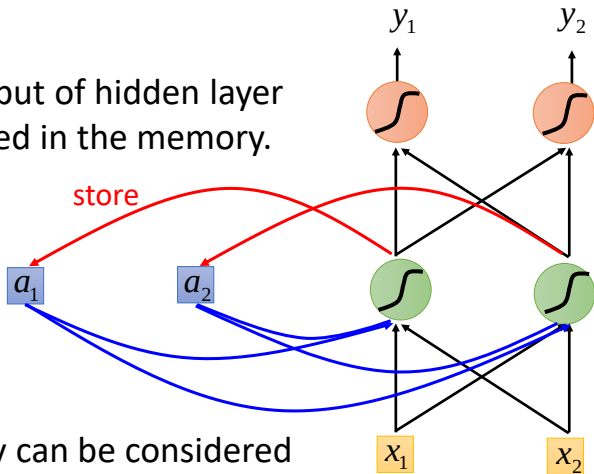
Learning: Nothing special, just gradient descent

循环神经网络

Recurrent Neural Network (RNN)

Recurrent Neural Network (RNN)

The output of hidden layer are stored in the memory.



Memory can be considered as another input.

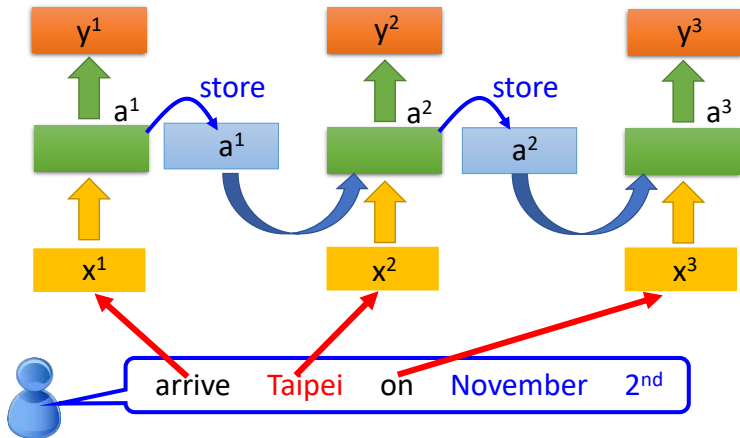
RNN

The same network is used again and again.

Probability of
“arrive” in each slot

Probability of
“**Taipei**” in each slot

Probability of
“on” in each slot



RNN

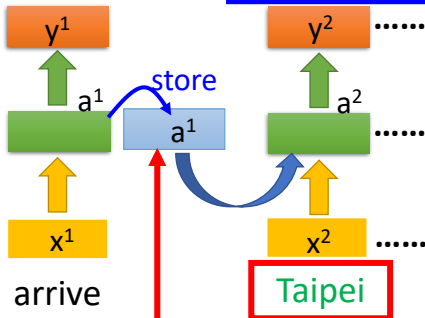
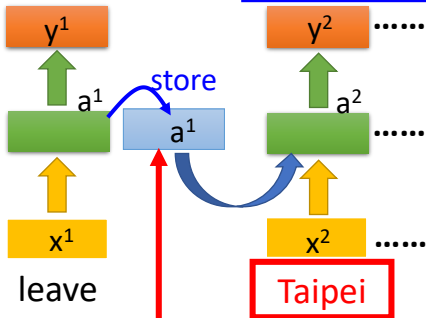
Different

Prob of "leave"
in each slot

Prob of "Taipei"
in each slot

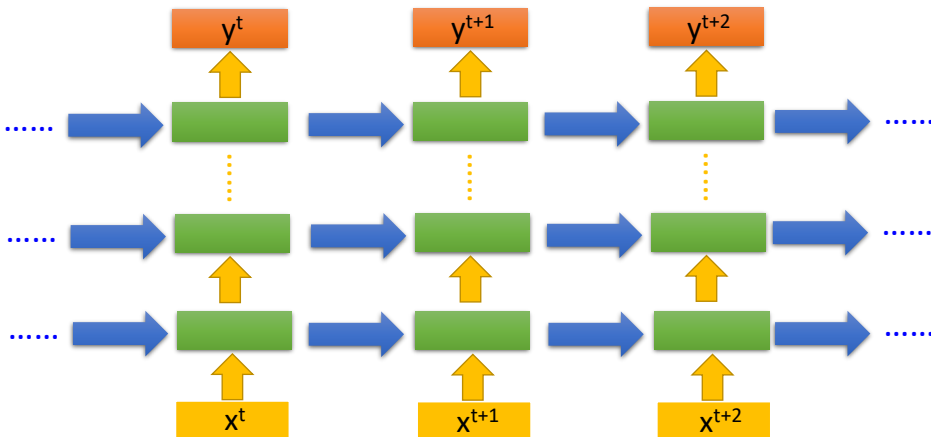
Prob of "arrive"
in each slot

Prob of "Taipei"
in each slot

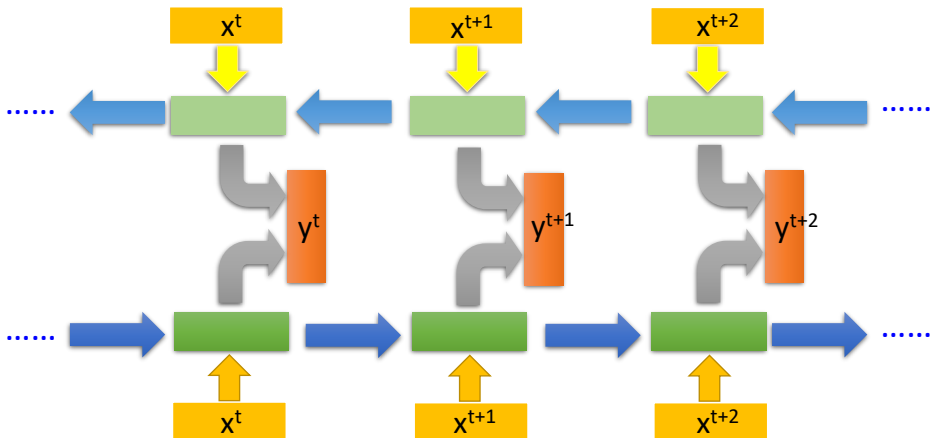


The values stored in the memory is different.

Of course it can be deep ...



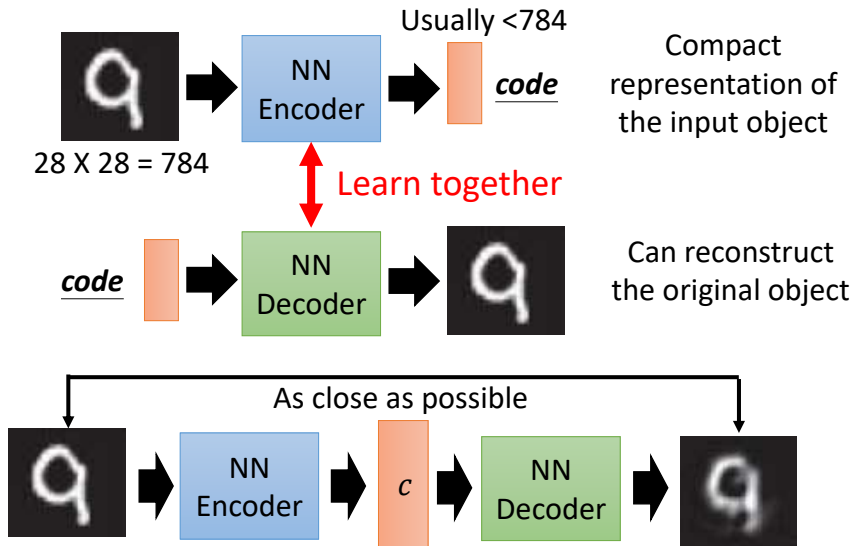
Bidirectional RNN



AutoEncoder (AE)

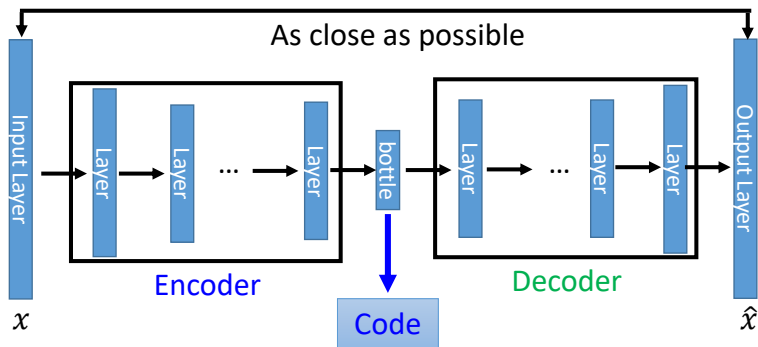
(Unsupervised Learning)

Auto-encoder



Deep Auto-encoder

- NN encoder + NN decoder = a deep network



Reference: Hinton, Geoffrey E., and Ruslan R. Salakhutdinov. "Reducing the dimensionality of data with neural networks." *Science* 313.5786 (2006): 504-507

Deep Auto-encoder

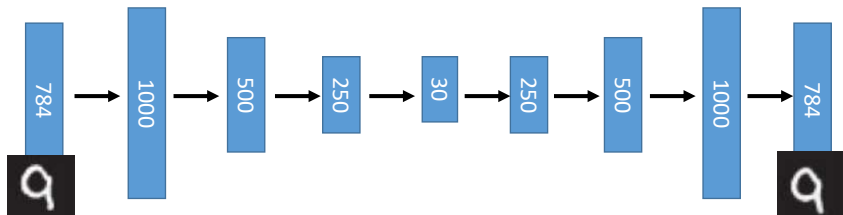
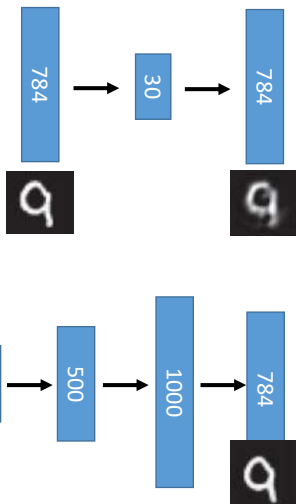
Original Image

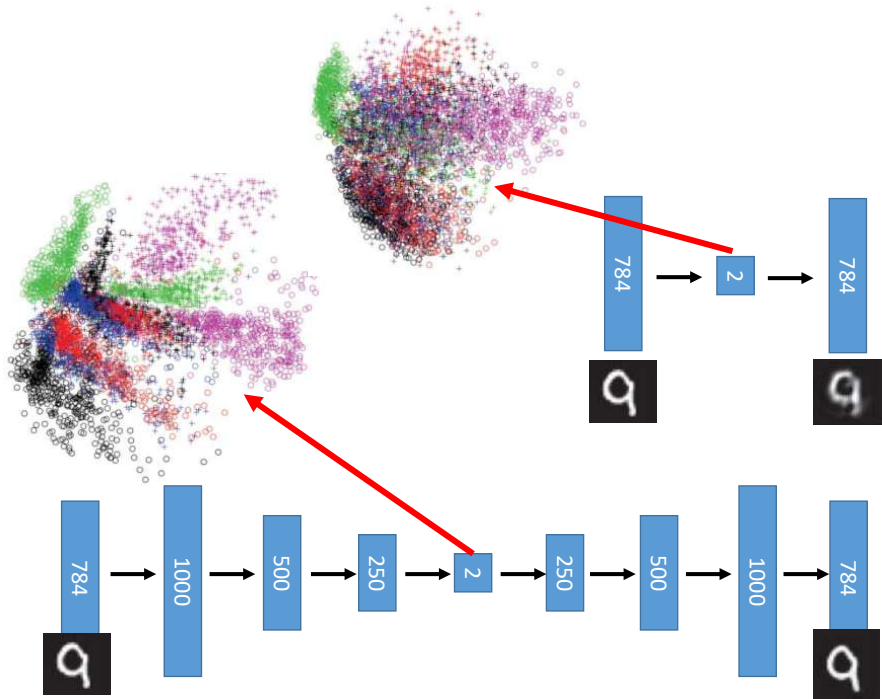


PCA



Deep Auto-encoder

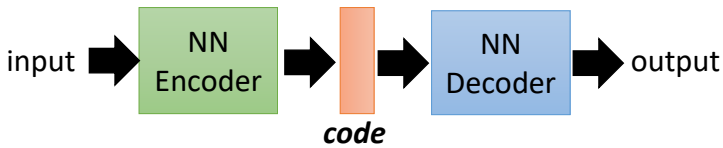




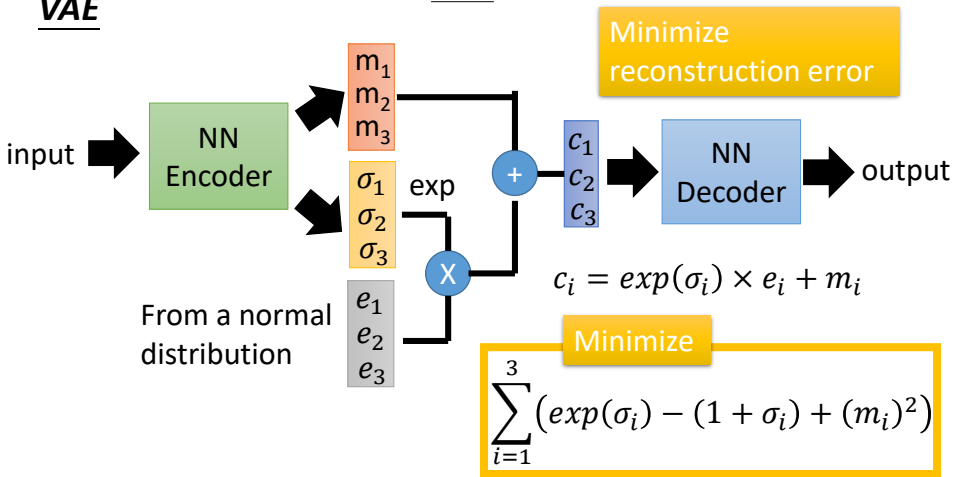
生成-对抗网络

Generative Adversarial Network (GAN)

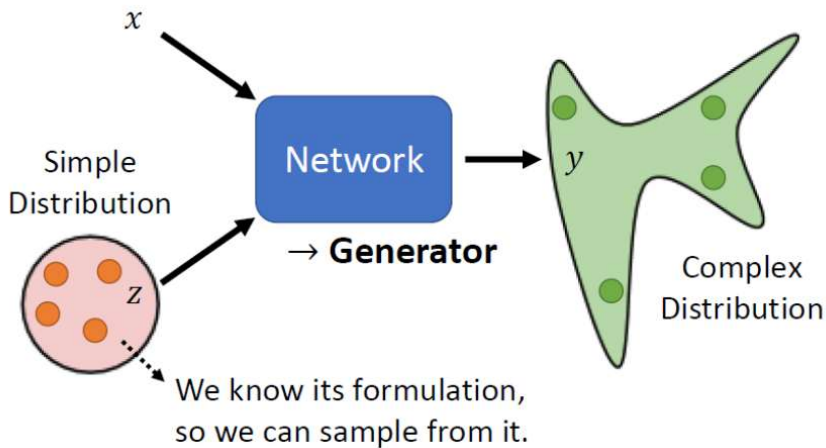
Auto-encoder



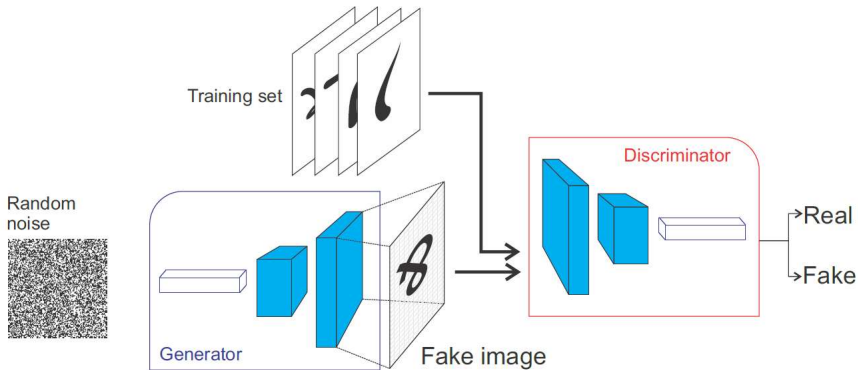
VAE



Network as a Generator



Generative Adversarial Network (GAN)



Images Generated by GAN



Images Generated by GAN



All Kinds of GAN ...

<https://github.com/hindupuravinash/the-gan-zoo>

GAN

ACGAN

BGAN

CGAN

DCGAN

EBGAN

fGAN

GoGAN

⋮

- SeUDA - Semantic-Aware Generative Adversarial Nets for Unsupervised Domain Adaptation
- SG-GAN - Semantic-aware Grad-GAN for Virtual-to-Real Urban Scene Adaption (github)
- SG-GAN - Sparsely Grouped Multi-task Generative Adversarial Networks for Facial Attribute Transfer
- SGAN - Texture Synthesis with Spatial Generative Adversarial Networks
- SGAN - Stacked Generative Adversarial Networks (github)
- SGAN - Steganographic Generative Adversarial Networks
- SGAN - SGAN: An Alternative Training of Generative Adversarial Networks
- SGAN - CT Image Enhancement Using Stacked Generative Adversarial Networks and U-Net
- sGAN - Generative Adversarial Training for MRA Image Synthesis Using Multi-Contrast Self-Supervised Learning
- SiftingGAN - SiftingGAN: Generating and Sifting Labeled Samples to Improve the Renal Lesion Classification Baseline in vitro
- SiGAN - SiGAN: Siamese Generative Adversarial Network for Identity-Preserving Face Image Synthesis
- SimGAN - Learning from Simulated and Unsupervised Images through Adversarial Training
- SisGAN - Semantic Image Synthesis via Adversarial Learning

Mihaela Rosca, Balaji Lakshminarayanan, David Warde-Farley, Shakir Mohamed, "Variational Approaches for Auto-Encoding Generative Adversarial Networks", arXiv, 2017

强化学习

Reinforcement Learning

Scenario of Reinforcement Learning



Scenario of Reinforcement Learning

Agent learns to take actions to maximize expected reward.



Supervised v.s. Reinforcement

- Supervised

Learning from
teacher



"Hello"

Say "Hi"



"Bye bye"

Say "Good bye"

- Reinforcement



.....



.....

.....

Hello 😊

Agent

.....

Agent

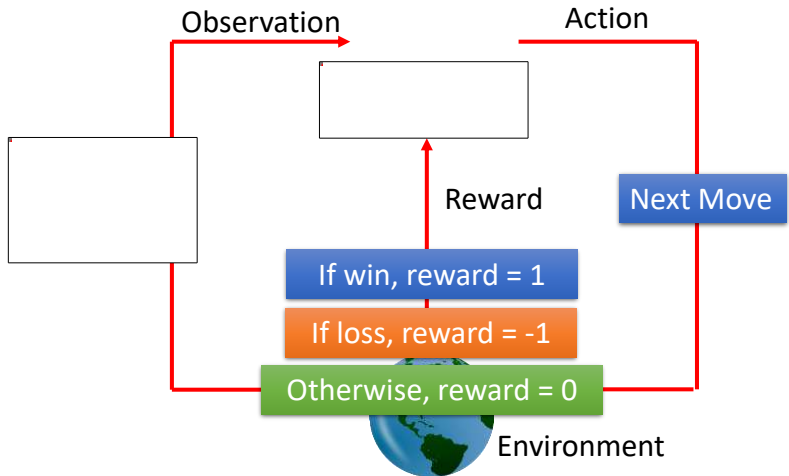


Bad

Learning from
critics

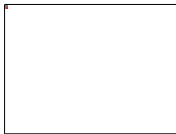
Scenario of Reinforcement Learning

Agent learns to take actions to maximize expected reward.

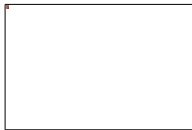


Supervised v.s. Reinforcement

- Supervised:



Next move:
"5-5"



Next move:
"3-3"

- Reinforcement Learning

First move → many moves → Win!

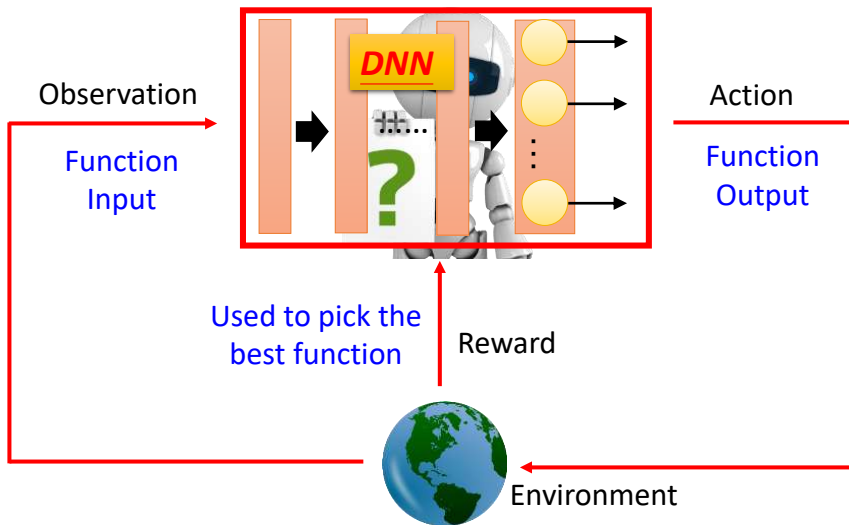
Alpha Go is supervised learning + reinforcement learning.

Difficulties of Reinforcement Learning

- It may be better to sacrifice immediate reward to gain more long-term reward
 - E.g. Playing Go
- Agent's actions affect the subsequent data it receives
 - E.g. Exploration



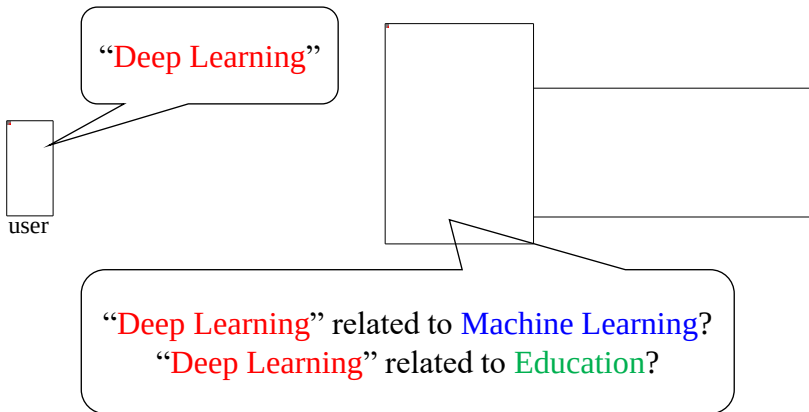
Deep Reinforcement Learning



Application: Interactive Retrieval

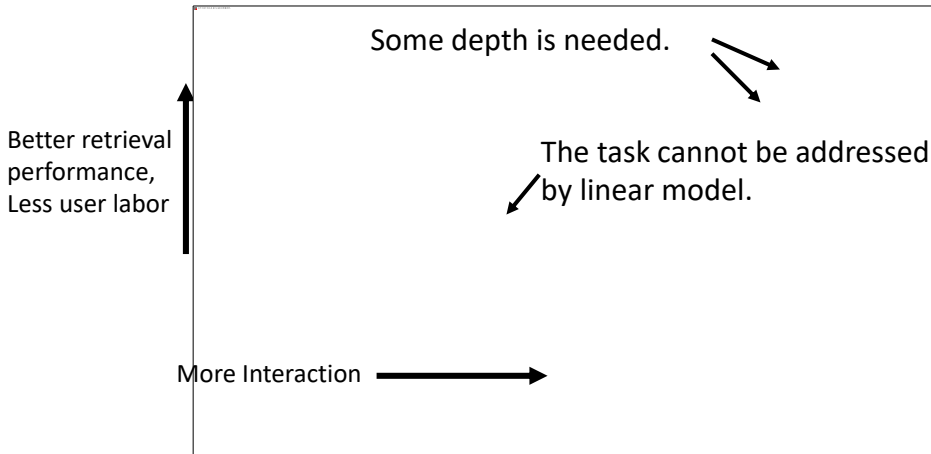
- Interactive retrieval is helpful.

[Wu & Lee, INTERSPEECH 16]



Deep Reinforcement Learning

- Different network depth



More applications

- Alpha Go, Playing Video Games, Dialogue
- Flying Helicopter
 - <https://www.youtube.com/watch?v=0JL04JJjocc>
- Driving
 - <https://www.youtube.com/watch?v=0xo1Ldx3L5Q>
- Google Cuts Its Giant Electricity Bill With DeepMind-Powered AI
 - <http://www.bloomberg.com/news/articles/2016-07-19/google-cuts-its-giant-electricity-bill-with-deepmind-powered-ai>

To learn deep reinforcement learning

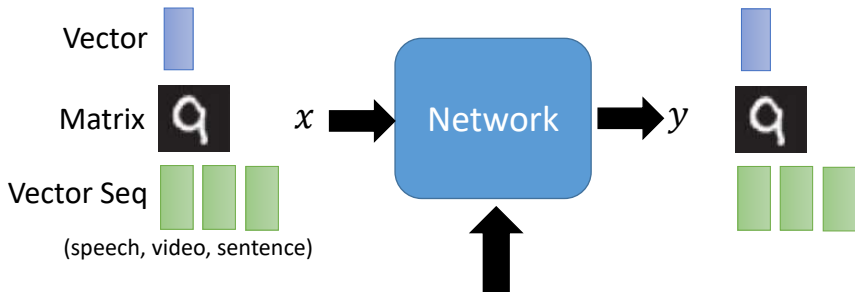
- Lectures of David Silver
 - <http://www0.cs.ucl.ac.uk/staff/D.Silver/web/Teaching.html>
 - 10 lectures (1:30 each)
- Deep Reinforcement Learning
 - http://videolectures.net/rldm2015_silver_reinforcement_learning/

Structured Learning

Deep Learning in One Slide (Review)

- Many kinds of network structures:
- Fully connected feedforward network
 - Convolutional neural network (CNN)
 - Recurrent neural network (RNN)

Different networks can take different kinds of input/output.



How to find
the function?

Given the example inputs/outputs as
training data: $\{(x_1, y_1), (x_2, y_2), \dots, (x_{1000}, y_{1000})\}$

Structured Learning

Machine learning is to find a function f

$$f : X \rightarrow Y$$

Regression: output a scalar

Classification: output a “class” (one-hot vector)

1	0	0
---	---	---

Class 1

0	1	0
---	---	---

Class 2

0	0	1
---	---	---

Class 3

Structured Learning/Prediction: output a sequence, a matrix, a graph, a tree

Output is composed of components with dependency

Output Sequence

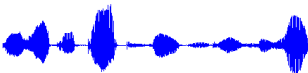
$$f : X \rightarrow Y$$

Machine Translation

X : “機器學習及其深層與結構化”
(sentence of language 1)

Y : “Machine learning and having it deep and structured”
(sentence of language 2)

Speech Recognition

X : 
(speech)

Y : 感謝大家來上課”
(transcription)

Chat-bot

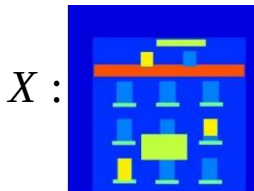
X : “How are you?”
(what a user says)

Y : “I’m fine.”
(response of machine)

Output Matrix

$$f : X \rightarrow Y$$

Image to Image



Colorization:



Ref: <https://arxiv.org/pdf/1611.07004v1.pdf>

Text to Image

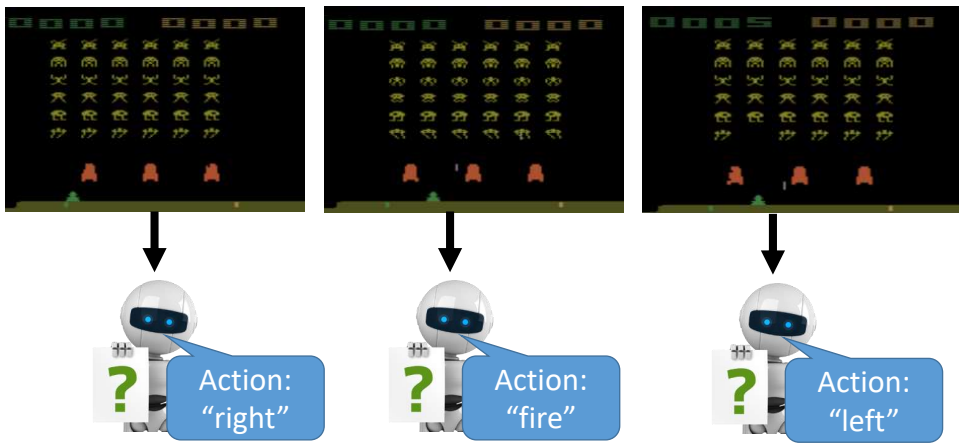
$X :$ “this white and yellow flower have thin white petals and a round yellow stamen”

$Y :$



ref: <https://arxiv.org/pdf/1605.05396.pdf>

Decision Making and Control



GO Playing is the same.

学习资料

- 李宏毅（台湾大学），《机器学习》
 - <https://speech.ee.ntu.edu.tw/~hylee/ml/2021-spring.html>
 - B站视频（李宏毅2021春机器学习课程）：
<https://www.bilibili.com/video/BV1Wv411h7kN?p=1>
- Andrew Ng（Coursera，神经网络与深度学习）
 - <https://www.coursera.org/learn/neural-networks-deep-learning>



中国科学技术大学
University of Science and Technology of China

谢谢！



中国科学技术大学
University of Science and Technology of China

数学建模 Mathematical Modeling

刘利刚
中国科学技术大学

深度学习的理论基础

Mathematical Theory of Deep Learning

Two fundamental approaches of AI

- Rule-based (logical approach, symbolism, expert system, ...)
 - very effective for easy problems
 - interpretable
 - does not require a lot of training data
 - ineffective for hard problems: combinatorial explosion
- Learning-based (particularly **deep learning**)
 - much more accurate
 - not very interpretable
 - requires a lot of training data
 - vulnerable to attacks

高维问题：维数灾难！

高维函数的表达 (神经网络之前)

- Hartree逼近（分离变量法）
 - 局限：过于简单

$$f(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_d) \sim f_1(\mathbf{x}_1)f_2(\mathbf{x}_2) \cdots f_d(\mathbf{x}_d)$$

- 基函数的线性组合
 - 基函数的构造也不简单
 - 张量积函数的维数灾难

$$f(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_d) \sim \sum_{k=1}^m a_k f_k(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_d)$$

神经网络模型

- 从数学上看，本质是**高维函数**的一种表达
 - 数据驱动的方法
 - “学习” → “拟合”
- 优势：**能够表达与逼近任何维数（高维）的函数**、超大规模数据拟合、可优化求解
 - 避免了“维数灾难”！
 - “可微优化” (differential rendering/simulation...)
- 并不是真正意义上的“智能”
 - 能替代人们做不少事情
- 更确切：神经网络计算

Supervised learning: Approximating functions using samples

- Object of interest: (f^*, μ) , where $f^* : \mathbb{R}^d \rightarrow \mathbb{R}^1$, μ is a prob measure on $\mathbb{R}^d$.
- Given a set of samples from μ , $\{x_j\}_{j=1}^n$, and $\{y_j = f^*(x_j)\}_{j=1}^n$
- Task: Approximate f^* using $S = \{(x_j, y_j)\}_{j=1}^n$.
- Strategy: Construct some “hypothesis space” (space of functions) $\mathcal{H}_m$ ($m \sim$ the dimension of $\mathcal{H}_m$).
 - linear regression: $f(x) = \beta \cdot x + \beta_0$
 - generalized linear models: $f(x) = \sum_{k=1}^m c_k \phi_k(x)$, where $\{\phi_k\}$ are linearly independent functions.
 - two-layer neural networks: $f(x) = \sum_k a_k \sigma(b_k \cdot x + c_k)$, where σ is some nonlinear function, e.g. $\sigma(z) = \max(z, 0)$.
 - deep neural networks (DNN) : compositions of functions of the form above.
- Minimize the “empirical risk”:

$$\hat{\mathcal{R}}_n(\theta) = \frac{1}{n} \sum_j (f(x_j) - y_j)^2 = \frac{1}{n} \sum_j (f(x_j) - f^*(x_j))^2$$

Three important aspects to study

- hypothesis space: what kind of functions can be approximated efficiently, generalization gap (= difference between training and testing errors)
- loss function (the variational problem): landscape?
- training: can we optimize? does the solution generalize?

Important parameters:

- m =: number of free parameters
- n =: size of training dataset
- t =: training steps
- d =: dimensionality

typically interested in the case: $m, n, t \rightarrow \infty, d \gg 1$.

Main goal: Error estimates

$$\mathcal{R}(f_{m,n,t}) \lesssim m^{-\alpha} + n^{-\beta} + t^{-\gamma}$$

Want: free of **curse of dimensionality (CoD)**: α, β, γ are independent of d .

Classical numerical analysis (approximation theory)

- Define a “well-posed” math model (the hypothesis space, the loss function, etc)
 - splines: hypothesis space = C^1 piecewise cubic polynomials the data

$$I_n(f) = \frac{1}{n} \sum_{j=1}^n (f(x_j) - y_j)^2 + \lambda \int |D^2 f(x)|^2 dx$$

- finite elements: hypothesis space = C^0 piecewise polynomials
- Identify the right function spaces, e.g. Sobolev/Besov spaces
 - direct and inverse approximation theorem (Bernstein and Jackson type theorems):
 f can be approximated by trig polynomials in L^2 to order s iff $f \in H^s$, $\|f\|_{H^s}^2 = \sum_{k=0}^s \|\nabla^k f\|_{L^2}^2$.
 - functions of interest are in the right spaces (PDE theory, real analysis, etc).
- Optimal error estimates
 - A priori estimates (for piecewise linear finite elements, $\alpha = 1/d, s = 2$)

$$\|f_m - f^*\|_{H^1} \leq C m^{-\alpha} \|f^*\|_{H^s}$$

- A posteriori estimates (say in finite elements):

$$\|f_m - f^*\|_{H^1} \leq C m^{-\alpha} \|f_m\|_h$$

Classical approximation theory: approximating functions by polynomials or piecewise polynomials

Example: Piecewise linear approximation

- d = dimensionality
- m = number of free parameters
- h = size of the mesh: $h \sim m^{-1/d}$

$$\inf_{f_m \in \mathcal{H}_m} \|f^* - f_m\|_{L^2(X)} \lesssim h \|\nabla^2 f^*\|_{L^2(X)}$$

- **Curse of Dimensionality (CoD):** If we want to reduce the error by a factor of 10, we have to increase m by a factor of 10^d .

This is the case for all grid-based or fixed basis-based algorithms.

Classical approximation theory: approximating functions by polynomials or piecewise polynomials

Approximation by piecewise polynomials: (m = number of free parameters)

$$\inf_{f \in \mathcal{H}_m} \|f - f_m\|_{L^2(X)} \leq C_0 h^\alpha \|f\|_{H^\alpha(X)}, \quad h \sim m^{-1/d}$$

- Sobolev (Besov) norm is the right quantity for the right hand side.
- They suffer from CoD: $m \sim \epsilon^{-d}$ where ϵ is the error tolerance.

The number of monomials of degree p in dimension d is C_{p+d}^d

高维积分：Monte Carlo方法

Example: Monte Carlo methods for integration

$$I(g) = \mathbb{E}_{x \sim \mu} g(x), \quad I_m(g) = \frac{1}{m} \sum_j g(x_j)$$

$\{x_j, j \in [m]\}$ is i.i.d samples of μ .

$$\mathbb{E}(I(g) - I_m(g))^2 = \frac{\text{var}(g)}{m}, \quad \text{var}(g) = \mathbb{E}_{x \sim \mu} g^2(x) - (\mathbb{E}_{x \sim \mu} g(x))^2$$

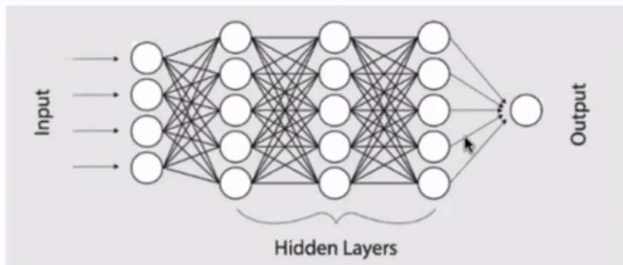
The best we can expect for function approximation in high D:

$$\inf_{f \in \mathcal{H}_m} \mathcal{R}(f) = \inf_{f \in \mathcal{H}_m} \|f - f^*\|_{L^2(d\mu)}^2 \lesssim \frac{\|f^*\|_*^2}{m}$$

- What should be the norm $\|\cdot\|_*$ (associated with the choice of $\mathcal{H}_m$)?

无维数灾难问题！

Multi-layer Neural Networks



$$f(\mathbf{x}, \theta) = \mathbf{W}_L \sigma \circ (\mathbf{W}_{L-1} \sigma \circ (\cdots \sigma \circ (\mathbf{W}_0 \mathbf{x}))), \quad \theta = (\mathbf{W}_0, \mathbf{W}_1, \cdots, \mathbf{W}_L)$$

σ is a scalar function.

" $\circ$ " means acting on each components, the $\mathbf{W}$'s are matrices.

Exploding and vanishing gradients

$$f(\mathbf{x}, \theta) = \mathbf{W}_L \sigma \circ (\mathbf{W}_{L-1} \sigma \circ (\cdots \sigma \circ (\mathbf{W}_0 \mathbf{x}))), \quad \theta = (\mathbf{W}_0, \mathbf{W}_1, \cdots, \mathbf{W}_L)$$

$$\nabla_{\theta} f \approx \mathbf{W}_L \cdot \mathbf{W}_{L-1} \cdots \mathbf{W}_0 \sim \kappa^L, \quad L \gg 1$$

exploding or vanishing gradients problem (see Hanin (2018)).

Solution: Using residual networks (He et al. (2016))

$$\begin{aligned} \mathbf{z}_{0,L}(\mathbf{x}) &= \mathbf{V} \mathbf{x}, \\ \mathbf{z}_{l+1,L}(\mathbf{x}) &= \mathbf{z}_{l,L}(\mathbf{x}) + \frac{1}{L} \mathbf{U}_l \sigma \circ (\mathbf{W}_l \mathbf{z}_{l,L}(\mathbf{x})), \quad l = 0, 1, \cdots, L-1 \end{aligned}$$

$$f(\mathbf{x}, \theta) = \alpha \cdot \mathbf{z}_{L,L}(\mathbf{x})$$

Approximating functions using two-layer NN: An Illustrative Example

Traditional approach for Fourier transform:

$$f(x) = \int_{\mathbb{R}^d} a(\omega) e^{i(\omega, x)} d\omega, \quad f_m(x) = \frac{1}{m} \sum_j a(\omega_j) e^{i(\omega_j, x)}$$

$\{\omega_j\}$ is a fixed grid, e.g. uniform.

$$\|f - f_m\|_{L^2(X)} \leq C_0 m^{-\alpha/d} \|f\|_{H^\alpha(X)}$$

"New" approach: Let π be a probability distribution and

$$f(x) = \int_{\mathbb{R}^d} a(\omega) e^{i(\omega, x)} \pi(d\omega) = \mathbb{E}_{\omega \sim \pi} a(\omega) e^{i(\omega, x)}$$

Let $\{\omega_j\}$ be an i.i.d. sample of π , $f_m(x) = \frac{1}{m} \sum_{j=1}^m a(\omega_j) e^{i(\omega_j, x)}$,

$$\mathbb{E}|f(x) - f_m(x)|^2 = m^{-1} \text{var}(f) \quad \text{无维数灾难问题!}$$

$f_m(x) = \frac{1}{m} \sum_{j=1}^m a_j \sigma(\omega_j^T x) =$ two-layer neural network with activation function $\sigma(z) = e^{iz}$.

Two-layer neural network model: Barron spaces

E, Ma and Wu (2018, 2019), Bach (2017)

$$\mathcal{H}_m = \{f_m(x) = \frac{1}{m} \sum_j a_j \sigma(w_j^T x)\}, \theta = \{(a_j, w_j), j \in [m]\}$$

Consider the function $f : X = [0, 1]^d \mapsto \mathbb{R}$ of the following form

$$f(x) = \int_{\Omega} a \sigma(w^T x) \rho(da, dw) = \mathbb{E}_{(a,w) \sim \rho}[a \sigma(w^T x)], \quad x \in X$$

— 概率密度函数

$\Omega = \mathbb{R}^1 \times \mathbb{R}^{d+1}$, ρ is a probability distribution on Ω .

$$\|f\|_{\mathcal{B}} = \inf_{\rho \in P_f} (\mathbb{E}_{\rho}[a^2 \|w\|_1^2])^{1/2}$$

where $P_f := \{\rho : f(x) = \mathbb{E}_{\rho}[a \sigma(w^T x)]\}$.

$$\mathcal{B} = \{f \in C^0 : \|f\|_{\mathcal{B}} < \infty\}$$

Related work in Barron (1993), Klusowski and Barron (2016), E and Wojtowytsch (2020)

What kind of functions admit such a representation?

Theorem (Barron and Klusowski (2016)): If $\int_{\mathbb{R}^d} \|\omega\|_1^2 |\hat{f}(\omega)| d\omega < \infty$, where $\hat{f}$ is the Fourier transform of f , then f can be represented as

$$\tilde{f}(x) = f(x) - (f(0) + x \cdot \nabla f(0)) = \int_{\Omega} a \sigma(b^T x + c) \rho(da, db, dc)$$

where $\sigma(x) = \max(0, x)$. Moreover $f \in \mathcal{B}_{\infty}$. Furthermore, we have

$$\|\tilde{f}\|_{\mathcal{B}_{\infty}} \leq 2 \int_{\mathbb{R}^d} \|\omega\|_1^2 |\hat{f}(\omega)| d\omega$$

- $\Gamma(f) = \int_{\mathbb{R}^d} \|\omega\|_1^2 |\hat{f}(\omega)| d\omega$ is Barron's spectral norm. This condition depends on dimensionality.
- Let $f_1(x) = \|\mathbf{x}\|$, $f_2(x) = \text{dist}(x, \mathcal{S}^{d-1})$, then f_1 is in $\mathcal{B}$, f_2 is not in $\mathcal{B}$.

关于Barron函数的逼近论定理

Theorem (Direct Approximation Theorem)

There exists an absolute constant C_0 such that

$$\|f - f_m\|_{L^2(X)} \leq \frac{C_0 \|f\|_{\mathcal{B}_2}}{\sqrt{m}}$$

Theorem (Inverse Approximation Theorem)

For $p > 1$, let

$$\mathcal{N}_{p,C} = \left\{ \frac{1}{m} \sum_{k=1}^m a_k \sigma(b_k^T \mathbf{x} + c_k) : \left(\frac{1}{m} \sum_{k=1}^m |a_k|^p (\|b_k\|_1 + c_k)^p \right)^{1/p} \leq C, m \in \mathbb{N}^+ \right\}.$$

Let f^* be a continuous function. Assume there exists a constant C and a sequence of functions $f_m \in \mathcal{N}_{p,C}$ such that

$$f_m(\mathbf{x}) \rightarrow f^*(\mathbf{x})$$

for all $\mathbf{x} \in X$. Then there exists a probability distribution ρ on Ω , such that

$$f^*(\mathbf{x}) = \int a \sigma(\mathbf{b}^T \mathbf{x} + c) \rho(da, db, dc),$$

for all $\mathbf{x} \in X$.

Estimation of error: overfitting

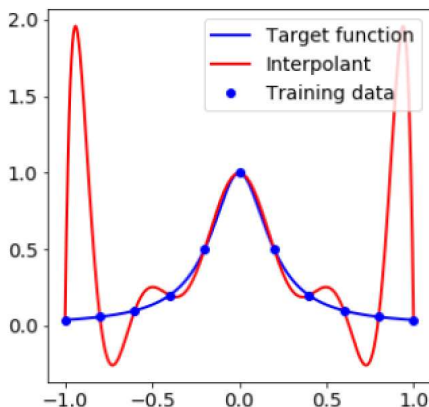


Figure: The Runge phenomenon: $f^*(x) = \frac{1}{1+25x^2}$

Training and testing errors

In practice, we minimize the training error:

$$\hat{\mathcal{R}}_n(\theta) = \frac{1}{n} \sum_j (f(x_j, \theta) - f^*(x_j))^2$$

but we are interested in the testing error:

$$\mathcal{R}(\theta) = \mathbb{E}_{x \sim \mu} (f(x, \theta) - f^*(x))^2$$

$\mathcal{H}$ = a set of functions, $S = (x_1, x_2, \dots, x_n)$ = dataset. Upto log terms,

$$\sup_{h \in \mathcal{H}} \left| \mathbb{E}_x [h(x)] - \frac{1}{n} \sum_{i=1}^n h(x_i) \right| \sim \text{Rad}_S(\mathcal{H})$$

where the **Rademacher complexity** of $\mathcal{H}$ with respect to S is defined as

$$\text{Rad}_S(\mathcal{H}) = \frac{1}{n} \mathbb{E}_{\xi} \left[\sup_{h \in \mathcal{H}} \sum_{i=1}^n \xi_i h(x_i) \right],$$

where $\{\xi_i\}_{i=1}^n$ are i.i.d. random variables taking values ± 1 with equal probability.

Complexity Estimates

Theorem (Bach, 2017)

Let $\mathcal{F}_Q = \{f \in \mathcal{B}, \|f\|_{\mathcal{B}} \leq Q\}$. Then we have

$$\text{Rad}_S(\mathcal{F}_Q) \leq 2Q \sqrt{\frac{2 \ln(2d)}{n}}$$

where $n = |S|$, the size of the dataset S .

In contrast, if $\mathcal{F} = \{f \in C^0, \|f\|_{\text{Lip}} \leq 1\}$, then

$$\text{Rad}_S(\mathcal{F}) \leq \frac{C_d}{n^{1/d}} \quad \mathbb{I}$$

This is another form of CoD: the size of the dataset needed.

A priori estimates for regularized model

$$L_n(\theta) = \hat{\mathcal{R}}_n(\theta) + \lambda \sqrt{\frac{\log(2d)}{n}} \|\theta\|_{\mathcal{P}}, \quad \hat{\theta}_n = \operatorname{argmin} L_n(\theta)$$

where the path norm is defined by:

$$\|\theta\|_{\mathcal{P}} = \frac{1}{m} \sum_{k=1}^m |a_k| (\|\mathbf{b}_k\|_1 + |c_k|) \quad (= \|f(\cdot; \theta)\|_{\mathcal{B}_1})$$

Theorem (Weinan E, Chao Ma, Lei Wu, submitted)

Assume that the target function $f^ : [0, 1]^d \mapsto [0, 1] \in \mathcal{B}_2$. There exist constants C_0, C_1, C_2 , such that for any $\delta > 0$, if $\lambda \geq C_0$, then with probability at least $1 - \delta$ over the choice of training set, we have*

$$\mathcal{R}(\hat{\theta}_n) \leq C_1 \left(\frac{\|f^*\|_{\mathcal{B}_2}^2}{m} + \|f^*\|_{\mathcal{B}_2} \sqrt{\frac{\log(2d)}{n}} \right) + C_2 \sqrt{\frac{\log(4C_2/\delta) + \log(n)}{n}}.$$

Approximation theory and function spaces for other ML models

- random feature model: Reproducing kernel Hilbert space (RKHS)
- Residual networks (ResNets): Flow-induced space (E, Ma and Wu (2019))
- Multi-layer neural networks: Multi-layer spaces (E and Wojtowysch (2020))

Up to log terms, we have

$$\mathcal{R}(\hat{f}) \lesssim \frac{\|f^*\|_*^2}{m} + \frac{\|f^*\|_*}{\sqrt{n}}$$

where m = number of free parameters, n = size of training dataset.

The error constant (the “norm”) decreases as the depth increases. This partly explains why deeper neural networks tend to perform better (although due to other reasons, they are also harder to train in general).

Summary

- One key reason behind the success of deep neural networks is that they provide an effective tool for approximating functions in **high dimension** for **numerical analysis**.
 - Theoretical foundation of machine learning = high dimensional numerical analysis
- We can then use deep neural networks to solve many problems in high dimension.
- A reasonable mathematical picture for deep learning is emerging, from the perspective of numerical analysis.
- The key to this understanding is CoD: deep neural networks are the right replacement of polynomials in high dimensions.

What we don't know?

- Why does GD/SGD converge to good minima?
- Other general representations of functions?
 - It has to be some kind of expectation.
 - Each such representation may lead to a new kind of ML model.
- Note: In a way, neural network models are also very natural: They arise naturally when considering continuous formulations in high dimensions.

应用数学迎来了新的时代

- 机器学习的基础本质上是应用数学问题
- 应用数学能够帮助解决实际（高维）问题了
- 纯数学：
 - 代数、分析、几何拓扑
- 应用数学：
 - 算法、物理模型、机器学习（数据科学）

基础数学：高维分析

- 监督学习：高维函数
- 生成模型：高维概率分布
- 时间序列：高维动力系统
- 强化学习：高维微分和差分方程

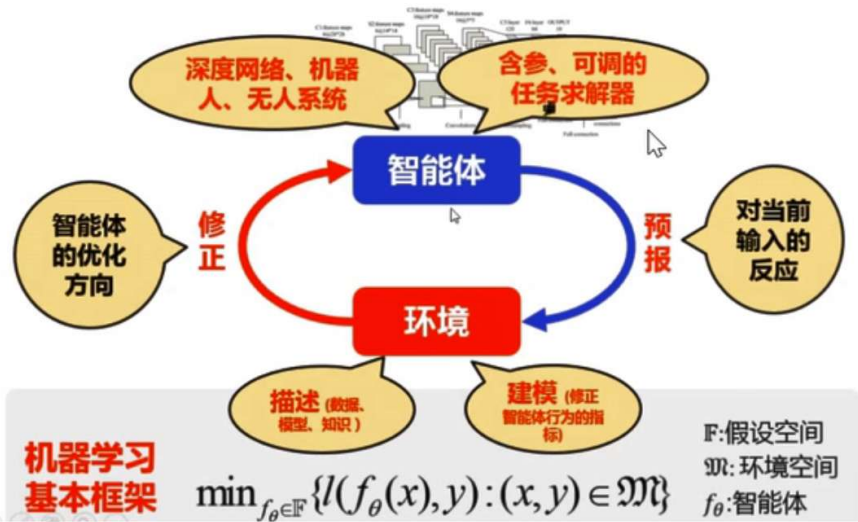
数学的新局面

- 三次重大里程碑
 - 牛顿时代：数学成为自然科学的语言
 - Von Neumann时代：科学计算成为通往应用的主要桥梁
 - 当今时代：作为人工智能的基础，数学走到科学和技术创新的最前沿
- 计算数学与应用数学解决实际高维问题！

机器学习

Machine Learning

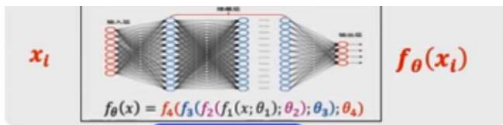
机器学习的范式



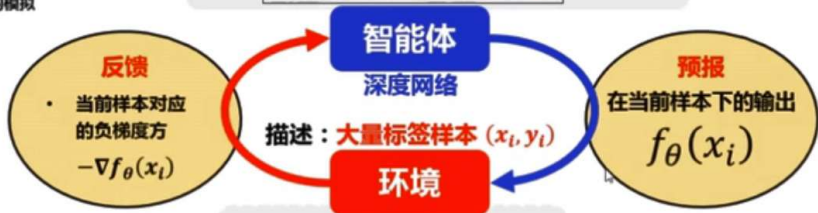
模型1：深度学习

深度学习

- 智能体是由多个非线性变换复合而成的深度网络
- 用于对数据生成机制、映照关系的模拟



通过反馈
修正参数



主要关注 **网络拓扑、学习算法、收敛性和泛化性** 等基本问题。

$$loss_{\theta} = \sum_{i=1}^m l(f_{\theta}(x_i), y_i)$$

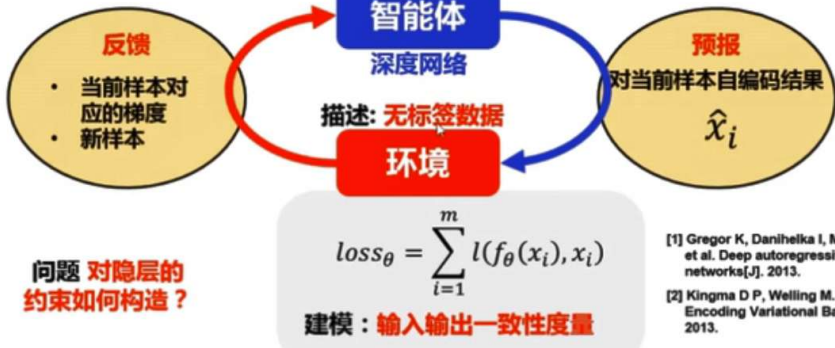
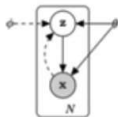
环境建模: **损失函数**

- [1] Krizhevsky A, et al. Imagenet classification with deep convolutional neural networks[C] 2012: 1097-1105
- [2] He K, et al. Deep Residual Learning for Image Recognition[J]. 2015
- [3] Jia Y, et al. Caffe: Convolutional Architecture for Fast Feature Embedding[J]. 2014

模型2：自编码器

自编码器

利用一个智能体对数据进行自编码
(特征提取、降维)



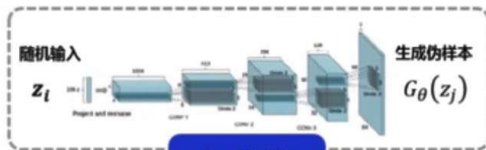
[1] Gregor K, Danihelka I, Mnih A, et al. Deep autoregressive networks[J]. 2013.

[2] Kingma D P, Welling M. Auto-Encoding Variational Bayes[J]. 2013.

模型3：生成网络

生成网络

- 目标：学习数据生成机制,生成样本
- 特征：生成器和判别器对抗学习



[1] Ian J. Goodfellow, et. al. Generative Adversarial Networks : Cornell University Library , 2014

[2] Radford A , et al. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks[J]. Computer Science, 2015.



学习判别器 D_{α^*} :

$$\max_{\alpha} \sum_{i=1}^m \log D_{\alpha}(x_i) + \sum_{j=1}^m \log (1 - D_{\alpha}(G_{\theta}(z_j)))$$

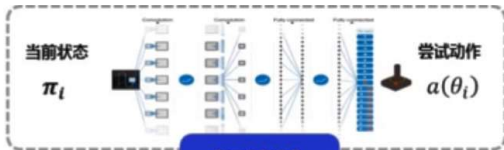
由判别器 D_{α^*} 定义损失函数： $f_{\theta}(z_i) = \log (1 - D_{\alpha^*}(G_{\theta}(z_j)))$

由于没有标签数据，环境建模为一个网络，以学习一个能区分真实样本与生成样本的判别函数 D_{α}

模型4：强化学习

强化学习

一个智能体通过与环境的交互不断提升自身行为的过程



- [1] Silver D, et al. Mastering the game of Go without human knowledge[J]. Nature, 2017, 550(7676):354-359.
- [2] Human-level control through deep reinforcement learning[J]. Nature, 2015, 518(7540):629-633.



强化学习通常表现为一个过程。



建模：据Bellman最优原理定义一个可行目标函数

$$f_{\theta} = E_{(s,a,r,s') \sim U(D)}[(r + \gamma \max Q(s, a(\theta_i)))]$$

为智能体设计一套策略(产生动作序列 $\tau = \{a_t, s_t\}_{t=1}^T$) 使得该序列所对应的整体奖励最大化

机器学习的数学框架

机器学习

建模“智能体通过与环境的交互、自省，提升自身能力”这一智能。形式化：给定一个环境，从一个预设的函数族内找一个最优的映射函数 f_{θ}^* ，使得这一映射能实现在给定的环境中最优地完成任务。

典型
应用

- 逼近
- 分类（识别）
- 优化
- 降维（特征提取）

数学 框架

函数空间形式

$$\min_{f_{\theta} \in \mathfrak{F}} \{l(f_{\theta}(x), y) : (x, y) \in \mathfrak{M}\} + \lambda R(\|f_{\theta}\|)$$

$\mathfrak{F}$ ：假设空间

$\mathfrak{M}$ ：环境空间

f_{θ} ：智能体

参数空间形式

$$\min_{\theta} \mathbb{E}_{(x,y)} \{l(f_{\theta}(x), y)\} + \lambda R(\theta)$$

机器学习的数学模型

Machine Learning: Problem and Formulation

Given a dataset $\mathcal{D} = \{x_i, y_i\}$, find the possibly existed mapping $f^*: x \rightarrow y$ to fit $\mathcal{D}$. To formulate the problem, choose a hypothetical space $\mathcal{F}$, a loss function l , and a regularizer p , we find the approximate mapping f_{θ^*} such that


$$f_{\theta^*} = \operatorname{argmin}_{f_{\theta} \in \mathcal{F}} \{ \mathbb{E}_{\mathcal{D}} [l(f_{\theta}(x), y)] + \lambda p(f_{\theta}) \}$$

where f_{θ} : A function in $\mathcal{F}$ with parameter θ

激活 W
神经网络

机器学习的数学模型：数据

Machine Learning: Problem and Formulation

hypothetical
space

loss
fun.

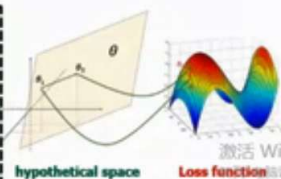
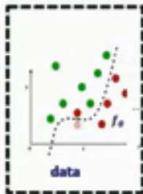
data

regularizer

$$f_{\theta^*} = \operatorname{argmin}_{f_{\theta} \in \mathcal{F}} \{ \mathbb{E}_{\mathcal{D}} [l(f_{\theta}(x), y)] + \lambda p(f_{\theta}) \}$$

Data: Pairwisely collected, corresponding to supervised learning mode (otherwise, unsupervised). A pair (x_i, y_i) represents a sample consisted of an input-output observation.

- $y_i \in \mathbb{R}^1$ normally is label
- $x_i = [x_{i1}, x_{i2}, \dots, x_{im}] \in \mathbb{R}^m$, with each x_{ij} being j -th feature of x_i



机器学习的数学模型：假设空间

Machine Learning: Problem and Formulation

hypothetical
space

loss
fun.

data

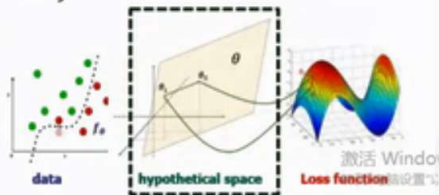
regularizer

$$f_{\theta^*} = \operatorname{argmin}_{f_{\theta} \in \mathfrak{F}} \{ \mathbb{E}_{\mathcal{D}} [l(f_{\theta}(x), y)] + \lambda p(f_{\theta}) \}$$

Hypothetical space $\mathfrak{F}$: A parameterized family of functions,
 $\mathfrak{F} = \{f_{\theta}(x); \theta \in \Omega \subset \mathbb{R}^n\}$

$f_{\theta}(x)$ may be:

- Linear functions
- Polynomial functions
- Deep neural networks
- ...



机器学习的数学模型：损失函数

Machine Learning: Problem and Formulation

hypothetical
space

loss
fun.

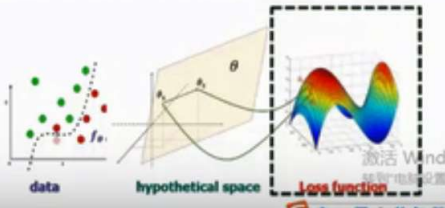
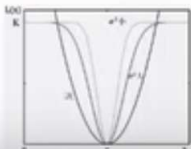
data

regularizer

$$f_{\theta^*} = \operatorname{argmin}_{f_{\theta} \in \mathcal{H}} \{ \mathbb{E}_{\mathcal{D}} [l(f_{\theta}(x), y)] + \lambda p(f_{\theta}) \}$$

Loss function l : A measure quantifies the amount of the prediction deviates from the actual values. It determines the sense of optimality with which the solution finds.

- Lest square loss
- Cross entropy
- L_p loss
- Hinge loss
- ...



机器学习的数学模型：正则约束

Machine Learning: Problem and Formulation

hypothetical
space

loss
fun.

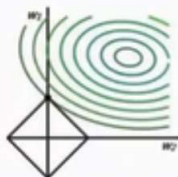
data

regularizer

$$f_{\theta^*} = \operatorname{argmin}_{f_{\theta} \in \mathcal{H}} \{ \mathbb{E}_{\mathcal{D}} [l(f_{\theta}(x), y)] + \lambda p(f_{\theta}) \}$$

Regularizer p : A term to regularize the property of solution to be found, which is determined principally by the prior of problem.

- Lest square regularizer (Tikhonov regularization)
- L_1 regularizer (Lasso)
- TV norm regularizer
- ...



(a) L_1 Regularization



(b) L_2 Regularization

机器学习的假设1

Hypothesis I

The Independence Hypothesis on loss function

The loss function l is preset before implementation, independent of the data (the data distribution) in problem (**application-independent setting**).

Typical preassumption:

Least square loss:

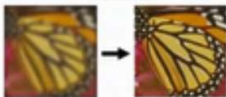
$$L(x, y) = \|x - y\|_F^2$$

Processing Task:

- Denoising
- Derain
- Deblur
- Super resolution



Derain



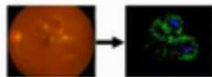
Deblur

Cross entropy :

$$L(x, y) = - \sum_i x_i \log(y_i)$$

Analysis Task:

- Classification
- Segmentation
-



Segmentation



Classification

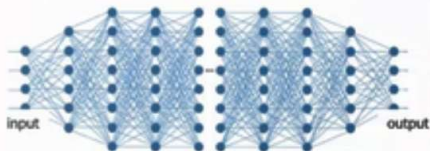
机器学习的假设2

Hypothesis II

The Large Capacity Hypothesis on hypothetical space

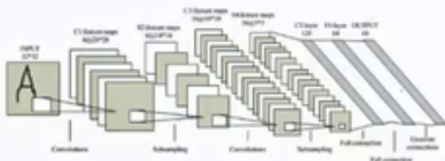
The hypothetical space $\mathcal{H}$ should be large enough in capacity to embody the optimal solution. It is preset independent of the problem.

Typical preassumption:



$$f_{\theta} \in \mathcal{H} = \{h(w_n \cdot (\dots w_2 \cdot h(w_1 \cdot x))) \mid \theta = (w_1, w_2, \dots, w_n)\}$$

Full connected neural network
(assuming **sufficient layers**)



$$f_{\theta} \in \mathcal{H} = \{h(c_n \otimes (\dots c_2 \otimes h(c_1 \otimes x))) \mid \theta = (c_1, c_2, \dots, c_n)\}$$

Convolutional network
(assuming **sufficient layers**)

机器学习的假设3

Hypothesis III

The Completeness Hypothesis of training data

Samples in training data should be well labeled, low level noised, class balanced, sufficient number (**high quality labeled**).

Typical preassumption:



Well labeled data



Enough number of samples

机器学习的假设4

Hypothesis IV

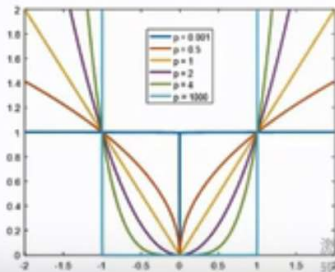
The Regularizer Hypothesis

Regularizer p is fixed and preset by a prior, while only the parameter λ is adjusted.

$$f_{\theta^*} = \operatorname{argmin}_{f_{\theta} \in \mathcal{F}} \{ \mathbb{E}_{\mathcal{D}} [l(f_{\theta}(x), y)] + \lambda p(f_{\theta}) \}$$

Typical preassumption:

- Lest square regularizer (Tikhonov regularization)
- Sparsity promotion
- TV norm regularizer
- L(P) norms
- ...



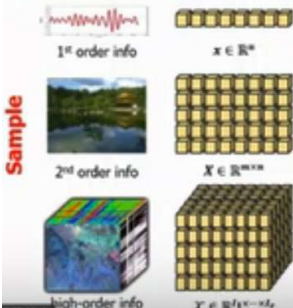
机器学习的假设5

Hypothesis V

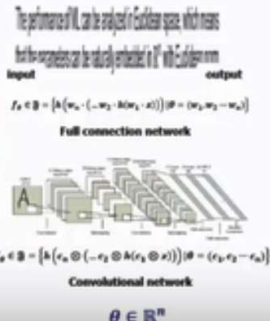
The Euclidean hypothesis on analysis

The performance of ML can be analyzed in Euclidean space, which means that the parameters can be naturally embedded in $\mathbb{R}^n$ with Euclidean norm.

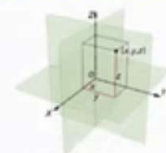
Typical preassumption:



Parameter



Euclidean Space



- $\exists g(\cdot, \cdot), s.t., \forall x, y, z \in V$
- $g(x, y) = g(y, x)$
 - $g(x + y, z) = g(x, z) + g(y, z)$
 - $g(kx, y) = kg(x, y)$
 - $g(x, x) \geq 0$
 - $g(x, x) = 0 \Leftrightarrow x = 0$

机器学习在数学的应用

利用AI求解PDE问题(1)

PDE的一般形式：

$$\begin{aligned}\frac{\partial u}{\partial t}(t, \mathbf{x}) + \mathcal{L}u(t, \mathbf{x}) &= 0, (t, \mathbf{x}) \in [0, T] \times \Omega, \\ u(t = 0, \mathbf{x}) &= u_0(\mathbf{x}), \\ u(t, \mathbf{x}) &= g(t, \mathbf{x}), \mathbf{x} \in \partial\Omega.\end{aligned}$$

传统数值方法：

- 有限差分方法
- 有限元法
- 有限体方法

传统数值方法基本原理：

- 对原定解问题或其等价形式**离散化**，并归结为一个**线性代数方程组**，最终在计算机上求得精确解在离散网格点上的近似值（微分的近似）

传统数值方法的局限：

- 对维数、阶数依赖
- 受限于复杂边界条件

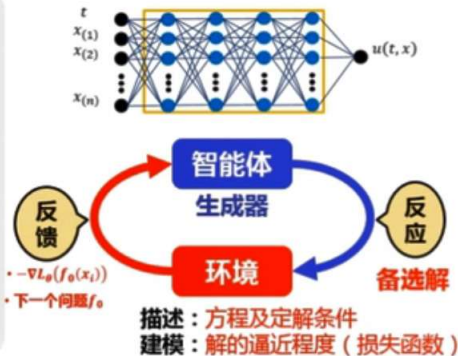
人工智能方法：

- 利用对空间的采样代替离散化，（积分的近似）

利用AI求解PDE问题(2)

基本思想：

1. 利用神经网络（**生成器**）近似微分方程的解，网络的输入为微分方程解的参数（时间和空间），输出为微分方程解的函数值（**备选解**）；
2. 利用微分方程及边界条件（**环境**）设计损失函数（**评价**）；
3. 从参数空间中采样输入网络，调整网络权重以使网络输出极小化（**反馈**），使得网络逼近微分方程的解。



步1、由神经网络 $N(t, x; w)$ 生成近似解 $u(t, x)$:

- [1] Sirignano, Justin, and Konstantinos Spiliopoulos. "DGM: A deep learning algorithm for solving partial differential equations." *Journal of Computational Physics* 375 (2018): 1339-1364.
[2] Lagaris I E, Likas A, Fotiadis D I. Artificial neural networks for solving ordinary and partial differential equations. *IEEE Transactions on Neural Networks* 9(5)(1998): 987-1000

$$\begin{aligned}\frac{\partial u}{\partial t}(t, x) + \mathcal{L}u(t, x) &= 0 \\ u(t = 0, x) &= u_0(x), \\ u(t, x) &= g(t, x), x \in \partial\Omega.\end{aligned}$$

I. Lagaris et al. Artificial neural networks for solving ordinary and partial differential equations. 1997.

<https://arxiv.org/abs/physics/9705023>

J. Sirignano, K. Spiliopoulos. DGM: A deep learning algorithm for solving partial differential equations. 2018

<https://arxiv.org/abs/1708.07469>

利用AI求解PDE问题(3)

基本思想：

1. 利用神经网络（**生成器**）近似微分方程的解，网络的输入为微分方程解的参数（时间和空间），输出为微分方程解的函数值（**备选解**）；
2. 利用微分方程及边界条件（**环境**）设计损失函数（**评价**）；
3. 从参数空间中采样输入网络，调整网络权重以使网络输出极小化损失函数（**反馈**），使得网络逼近微分方程的解。



步2. 由方程及定解条件定义损失函数：

$$J(t, x; w) = \left\| \frac{\partial N}{\partial t}(t, x; w) + \mathcal{L}N(t, x; w) \right\|_{[0, T] \times \Omega, v_1}^2 + \|N(t, x; w) - g(t, x)\|_{\partial\Omega, v_2}^2 + \|N(0, x; w) - u_0(x)\|_{\Omega, v_3}^2$$

利用AI求解PDE问题(4)

基本思想：

1. 利用神经网络（**生成器**）近似微分方程的解，网络的输入为微分方程解的参数（时间和空间），输出为微分方程解的函数值（**备选解**）；
2. 利用微分方程及边界条件（**环境**）设计损失函数（**评价**）；
3. 从参数空间中采样输入网络，调整网络权重以使网络输出极小化损失函数（**反馈**），使得网络逼近微分方程的解。



步3、采样近似及权重调整：

$$\hat{J}(s_n, w) = \frac{1}{n} \sum_{i=1}^n \left(\frac{\partial N}{\partial t}(t_i, x_i; w) + \mathcal{L}N(t_i, x_i; w) \right)^2 + \frac{1}{n} \sum_{i=1}^n (N(\tau_i, z_i; w) - g(\tau_i, z_i))^2 + \frac{1}{n} \sum_{i=1}^n (N(0, y_i; w) - u_0(y_i))^2$$

其中 $s_n = \{(t_i, x_i), (\tau_i, z_i), y_i\}_{i=1}^n$ 为从 v_1, v_2, v_3 中采样所得样本。 $w \leftarrow \arg \min_w \hat{J}(s_n, w)$

数值方法驱动的DL方法(1)

基本思想：

1. 将时间离散化，利用神经网络近似数值方法（如欧拉方法等）中难以计算的部分；
2. 数值方法诱导深度神经网络结构；
3. 设计损失函数训练神经网络，使其逼近微分方程的解。

1. 网络近似数值方法中难以计算的项：

考虑微分方程

$$\frac{\partial u}{\partial t}(t, x) + \mathcal{L}u(t, x) = 0, (t, x)$$

$$\in [0, T] \times \Omega$$

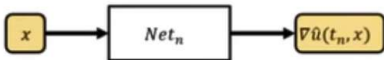
$$u(T, x) = g(x)$$

欧拉方法为

$$u_{n+1} = u_n + \Delta t(-\mathcal{L}u(t_n, x))$$

其中 $\mathcal{L}u$ 中通常含有 u 的梯度 ∇u 等难以计算的项。

利用网络近似梯度 ∇u 等难以计算的项



网络参数为 w_n 。

从数据中发现微分方程

研究问题：

假设微分方程未知，已有观测数据，希望从数据中发现微分方程。设观测数据为 $\{u(t, \cdot): t = t_0, t_1, \dots\}$ ，数据由下面的潜在微分方程支配

$$u_t(t, x, y) = F(x, y, u, u_x, u_y, u_{xx}, u_{xy}, u_{yy}, \dots), (x, y) \in \Omega \subset \mathbb{R}^2, t \in [0, T]$$

其中 F 为待发现的未知算子。

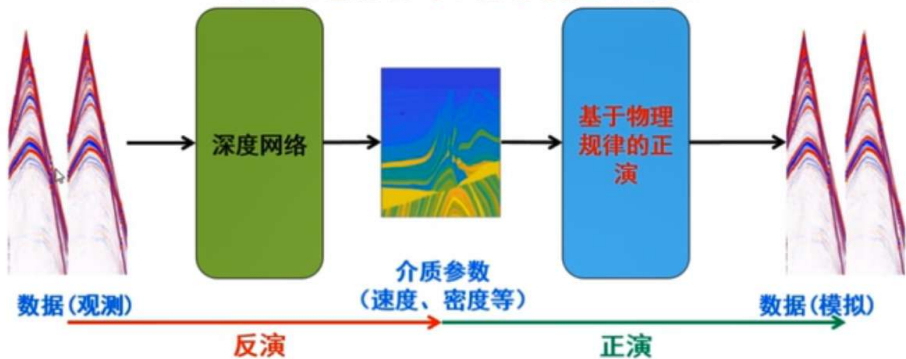
如何从数据中发现方程的基本思想：

1. 建立卷积操作与微分算符的关系；
2. 利用数值方法驱动的深度神经网络学习数据所蕴含的规律；
3. 将学习后的深度神经网络翻译成微分方程。

[5] Bin Dong, Qingtang Jiang, and Zuowei Shen. Image restoration: wavelet frame shrinkage, nonlinear evolution pdes, and beyond. *Multiscale Modeling & Simulation*, 15(1):606–660, 2017.

反问题

带有物理机制约束的深度网络结构



展望

- 从认识论和方法论上看，数学与AI交互作用的本质是提倡“模型驱动”与“数据驱动”的结合。这种结合具有巨大潜力。一些重要的结合方式包括：
 - **数据不够模型补**（如：模型驱动的学习）
 - **模型不精数据上**（如：数据驱动的控制、模型+网络）
 - **机理启发、知识融入**（如：物理机理融入的反问题求解）

[8] Xu ZB and Sun J, Model driven deep learning, National Science Record (NSR), 5: 1 (2018), 22-24

[9] Xu ZB, Gao ZQ & Yang Y, Physical principle incorporated autoencoder for inverse problems (2019)

学习方法论

Simulate Learning Methodology (SLM)

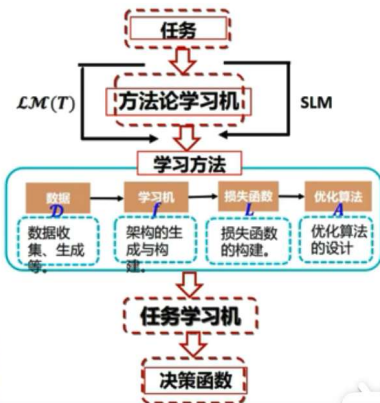
- **学习方法论**: 是从任务空间到学习空间的一个映照 $\mathcal{LM}: \mathcal{T} \rightarrow \mathcal{K}$.
- 学习方法论是对学习方法的一个超参赋值规则。
- **SLM的任务**: 学习从任务空间到学习空间的映射 $\mathcal{LM}$ (即学习超参赋值规则)。
- 为机器学习自动化提供了模型化/科学化基础。

$\mathcal{LM}(\cdot)$

学习方法论

$\mathcal{LM}: \mathcal{T} \rightarrow \mathcal{K}$

$\mathcal{LM}(T) = (\{a_i\}, \{b_i\}, \{c_i\}, \{d_i\})$



基于SLM的任务求解过程

References

- 徐宗本院士（B站）

- https://www.bilibili.com/video/BV1pa4y1v7zG?p=1&share_medium=android&share_plat=android&share_session_id=ec38f98e-ef30-4264-b73e-e9bfee51642a&share_source=WEIXIN&share_tag=s_i×tamp=1639386106&unique_k=CfOQ8yf
- https://www.bilibili.com/video/BV1Cv41147uh?p=1&share_medium=android&share_plat=android&share_session_id=ce4a705b-4cf9-46c7-bddd-6df513e82092&share_source=WEIXIN&share_tag=s_i×tamp=1639317685&unique_k=vP4pFx8
- https://www.bilibili.com/video/BV1NF41187jz?p=1&share_medium=android&share_plat=android&share_session_id=a28061bd-edfe-4751-b693-c26dbe25dd6f&share_source=WEIXIN&share_tag=s_i×tamp=1639317664&unique_k=K406u6r

AI for Sceince

Nature, Dec. 2012

- DeepMind + 数学家
- “首次证明了人工智能可以走在纯数学研究的前沿”
 - 「AI与人类合作」
 - 「AI指引人类直觉」

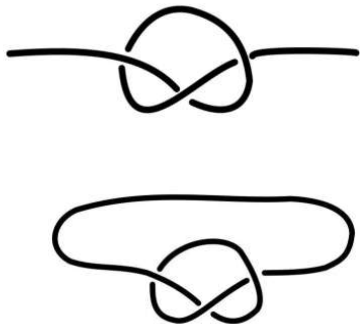


<https://mp.weixin.qq.com/s/yF50NJ8BTd6sSfFUпкиO4A>

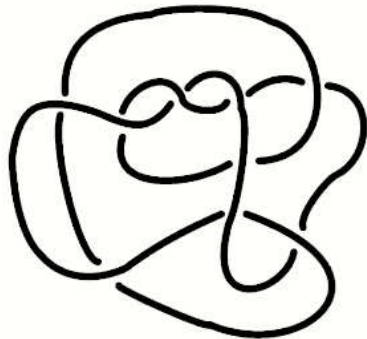
大数据规模与高复杂度问题

- 近代数学的三大猜想
 - 费马猜想、四色猜想和哥德巴赫猜想
 - 靠少数科学家的洞察力和个人经验积累，比如历史上两位天才，物理学家爱因斯坦和数学家拉马努金。
- 需要研究的问题复杂程度逐渐超出人类能力极限
 - 有的问题涉及的数据规模，是一个人一辈子也研究不完的。
 - 有的研究对象复杂程度之高，甚至可以有几千个维度，超出了一般人类大脑从直觉上可以理解的能力。

问题



纽结 (Knot)



一个复杂的纽结能不能被还原成简单的纽结，如果能就说明这两种纽结在拓扑上是等价的。

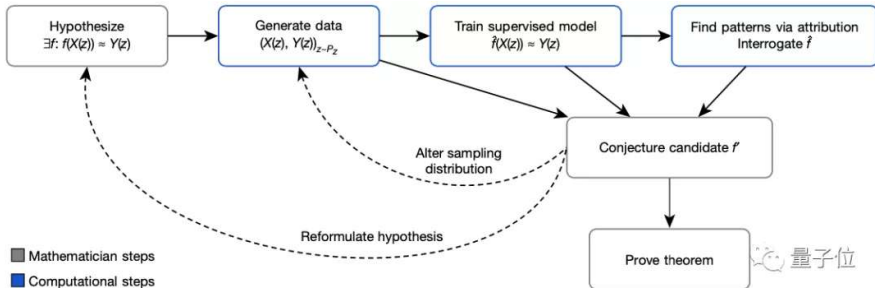
AI协助数学家

- 问题难就难在纽结的种类太多，自19世纪以来人类已经收集了无数种，如果用上计算机自动生成，现在每天都能生成几十亿种。
- 普通人难以从海量数据中发现隐藏的模式，AI这次却做到了。
- AI的贡献是发现了纽结的几何特征和代数特征之间存在直接的关联：数学家由此发现提出猜想，再给出严格证明，为纽结问题研究开辟了新的方向。

z: Knot	X(z): Geometric invariants				Y(z): Algebraic invariants		
	Volume	Chern-Simons	Meridional translation	...	Signature	Jones polynomial	...
	2.0299	0	i	...	0	$t^{-2} - t^{-1} + 1 - t + t^2$	...
	2.8281	-0.1532	$0.7381 + 0.8831i$	...	-2	$t - t^2 + 2t^3 - t^4 + t^5 - t^6$	...
	3.1640	0.1560	$-0.7237 + 1.0160i$	...	0	$t^{-2} - t^{-1} + 2 - 2t + t^2$	...

方法论：AI引导数学家直觉

- 提出一种研究框架，用来快速验证对两个量之间关系的猜想（直觉）是否值得继续探索：
 - 先通过监督学习来验证数学对象中的某一结构/模式的假设是存在的。
 - 然后，再使用归因技术来深入理解这些模式。
 - 在这个过程中，AI能够以人类无法比拟的规模输出数据，并从数据中挑选出人类无法检测到的模式。



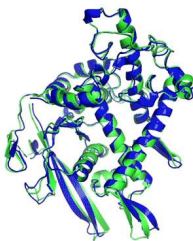
Science, Aug. 2021

- AI预测RNA分子结构
 - 三级结构
- 基于ARES (Atomic Rotationally Equivariant Scorer) 的几何深度学习学习方法

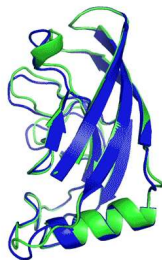


Science, July 2021

- AlphaFold2
 - 由给定的氨基酸序列，预测出蛋白质的3D结构
- 入选Science年度十大突破，被称作结构生物学“革命性”的突破、蛋白质研究领域的里程碑。



T1037 / 6vr4
90.7 GDT
(RNA polymerase domain)



T1049 / 6y4f
93.3 GDT
(adhesin tip)

● Experimental result
● Computational prediction

<https://mp.weixin.qq.com/s/95rG-vBJD0tdP4ymIECPgw>

References

- <http://web.math.princeton.edu/~weinan/>
- 鄂维南院士（B站）
 - https://b23.tv/aUk6UCD?share_medium=android&share_source=weixin&bbid=XYF13A7F3957C3B231E5D57BB1332806A7CCF&ts=1638976908236
 - https://www.bilibili.com/video/BV1Jz411e7RP/?spm_id_from=333.788.recommend_more_video.-1
 - https://www.bilibili.com/video/BV1YL411p7ny/?spm_id_from=333.788.recommend_more_video.5
 - https://www.bilibili.com/video/BV1Ph41187tU?p=1&share_medium=android&share_plat=android&share_session_id=e4b5beab-9a55-4c39-8973-78527e961b1d&share_source=WEIXIN&share_tag=s_i×tamp=1639299991&unique_k=7QybJlt
 - <https://www.bilibili.com/read/cv11535400>



中国科学技术大学
University of Science and Technology of China

谢谢！



中国科学技术大学
University of Science and Technology of China

数学建模 Mathematical Modeling

刘利刚

中国科学技术大学

高阶拟合模型

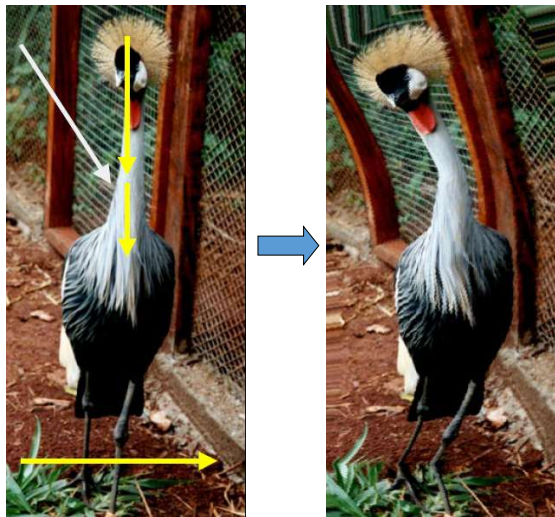
数据拟合/插值再思考

- 插值数据（函数值）
- 还有哪些属性可以插值？

超限插值

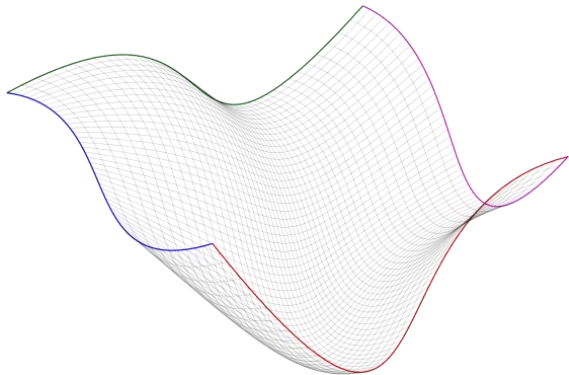
Transfinite Interpolation

问题：插值曲线？



Transfinite Interpolation

- 问题：给定4条边界曲线，构造插值这4条曲线的一张曲面



边界插值问题

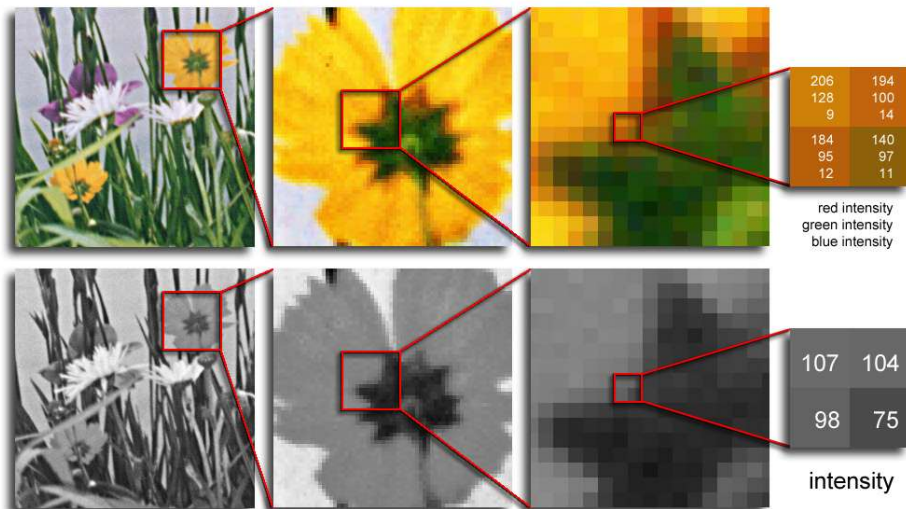
- 曲线
 - Hermite插值（3次、5次）
 - Lagrange插值
 - 几何样条
- 曲面
 - 线性混合曲面：插值2条曲线
 - Coons曲面（双线性混合曲面）：插值4条曲线
 - Bezier曲面
 - B样条曲面
 - Bernstein-Bezier三角曲面

高阶插值

高阶插值

- 插值高阶导数
- 多元：偏导数
- 更高阶光滑性

Color Image and Gray Image



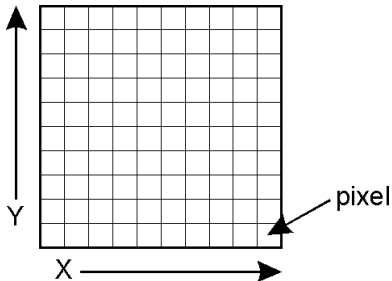
图像的另一个模型：函数

- 2D区域上的向量值函数

$$I = I(x, y)$$

Discrete sampling

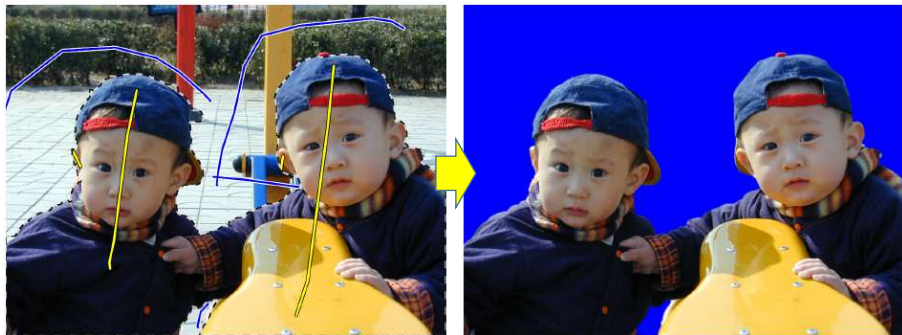
$$I = f(i, j)$$



思考：图像分割



前景/背景交互式图像分割

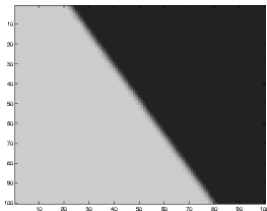


思考：图像修复

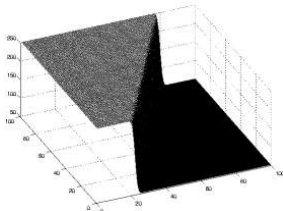


Partial Differential Equation (PDE)

Image as a Surface



single-edge image



3D visualization

If image can be viewed as a surface, it is then natural to ask: can we apply *geometric* tools to process this surface (or its equivalent image signals)?

Geometric Formulation

- Image I: $\mathbb{R}^2 \rightarrow \mathbb{R}$ may be viewed as a two-dimensional surface in three-dimensional space, i.e.,

$$M : S(x, y) = (x, y, I(x, y)) \subset \mathbb{R}^3$$



$$\begin{aligned} ds^2 &= dx^2 + dy^2 + dI^2 = dx^2 + dy^2 + (I_x dx + I_y dy)^2 \\ &= (1 + I_x^2) dx^2 + 2I_x I_y dx dy + (1 + I_y^2) dy^2 \end{aligned}$$



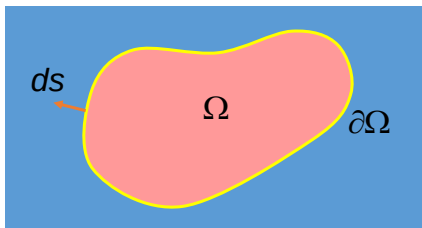
$$ds^2 = [dx \quad dy] G \begin{bmatrix} dx \\ dy \end{bmatrix}, G = \begin{bmatrix} 1 + I_x^2 & I_x I_y \\ I_x I_y & 1 + I_y^2 \end{bmatrix}$$

G: symmetric and positive definite matrix

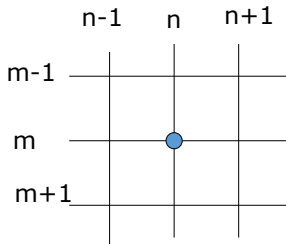
PDE-based Image Processing

- Image as a function (surface)

The solution of a PDE Equation is **uniquely** determined in Ω , if *Dirichlet* boundary conditions or *Neumann* boundary conditions are specified on $\partial\Omega$



Numerical Implementation of PDEs



$$\left. \frac{\partial I}{\partial x} \right|_{(m,n)} \approx \frac{I(m+1,n) - I(m-1,n)}{2h}$$

or

$$\left. \frac{\partial I}{\partial x} \right|_{(m,n)} \approx \lambda \frac{I(m+1,n) - I(m-1,n)}{2h}$$

$$+ \frac{1-\lambda}{2} \left(\frac{I(m+1,n+1) - I(m-1,n+1)}{2h} + \frac{I(m+1,n-1) - I(m-1,n-1)}{2h} \right)$$

$$\Delta I|_{(m,n)} \approx \frac{I(m+1,n) + I(m-1,n) + I(m,n+1) + I(m,n-1) - 4I(m,n)}{h^2}$$

or

$$\Delta I|_{(m,n)} \approx \lambda \frac{I(m+1,n) + I(m-1,n) + I(m,n+1) + I(m,n-1) - 4I(m,n)}{h^2}$$

$$+ (1-\lambda) \frac{I(m+1,n+1) + I(m-1,n+1) + I(m+1,n-1) + I(m-1,n-1) - 4I(m,n)}{h^2}$$

高阶插值例子：

Poisson Image Editing

Perez et al. Poisson image editing. Siggraph 2003.

<https://www.cs.jhu.edu/~misha/Fall07/Papers/Perez03.pdf>

图像移植

- 能否做到无缝移植？



Demo

如何做到的？



sources/destinations



cloning



seamless cloning

Poisson方程

Poisson Equation

$$\Delta\Phi = -4\pi G\rho(x, y)$$

$$\Delta \equiv \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$$

Siméon Denis Poisson (泊松)

- His teachers:
 - *Laplace*
 - *Lagrange*
- Poisson's terms:
 - Poisson's equation
 - Poisson's integral
 - Poisson distribution
 - Poisson brackets
 - Poisson's ratio
 - Poisson's constant



1781-1840, France

Siméon Denis Poisson

“Life is good for only two things:

*to **study** mathematics
and to **teach** it.”*



1781-1840, France

Background

- Partial Differential Equations (PDE)

$$E(f, f_x, f_y, f_{xx}, f_{xy}, f_{yy}) = 0$$

- The PDE's which occur in physics are mostly second order and linear:

$$A \cdot f_{xx} + 2B \cdot f_{xy} + C \cdot f_{yy} + D \cdot f_x + E \cdot f_y + F \cdot f + G = 0$$

$$A \cdot f_{xx} + 2B \cdot f_{xy} + C \cdot f_{yy} + D \cdot f_x + E \cdot f_y + F \cdot f + G = 0$$

$A \cdot C < B^2$: • Hyperbolic

- wave equation:

$$\Delta f = \frac{1}{v^2} \frac{\partial^2 f}{\partial t^2}$$

$A \cdot C = B^2$: • Parabolic

- heat equation:

$$\frac{\partial f}{\partial t} = k \cdot \Delta f$$

$A \cdot C > B^2$: • Elliptic

- Laplace equation:

$$\Delta f = 0$$

- Poisson equation:

$$\Delta f = -\rho$$

Poisson Equation

$$\Delta f = -\rho$$

$$\Delta \equiv \frac{\partial^2}{\partial^2 x} + \frac{\partial^2}{\partial^2 y}$$

$$\rho = \rho(x, y)$$

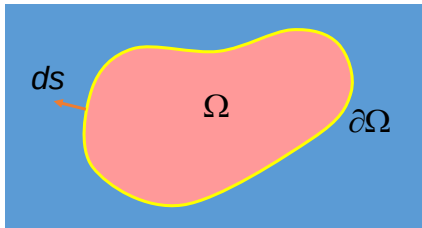
Boundary conditions

- *Dirichlet* boundary conditions:

$$f|_{\partial\Omega}$$

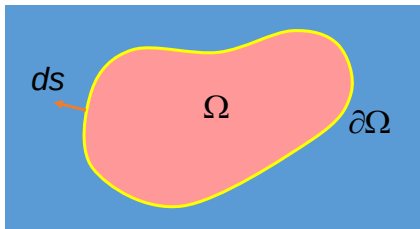
- *Neumann* boundary conditions:

$$\frac{\partial f}{\partial \mathbf{s}}|_{\partial\Omega}$$

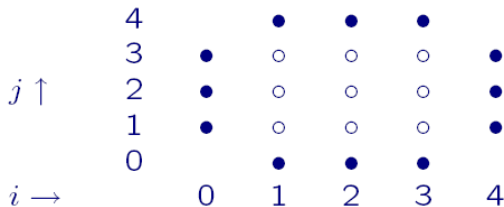


Existence of solution

The solution of an Poisson Equation is **uniquely** determined in Ω , if *Dirichlet* boundary conditions or *Neumann* boundary conditions are specified on $\partial\Omega$



Discrete Poisson Equation



$$\Delta f = -\rho$$



$$\frac{f_{i+1,j} + f_{i-1,j} - 2f_{i,j}}{h^2} + \frac{f_{i,j+1} + f_{i,j-1} - 2f_{i,j}}{h^2} = \rho_{i,j}$$

Matrix Nature of Poisson Equation

- Sparse linear system

$$\begin{bmatrix} -4 & 1 & & & & & & & \\ 1 & -4 & 1 & & & & & & \\ & 1 & -4 & 1 & & & & & \\ & & 1 & -4 & 1 & & & & \\ & & & 1 & -4 & 1 & & & \\ & & & & 1 & -4 & 1 & & \\ & & & & & 1 & -4 & 1 & \\ & & & & & & 1 & -4 & 1 \\ & & & & & & & 1 & -4 \end{bmatrix} \begin{bmatrix} f_{11} \\ f_{21} \\ f_{31} \\ f_{12} \\ f_{22} \\ f_{32} \\ f_{13} \\ f_{23} \\ f_{33} \end{bmatrix} = \begin{bmatrix} b_{11} \\ b_{21} \\ b_{31} \\ b_{12} \\ b_{22} \\ b_{32} \\ b_{13} \\ b_{23} \\ b_{33} \end{bmatrix}$$

Poisson Equation Solver

- Direct method
- Iterative methods
 - Jacobi, Gauss-Seidel, SOR
- Multigrid method

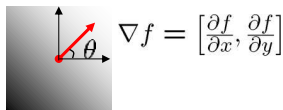
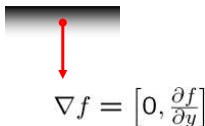
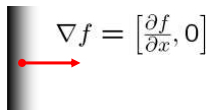
Gradient Image Processing

Image gradient

- The gradient of an image:

$$\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right]$$

- The gradient points in the direction of most rapid change in intensity



The gradient direction is given by:

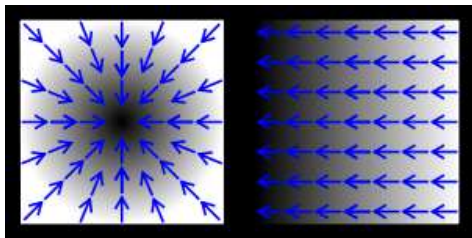
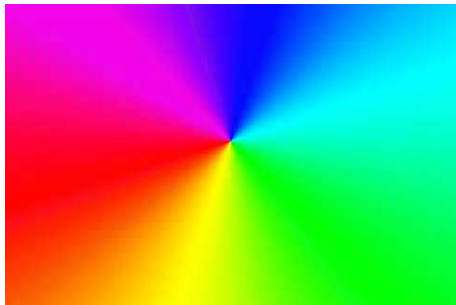
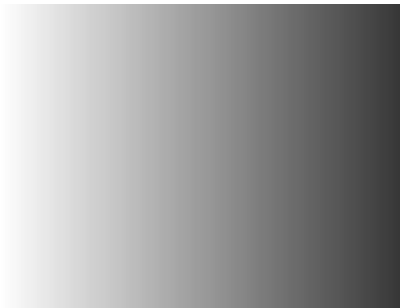
$$\theta = \tan^{-1} \left(\frac{\partial f}{\partial y} / \frac{\partial f}{\partial x} \right)$$

- how does this relate to the direction of the edge?

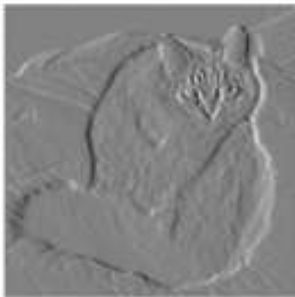
The *edge strength* is given by the gradient magnitude

$$\|\nabla f\| = \sqrt{\left(\frac{\partial f}{\partial x} \right)^2 + \left(\frac{\partial f}{\partial y} \right)^2}$$

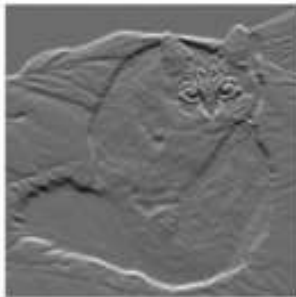
Gradient (梯度)



Gradients



X-gradient



Y-gradient

Why gradients?

- Human visual system is very sensitive to gradient
- Gradient encode edges and local contrast quite well
- Method
 - Do your editing in the gradient domain
 - Reconstruct image from gradient

Cloning Gradient



sources/destinations



seamless cloning

Variational interpretation

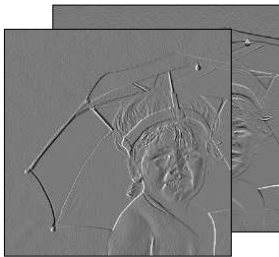
$$f^* = \arg \min_f \iint_{\Omega} \underbrace{\|\nabla f - \mathbf{v}\|^2}_F \quad \text{s.t. } f^*|_{\partial\Omega} = f|_{\partial\Omega}$$



$$\text{Euler Equation: } F_f - \frac{\partial}{\partial x} F_{f_x} - \frac{\partial}{\partial y} F_{f_y} = 0$$

$$\Delta f = \text{div}(\mathbf{v}) \quad \text{s.t. } f^*|_{\partial\Omega} = f|_{\partial\Omega}$$

$\mathbf{V}$ is a **guidance** field, needs not to be a gradient field.



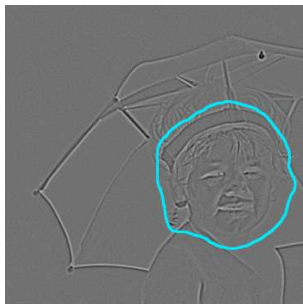
$$\mathbf{g} = -\nabla I$$

$$\rho(\mathbf{x}) = \text{div}(\mathbf{g}) \equiv \lim_{V \rightarrow 0} \frac{\oint_S \mathbf{g} \cdot d\mathbf{s}}{V}$$



$$\Delta I = -\rho(\mathbf{x})$$





$$\Delta I = -\rho(\mathbf{x})$$



boundary



div of gradient

Poisson Image Editing

[Perez et al., Siggraph 2003]

Seamless Cloning

- Precise selection: tedious and unsatisfactory
- Alpha-Matting: powerful but involved
- **Seamless cloning**: loose selection but no seams?



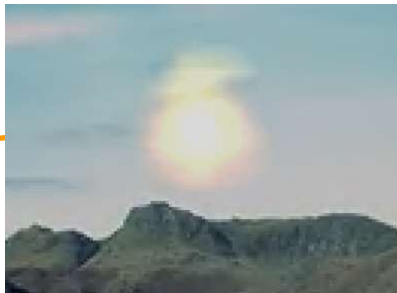
Cloning by solving Poisson Equation

$$\Delta I = \text{div}(\nabla I_A) \quad \text{s.t.} \quad I|_{\partial\Omega} = I_B|_{\partial\Omega}$$

I_A



I_B



compose



Cloning Gradient



source/destination



cloning



seamless cloning

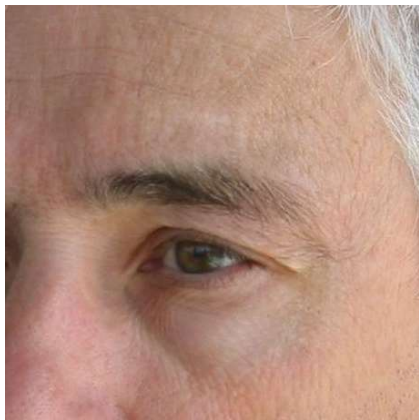
Cloning Gradient



change texture



conceal



mix lights



change colors



change colors



Seamless Tiling

Multiple
images
tiled at
random



More on Gradient-based Method

Gradients and grayscale images

- **Grayscale image: $n \times n$ scalars**
- **Gradient: $n \times n$ *2D vectors***
- **Overcomplete!**
- **What's up with this?**
- **Not all vector fields are the gradient of an image!**
- **Only if they are curl-free (a.k.a. conservative)**
 - But it does not matter for us

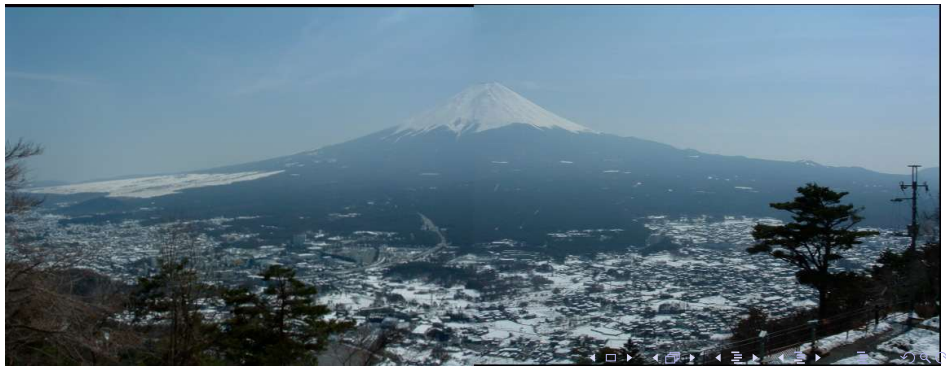
Example: Image Stitching

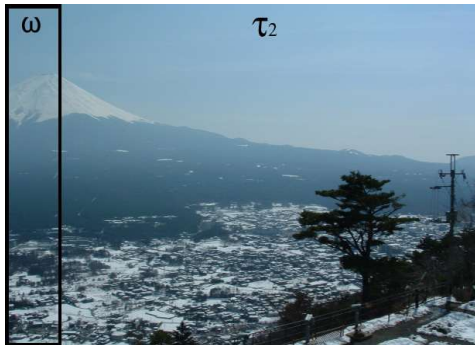


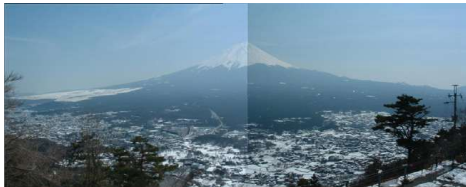
<http://www.cs.huji.ac.il/~alevin/papers/eccv04-blending.pdf>

<http://eprints.pascal-network.org/archive/00001062/01/tips05=blending.pdf>



τ_1 ω ω τ_2 

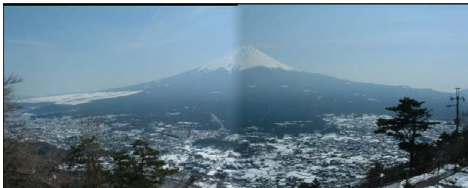
τ_1 ω ω τ_2 



Optimal seam



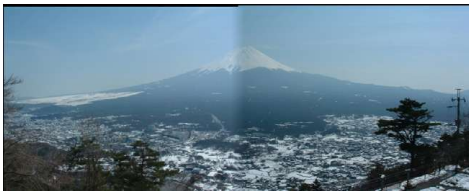
Optimal seam on the gradients



Feathering



Pyramid blending on the gradients



Pyramid blending



GIST1

PDE based Image Processing

1. Mean Curvature Evolution

Mean Curvature

Unit normal of this surface is

$$N(x, y) = \frac{S_x \times S_y}{\|S_x \times S_y\|} = \frac{(-I_x, -I_y, 1)}{\sqrt{1 + I_x^2 + I_y^2}}$$

Mean curvature is

$$H(x, y) = \frac{I_{xx}(1 + I_y^2) - 2I_x I_y I_{xy} + I_{yy}(1 + I_x^2)}{2(1 + I_x^2 + I_y^2)^{3/2}}$$

Theorem

Surfaces of zero mean curvature have minimal areas

$$\frac{I_{xx}(1 + I_y^2) - 2I_x I_y I_{xy} + I_{yy}(1 + I_x^2)}{2(1 + I_x^2 + I_y^2)^{3/2}} = 0$$

Discrete Mean Curvature

$$S(M) = \iint_{\Omega} \sqrt{\det(G)} dx dy = \iint_{\Omega} \sqrt{1 + I_x^2 + I_y^2} dx dy$$

Euler-Lagrange Equation:  minimize $S(M)$ leads to

$$\frac{\partial}{\partial x} \left(\frac{I_x}{\sqrt{1 + I_x^2 + I_y^2}} \right) + \frac{\partial}{\partial y} \left(\frac{I_y}{\sqrt{1 + I_x^2 + I_y^2}} \right) = 0 \quad (A)$$


$$\frac{I_{xx}(1 + I_y^2) - 2I_x I_y I_{xy} + I_{yy}(1 + I_x^2)}{2(1 + I_x^2 + I_y^2)^{3/2}} = 0 \quad (B)$$

Mean Curvature Diffusion

Diffusion equation

$$I_t = \frac{I_{xx}(1+I_y^2) - 2I_xI_yI_{xy} + I_{yy}(1+I_x^2)}{2(1+I_x^2 + I_y^2)^{3/2}}$$



Before post-processing



After post-processing

Further Diffusion



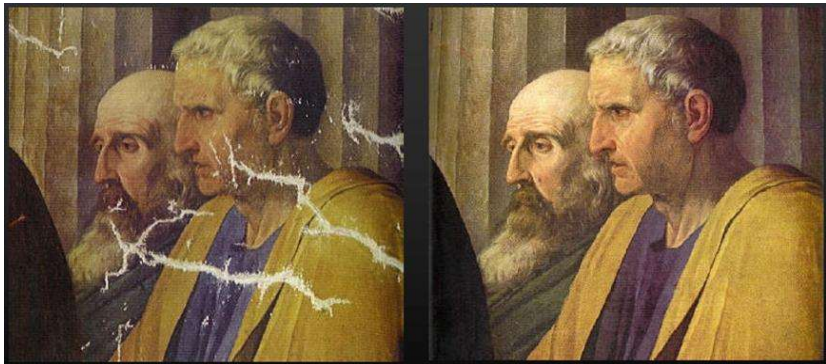
After 3 iterations



After 10 iterations

2. Image Inpainting

Image Inpainting



"Image Inpainting : An Overview",
Guillermo Sapiro

Image Inpainting

“Digital Image Inpainting is an iterative method for repairing damaged pictures or removing unnecessary elements from pictures”



“Fast Digital Image Inpainting”,

Manuel M. Oliveira, Brian Bowen, Richard McKenna and Yu-Sung Chang

Image Inpainting: Modeling

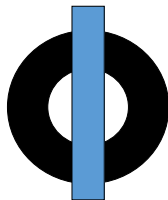
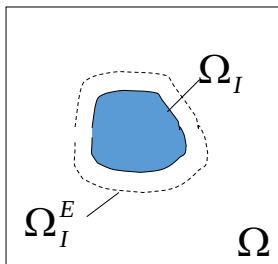


Image example

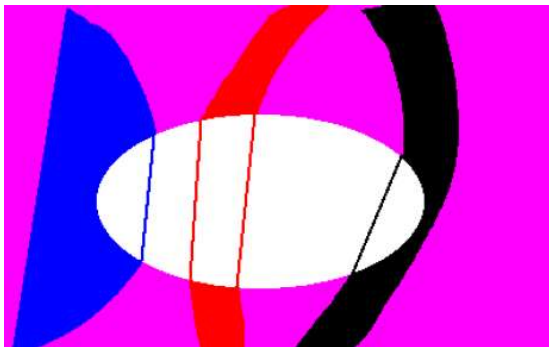
Ω_I^E Extended inpainting domain

Assumption: inpainting domain is local and does not contain texture (complimentary to texture-synthesis based inpainting techniques)

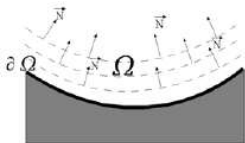
2.1 Level Line Based Method

- Joining with geodesic curves the points of the isophotes (lines of equal gray values) arriving at the boundary of the region to be inpainted
- Drawbacks
 - Inpainted region should have simple topology
 - Angle of level lines is not preserved

Example



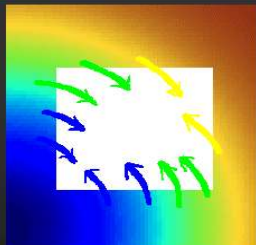
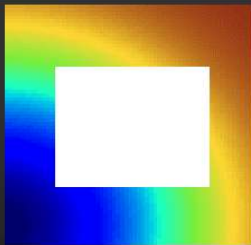
Extension



[Siggraph 2000]

- Apply diffusion to the original image to avoid noise
- Updates to the values of pixels inside the region are made, information propagated in the direction of the isophotes
- After every few iterations, diffusion process is applied
- Propagation of gray values and the isophotes direction is critical
- Color images are considered as a group of 3 images and this technique is applied independently to them

Idea



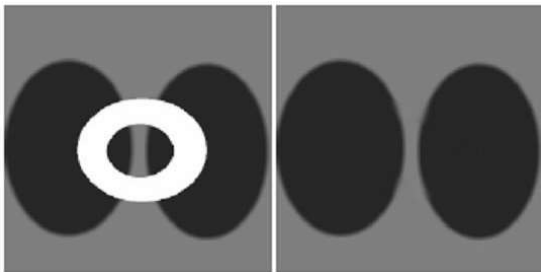
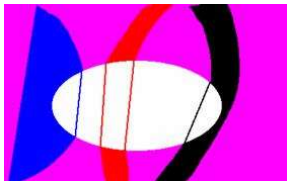
- **Propagate information**

$$\nabla L \bullet \vec{N} = 0$$

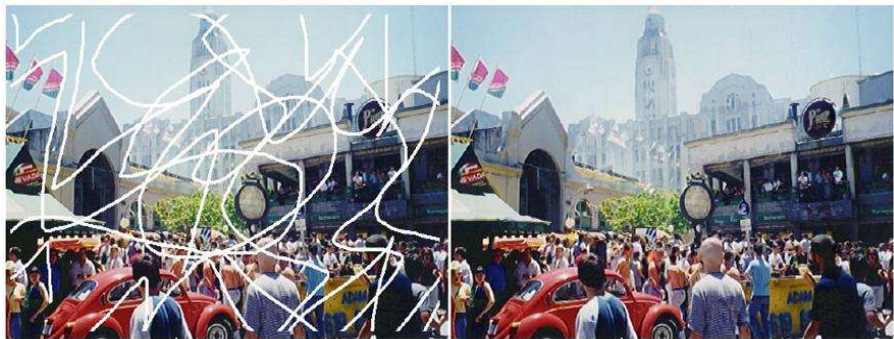
- **Evolutionary form**

$$\frac{\partial I}{\partial \mathbf{t}} = \nabla L \bullet \vec{N}$$

Comparison



Examples



Examples



2.2 Variational Method

$$\min J(\hat{u}) = \int_{\Omega_I^E} |\nabla \hat{u}| \, dx dy + \frac{\lambda}{2} \int_{\Omega_I^E \setminus \Omega_I} |\hat{u} - u|^2 \, dx dy$$

↓
Total variation (TV)

$\hat{u}$ Restored image u degraded image

Rational:

The first term describes the smoothness constraint within the extend inpainting domain

The second term describes the observation constraint

How to obtain the corresponding PDE?

Euler-Lagrangian Equation

$$-\nabla \cdot \left(\frac{\nabla \hat{u}}{|\nabla \hat{u}|} \right) + \lambda_E (\hat{u} - u) = 0$$

Where

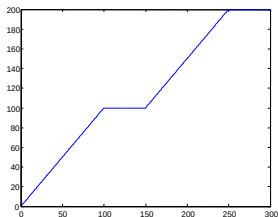
$$\lambda_E = \begin{cases} \lambda & (x, y) \in \Omega_I^E \setminus \Omega_I \\ 0 & (x, y) \in \Omega_I \end{cases}$$



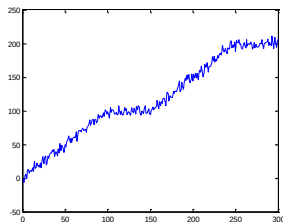
TV-inpainting:
$$\frac{\partial \hat{u}}{\partial t} = \nabla \cdot \left(\frac{\nabla \hat{u}}{|\nabla \hat{u}|} \right) + \lambda_E (u - \hat{u})$$

Total Variation (TV)

$$V(f, \Omega) = \int_{\Omega} |\nabla f|$$



Clean (TV small)



noisy (TV large)

Key idea: it is L_1 instead of L_2 norm
(minimizing L_2 will not preserve edges)

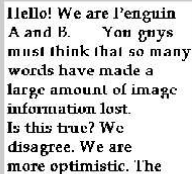
Inpainting Example

Image to be inpainted



Hello! We are Penguin
A and B. You guys
must think that so many
words have made a
large amount of image
information lost.
Is this true? We
disagree. We are
more optimistic. The

Inpainting domain (or mask)



Hello! We are Penguin
A and B. You guys
must think that so many
words have made a
large amount of image
information lost.
Is this true? We
disagree. We are
more optimistic. The

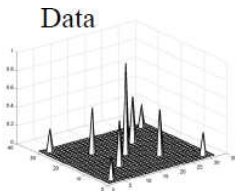


(Courtesy: Jackie Shen, UMN MATH)

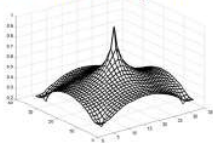
PDE: Thin Plate Energy

- **Thin plate:**
minimize *second* derivative

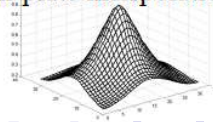
$$\min_f \int \int f_{xx}^2 + 2f_{xy}^2 + f_{yy}^2 dx dy$$



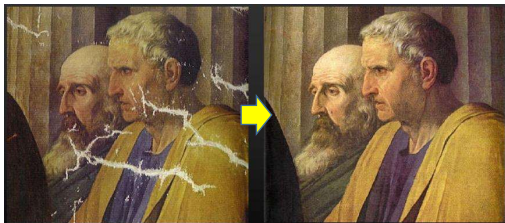
Membrane interpolation



Thin-plate interpolation



偏微分方程的广泛应用



泊小松

“Life is good for only two things:

*to **study** mathematics
and to **apply** it.”*





中国科学技术大学
University of Science and Technology of China

谢谢！



中国科学技术大学

University of Science and Technology of China

数学建模

Mathematical Modeling

刘利刚

中国科学技术大学

微分方程模型

微分方程模型

此类模型的建模思想是：

- 模型的因素随时间、空间而变化，可以看作是时间、空间的连续可微函数。
- 因素的变化与时间、空间以及因素本身有关，可用微分方程（或微分方程组）来表示。
- 微分方程的建立与求解是解决实际问题的核心。
- 当微分方程无解析解的时候，需要使用数值算法近似求解。
- 对于某些求泛函的极值问题，可以离散化，或者用变分法求解。

例2：（谭永基《数学模型》第17章）生物群体模型。
考虑生物群体数量 x 随时间 t 的变化。

【单物种模型】

- $\frac{dx}{dt} = (a + bx)x$ ，其中 a 与物种的生长资源及人类的捕杀有关， b 与物种内部的合作与竞争有关， a, b 都可能是 t 的函数。

【双物种模型】

- $\begin{cases} x'(t) = f(x, y) \cdot x = (a_{11}x + a_{12}y + b_1) \cdot x \\ y'(t) = g(x, y) \cdot y = (a_{21}x + a_{22}y + b_2) \cdot y \end{cases}$ ，其中 $a_{12}, a_{21} > 0$ 表示双方互助， $a_{12}, a_{21} < 0$ 表示双方竞争， $a_{12}a_{21} < 0$ 表示捕食关系。
- 达到平衡时， $xy = 0$ 或 $\begin{cases} a_{11}x + a_{12}y + b_1 = 0 \\ a_{21}x + a_{22}y + b_2 = 0 \end{cases}$
- f, g 可能是其它形式的函数，此处仅作局部线性近似。

例3：（姜启源《数学模型》§ 5.10）传染病的传播。

【模型一】

- 假设被发现的感染者都被隔离治疗，治愈后不再传播病菌。
- 设 $x(t)$ 是尚未感染的易感者比例， $y(t)$ 是未被发现的感染者比例， $z(t)$ 是已确诊的感染者比例， $x + y + z = 1$ 。

$$\begin{cases} x'(t) = -axy \\ y'(t) = axy - by \\ z'(t) = by \end{cases}$$

其中 a 称为传染率， b 称为确诊率， a, b 可以是 t 的函数。

- 每日新增确诊病例数与 $y(t)$ 成正比例，积累确诊病例数与 $z(t)$ 成正比例。由此可以估计出 a, b 。
- 思考：求 $t \rightarrow \infty$ 时， $x(t)$ 和 $z(t)$ 的极限。

【模型二】

- 假设被发现的感染者都被隔离治疗，治愈后部分人免疫。
- 设 $x(t)$ 是尚未感染的易感者比例， $y(t)$ 是未被发现的感染者比例， $z(t)$ 是正在隔离治疗中的感染者比例， $x + y + z < 1$ 。

$$\begin{cases} x'(t) = -axy + dz \\ y'(t) = axy - by \\ z'(t) = by - cz \end{cases}$$

其中 a 称为传染率， b 称为确诊率， c 称为出院率， $0 < d < c$ 。

【思考】

- 参数 a, b, c, d 对于 $y(t)$ 的影响？
- 目前的一些防疫措施对 a, b, c, d 和 $y(t)$ 的影响？
- 如果希望把 $y(t)$ 迅速降至某一阈值之下，应如何调整参数？
- 如果考虑到调整参数所造成的社会经济影响，又如何处理？

例4：（CUMCM2007A）中国人口增长预测。

中国是一个人口大国，人口问题始终是制约我国发展的关键因素之一。根据已有数据，运用数学建模的方法，对中国人口做出分析和预测是一个重要问题。

近年来中国的人口发展出现了一些新的特点，例如，老龄化进程加速、出生人口性别比持续升高，以及乡村人口城镇化等因素，这些都影响着中国人口的增长。2007年初发布的《国家人口发展战略研究报告》(附录1)还做出了进一步的分析。

关于中国人口问题已有多方面的研究，并积累了大量数据资料。附录2就是从《中国人口统计年鉴》上收集到的部分数据。

试从中国的实际情况和人口增长的上述特点出发，参考附录2中的相关数据（也可以搜索相关文献和补充新的数据），建立中国人口增长的数学模型，并由此对中国人口增长的中短期和长期趋势做出预测；特别要指出你们模型中的优点与不足之处。

【模型建立】

- （人口的总数）设时刻 t 时，人口总数 $x(t)$ 。

$$x'(t) = \lambda(t)x(t) \Rightarrow x(t) = x(0)e^{\int_0^t \lambda(s)ds}$$

其中 $\lambda(t)$ 称为人口的净增长率。 $\lambda(t)$ 与多种因素有关，易受社会经济条件、国家政策、文化观念等影响。

- （人口的年龄结构）设第 t 年时，年龄 k 的人口总数 $x_k(t)$ 。

$$\begin{cases} x_0(t+1) = \sum_i b_i(t)x_i(t) \\ x_{k+1}(t+1) = (1-d_k(t))x_k(t), \quad k \in \mathbf{N} \end{cases}$$

其中 $b_k(t)$ 和 $d_k(t)$ 分别是年龄 k 的人口的生育率和死亡率。

- 上述 $\lambda(t), b_k(t), d_k(t)$ 都可根据历史抽样数据估计得到。
- 上式可以表示成矩阵乘积的形式

$$X(t+1) = A(t)X(t) \Rightarrow X(t+1) = A(t) \cdots A(0)X(0)$$

其中

$$A(t) = \begin{pmatrix} 0 & b_1(t) & \cdots & b_{n-1}(t) \\ 1 - d_0(t) & \ddots & & \\ & \ddots & 0 & \\ & & 1 - d_{n-2}(t) & 0 \end{pmatrix}$$

- 上式还可以表示成偏微分方程的形式

$$\frac{\partial p(s, t)}{\partial s} + \frac{\partial p(s, t)}{\partial t} = -d(s, t)p(s, t)$$

其中 $p(s, t)$ 是 t 时刻年龄 s 的人口密度函数。 $\int_0^\infty p(s, t)ds = 1, \forall t$ 。

这种表示形式仅具有理论意义，对于人口问题的计算没有任何优势。

- （老龄化问题）随着人均寿命的延长，老龄化进程不可逆转。国家和社会必须适应老龄化现象，而非试图改变老龄化现象。
- （人口的性别比例）随着男女平等观念的深入，无需特别考虑。
- （人口的城乡结构）这是政治和经济问题，不属于人口问题。

例5：（CUMCM2018A）高温作业专用服装设计。

在高温环境下工作时，人们需要穿着专用服装以避免灼伤。专用服装通常由三层织物材料构成，记为I、II、III层，其中I层与外界环境接触，III层与皮肤之间还存在空隙，将此空隙记为IV层。

为设计专用服装，将体内温度控制在 37°C 的假人放置在实验室的高温环境中，测量假人皮肤外侧的温度。为了降低研发成本、缩短研发周期，请你们利用数学模型来确定假人皮肤外侧的温度变化情况，并解决以下问题：

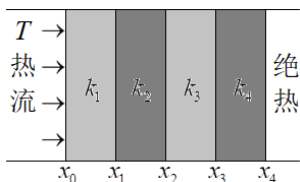
(1) 专用服装材料的某些参数值由附件1给出，对环境温度为 75°C 、II层厚度为6mm、IV层厚度为5mm、工作时间为90分钟的情形开展实验，测量得到假人皮肤外侧的温度。建立数学模型，计算温度分布，并生成温度分布的Excel文件。

(2) 当环境温度为 65°C 、IV层的厚度为5.5mm时，确定II层的最优厚度，确保工作60分钟时，假人皮肤外侧温度不超过 47°C ，且超过 44°C 的时间不超过5分钟。

(3) 当环境温度为 80°C 时，确定II层和IV层的最优厚度，确保工作30分钟时，假人皮肤外侧温度不超过 47°C ，且超过 44°C 的时间不超过5分钟。

【问题1模型建立】

- 一维热传导方程 $u_t = (ku_x)_x$ 。 $x \in [0, L]$, $t \in [0, \infty)$ 。
- $k(x) = \frac{\text{热传导率}}{\text{密度} \times \text{比热}} = \text{const}$, $x \in (x_{i-1}, x_i)$ 。 $x_0 = 0$, $x_4 = L$ 。
- 边界条件 $u(x, 0) = 37$, $u_x(0, t) = \lambda \cdot (u(0, t) - T)$, $u_x(L, t) = 0$ 。
 T 是环境温度, λ 是待求常数。



【模型求解】

k_1	$\dots$	k_i	k_{i+1}	$\dots$	k_n	
u_0	u_1	u_{i-1}	u_i	u_{i+1}	u_{n-1}	u_n

- 把定义域 n 等分, 记 $u_i(t) = u(x_i, t)$, 则有 $u_i(0) = 37$, 并且

$$\begin{cases} k_{i+1}(u_{i+1} - u_i) - k_i(u_i - u_{i-1}) \approx \frac{L^2}{n^2} u'_i(t), & 1 \leq i \leq n-1 \\ u_1 - u_0 \approx \frac{\lambda L}{n} (u_0 - T), & u_n - u_{n-1} \approx 0 \end{cases}$$

- 记 $\mathbf{v}(t) = (u_1, \dots, u_{n-1})$, 上式可表示为 $\mathbf{v}'(t) + A\mathbf{v} = \mathbf{b}$, 其中

$$A = \frac{n^2}{L^2} \begin{pmatrix} \frac{\lambda L k_1}{n + \lambda L} + k_2 & -k_2 & & & \\ -k_2 & k_2 + k_3 & -k_3 & & \\ & \ddots & \ddots & \ddots & \\ & & -k_{n-2} & k_{n-2} + k_{n-1} & -k_{n-1} \\ & & & -k_{n-1} & k_{n-1} \end{pmatrix},$$

$$\mathbf{b} = \left(\frac{n^2}{L^2} \frac{\lambda L k_1}{n + \lambda L} T, 0, \dots, 0 \right). A \text{ 是对称、对角优、正定的。}$$

- 解得 $\mathbf{v}(t) = e^{-At}(\mathbf{v}(0) - A^{-1}\mathbf{b}) + A^{-1}\mathbf{b}$ 。设 $\mathbf{c}$ 是全一向量。注意到 $A\mathbf{c} = \frac{1}{T}\mathbf{b}$, $\mathbf{v}(0) = 37\mathbf{c}$ 。故 $\mathbf{v}(t) = T\mathbf{c} - (T - 37)e^{-At}\mathbf{c}$, $u_n(t)$ 形如 $T - \sum_{i=1}^n a_i e^{-b_i t}$
- 取定 n , 求 λ 使 $u(L, t) \approx u_n(t)$ 的图像与数据拟合。

【问题2,3模型建立】

- 求II层的厚度 d , 使得 $u(L, 3300) \leq 44$ 且 $u(L, 3600) \leq 47$ 。
- 求II、IV层的厚度 d_1, d_2 , 使得 $u(L, 1500) \leq 44$ 且 $u(L, 1800) \leq 47$ 。

【问题求解】

- 对给定的 x, t , 视 $u(x, t)$ 为 d 或 d_1, d_2 的函数。

例6：（姜启源《数学模型》§5.8）赛跑的速度。
如何分配体能使得成绩最好？

【模型建立】

- 设赛程 L ，速度 $v(x)$ ，则成绩 $T = \int_0^L \frac{dx}{v(x)}$ 。
- 设空气阻力 f 关于 v 单调增，并且不超过运动员的最大爆发力 F 。
- 运动员消耗的体能 $\int_0^L f(x)dx$ 也有上限 E 。
- 问题化为求 $\int_0^L \frac{dx}{v(x)}$ 在上述两个约束下的最小值。

【模型求解】

- 由均值不等式，当且仅当 $v(x)$ 是常数时， $\int_0^L \frac{dx}{v(x)} \int_0^L f(x)dx$ 最小。
- 比赛策略：先全力加速至最大，再保持匀速，最后耗尽体能冲刺。

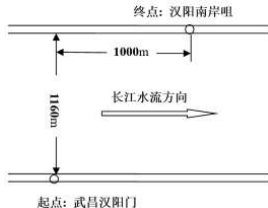
例7：（CUMCM2003D）抢渡长江。

“渡江”是武汉城市的一张名片。1934年9月9日，武汉警备旅官兵与体育界人士联手，在武汉第一次举办横渡长江游泳竞赛活动，起点为武昌汉阳门码头，终点设在汉口三北码头，全程约5000米。有44人参加横渡，40人达到终点，张学良将军特意向冠军获得者赠送了一块银盾，上书“力挽狂澜”。

2001年，“武汉抢渡长江挑战赛”重现江城。2002年，正式命名为“武汉国际抢渡长江挑战赛”，于每年的5月1日进行。由于水情、水性的不可预测性，这种竞赛更富有挑战性和观赏性。

2002年5月1日，抢渡的起点设在武昌汉阳门码头，终点设在汉阳南岸咀，江面宽约1160米。据报载，当日的平均水温 16.8°C ，江水的平均流速为1.89米/秒。参赛的国内外选手共186人（其中专业人员将近一半），仅34人到达终点，第一名的成绩为14分8秒。除了气象条件外，大部分选手由于路线选择错误，被滚滚的江水冲到下游，而未能准确到达终点。

假设在竞渡区域两岸为平行直线，它们之间的垂直距离为1160米，从武昌汉阳门的正对岸到汉阳南岸咀的距离为1000米，见示意图。



请你们通过数学建模来分析上述情况，并回答以下问题：

1. 假定在竞渡过程中游泳者的速度大小和方向不变，且竞渡区域每点的流速均为1.89米/秒。试说明2002年第一名是沿着怎样的路线前进的，求他游泳速度的大小和方向。如何根据游泳者自己的速度选择游泳方向，试为一个速度能保持在1.5米/秒的人选择游泳方向，并估计他的成绩。
2. 在（1）的假设下，如果游泳者始终以和岸边垂直的方向游，他们能否到达终点？根据你们的数学模型说明为什么1934年和2002年能游到终点的人数的百分比有如此大的差别；给出能够成功到达终点的选手的条件。
3. 若流速沿离岸边距离的分布为 $v(y) = \begin{cases} 1.47, & 0 \leq y \leq 200 \\ 2.11, & 200 \leq y \leq 960 \\ 1.47, & 960 \leq y \leq 1160 \end{cases}$
(设从武昌汉阳门垂直向上为 y 轴正向)
游泳者的速度大小（1.5米/秒）仍全程保持不变，
试为他选择游泳方向和路线，估计他的成绩。
4. 若流速沿离岸边距离为连续分布，例如 $v(y) = \begin{cases} \frac{2.28}{200}y, & 0 \leq y \leq 200 \\ 2.28, & 200 \leq y \leq 960 \\ \frac{2.28}{200}(1160 - y), & 960 \leq y \leq 1160 \end{cases}$
或你们认为合适的连续分布，如何处理这个问题。
5. 用普通人能懂的语言，给有意参加竞渡的游泳爱好者写一份竞渡策略的短文。
6. 你们的模型还可能有什么其他的应用？

【模型建立】

- 设游泳方向角 θ ，则游泳路线 (x, y) 满足 $\begin{cases} x'(t) = 1.5 \cos \theta + v \\ y'(t) = 1.5 \sin \theta \end{cases}$ 。
- 求 $\theta(y)$ 使得 $\int_0^{1160} \frac{1.5 \cos \theta + v}{1.5 \sin \theta} dy = 1000$ 并且 $T = \int_0^{1160} \frac{1}{1.5 \sin \theta} dy$ 最小。

【模型求解】

- 当 T 最小时， $\frac{\partial}{\partial \theta} \left(\frac{1.5 \cos \theta + v}{1.5 \sin \theta} \right) = \lambda \frac{\partial}{\partial \theta} \left(\frac{1}{1.5 \sin \theta} \right)$ ，其中 λ 是待定常数。
- 化简上式，得 $\cos \theta = \frac{1.5}{\lambda - v}$ 。再由约束条件，解得 λ ，进而得 θ, T 。

微分方程模型

- 1 传染病模型
- 2 经济增长模型
- 3 正规战与游击战
- 4 药物在体内的分布与排除
- 5 香烟过滤嘴的作用
- 6 人口预测和控制
- 7 烟雾的扩散与消失
- 8 万有引力定律的发现

动态 模型

- 描述对象特征随时间(空间)的演变过程
- 分析对象特征的变化规律
- 预报对象特征的未来性态
- 研究控制对象特征的手段

微分 方程 建模

- 根据函数及其变化率之间的关系确定函数
- 根据建模目的和问题分析作出简化假设
- 按照内在规律或用类比法建立微分方程

1. 传染病模型

问题

- 描述传染病的传播过程
- 分析受感染人数的变化规律
- 预报传染病高潮到来的时刻
- 预防传染病蔓延的手段
- 按照传播过程的一般规律，
用机理分析方法建立模型

模型1

已感染人数 (病人) $i(t)$

假设

- 每个病人每天有效接触 (足以使人致病) 人数为 λ

建模

$$i(t + \Delta t) - i(t) = \lambda i(t) \Delta t$$

$$\Rightarrow \frac{di}{dt} = \lambda i$$

$$i(0) = i_0$$



$$i(t) = i_0 e^{\lambda t}$$



$$t \rightarrow \infty \Rightarrow i \rightarrow \infty \quad ?$$

若有效接触的是病人，
则不能使病人数增加



必须区分已感染者(病人)和未感染者(健康人)

模型2

区分已感染者(病人)和未感染者(健康人)

假设

1) 总人数 N 不变, 病人和健康人的比例分别为 $i(t), s(t)$

SI 模型

2) 每个病人每天有效接触人数为 λ , 且使接触的健康人致病

$\lambda \sim$ 日接触率

建模

$$N[i(t + \Delta t) - i(t)] = [\lambda s(t)]Ni(t)\Delta t$$

$$\frac{di}{dt} = \lambda si$$

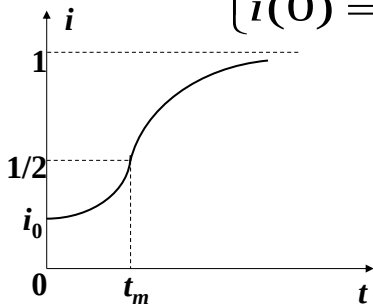
$$s(t) + i(t) = 1$$



$$\begin{cases} \frac{di}{dt} = \lambda i(1-i) \\ i(0) = i_0 \end{cases}$$

模型2

$$\begin{cases} \frac{di}{dt} = \lambda i(1-i) \\ i(0) = i_0 \end{cases} \Rightarrow \text{Logistic 模型}$$



$t=t_m$, di/dt 最大

$t_m \sim$ 传染病高潮到来时刻

λ (曰接触率) $\downarrow \rightarrow t_m \uparrow$

$$i(t) = \frac{1}{1 + \left(\frac{1}{i_0} - 1 \right) e^{-\lambda t}}$$

$$t_m = \lambda^{-1} \ln \left(\frac{1}{i_0} - 1 \right)$$

$t \rightarrow \infty \Rightarrow i \rightarrow 1$?

病人可以治愈!

模型3

传染病无免疫性——病人治愈成为健康人，健康人可再次被感染

SIS 模型

增加假设

3) 病人每天治愈的比例为 μ $\mu \sim$ 日治愈率

建模

$$N[i(t + \Delta t) - i(t)] = \lambda Ns(t)i(t)\Delta t - \mu Ni(t)\Delta t$$



$$\begin{cases} \frac{di}{dt} = \lambda i(1-i) - \mu i \\ i(0) = i_0 \end{cases}$$

$\lambda \sim$ 日接触率

$1/\mu \sim$ 感染期

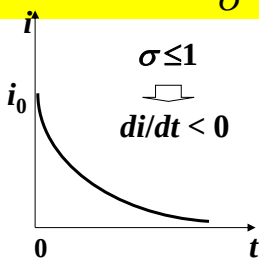
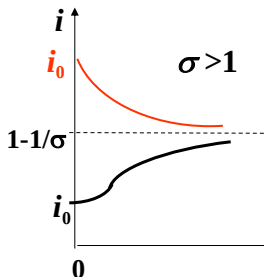
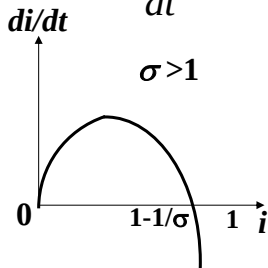
$$\sigma = \lambda / \mu$$

$\sigma \sim$ 一个感染期内每个病人的有效接触人数，称为**接触数**。

模型3

$$\frac{di}{dt} = \lambda i(1-i) - \mu i \quad \sigma = \lambda / \mu$$

$$\frac{di}{dt} = -\lambda i \left[i - \left(1 - \frac{1}{\sigma} \right) \right]$$



$$i(\infty) = \begin{cases} 1 - \frac{1}{\sigma}, & \sigma > 1 \\ 0, & \sigma \leq 1 \end{cases}$$

接触数 $\sigma = 1 \sim$ 阈值

$$\sigma \leq 1 \Rightarrow i(t) \downarrow$$

$\sigma > 1$

i_0 小

$\Rightarrow i(t)$ 按 S 形曲线增长

感染期内有效接触感染的健康者人数不超过病人数

模型2(SI模型)如何看作模型3(SIS模型)的特例

模型4

传染病有免疫性——病人治愈后即移出感染系统，称移出者

SIR模型

假设

1) 总人数 N 不变，病人、健康人和移出者的比例分别为 $i(t), s(t), r(t)$

2) 病人的日接触率 λ ，日治愈率 μ ，
接触数 $\sigma = \lambda / \mu$

建模

$$s(t) + i(t) + r(t) = 1$$

需建立 $i(t), s(t), r(t)$ 的两个方程

模型4

SIR模型

$$N[i(t + \Delta t) - i(t)] = \lambda N s(t) i(t) \Delta t - \mu N i(t) \Delta t$$

$$N[s(t + \Delta t) - s(t)] = -\lambda N s(t) i(t) \Delta t$$

$$\Rightarrow \begin{cases} \frac{di}{dt} = \lambda si - \mu i \\ \frac{ds}{dt} = -\lambda si \\ i(0) = i_0, s(0) = s_0 \end{cases} \Rightarrow$$

无法求出 $i(t), s(t)$
的解析解



在相平面 $s \sim i$ 上
研究解的性质

$i_0 + s_0 \approx 1$ (通常 $r(0) = r_0$ 很小)

模型4

$$\begin{cases} \frac{di}{dt} = \lambda si - \mu i \\ \frac{ds}{dt} = -\lambda si \\ i(0) = i_0, s(0) = s_0 \end{cases}$$

消去 dt

$$\sigma = \lambda / \mu$$



SIR模型

$$\begin{cases} \frac{di}{ds} = \frac{1}{\sigma s} - 1 \\ i|_{s=s_0} = i_0 \end{cases}$$

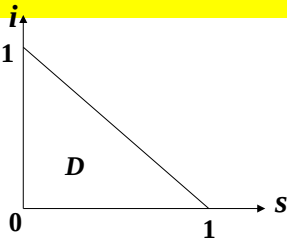
相轨线

$$i(s) = (s_0 + i_0) - s + \frac{1}{\sigma} \ln \frac{s}{s_0}$$

相轨线 $i(s)$ 的定义域

$$D = \{(s, i) | s \geq 0, i \geq 0, s + i \leq 1\}$$

在 D 内作相轨线 $i(s)$
的图形，进行分析



模型4

相轨线 $i(s)$ 及其分析

SIR模型

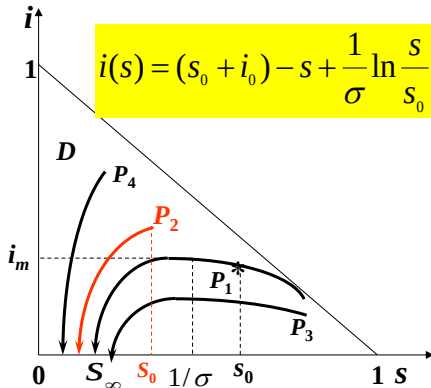
$$\begin{cases} \frac{di}{dt} = \lambda si - \mu i \\ \frac{ds}{dt} = -\lambda si \\ i(0) = i_0, s(0) = s_0 \end{cases}$$

$$\begin{cases} \frac{di}{ds} = \frac{1}{\sigma} - 1 \\ i|_{s=s_0} = i_0 \end{cases}$$

$s(t)$ 单调减 $\rightarrow$ 相轨线的方向

$$s = 1/\sigma, i = i_m \quad t \rightarrow \infty, i \rightarrow 0$$

$$s_\infty \text{ 满足 } s_0 + i_0 - s_\infty + \frac{1}{\sigma} \ln \frac{s_\infty}{s_0} = 0$$



$P_1: s_0 > 1/\sigma \rightarrow i(t)$ 先升后降至 0

$\Rightarrow$ 传染病蔓延

$P_2: s_0 < 1/\sigma \rightarrow i(t)$ 单调降至 0

$\Rightarrow$ 传染病不蔓延

1/
~ 阈
值

模型4

预防传染病蔓延的手段

SIR模型

传染病不蔓延的条件—— $s_0 < 1/\sigma$

- 提高阈值 $1/\sigma$ $\Leftrightarrow$ 降低 $\sigma (= \lambda/\mu)$ $\Leftrightarrow \lambda \downarrow, \mu \uparrow$

λ (日接触率) $\downarrow \Rightarrow$ 卫生水平 $\uparrow$

μ (日治愈率) $\uparrow \Rightarrow$ 医疗水平 $\uparrow$



- 降低 s_0 $\Rightarrow$ 提高 r_0 $\Rightarrow$ 群体免疫

$$s_0 + i_0 + r_0 = 1$$

σ 的估计

$$s_0 + i_0 - s_\infty + \frac{1}{\sigma} \ln \frac{s_\infty}{s_0} = 0 \quad \text{忽略 } i_0$$

$$\sigma = \frac{\ln s_0 - \ln s_\infty}{s_0 - s_\infty}$$

模型4

被传染人数的估计

SIR模型

记被传染人数比例 $x = s_0 - s_\infty$

$$s_0 + i_0 - s_\infty + \frac{1}{\sigma} \ln \frac{s_\infty}{s_0} = 0$$



$$i_0 \cong 0, s_0 \cong 1$$

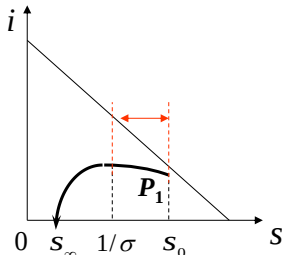
$$x + \frac{1}{\sigma} \ln(1 - \frac{x}{s_0}) \cong 0$$



$$x \ll s_0$$

$$x(1 - \frac{1}{s_0\sigma} - \frac{x}{2s_0^2\sigma}) \cong 0$$

$$\Rightarrow x \approx 2s_0\sigma(s_0 - \frac{1}{\sigma})$$



$$s_0 - 1/\sigma = \delta$$

δ 小, $s_0\sigma \cong 1$

$$x \cong 2\delta$$

提高阈值 $1/\sigma \rightarrow$ 降低
被传染人数比例 x

Discussions

2. 经济增长模型

增加生产 发展经济 增加投资 增加劳动力 提高技术

- 建立产值与资金、劳动力之间的关系
- 研究资金与劳动力的最佳分配，使投资效益最大
- 调节资金与劳动力的增长率，使经济(生产率)增长

1. 道格拉斯(Douglas)生产函数

产值 $Q(t)$ 资金 $K(t)$ 劳动力 $L(t)$
 $\hookleftarrow$ 技术 $f(t) = f_0$

$$Q(t) = f_0 F(K(t), L(t)) \quad F \text{ 为待定函数}$$

1. 道格拉斯(Douglas)生产函数

静态模型

$$Q(K, L) = f_0 F(K, L)$$

每个劳动力的产值 $z = \frac{Q}{L}$

每个劳动力的投资 $y = \frac{K}{L}$

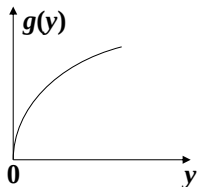
模型假设 z 随着 y 的增加而增长, 但增长速度递减

$$z = Q/L = f_0 g(y) \quad g(y) = y^\alpha, \quad 0 < \alpha < 1$$

$$\Rightarrow Q = f_0 L (K/L)^\alpha$$

$$\Rightarrow Q(K, L) = f_0 K^\alpha L^{1-\alpha} \quad \text{Douglas生产函数}$$

$$\frac{\partial Q}{\partial K}, \frac{\partial Q}{\partial L} > 0 \quad \frac{\partial^2 Q}{\partial K^2}, \frac{\partial^2 Q}{\partial L^2} < 0 \quad \text{含义?}$$



1. Douglas生产函数 $Q(K, L) = f_0 K^\alpha L^{1-\alpha}$

$Q_K \sim$ 单位资金创造的产值

$Q_L \sim$ 单位劳动力创造的产值

$$\frac{KQ_K}{Q} = \alpha, \quad \frac{LQ_L}{Q} = 1 - \alpha$$

$$KQ_K + LQ_L = Q$$

$\alpha \sim$ 资金在产值中的份额

$1-\alpha \sim$ 劳动力在产值中的份额

更一般的道格拉斯(Douglas)生产函数

$$Q(K, L) = f_0 K^\alpha L^\beta, \quad 0 < \alpha, \beta < 1, \quad f_0 > 0$$

2) 资金与劳动力的最佳分配 (静态模型)

资金来自贷款, 利率 r 劳动力付工资 w

资金和劳动力创造的效益 $S = Q - rK - wL$

求资金与劳动力的分配比例 K/L (每个劳动力占有的资金), 使效益 S 最大

$$\begin{aligned} \frac{\partial S}{\partial K} = 0, \frac{\partial S}{\partial L} = 0 &\Rightarrow \frac{Q_K}{Q_L} = \frac{r}{w} \\ \frac{KQ_K}{Q} = \alpha, \frac{LQ_L}{Q} = 1 - \alpha &\Rightarrow \frac{Q_K}{Q_L} = \frac{L}{K} \frac{\alpha}{1 - \alpha} \end{aligned} \Rightarrow \frac{K}{L} = \frac{\alpha}{1 - \alpha} \frac{w}{r}$$

$w \uparrow, r \downarrow, \alpha \uparrow$
 $\Rightarrow K/L \uparrow$

3) 经济(生产率)增长的条件 (动态模型)

要使 $Q(t)$ 或 $Z(t)=Q(t)/L(t)$ 增长, $K(t), L(t)$ 应满足的条件

模型
假设

• 投资增长率与产值成正比
(用一定比例扩大再生产)

$$\Rightarrow \frac{dK}{dt} = \lambda Q, \lambda > 0$$

• 劳动力相对增长率为常数

$$\Rightarrow \frac{dL}{dt} = \mu L \quad \Rightarrow L(t) = L_0 e^{\mu t}$$

$$Q = f_0 L g(y) \quad g(y) = y^\alpha \quad \Rightarrow \frac{dK}{dt} = \lambda f_0 L y^\alpha$$

$$y = \frac{K}{L}, K = Ly \quad \Rightarrow \frac{dK}{dt} = L \frac{dy}{dt} + \mu Ly$$

$$\frac{dK}{dt} = \lambda f_0 L y^\alpha$$



$$\frac{dy}{dt} + \mu y = f_0 \lambda y^\alpha$$

$$\frac{dK}{dt} = L \frac{dy}{dt} + \mu L y$$

Bernoulli方程

$$\Rightarrow y(t) = \left(\frac{f_0 \lambda}{\mu} + (y_0^{1-\alpha} - \frac{f_0 \lambda}{\mu}) e^{-(1-\alpha)\mu t} \right)^{\frac{1}{1-\alpha}}$$

$$y_0 = K_0 / L_0, Q_0 = f_0 K_0^\alpha L_0^{1-\alpha}, \dot{K}_0 = \lambda Q_0 \Rightarrow y_0^{1-\alpha} = f_0 \lambda \frac{K_0}{\dot{K}_0}$$

$$\Rightarrow y(t) = \left\{ \frac{f_0 \lambda}{\mu} \left[1 - \left(1 - \mu \frac{K_0}{\dot{K}_0} \right) e^{-(1-\alpha)\mu t} \right] \right\}^{\frac{1}{1-\alpha}}$$

3) 经济增长的条件

产值 $Q(t)$ 增长



$$dQ/dt > 0$$

$$\begin{aligned} Q &= f_0 L g(y) & \frac{dQ}{dt} &= f_0 L g'(y) \frac{dy}{dt} + f_0 g(y) \frac{dL}{dt} \\ g(y) &= y^\alpha & &= f_0 L y^{2\alpha-1} [f_0 \alpha \lambda + \mu(1-\alpha)y^{1-\alpha}] \end{aligned}$$

$$\frac{dQ}{dt} > 0 \Leftrightarrow \left(1 - \frac{\mu}{\dot{K}_0 / K_0}\right) e^{-(1-\alpha)\mu t} < \frac{1}{1-\alpha} \quad (A)$$

$$\mu > 0 \Rightarrow A \text{ 成立}$$

$$\mu < 0 \Rightarrow \text{当 } t < \frac{1}{(1-\alpha)\mu} \ln(1-\alpha) \left(1 - \frac{\mu}{\dot{K}_0 / K_0}\right), A \text{ 成立}$$

3) 经济增长的条件

每个劳动力的产值 $Z(t)=Q(t)/L(t)$ 增长



$$dZ/dt > 0$$

$$Z(t) = \frac{f_0 L y^\alpha}{L} = f_0 y^\alpha = f_0 \left(\frac{K}{L}\right)^\alpha \Rightarrow \frac{dZ}{dt} = f_0 \alpha y^{\alpha-1} \frac{dy}{dt}$$

$$\frac{dZ}{dt} > 0 \Leftrightarrow \frac{dy}{dt} > 0 \Leftrightarrow \left(1 - \frac{\mu}{\dot{K}_0 / K_0}\right) e^{-(1-\alpha)\mu t} > 0 \quad (B)$$

$$\mu < 0 \Rightarrow B \text{ 成立} \quad \mu > 0 \Rightarrow \text{当 } \frac{\mu}{\dot{K}_0 / K_0} < 1 \text{ 时, } B \text{ 成立}$$

劳动力增长率小于初始投资增长率

Discussions

3. 正规战与游击战

第一次世界大战Lanchester提出

预测战役结局的模型

战争分类：正规战争，游击战争，混合战争

只考虑双方兵力多少和战斗力强弱

兵力因战斗及非战斗减员而减少，因增援而增加

战斗力与射击次数及命中率有关

建模思路和方法为用数学模型讨论社会领域的实际问题提供了可借鉴的示例

一般模型

$x(t)$ ~ 甲方兵力, $y(t)$ ~ 乙方兵力

模型假设

- 每方战斗减员率取决于双方的兵力和战斗力
- 每方非战斗减员率与本方兵力成正比
- 甲乙双方的增援率为 $u(t), v(t)$

模型

$$\begin{cases} \dot{x}(t) = f(x, y) - \alpha x + u(t), & \alpha > 0 \\ \dot{y}(t) = g(x, y) - \beta y + v(t), & \beta > 0 \end{cases}$$

f, g 取决于战争类型

正规战争模型

双方均以正规部队作战

- 甲方战斗减员率只取决于乙方的兵力和战斗力

$f(x, y) = -ay$, $a \sim$ 乙方每个士兵的杀伤率

$a = r_y p_y$, $r_y \sim$ 射击率, $p_y \sim$ 命中率

$$\begin{cases} \dot{x} = -ay - \alpha x + u(t) \\ \dot{y} = -bx - \beta y + v(t) \end{cases} \quad g = -bx, b = r_x p_x$$

- 忽略非战斗减员
- 假设没有增援



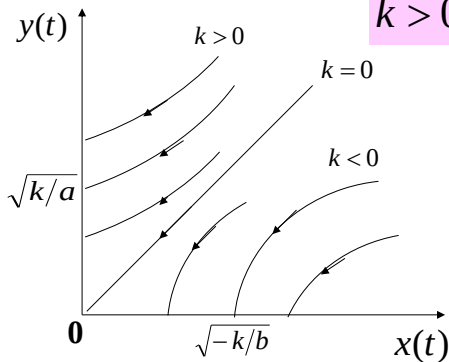
$$\begin{cases} \dot{x} = -ay \\ \dot{y} = -bx \\ x(0) = x_0, y(0) = y_0 \end{cases}$$

正规战争模型

$$\begin{cases} \dot{x} = -ay \\ \dot{y} = -bx \\ x(0) = x_0, y(0) = y_0 \end{cases}$$

为判断战争的结局，不求 $x(t), y(t)$ 而在相平面上讨论 x 与 y 的关系

$$\Rightarrow \frac{dy}{dx} = \frac{bx}{ay} \Rightarrow ay^2 - bx^2 = k$$
$$k = ay_0^2 - bx_0^2$$



$k > 0 \Rightarrow x = 0$ 时 $y > 0$ 乙方胜

$$\left(\frac{y_0}{x_0} \right)^2 > \frac{b}{a} = \frac{r_x p_x}{r_y p_y}$$

平方律模型

$k < 0 \Rightarrow$ 甲方胜

$k = 0 \Rightarrow$ 平局

游击战争模型

双方都用游击部队作战

- 甲方战斗减员率还随着甲方兵力的增加而增加

$f(x, y) = -cxy$, $c \sim$ 乙方每个士兵的杀伤率

$$c = r_y p_y$$

$r_y \sim$ 射击率

$p_y \sim$ 命中率

$$p_y = s_{ry} / s_x$$

$s_x \sim$ 甲方活动面积

$s_{ry} \sim$ 乙方射击有效面积

$$g(x, y) = -dxy, \quad d = r_x p_x = r_x s_{rx} / s_y$$

- 忽略非战斗减员
- 假设没有增援

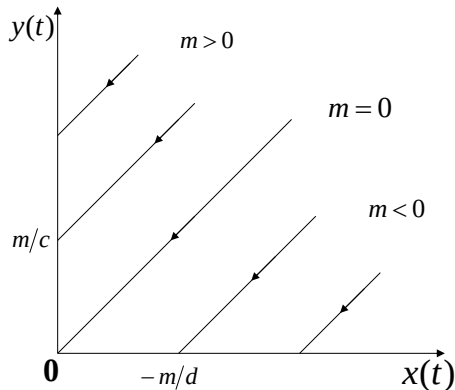


$$\begin{cases} \dot{x} = -cxy \\ \dot{y} = -dxy \\ x(0) = x_0, y(0) = y_0 \end{cases}$$

游击战争模型

$$\begin{cases} \dot{x} = -cxy \\ \dot{y} = -dxy \\ x(0) = x_0, y(0) = y_0 \end{cases}$$

$$\Rightarrow \frac{dy}{dx} = \frac{d}{c} \Rightarrow \begin{cases} cy - dx = m \\ m = cy_0 - dx_0 \end{cases}$$



$m > 0 \Rightarrow x = 0$ 时 $y > 0$
 $\Rightarrow$ 乙方胜



$$\frac{y_0}{x_0} > \frac{d}{c} = \frac{r_x s_{rx} s_x}{r_y s_{ry} s_y}$$

线性律
模型

$m < 0 \Rightarrow$ 甲方胜

$m = 0 \Rightarrow$ 平局

混合战争模型

甲方为游击部队，乙方为正规部队

$$\begin{cases} \dot{x} = -cxy \\ \dot{y} = -bx \\ x(0) = x_0, y(0) = y_0 \end{cases}$$

$$\Rightarrow \begin{aligned} cy^2 - 2bx &= n \\ n &= cy_0^2 - 2bx_0 \end{aligned}$$

$n > 0$
乙方胜

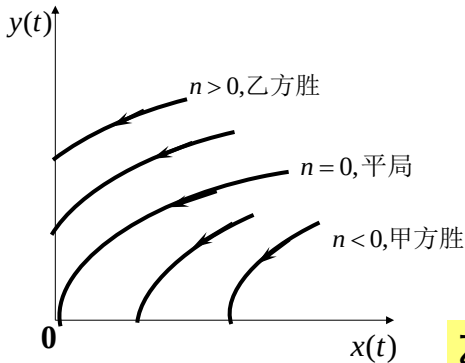
$$\Rightarrow \left(\frac{y_0}{x_0} \right)^2 > \frac{2b}{cx_0}$$

$$\Rightarrow \left(\frac{y_0}{x_0} \right)^2 > \frac{2r_x p_x s_x}{r_y s_{ry} x_0}$$

设 $x_0=100, r_x/r_y=1/2, p_x=0.1,$
 $s_x=1(\text{km}^2), s_{ry}=1(\text{m}^2)$

$$\Rightarrow (y_0 / x_0)^2 > 100$$

乙方必须10倍于甲方的兵力



Discussions

4. 药物在体内的分布 与排除

- 药物进入机体形成**血药浓度**（单位体积血液的药物量）
- 血药浓度需保持在一定范围内——**给药方案设计**
- 药物在体内吸收、分布和排除过程 ——药物动力学
- 建立**房室模型**——药物动力学的基本步骤
- 房室——机体的一部分，药物在一个房室内均匀分布（血药浓度为常数），在房室间按一定规律转移
- 本节讨论**二室模型**——中心室（心、肺、肾等）和周边室（四肢、肌肉等）

模型假设

- 中心室(1)和周边室(2), 容积不变
- 药物从体外进入中心室, 在二室间相互转移, 从中心室排出体外
- 药物在房室间转移速率及向体外排除速率, 与该室血药浓度成正比

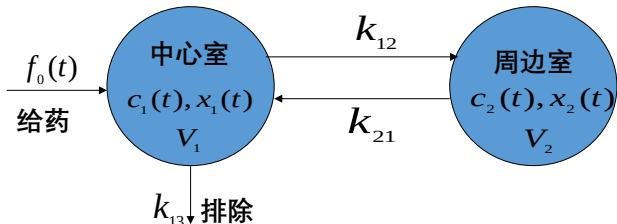
模型建立

$x_i(t) \sim$ 药量

$c_i(t) \sim$ 浓度

$V_i \sim$ 容积

$i=1,2$



$$\dot{x}_1(t) = -k_{12}x_1 - k_{13}x_1 + k_{21}x_2 + f_0(t)$$

$$\dot{x}_2(t) = k_{12}x_1 - k_{21}x_2$$

$$f_0 \sim \text{给药速率}$$

模型建立

$$x_i(t) = V_i c_i(t), i = 1, 2$$

$$\begin{cases} \dot{c}_1(t) = -(k_{12} + k_{13})c_1 + \frac{V_2}{V_1}k_{21}c_2 + \frac{f_0(t)}{V_1} \\ \dot{c}_2(t) = \frac{V_1}{V_2}k_{12}c_1 - k_{21}c_2 \end{cases}$$

线性常系数
非齐次方程

对应齐次
方程通解

$$\begin{cases} \bar{c}_1(t) = A_1 e^{-\alpha t} + B_1 e^{-\beta t} \\ \bar{c}_2(t) = A_2 e^{-\alpha t} + B_2 e^{-\beta t} \end{cases}$$

$$\begin{cases} \alpha + \beta = k_{12} + k_{21} + k_{13} \\ \alpha\beta = k_{21}k_{13} \end{cases}$$

几种常见的给药方式

给药速率 $f_0(t)$
和初始条件

1. 快速静脉注射

$$\begin{cases} \dot{c}_1(t) = -(k_{12} + k_{13})c_1 + \frac{V_2}{V_1}k_{21}c_2 + \frac{f_0(t)}{V_1} \\ \dot{c}_2(t) = \frac{V_1}{V_2}k_{12}c_1 - k_{21}c_2 \end{cases}$$

$t=0$ 瞬时注射剂量 D_0
的药物进入中心室, 血
药浓度立即为 D_0/V_1

$$f_0(t) = 0, c_1(0) = \frac{D_0}{V_1}, c_2(0) = 0$$

$$c_1(t) = \frac{D_0}{V_1(\beta - \alpha)} [(k_{21} - \alpha)e^{-\alpha t} + (\beta - k_{21})e^{-\beta t}]$$

$$c_2(t) = \frac{D_0 k_{12}}{V_2(\beta - \alpha)} (e^{-\alpha t} - e^{-\beta t})$$

$$\begin{cases} \alpha + \beta = k_{12} + k_{21} + k_{13} \\ \alpha\beta = k_{21}k_{13} \end{cases}$$

2. 恒速静脉滴注

$0 \leq t \leq T$ 药物以速率 k_0 进入中心室

$$\begin{cases} \dot{c}_1(t) = -(k_{12} + k_{13})c_1 + \frac{V_2}{V_1}k_{21}c_2 + \frac{f_0(t)}{V_1} \\ \dot{c}_2(t) = \frac{V_1}{V_2}k_{12}c_1 - k_{21}c_2 \end{cases}$$

$$f_0(t) = k_0, c_1(0) = 0, c_2(0) = 0$$

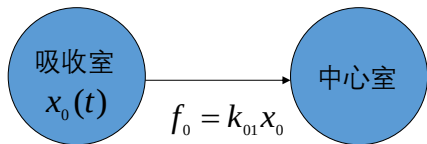
$$\begin{cases} c_1(t) = A_1 e^{-\alpha t} + B_1 e^{-\beta t} + \frac{k_0}{k_{13} V_1}, & 0 \leq t \leq T \\ c_2(t) = A_2 e^{-\alpha t} + B_2 e^{-\beta t} + \frac{k_{12} k_0}{k_{21} k_{13} V_2}, & 0 \leq t \leq T \\ A_2 = \frac{V_1(k_{12} + k_{13} - \alpha)}{k_{21} V_2} A_1, B_2 = \frac{V_1(k_{12} + k_{13} - \beta)}{k_{21} V_2} B_1 \end{cases}$$

$t > T$, $c_1(t)$ 和 $c_2(t)$ 按指数规律趋于零

3. 口服或肌肉注射



相当于药物(剂量 D_0)先进入吸收室，吸收后进入中心室



吸收室药量 $x_0(t)$

$$\begin{cases} \dot{x}_0(t) = -k_{01}x_0 \\ x_0(0) = D_0 \end{cases}$$

$$\begin{cases} \dot{c}_1(t) = -(k_{12} + k_{13})c_1 + \frac{V_2}{V_1}k_{21}c_2 + \frac{f_0(t)}{V_1} \\ \dot{c}_2(t) = \frac{V_1}{V_2}k_{12}c_1 - k_{21}c_2 \end{cases}$$

$$x_0(t) = D_0 e^{-k_{01}t} \quad f_0(t) = k_{01}x_0(t) = D_0 k_{01} e^{-k_{01}t}$$

$$c_1(t) = Ae^{-\alpha t} + Be^{-\beta t} + Ee^{-k_{01}t}$$

$$c_1(0) = 0, c_2(0) = 0 \Rightarrow A, B, E$$

参数估计

各种给药方式下的 $c_1(t)$, $c_2(t)$
取决于参数 k_{12} , k_{21} , k_{13} , V_1, V_2

$t=0$ 快速静脉注射 D_0 , 在 $t_i (i=1,2,\dots,n)$ 测得 $c_1(t_i)$

$$c_1(t) = \frac{D_0}{V_1(\beta - \alpha)} [(k_{21} - \alpha)e^{-\alpha t} + (\beta - k_{21})e^{-\beta t}]$$

设 $\alpha < \beta$, t 充分大 $\Rightarrow c_1(t) = \frac{D_0(k_{21} - \alpha)}{V_1(\beta - \alpha)} e^{-\alpha t} = Ae^{-\alpha t}$

由较大的 t_i , $c_1(t_i)$ 用最小二乘法定 A, α

$$\tilde{c}_1(t) = c_1(t) - Ae^{-\alpha t} = Be^{-\beta t}$$

由较小的 t_i , $\tilde{c}_1(t_i)$ 用最小二乘法定 B, β

参数估计

$t \rightarrow \infty, c_1, c_2 \rightarrow 0$ $\Rightarrow$ 进入中心室的药物全部排除

$$D_0 = k_{13} V_1 \int_0^{\infty} c_1(t) dt \Rightarrow D_0 = k_{13} V_1 \left(\frac{A}{\alpha} + \frac{B}{\beta} \right)$$

$$c_1(0) = \frac{D_0}{V_1} = A + B \Rightarrow k_{13} = \frac{\alpha\beta(A + B)}{\alpha B + \beta A}$$

$$\begin{cases} \alpha + \beta = k_{12} + k_{21} + k_{13} \\ \alpha\beta = k_{21}k_{13} \end{cases} \Rightarrow k_{21} = \frac{\alpha\beta}{k_{13}}$$

$$k_{12} = \alpha + \beta - k_{13} - k_{21}$$

Discussions

5. 香烟过滤嘴的作用

问题

- 过滤嘴的作用与它的材料和长度有什么关系
- 人体吸入的毒物量与哪些因素有关，其中哪些因素影响大，哪些因素影响小。

模型分析

- 分析吸烟时毒物进入人体的过程，建立吸烟过程的数学模型。
- 设想一个“机器人”在典型环境下吸烟，吸烟方式和外部环境认为是不变的。

模型假设

- 1) $l_1 \sim$ 烟草长, $l_2 \sim$ 过滤嘴长, $l = l_1 + l_2$, 毒物量 M 均匀分布, 密度 $w_0 = M/l_1$
- 2) 点燃处毒物随烟雾进入空气和沿香烟穿行的数量比是 $a' : a$, $a' + a = 1$
- 3) 未点燃的烟草和过滤嘴对随烟雾穿行的毒物的(单位时间)吸收率分别是 b 和 β
- 4) 烟雾沿香烟穿行速度是常数 v , 香烟燃烧速度是常数 u , $v \gg u$

定性分析

$Q \sim$ 吸一支烟毒物进入人体总量

$$\beta \uparrow, l_2 \uparrow, M \downarrow, a \downarrow, v \downarrow \Rightarrow Q \downarrow$$

$$b \uparrow, l_1 \uparrow \Rightarrow Q \downarrow? \quad u \uparrow \Rightarrow Q \uparrow \downarrow?$$

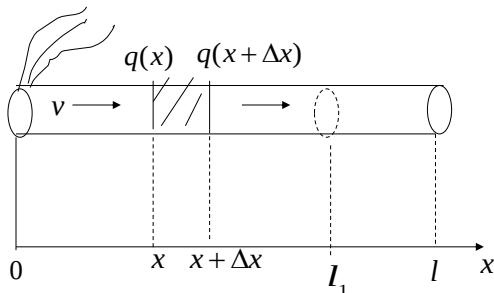
模型建立

$t=0, x=0$, 点燃香烟

$q(x,t) \sim$ 毒物流量

$w(x,t) \sim$ 毒物密度

$w(x,0) = w_0$



$$Q = \int_0^T q(l, t) dt, \quad T = l_1 / u$$

1) 求 $q(x,0)=q(x)$

$$q(x) - q(x + \Delta x) = \begin{cases} bq(x)\Delta\tau, & 0 \leq x \leq l_1, \\ \beta q(x)\Delta\tau, & l_1 \leq x \leq l, \end{cases} \quad \Delta\tau = \frac{\Delta x}{v}$$

$$\frac{dq}{dx} = \begin{cases} -\frac{b}{v} q(x), & 0 \leq x \leq l_1 \\ -\frac{\beta}{v} q(x), & l_1 \leq x \leq l \end{cases}$$

$$q(0) = aH_0$$

$$H_0 = uw_0$$

1) 求 $q(x,0)=q(x)$

$$q(x) = \begin{cases} aH_0 e^{-\frac{bx}{v}}, & 0 \leq x \leq l_1 \\ aH_0 e^{-\frac{bl_1}{v}} e^{-\frac{\beta(x-l_1)}{v}}, & l_1 \leq x \leq l \end{cases}$$

2) 求 $q(l,t)$

t 时刻, 香烟燃至 $x=ut$

$$H(t) = uw(ut, t)$$

$$q(x, t) = \begin{cases} aH(t) e^{-\frac{b(x-ut)}{v}}, & ut \leq x \leq l_1 \\ aH(t) e^{-\frac{b(l_1-ut)}{v}} e^{-\frac{\beta(x-l_1)}{v}}, & l_1 \leq x \leq l \end{cases}$$

$$q(l, t) = auw(ut, t) e^{-\frac{b(l_1-ut)}{v}} e^{-\frac{\beta l_2}{v}}$$

3) 求 $w(ut, t)$

$$w(x, t + \Delta t) - w(x, t) = b \frac{q(x, t)}{v} \Delta t$$

$$\begin{cases} \frac{\partial w}{\partial t} = \frac{b}{v} a u w(ut, t) e^{-\frac{b(x-ut)}{v}} \\ w(x, 0) = w_0 \end{cases}$$

$$w(ut, t) = \frac{w_0}{a'} \left(1 - a e^{-\frac{a' b u t}{v}} \right), \quad a' = 1 - a$$

4) 计算 Q

$$w(ut, t) = \frac{w_0}{a'} \left(1 - ae^{-\frac{a'but}{v}} \right)$$

$$q(l, t) = auw(ut, t)e^{-\frac{b(l_1-ut)}{v}} e^{-\frac{\beta l_2}{v}}$$

$$q(l, t) = \frac{auw_0}{a'} e^{-\frac{bl_1}{v}} e^{-\frac{\beta l_2}{v}} \left(e^{-\frac{but}{v}} - ae^{-\frac{abut}{v}} \right)$$

$$Q = \int_0^{l_1/u} q(l, t) dt = \frac{aw_0v}{a'b} e^{-\frac{\beta l_2}{v}} \left(1 - e^{-\frac{a'bl_1}{v}} \right)$$

$$Q = aMe^{-\frac{\beta l_2}{v}} \varphi(r),$$

$$r = \frac{a'bl_1}{v}, \varphi(r) = \frac{1 - e^{-r}}{r}$$

结果分析

$$Q = aMe^{-\frac{\beta l_2}{v}} \varphi(r), \quad r = \frac{a'bl_1}{v}, \quad \varphi(r) = \frac{1-e^{-r}}{r}$$

1) Q 与 a, M 成正比, aM 是毒物集中在 $x=l$ 处的吸入量

2) $e^{-\frac{\beta l_2}{v}} \sim$ 过滤嘴因素, $\beta, l_2 \sim$ 负指数作用

$aMe^{-\frac{\beta l_2}{v}}$ 是毒物集中在 $x=l_1$ 处的吸入量

3) $\varphi(r) \sim$ 烟草的吸收作用

烟草为什么有作用?

$$r = \frac{a'bl_1}{v} \ll 1 \quad \varphi(r) \doteq 1 - r/2$$

$$Q \doteq aMe^{-\frac{\beta l_2}{v}} \left(1 - \frac{a'bl_1}{2v} \right)$$

$b, l_1 \sim$ 线性作用

结果 分析

4) 与另一支不带过滤嘴的香烟比较, w_0, b, a, v, l 均相同, 吸至 $x=l_1$ 扔掉

带过滤嘴

$$Q_1 = \frac{aw_0v}{a'b} e^{-\frac{\beta l_2}{v}} \left(1 - e^{-\frac{a'bl_1}{v}} \right)$$

不带过滤嘴

$$Q_2 = \frac{aw_0v}{a'b} e^{-\frac{bl_2}{v}} \left(1 - e^{-\frac{a'bl_1}{v}} \right)$$

$$\frac{Q_1}{Q_2} = e^{-\frac{(\beta-b)l_2}{v}}$$

$$\beta > b \Rightarrow Q_1 < Q_2$$

提高 βb 与加长 l_2 , 效果相同

Discussions

6. 人口预测和控制

人口 发展 方程

- 年龄分布对于人口预测的重要性
- 只考虑自然出生与死亡，不计迁移

$F(r, t) \sim$ 人口分布函数 (年龄 $< r$ 的人口)

$p(r, t) \sim$ 人口密度函数

$N(t) \sim$ 人口总数

$r_m (\rightarrow \infty) \sim$ 最高年龄

$$F(0, t) = 0, F(r_m, t) = N(t)$$

$$p(r, t) = \frac{\partial F}{\partial r}$$

人口发展方程

$\mu(r,t) \sim$ 死亡率

$$\begin{array}{ccc} \begin{array}{c} t, \text{ 年龄 } [r, r \\ + dr] \text{ 人数} \end{array} & - & \begin{array}{c} t + dt, \text{ 年龄 } [r + dr_1, \\ r + dr_1 + dr] \text{ 人数} \end{array} \end{array} \quad \frac{\text{——}}{dt = dr_1} \quad \begin{array}{c} (t, t + dt) \text{ 内} \\ \text{死亡人数} \end{array}$$

$$p(r,t)dr - p(r + dr_1, t + dt)dr = \mu(r,t)p(r,t)drdt$$

$$\begin{aligned} & [p(r + dr_1, t + dt) - p(r, t + dt)] + [p(r, t + dt) - p(r, t)] \\ & = -\mu(r,t)p(r,t)dt, \quad dt = dr_1 \end{aligned}$$

$$\frac{\partial p}{\partial r} + \frac{\partial p}{\partial t} = -\mu(r,t)p(r,t)$$

一阶偏微分方程

人口发展方程

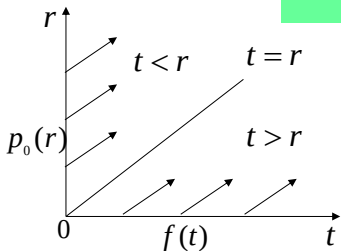
$$\begin{cases} \frac{\partial p}{\partial r} + \frac{\partial p}{\partial t} = -\mu(r,t)p(r,t) \\ p(r,0) = p_0(r), \quad r \geq 0 \\ p(0,t) = f(t), \quad t \geq 0 \end{cases}$$

~已知函数（人口调查）

~生育率（控制人口手段）

$$\mu(r,t) = \mu(r) \Rightarrow$$

$$p(r,t) = \begin{cases} p_0(r-t)e^{-\int_{r-t}^r \mu(s)ds}, & 0 \leq t \leq r \\ f(t-r)e^{-\int_0^{t-r} \mu(s)ds}, & t > r \end{cases}$$



$$F(r,t) = \int_0^r p(s,t)ds$$

$$N(t) = \int_0^{r_m} p(s,t)ds$$

生育率的分解



$k(r, t) \sim$ (女性)性别比函数

$b(r, t) \sim$ (女性)生育数 $[r_1, r_2] \sim$ 育龄区间

$$f(t) = \int_{r_1}^{r_2} b(r, t) k(r, t) p(r, t) dr$$

$$b(r, t) = \beta(t) h(r, t)$$

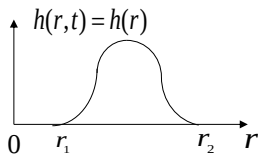
$$\int_{r_1}^{r_2} h(r, t) dr = 1$$

$h \sim$ 生育模式

$$\beta(t) = \int_{r_1}^{r_2} b(r, t) dr$$

$\beta \sim$ 总和生育率

$$f(t) = \beta(t) \int_{r_1}^{r_2} h(r, t) k(r, t) p(r, t) dr$$



人口发展方程和生育率

$$f(t) = \beta(t) \int_{r_1}^{r_2} h(r, t) k(r, t) p(r, t) dr$$

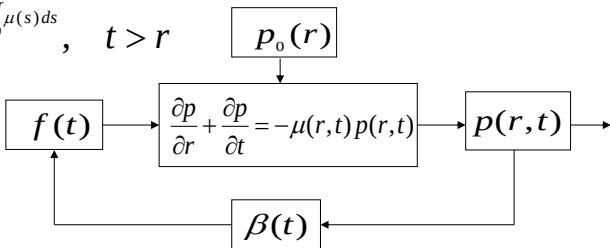
$\beta(t)$ ~ 总和生育率——控制生育的多少

$h(r, t)$ ~ 生育模式——控制生育的早晚和疏密

$$p(r, t) = \begin{cases} p_0(r-t) e^{-\int_{r-t}^r \mu(s) ds}, & 0 \leq t \leq r \\ f(t-r) e^{-\int_0^r \mu(s) ds}, & t > r \end{cases}$$

• 正反馈系统

• 滞后作用很大



人口指数

1) 人口总数 $N(t) = \int_0^{r_m} p(r, t) dr$

2) 平均年龄 $R(t) = \frac{1}{N(t)} \int_0^{r_m} r p(r, t) dr$

3) 平均寿命 $S(t) = \int_t^{\infty} e^{-\int_0^{\tau-t} \mu(r, t) dr} d\tau$

t 时刻出生的人，死亡率按 $\mu(r, t)$ 计算的平均存活时间

4) 老龄化指数 $\omega(t) = R(t) / S(t)$

控制生育率



控制 $N(t)$ 不过大

控制 $\omega(t)$ 不过高

Discussions

7. 烟雾的扩散与消失

现象 和 问题

炮弹在空中爆炸，烟雾向四周扩散，形成圆形不透光区域。

不透光区域不断扩大，然后区域边界逐渐明亮，区域缩小，最后烟雾消失。

建立模型描述烟雾扩散和消失过程，分析消失时间与各因素的关系。

问题 分析

无穷空间由瞬时点源导致的扩散过程，用二阶偏微分方程描述烟雾浓度的变化。

观察的烟雾消失与烟雾对光线的吸收，以及仪器对明暗的灵敏程度有关。

模型 假设

- 1) 烟雾在无穷空间扩散，不受大地和风的影响；扩散服从热传导定律。
- 2) 光线穿过烟雾时光强的减少与烟雾浓度成正比；无烟雾的大气不影响光强。
- 3) 穿过烟雾进入仪器的光线只有明暗之分，明暗界限由仪器灵敏度决定。

模型 建立

- 1) 烟雾浓度 $C(x, y, z, t)$ 的变化规律

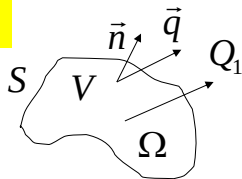
热传导定律：单位时间通过单位法向面积的流量与浓度梯度成正比

$$\vec{q} = -k \cdot \text{grad}C$$

1) 烟雾浓度 $C(x, y, z, t)$ 的变化规律

$[t, t + \Delta t]$ 通过 Ω 流量

$$Q_1 = \int_t^{t+\Delta t} \iint_S \vec{q} \cdot \vec{n} d\sigma dt$$



Ω 内烟雾改变量 $Q_2 = \iiint_V [C(x, y, z, t) - C(x, y, z, t + \Delta t)] dV$

曲面积分的奥氏公式

$$\iint_S \vec{q} \cdot \vec{n} d\sigma = \iiint_V \text{div} \vec{q} dV$$

$$Q_1 = Q_2$$

$$\vec{q} = -k \cdot \text{grad} C$$

$$\frac{\partial C}{\partial t} = k[\text{div}(\text{grad} C)] = k \left(\frac{\partial^2 C}{\partial x^2} + \frac{\partial^2 C}{\partial y^2} + \frac{\partial^2 C}{\partial z^2} \right)$$

1) 烟雾浓度 $C(x, y, z, t)$ 的变化规律

$$\frac{\partial C}{\partial t} = k \left(\frac{\partial^2 C}{\partial x^2} + \frac{\partial^2 C}{\partial y^2} + \frac{\partial^2 C}{\partial z^2} \right), \quad -\infty < x, y, z < \infty, \quad t > 0$$

初始条件

$$C(x, y, z, 0) = Q\delta(x, y, z)$$

Q ~炮弹释放的烟雾总量

δ ~单位强度的点源函数

$$C(x, y, z, t) = \frac{Q}{(4\pi kt)^{3/2}} e^{-\frac{x^2+y^2+z^2}{4kt}}$$

- 对任意 t , C 的等值面是球面 $x^2+y^2+z^2=R^2$; $R\uparrow \rightarrow C\downarrow$
- 仅当 $t\rightarrow\infty$, 对任意点 (x,y,z) , $C\rightarrow 0$

2) 穿过烟雾光强的变化规律

$I(l) \sim$ 沿 l 方向的光强

光强的减少与烟

$C(l) \sim$ 沿 l 方向的烟雾浓度

雾浓度成正比

$$\frac{dI}{dl} = -\alpha C(l) I(l)$$

$I(l_0) = I_0$ 未进入烟雾($l \leq l_0$)的光强为 I_0

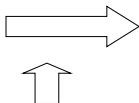
$$\Rightarrow I(l) = I_0 e^{-\alpha \int_{l_0}^l C(s) ds}$$

3) 仪器灵敏度与烟雾明暗界限

$$I(l) = I_0 e^{-\alpha \int_{l_0}^l C(s) ds}$$

烟雾浓度连续变化

烟雾中光强连续变化



不透光区域有扩大、
缩小、消失的过程

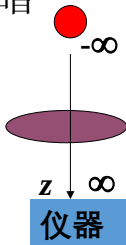
穿过烟雾进入仪器的光线只有明暗之分，明暗界限由仪器灵敏度决定。

$\mu \sim$ 仪器灵敏度，当 $I / I_0 < 1 - \mu$ ，观测结果为暗

设光源在 $z = -\infty$ ，仪器在 $z = \infty$ ，则观测到的明暗界限为

$$e^{-\alpha \int_{-\infty}^{\infty} C(x, y, z, t) dz} = 1 - \mu$$

$\sim$ 不透光区域边界



4) 不透光区域边界的变化规律

$$C(x, y, z, t) = \frac{Q}{(4\pi kt)^{3/2}} e^{-\frac{x^2+y^2+z^2}{4kt}} \quad e^{-\alpha \int_{-\infty}^{\infty} C(x, y, z, t) dz} = 1 - \mu$$

$$\Rightarrow \int_{-\infty}^{\infty} C(x, y, z, t) dz = \frac{1}{\alpha} \ln \frac{1}{1 - \mu} \approx \frac{\mu}{\alpha} \quad (\mu \text{ 很小})$$

$$\int_{-\infty}^{\infty} e^{-x^2/a} dx = \sqrt{\pi a} \quad \Rightarrow \quad \int_{-\infty}^{\infty} C(x, y, z, t) dz = \frac{Q}{4\pi kt} e^{-\frac{x^2+y^2}{4kt}}$$

$$\frac{Q}{4\pi kt} e^{-\frac{x^2+y^2}{4kt}} = \frac{\mu}{\alpha}$$

对任意 t , 不透光区域边界是圆周

$$x^2 + y^2 = r^2$$

不透光区域
边界半径

$$r(t) = \sqrt{4kt \ln \frac{\alpha Q}{4\pi k \mu t}}$$

结果分析

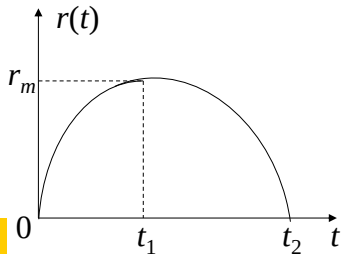
$$r(t) = \sqrt{4kt \ln \frac{\alpha Q}{4\pi k \mu t}}$$

$$t = t_1 = \frac{\alpha Q}{4\pi k \mu e}, \quad r = r_m = \sqrt{\frac{\alpha Q}{\pi \mu e}} \quad (\text{最大值})$$

$$t = t_2 = \frac{\alpha Q}{4\pi k \mu}, \quad r = 0$$



$$t_2 = t_1 \cdot e \doteq 2.7t_1$$



观测到不透光区域边界达到最大的时刻 t_1 ，可以预报烟雾消失的时刻 t_2

$$\alpha \uparrow, Q \uparrow, \mu \downarrow \Rightarrow t_1 \uparrow, r_m \uparrow \quad k \downarrow \Rightarrow t_1 \uparrow$$

Discussions

8. 万有引力定律的发现

背景

航海业发展

天文观测精确

“地心说”动摇

哥白尼：“日心说”

伽里略：落体运动

开普勒：行星运动三定律



变速运动的计算方法

牛顿：一切运动有力学原因

牛顿运动三定律

牛顿：研究变速运动，发明微积分（流数法）

开普勒三定律



牛顿运动第二定律

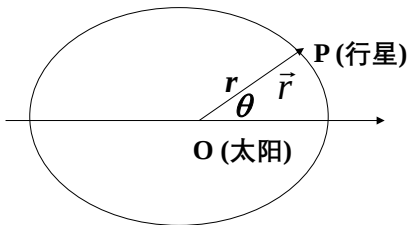
万有引力定律

《自然科学之数学原理》(1687)

模型假设

极坐标系 (r, θ) 太阳 $(0,0)$

行星位置：向径 $\vec{r}(t)(r(t), \theta(t))$



1. 行星轨道

$$r = \frac{p}{1 + e \cos \theta}, \quad p = \frac{b^2}{a}, \quad b^2 = a^2(1 - e^2)$$

$a \sim$ 长半轴, $b \sim$ 短半轴, $e \sim$ 离心率

2. 单位时间 $\vec{r}$ 扫过面积为常数 A

$$r^2 \dot{\theta} / 2 = A$$

3. 行星运行周期 T

$$T^2 = \lambda a^3$$

$\lambda \sim$ 绝对常数

4. 行星运行受力 $\vec{f}$

$$\vec{f} = m \ddot{\vec{r}}$$

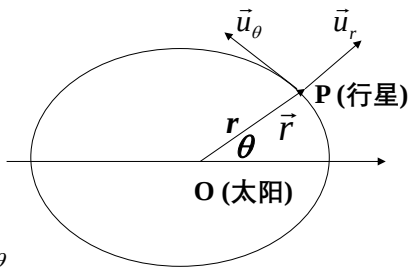
$m \sim$ 行星质量

模型建立 向径 $\vec{r}$ 的基向量

$$\vec{u}_r = \cos \theta \vec{i} + \sin \theta \vec{j}$$

$$\vec{u}_\theta = -\sin \theta \vec{i} + \cos \theta \vec{j}$$

$$\vec{r} = r\vec{u}_r$$



$$\dot{\vec{u}}_r = \dot{\theta} \vec{u}_\theta$$

$$\dot{\vec{r}} = \dot{r}\vec{u}_r + r\dot{\theta}\vec{u}_\theta$$

$$\dot{\vec{u}}_\theta = -\dot{\theta} \vec{u}_r$$

$$\ddot{\vec{r}} = (\ddot{r} - r\dot{\theta}^2)\vec{u}_r + (r\ddot{\theta} + 2\dot{r}\dot{\theta})\vec{u}_\theta$$

$$r^2\dot{\theta}/2 = A \quad \Rightarrow \quad \dot{\theta} = \frac{2A}{r^2}, \quad \ddot{\theta} = \frac{-4A\dot{r}}{r^3}$$

$$r\ddot{\theta} + 2\dot{r}\dot{\theta} = 0$$

$$\ddot{\vec{r}} = (\ddot{r} - r\dot{\theta}^2)\vec{u}_r$$

$$r = \frac{p}{1 + e \cos \theta} \quad \Rightarrow \quad \dot{r} = \frac{2Ae \sin \theta}{p}, \quad \ddot{r} = \frac{4A^2(p-r)}{pr^3} \quad \Rightarrow \quad \ddot{\vec{r}} = -\frac{4A^2}{pr^2}\vec{u}_r$$

$$\begin{aligned} \vec{f} &= m\ddot{\vec{r}} \\ \vec{r} &= r\vec{u}_r \end{aligned}$$

$$\vec{f} = -\frac{4A^2m}{pr^2}\vec{r}_0, \quad \vec{r}_0 = \frac{\vec{r}}{r}$$

模型建立

$$\vec{f} = -\frac{4A^2 m}{pr^2} \vec{r}_0, \quad \vec{r}_0 = \frac{\vec{r}}{r}$$

需证明 $4A^2/p = kM$
(与哪一颗行星无关)

$A \sim$ 单位时间 $\vec{r}$ 扫过面积



$$TA = \pi ab$$

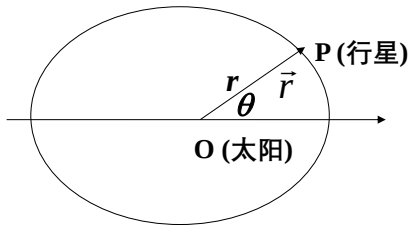
$$r = \frac{p}{1 + e \cos \theta}, \quad p = \frac{b^2}{a}, \quad b^2 = a^2(1 - e^2)$$



$$A^2 / p = \pi^2 / \lambda$$

万有引力定律

$$\vec{f} = -\frac{kMm}{r^2} \vec{r}_0$$



$$T^2 = \lambda a^3$$

$$4\pi^2 / \lambda = kM \quad (\text{习题})$$

Discussions



中国科学技术大学
University of Science and Technology of China

谢谢！



中国科学技术大学
University of Science and Technology of China

数学建模 Mathematical Modeling

刘利刚

中国科学技术大学

数码相机定位

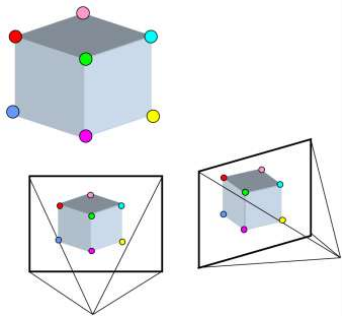
(课本第四章)

机理建模

- 根据问题的实际机理，比如物理规律、几何规律等，进行数学模型
- 机理：
 - 物理规律：牛顿运动学定律、材料力学、流体力学、光学、热力学...
 - 运动方程
 - 偏微分方程
 - 图理论
 - ...

数码相机定位问题

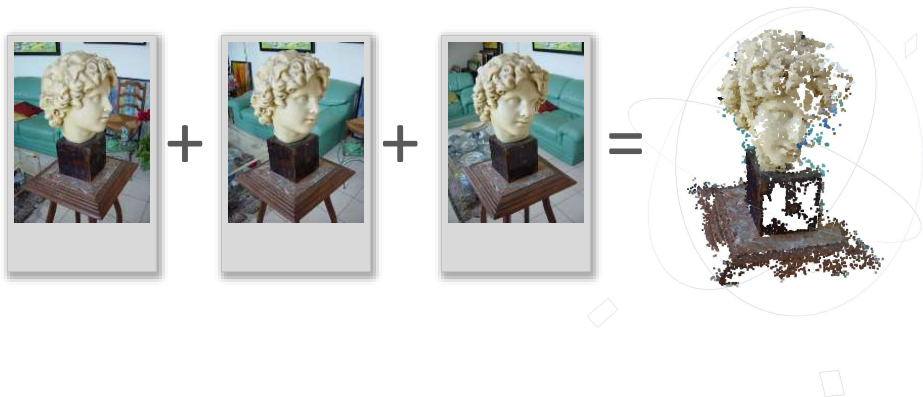
- 两部相机拍摄同一物体，得到两幅图像
- 根据两幅图像的公共特征，可以求得两部相机的相对位置关系
- 也可以求得对应公共特征点的三维坐标
- 机理：
 - 几何光学
 - 针孔成像原理



SFMedu Program with Code

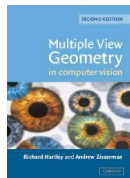
Download from:

<http://3dvision.princeton.edu/courses/SFMedu/>



How: a complete new way of learning

Standard Way



Reading: the MVG bible
(need ~2 years)



Coding



Crying: Not Working At All



Yes

Our Way



Run a program



See pictures



Listen stories

$$\mathbf{x} = \mathbf{P}\mathbf{x}$$

Play with math

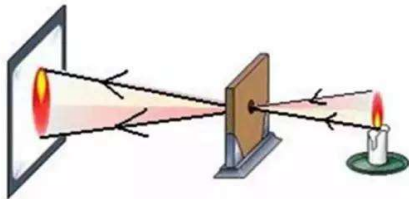


Enjoy&improve

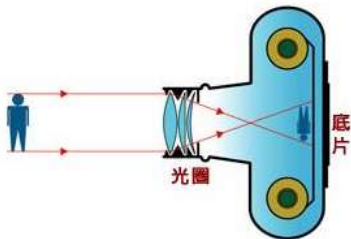
1 hour



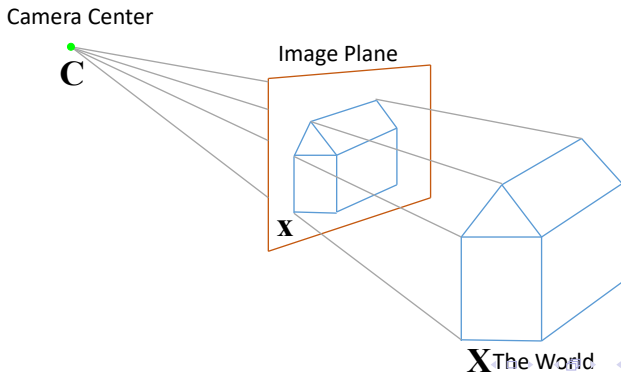
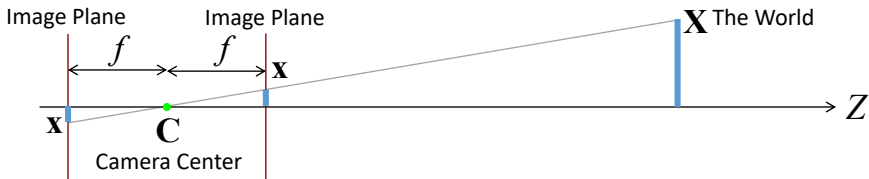
针孔成像



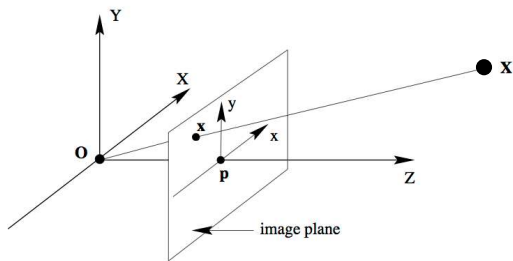
小孔成像



针孔成像的几何模型



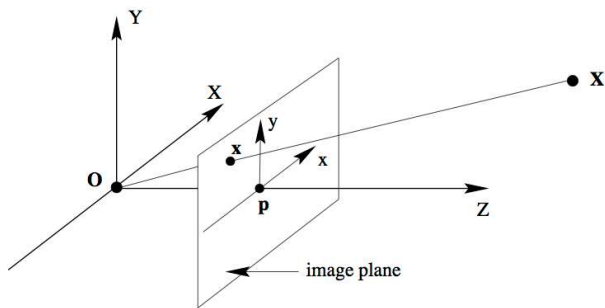
针孔成像的几何模型



$$\mathbf{x} = \begin{bmatrix} \hat{e}_x^T \mathbf{u} \\ \hat{e}_y^T \mathbf{u} \end{bmatrix}$$

$$\mathbf{X} = \begin{bmatrix} \hat{e}_x^T \mathbf{u} \\ \hat{e}_y^T \mathbf{u} \\ \hat{e}_z^T \mathbf{u} \end{bmatrix}$$

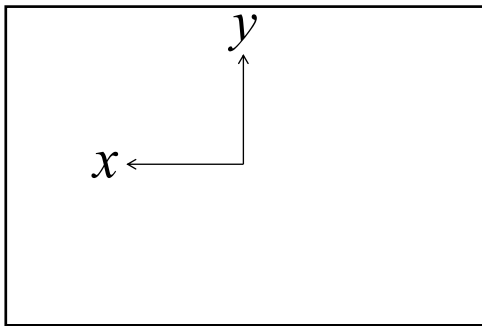
针孔成像的几何模型



$$\begin{bmatrix} x \\ y \\ f \end{bmatrix} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \iff \begin{bmatrix} x \\ y \\ f \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

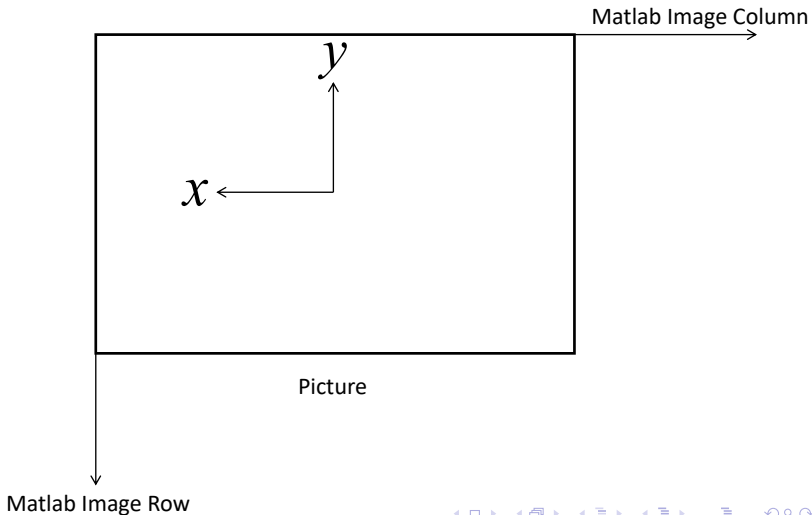
Up to a scale

When they take a picture:

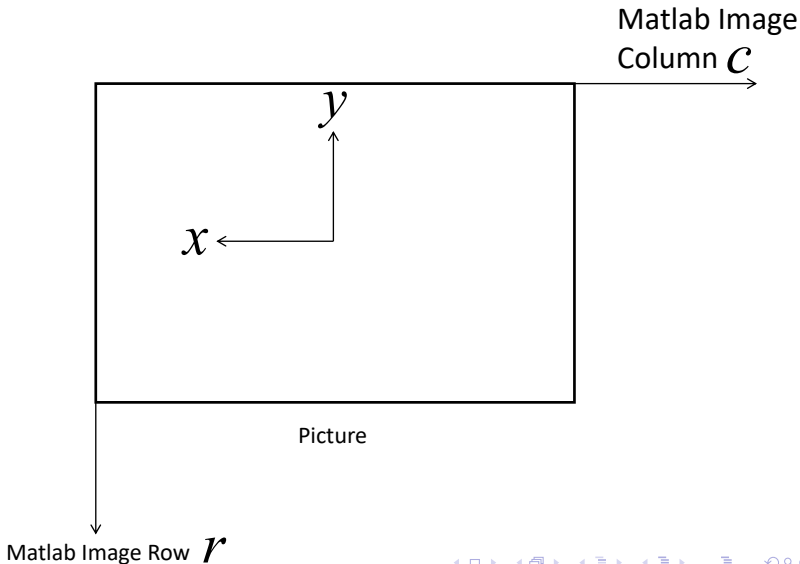


Picture

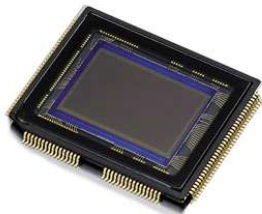
When they take a picture:



When they take a picture:



When they take a picture:



World Unit: e.g. Meters



We let f to take care of this as well.
Unit of f is pixel/world unit.



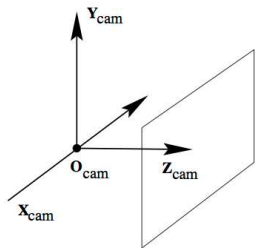
Image Unit: Pixels

$$\begin{array}{c}
 \begin{array}{cc} \text{é} & \text{x} \end{array} \begin{array}{c} \text{ù} \\ \text{é} \\ \text{è} \\ \text{ê} \end{array} = \begin{array}{cc} \begin{array}{c} \text{é} \\ \text{è} \\ \text{ê} \end{array} & \begin{array}{c} \text{ù} \\ \text{é} \\ \text{è} \end{array} \\
 \begin{array}{c} \text{é} \\ \text{è} \\ \text{ê} \end{array} & \begin{array}{c} \text{ù} \\ \text{é} \\ \text{è} \end{array}
 \end{array}
 \begin{array}{cccc}
 1 & 0 & 0 & 0 \\
 0 & 1 & 0 & 0 \\
 0 & 0 & 1 & 0
 \end{array}
 \begin{array}{c}
 \text{é} \text{x} \text{ù} \\
 \text{é} \text{é} \text{ù} \\
 \text{é} \text{è} \text{ù} \\
 \text{é} \text{ê} \text{ù} \\
 \text{é} \text{é} \text{ù} \\
 \text{é} \text{è} \text{ù} \\
 \text{é} \text{ê} \text{ù}
 \end{array}$$

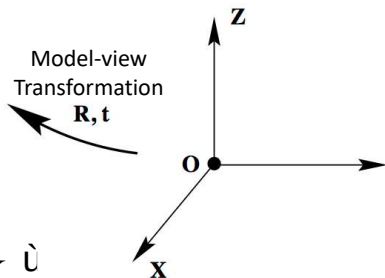


$$\begin{array}{c}
 \begin{array}{cc} \text{é} & \text{x} \end{array} \begin{array}{c} \text{ù} \\ \text{é} \\ \text{è} \\ \text{ê} \end{array} = \begin{array}{cc} \begin{array}{c} \text{é} \\ \text{è} \\ \text{ê} \end{array} & \begin{array}{c} \text{ù} \\ \text{é} \\ \text{è} \end{array} \\
 \begin{array}{c} \text{é} \\ \text{è} \\ \text{ê} \end{array} & \begin{array}{c} \text{ù} \\ \text{é} \\ \text{è} \end{array}
 \end{array}
 \begin{array}{cccc}
 f & 0 & 0 & 0 \\
 0 & f & 0 & 0 \\
 0 & 0 & 1 & 0
 \end{array}
 \begin{array}{c}
 \text{é} \text{x} \text{ù} \\
 \text{é} \text{é} \text{ù} \\
 \text{é} \text{è} \text{ù} \\
 \text{é} \text{ê} \text{ù} \\
 \text{é} \text{é} \text{ù} \\
 \text{é} \text{è} \text{ù} \\
 \text{é} \text{ê} \text{ù}
 \end{array}
 \begin{array}{cccc}
 1 & 0 & 0 & 0 \\
 0 & 1 & 0 & 0 \\
 0 & 0 & 1 & 0
 \end{array}
 \begin{array}{c}
 \text{é} \text{x} \text{ù} \\
 \text{é} \text{é} \text{ù} \\
 \text{é} \text{è} \text{ù} \\
 \text{é} \text{ê} \text{ù} \\
 \text{é} \text{é} \text{ù} \\
 \text{é} \text{è} \text{ù} \\
 \text{é} \text{ê} \text{ù}
 \end{array}$$

相机位姿：相对与世界坐标系的刚体变换 (旋转+平移)



Camera Coordinate System



World Coordinate System

$$\begin{bmatrix} x_{cam} \\ y_{cam} \\ z_{cam} \\ 1 \end{bmatrix} = \begin{bmatrix} \hat{e}_1^T R \\ \hat{e}_2^T R \\ \hat{e}_3^T R \\ \mathbf{0}^T \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \\ t_z \\ 0 \end{bmatrix}$$

世界坐标系到相机坐标系

$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R \\ t \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

$$K = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad x = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad X = \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Camera Parameter
Camera Projection Matrix

$$P = K[R|t]$$

Intrinsic

Extrinsic

$$x = K[R|t]X$$

$$x = PX$$

空间点在相机平面的成像位置

- 1个相机

$$\mathbf{x} = \mathbf{K} \mathbf{e} + \mathbf{t}$$

空间点在相机平面的成像位置

- 2个相机

$$\mathbf{x}_1 = \mathbf{K}_1 \mathbf{R}_1 | \mathbf{t}_1 | \mathbf{X}$$

$$\mathbf{x}_2 = \mathbf{K}_2 \mathbf{R}_2 | \mathbf{t}_2 | \mathbf{X}$$

空间点在相机平面的成像位置

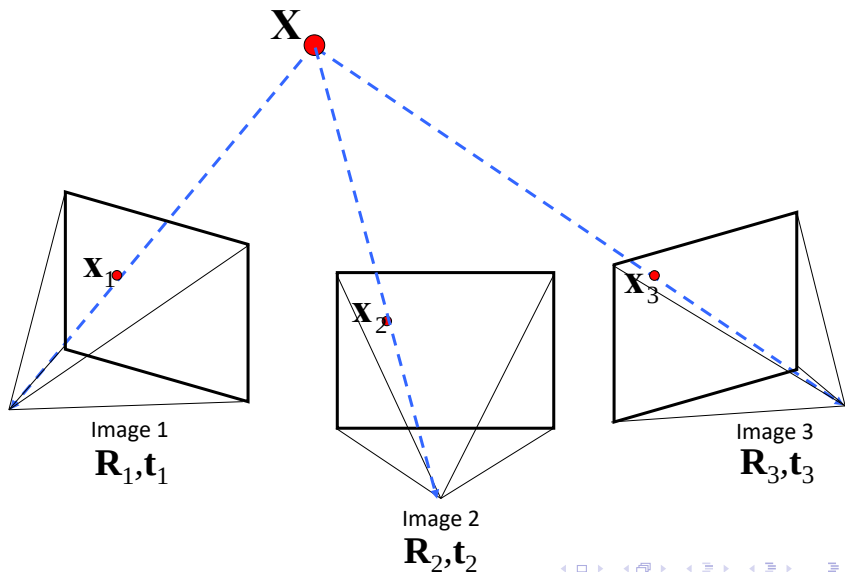
- 3个相机

$$\mathbf{x}_1 = \mathbf{K}_1 \mathbf{R}_1 | \mathbf{t}_1 | \mathbf{X}$$

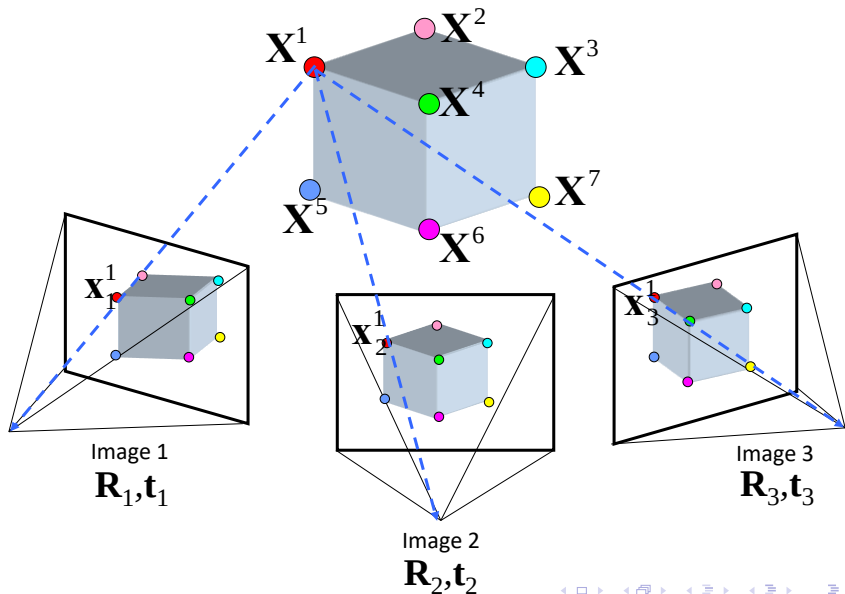
$$\mathbf{x}_2 = \mathbf{K}_2 \mathbf{R}_2 | \mathbf{t}_2 | \mathbf{X}$$

$$\mathbf{x}_3 = \mathbf{K}_3 \mathbf{R}_3 | \mathbf{t}_3 | \mathbf{X}$$

图像的公共对应点



图像的公共对应点



图像的公共对应点

	Point 1	Point 2	Point 3
Image 1	$\mathbf{x}_1^1 = \mathbf{K} \begin{bmatrix} \mathbf{R}_1 & \mathbf{t}_1 \end{bmatrix} \mathbf{X}^1$	$\mathbf{x}_1^2 = \mathbf{K} \begin{bmatrix} \mathbf{R}_1 & \mathbf{t}_1 \end{bmatrix} \mathbf{X}^2$	
Image 2	$\mathbf{x}_2^1 = \mathbf{K} \begin{bmatrix} \mathbf{R}_2 & \mathbf{t}_2 \end{bmatrix} \mathbf{X}^1$	$\mathbf{x}_2^2 = \mathbf{K} \begin{bmatrix} \mathbf{R}_2 & \mathbf{t}_2 \end{bmatrix} \mathbf{X}^2$	$\mathbf{x}_2^3 = \mathbf{K} \begin{bmatrix} \mathbf{R}_2 & \mathbf{t}_2 \end{bmatrix} \mathbf{X}^3$
Image 3	$\mathbf{x}_3^1 = \mathbf{K} \begin{bmatrix} \mathbf{R}_3 & \mathbf{t}_3 \end{bmatrix} \mathbf{X}^1$		$\mathbf{x}_3^3 = \mathbf{K} \begin{bmatrix} \mathbf{R}_3 & \mathbf{t}_3 \end{bmatrix} \mathbf{X}^3$

Same Camera Same Setting = Same $\mathbf{K}$

标定及重建问题

- Input: Observed 2D image position

$$\tilde{\mathbf{x}}_1^1 \quad \tilde{\mathbf{x}}_1^2$$

$$\tilde{\mathbf{x}}_2^1 \quad \tilde{\mathbf{x}}_2^2 \quad \tilde{\mathbf{x}}_2^3$$

- Output:

$$\tilde{\mathbf{x}}_3^1 \quad \tilde{\mathbf{x}}_3^3$$

Unknown Camera Parameters (with some guess)

$$\mathbf{R}_1 | \mathbf{t}_1 \cup \mathbf{R}_2 | \mathbf{t}_2 \cup \mathbf{R}_3 | \mathbf{t}_3$$

Unknown Point 3D coordinate (with some guess)

$$\mathbf{X}^1, \mathbf{X}^2, \mathbf{X}^3, \dots$$

标定及重建问题

A valid solution $\mathbf{e}_1 | \mathbf{t}_1, \mathbf{e}_2 | \mathbf{t}_2, \mathbf{e}_3 | \mathbf{t}_3$ and $\mathbf{X}^1, \mathbf{X}^2, \mathbf{X}^3, \dots$ must let

$$\text{Re-projection} = \left\{ \begin{array}{ll} \mathbf{x}_1^1 = \mathbf{K} \mathbf{e}_1 | \mathbf{t}_1 | \mathbf{X}^1 & \mathbf{x}_1^2 = \mathbf{K} \mathbf{e}_1 | \mathbf{t}_1 | \mathbf{X}^2 \\ \mathbf{x}_2^1 = \mathbf{K} \mathbf{e}_2 | \mathbf{t}_2 | \mathbf{X}^1 & \mathbf{x}_2^2 = \mathbf{K} \mathbf{e}_2 | \mathbf{t}_2 | \mathbf{X}^2 & \mathbf{x}_2^3 = \mathbf{K} \mathbf{e}_2 | \mathbf{t}_2 | \mathbf{X}^3 \\ \mathbf{x}_3^1 = \mathbf{K} \mathbf{e}_3 | \mathbf{t}_3 | \mathbf{X}^1 & & \mathbf{x}_3^3 = \mathbf{K} \mathbf{e}_3 | \mathbf{t}_3 | \mathbf{X}^3 \end{array} \right.$$

$$\text{Observation} = \left\{ \begin{array}{lll} \tilde{\mathbf{x}}_1^1 & \tilde{\mathbf{x}}_1^2 & \\ \tilde{\mathbf{x}}_2^1 & \tilde{\mathbf{x}}_2^2 & \tilde{\mathbf{x}}_2^3 \\ \tilde{\mathbf{x}}_3^1 & & \tilde{\mathbf{x}}_3^3 \end{array} \right.$$

标定及重建问题： 非线性最小二乘优化

A valid solution $\hat{\mathbf{R}}_1 | \mathbf{t}_1, \hat{\mathbf{R}}_2 | \mathbf{t}_2, \hat{\mathbf{R}}_3 | \mathbf{t}_3$ and $\mathbf{X}^1, \mathbf{X}^2, \mathbf{X}^3, \dots$ must let the Re-projection close to the Observation, i.e. to minimize the reprojection error

$$\min_{i,j} \left(\tilde{\mathbf{x}}_i^j - \mathbf{K} \hat{\mathbf{R}}_i | \mathbf{t}_i \mathbf{X}^j \right)^2$$

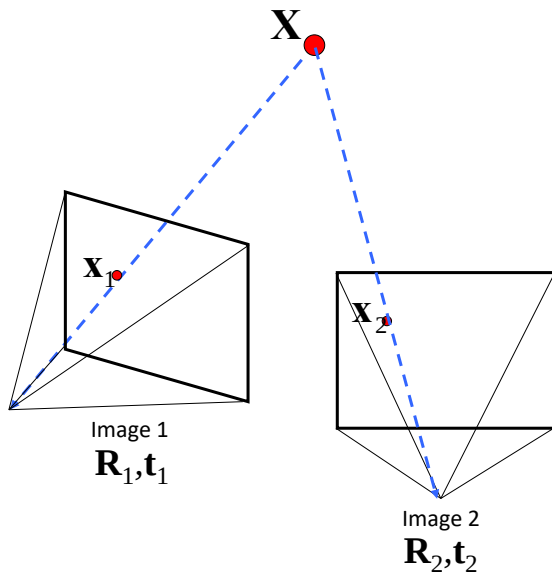
标定及重建问题： 非线性最小二乘优化

A valid solution $\hat{\mathbf{R}}_1 | \mathbf{t}_1, \hat{\mathbf{R}}_2 | \mathbf{t}_2, \hat{\mathbf{R}}_3 | \mathbf{t}_3$ and $\mathbf{X}^1, \mathbf{X}^2, \mathbf{X}^3, \dots$ must let the Re-projection close to the Observation, i.e. to minimize the reprojection error

$$\min_{i,j} \sum_i \sum_j \left(\tilde{\mathbf{x}}_i^j - \mathbf{K} \hat{\mathbf{R}}_i | \mathbf{t}_i \mathbf{X}^j \right)^2$$

Question: What is the unit of this objective function?

Fundamental Matrix



$$\mathbf{x}_1 \leftrightarrow \mathbf{x}_2$$

$$\mathbf{x}_1^T \mathbf{F} \mathbf{x}_2 = 0$$

Estimating Fundamental Matrix

- Given a correspondence

$$\mathbf{x}_1 \leftrightarrow \mathbf{x}_2$$

- The basic incidence relation is

$$\mathbf{x}_1^T \mathbf{F} \mathbf{x}_2 = 0 \quad \longleftrightarrow \quad [x_1 x_2, x_1 y_2, x_1 y_1 x_2, y_1 y_2, y_1, x_2, y_2, 1] \mathbf{f} = 0$$

Need 8 points

$$\begin{bmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \\ f_{33} \end{bmatrix}$$

Estimating Fundamental Matrix

$\mathbf{x}_1^T \mathbf{F} \mathbf{x}_2 = 0$ for 8 point correspondences:

$$\mathbf{x}_1^1 \leftrightarrow \mathbf{x}_2^1, \mathbf{x}_1^2 \leftrightarrow \mathbf{x}_2^2, \mathbf{x}_1^3 \leftrightarrow \mathbf{x}_2^3, \mathbf{x}_1^4 \leftrightarrow \mathbf{x}_2^4, \mathbf{x}_1^5 \leftrightarrow \mathbf{x}_2^5, \mathbf{x}_1^6 \leftrightarrow \mathbf{x}_2^6, \mathbf{x}_1^7 \leftrightarrow \mathbf{x}_2^7, \mathbf{x}_1^8 \leftrightarrow \mathbf{x}_2^8$$

$\hat{e}$	$x_1^1 x_2^1$	$x_1^1 y_2^1$	x_1^1	$y_1^1 x_2^1$	$y_1^1 y_2^1$	y_1^1	x_2^1	y_2^1	1	$\hat{e}$	f_{11}	$\hat{u}$
$\hat{e}$	$x_1^2 x_2^2$	$x_1^2 y_2^2$	x_1^2	$y_1^2 x_2^2$	$y_1^2 y_2^2$	y_1^2	x_2^2	y_2^2	1	$\hat{e}$	f_{12}	$\hat{u}$
$\hat{e}$	$x_1^3 x_2^3$	$x_1^3 y_2^3$	x_1^3	$y_1^3 x_2^3$	$y_1^3 y_2^3$	y_1^3	x_2^3	y_2^3	1	$\hat{e}$	f_{13}	$\hat{u}$
$\hat{e}$	$x_1^4 x_2^4$	$x_1^4 y_2^4$	x_1^4	$y_1^4 x_2^4$	$y_1^4 y_2^4$	y_1^4	x_2^4	y_2^4	1	$\hat{e}$	f_{21}	$\hat{u}$
$\hat{e}$	$x_1^5 x_2^5$	$x_1^5 y_2^5$	x_1^5	$y_1^5 x_2^5$	$y_1^5 y_2^5$	y_1^5	x_2^5	y_2^5	1	$\hat{e}$	f_{22}	$\hat{u}$
$\hat{e}$	$x_1^6 x_2^6$	$x_1^6 y_2^6$	x_1^6	$y_1^6 x_2^6$	$y_1^6 y_2^6$	y_1^6	x_2^6	y_2^6	1	$\hat{e}$	f_{23}	$\hat{u}$
$\hat{e}$	$x_1^7 x_2^7$	$x_1^7 y_2^7$	x_1^7	$y_1^7 x_2^7$	$y_1^7 y_2^7$	y_1^7	x_2^7	y_2^7	1	$\hat{e}$	f_{31}	$\hat{u}$
$\hat{e}$	$x_1^8 x_2^8$	$x_1^8 y_2^8$	x_1^8	$y_1^8 x_2^8$	$y_1^8 y_2^8$	y_1^8	x_2^8	y_2^8	1	$\hat{e}$	f_{32}	$\hat{u}$
$\hat{e}$										$\hat{e}$	f_{33}	$\hat{u}$

$\mathbf{Ax} = \mathbf{b}$

$\mathbf{Af} = \mathbf{0}$

$\mathbf{u} = \mathbf{0}$

Direct Linear Transformation (DLT)

Algebraic Error vs. Geometric Error

- Algebraic Error

$$\min \|\mathbf{A}\mathbf{f}\|$$

- Geometric Error (better) Unit: pixel

$$\min_j \mathring{\mathbf{a}} \quad d\left(\mathbf{x}_1^j, \mathbf{F}\mathbf{x}_2^j\right)^2 + d\left(\mathbf{x}_2^j, \mathbf{F}^T \mathbf{x}_1^j\right)^2$$

Solved by (non-linear) least square solver (e.g. Ceres)

Solving This Optimization Problem

- Theory:

The Levenberg–Marquardt algorithm

http://en.wikipedia.org/wiki/Levenberg-Marquardt_algorithm

- Practice:

The Ceres-Solver from Google

<http://code.google.com/p/ceres-solver/>

Ceres-solver: A Nonlinear Least Squares Minimizer

Toy problem to solve $\min(10 - x)^2$

```
class SimpleCostFunction
: public ceres::SizedCostFunction<1 /* number of residuals */,
                                  1 /* size of first parameter */> {
public:
    virtual ~SimpleCostFunction() {}
    virtual bool Evaluate(double const* const* parameters,
                          double* residuals,
                          double** jacobians) const {
        const double x = parameters[0][0];
        residuals[0] = 10 - x; // f(x) = 10 - x
        // Compute the Jacobian if asked for.
        if (jacobians != NULL && jacobians[0] != NULL) {
            jacobians[0][0] = -1;
        }
        return true;
    }
};
```

Ceres-solver: A Nonlinear Least Squares Minimizer

Toy problem to solve $\min(10 - x)^2$

```
int main(int argc, char** argv) {  
    double x = 5.0;  
    ceres::Problem problem;  
  
    // The problem object takes ownership of the newly allocated  
    // SimpleCostFunction and uses it to optimize the value of x.  
    problem.AddResidualBlock(new SimpleCostFunction, NULL, &x);  
  
    // Run the solver!  
    Solver::Options options;  
    options.max_num_iterations = 10;  
    options.linear_solver_type = ceres::DENSE_QR;  
    options.minimizer_progress_to_stdout = true;  
    Solver::Summary summary;  
    Solve(options, &problem, &summary);  
    std::cout << summary.BriefReport() << "\n";  
    std::cout << "x : 5.0 -> " << x << "\n";  
    return 0;  
}
```

Ceres-solver: A Nonlinear Least Squares Minimizer

Toy problem to solve $\min(10 - x)^2$

```
0: f: 1.250000e+01 d: 0.00e+00 g: 5.00e+00 h: 0.00e+00 rho: 0.00e+00 mu: 1.00e-04 li: 0
1: f: 1.249750e-07 d: 1.25e+01 g: 5.00e-04 h: 5.00e+00 rho: 1.00e+00 mu: 3.33e-05 li: 1
2: f: 1.388518e-16 d: 1.25e-07 g: 1.67e-08 h: 5.00e-04 rho: 1.00e+00 mu: 1.11e-05 li: 1
Ceres Solver Report: Iterations: 2, Initial cost: 1.250000e+01, \
Final cost: 1.388518e-16, Termination: PARAMETER_TOLERANCE.
x : 5 -> 10
```

图像对应点问题 (特征点匹配)

对应点

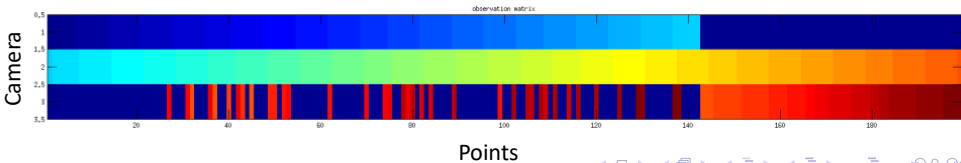


Linking



SIFT Matching

SIFT Matching



Keypoints Detection



→ keypoints

→ keypoints

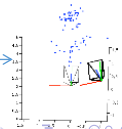
match

→ fundamental
matrix

→ essential
matrix

→ $[R|t]$

→ triangulation



Descriptor for each point



SIFT
descriptor

SIFT
descriptor



keypoints

keypoints

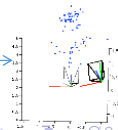
match

fundamental
matrix

essential
matrix

$[R|t]$

triangulation



Same for the other images



SIFT
descriptor

SIFT
descriptor



SIFT
descriptor

SIFT
descriptor



keypoints

keypoints

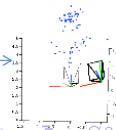
match

fundamental
matrix

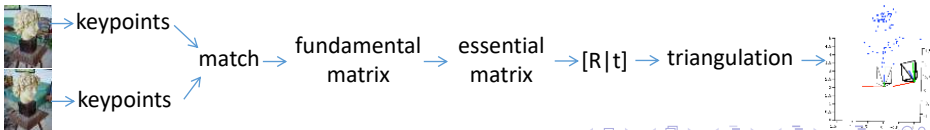
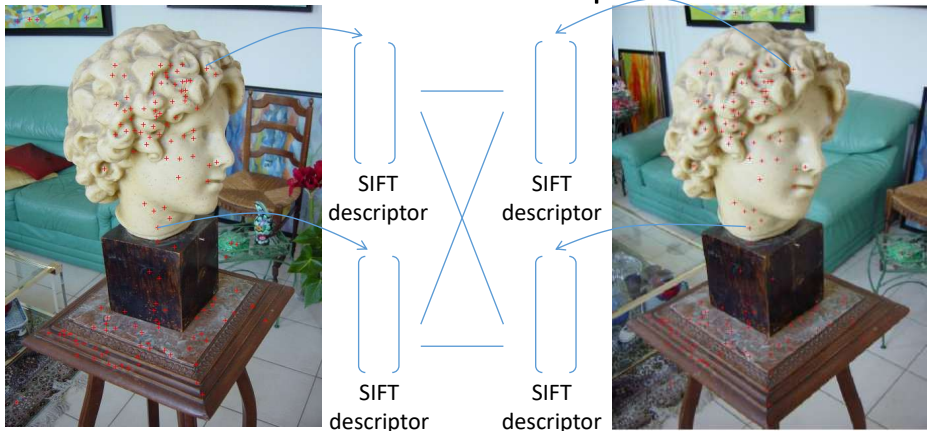
essential
matrix

$[R|t]$

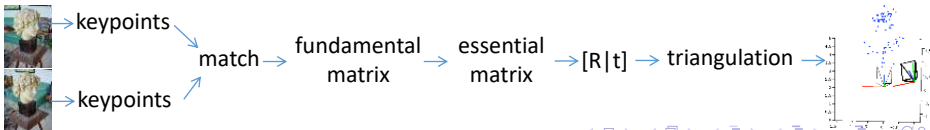
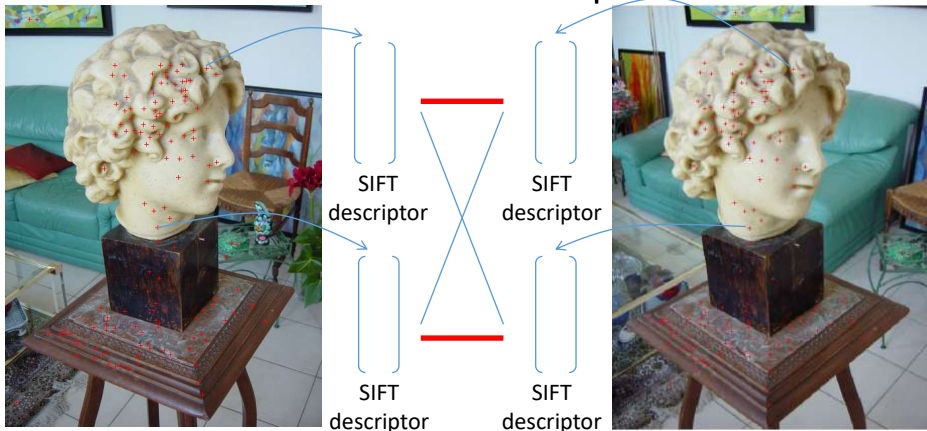
triangulation



Point Match for correspondences



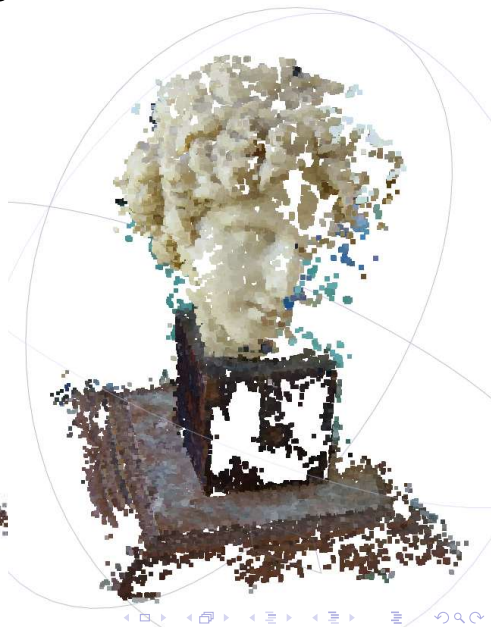
Point Match for correspondences



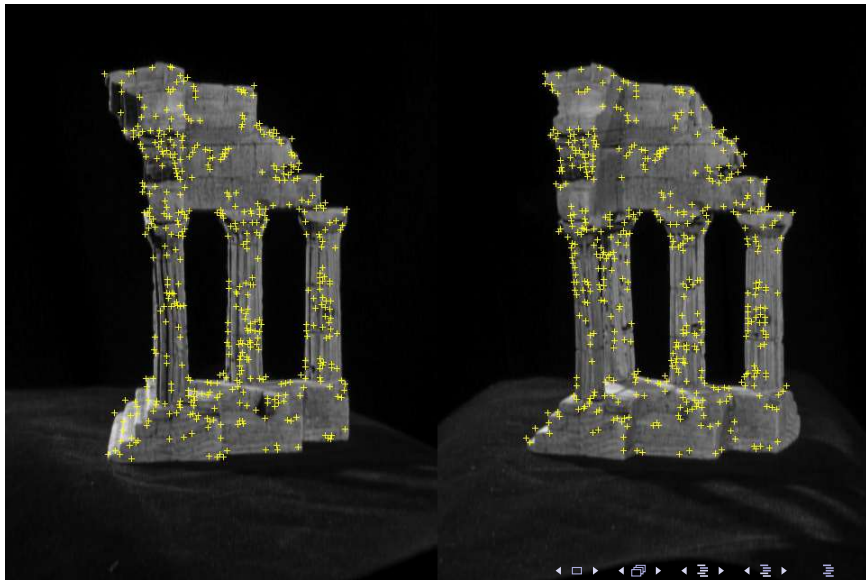
对应点



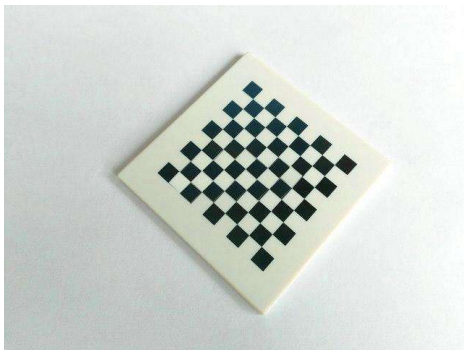
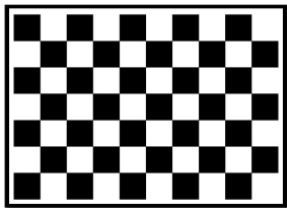
重建点及点颜色



Another Example



简化问题：使用简单的标定板



Other Intrinsic Camera Parameters

- Principle point offset
 - especially when images are cropped (Internet)
- Skew

$$\mathbf{K} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \longrightarrow \quad \mathbf{K} = \begin{bmatrix} f_x & s & p_x \\ 0 & f_y & p_y \\ 0 & 0 & 1 \end{bmatrix}$$

- Radial distortion (due to optics of the lens)

$$r^2 = \|\mathbf{x}\|^2 = x^2 + y^2$$

$$\mathbf{x}' = (1 + k_1 r^2 + k_2 r^4) \mathbf{x}$$



before



after

Other Camera Models

- Fisheye
- Mirror
- Panorama
- Tilt-Shift Lens
- Biological Eyes



http://www.popgadget.net/2006/07/fisheye_camera.php



<http://www.0-360.com/>



<http://sun360.csail.mit.edu>



Canon TS-E 24mm f/3.5L II

Recap: SFMedu Program with Code

Download from:

<http://3dvision.princeton.edu/courses/SFMedu/>



COLMAP: SfM和MVS的开源库

<https://github.com/colmap/colmap>

<https://colmap.github.io/tutorial.html>



中国科学技术大学
University of Science and Technology of China

谢谢！



中国科学技术大学
University of Science and Technology of China

数学建模 Mathematical Modeling

刘利刚

中国科学技术大学

平衡与稳定性模型

稳定性模型

- 对象是动态过程（随时间变化的），而建模目的是研究时间充分长以后过程的变化趋势——平衡状态是否稳定。
- 不求解微分方程，而是用微分方程稳定性理论研究平衡状态的稳定性。

稳定性模型

1 捕鱼业的持续收获

2 军备竞赛

3 种群的相互竞争

4 种群的相互依存

5 种群的弱肉强食

1 捕鱼业的持续收获

背景

- 再生资源（渔业、林业等）与非再生资源（矿业等）
- 再生资源应适度开发——在持续稳产前提下实现最大产量或最佳效益。

问题及分析

- 在**捕捞量稳定**的条件下，如何控制捕捞使产量最大或效益最佳。
- 如果使捕捞量等于自然增长量，**渔场鱼量将保持不变**，则捕捞量稳定。

产量模型

$x(t) \sim$ 渔场鱼量

假设

- 无捕捞时鱼的自然增长服从 Logistic 规律

$$\dot{x}(t) = f(x) = rx(1 - \frac{x}{N})$$

$r \sim$ 固有增长率, $N \sim$ 最大鱼量

- 单位时间捕捞量与渔场鱼量成正比

$$h(x) = Ex, E \sim \text{捕捞强度}$$

建模

$$\text{记 } F(x) = f(x) - h(x)$$

捕捞情况下
渔场鱼量满足

$$\dot{x}(t) = F(x) = rx(1 - \frac{x}{N}) - Ex$$

- 不需求解 $x(t)$, 只需知道 $x(t)$ 稳定的条件

一阶微分方程的平衡点及其稳定性

$$\dot{x} = F(x) \quad (1) \quad \text{一阶非线性（自治）方程}$$

$F(x)=0$ 的根 x_0 ~微分方程的平衡点

$$\dot{x} \Big|_{x=x_0} = 0 \Rightarrow x \equiv x_0$$

设 $x(t)$ 是方程的解，若从 x_0 某邻域的任一初值出发，都有 $\lim_{t \rightarrow \infty} x(t) = x_0$ ，称 x_0 是方程(1)的稳定平衡点

不求 $x(t)$, 判断 x_0 稳定性的方法——直接法

(1)的近似线性方程

$$\dot{x} = F'(x_0)(x - x_0) \quad (2)$$

$$F'(x_0) < 0 \Rightarrow x_0 \text{ 稳定(对(2), (1))}$$

$$F'(x_0) > 0 \Rightarrow x_0 \text{ 不稳定(对(2), (1))}$$

产量模型

$$\dot{x}(t) = F(x) = rx(1 - \frac{x}{N}) - Ex$$

$$F(x) = 0$$



平衡点

$$x_0 = N(1 - \frac{E}{r}), \quad x_1 = 0$$

稳定性判断

$$F'(x_0) = E - r, \quad F'(x_1) = r - E$$

$$E < r \Rightarrow F'(x_0) < 0, F'(x_1) > 0 \quad \Rightarrow \quad x_0 \text{ 稳定}, x_1 \text{ 不稳定}$$

$$E > r \Rightarrow F'(x_0) > 0, F'(x_1) < 0 \quad \Rightarrow \quad x_0 \text{ 不稳定}, x_1 \text{ 稳定}$$

$E \sim$ 捕捞强度

$r \sim$ 固有增长率

x_0 稳定, 可得到稳定产量

x_1 稳定, 渔场干枯

产量模型

在捕捞量稳定的条件下，
控制捕捞强度使产量最大

图解法

$$F(x) = f(x) - h(x)$$

$$f(x) = rx(1 - \frac{x}{N})$$

$$h(x) = Ex$$

$$F(x) = 0 \iff f \text{ 与 } h \text{ 交点 } P$$

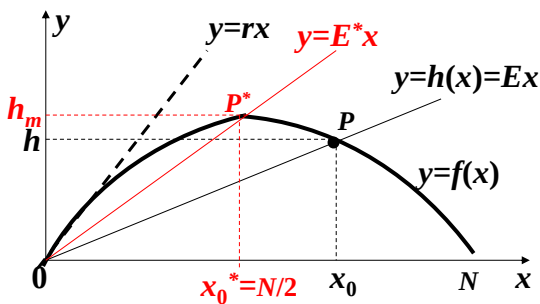
$$E < r \Rightarrow x_0 \text{ 稳定}$$

P 的横坐标 $x_0 \sim$ 平衡点

P 的纵坐标 $h \sim$ 产量

$$\text{产量最大 } P^*(x_0^* = N/2, h_m = rN/4) \quad E^* = h_m / x_0^* = r/2$$

控制渔场鱼量为最大鱼量的一半



效益模型

在捕捞量稳定的条件下，控制捕捞强度使效益最大。

假设 • 鱼销售价格 p • 单位捕捞强度费用 c

收入 $T = ph(x) = pEx$ 支出 $S = cE$

单位时间利润

$$R = T - S = pEx - cE$$

稳定平衡点 $x_0 = N(1 - E/r)$ $\Downarrow$

$$R(E) = T(E) - S(E) = pNE\left(1 - \frac{E}{r}\right) - cE$$

求 E 使 $R(E)$ 最大

$$\Rightarrow E_R = \frac{r}{2}\left(1 - \frac{c}{pN}\right) < E^* = \frac{r}{2}$$

渔场
鱼量

$$x_R = N\left(1 - \frac{E_R}{r}\right) = \frac{N}{2} + \frac{c}{2p} \quad h_R = \frac{rN}{4}\left(1 - \frac{c^2}{p^2 N^2}\right)$$

捕捞 过度

- 封闭式捕捞追求利润 $R(E)$ 最大
- 开放式捕捞只求利润 $R(E) > 0$

$$E_R = \frac{r}{2} \left(1 - \frac{c}{pN}\right)$$

$$R(E) = T(E) - S(E) = pNE \left(1 - \frac{E}{r}\right) - cE \stackrel{\text{令}}{=} 0 \quad \Rightarrow$$

$$E_s = r \left(1 - \frac{c}{pN}\right)$$

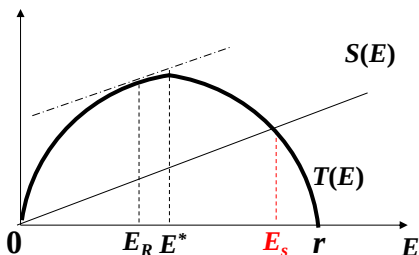
$R(E)=0$ 时的捕捞强度(临界强度) $E_s=2E_R$

临界强度下的渔场鱼量

$$x_s = N \left(1 - \frac{E_s}{r}\right) = \frac{c}{p}$$

$$p \uparrow, c \downarrow \Rightarrow E_s \uparrow, x_s \downarrow$$

捕捞过度



Discussion

2 军备竞赛

目的

- 描述双方(国家或国家集团)军备竞赛过程
- 解释(预测)双方军备竞赛的结局

假设

- 1) 由于相互不信任，一方军备越大，另一方军备增加越快；
- 2) 由于经济实力限制，一方军备越大，对自己军备增长的制约越大；
- 3) 由于相互敌视或领土争端，每一方都存在增加军备的潜力。

进一步 假设

- 1) 2) 的作用为线性；3) 的作用为常数

建模

$x(t)$ ~甲方军备数量, $y(t)$ ~乙方军备数量

$$\dot{x}(t) = -\alpha x + ky + g$$

$$\dot{y}(t) = lx - \beta y + h$$

α, β ~ 本方经济实力的制约;

k, l ~ 对方军备数量的刺激;

g, h ~ 本方军备竞赛的潜力。

军备竞赛的结局



$t \rightarrow \infty$ 时的 $x(t)$, $y(t)$



微分方程的平衡点及其稳定性

线性常系数
微分方程组

$$\begin{aligned}\dot{x}(t) &= ax + by \\ \dot{y}(t) &= cx + dy\end{aligned}$$

的平衡点及其稳定性

平衡点 $P_0(x_0, y_0) = (0, 0) \sim$ 代数方程

$$\begin{aligned}ax + by &= 0 \\ cx + dy &= 0\end{aligned}$$
 的根

若从 P_0 某邻域的任一初值出发, 都有 $\lim_{t \rightarrow \infty} x(t) = x_0$,

$\lim_{t \rightarrow \infty} y(t) = y_0$, 称 P_0 是微分方程的 **稳定平衡点**

记系数矩阵 $A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$

特征方程 $\det(A - \lambda I) = 0$

特征根

$$\begin{cases} \lambda^2 + p\lambda + q = 0 \\ p = -(a + d) \\ q = \det A \end{cases}$$

$$\lambda_{1,2} = (-p \pm \sqrt{p^2 - 4q})/2$$

线性常系数
微分方程组

$$\begin{aligned}\dot{x}(t) &= ax + by \\ \dot{y}(t) &= cx + dy\end{aligned}$$

的平衡点及其稳定性

平衡点 $P_0(0,0)$ 特征根 $\lambda_{1,2} = (-p \pm \sqrt{p^2 - 4q})/2$

微分方程一般解形式 $c_1 e^{\lambda_1 t} + c_2 e^{\lambda_2 t}$

$\lambda_{1,2}$ 为负数或有负实部



$p > 0$ 且 $q > 0$



平衡点 $P_0(0,0)$ 稳定

$p < 0$ 或 $q < 0$



平衡点 $P_0(0,0)$ 不稳定

军备竞赛

模型

$$\begin{cases} \dot{x}(t) = -\alpha x + ky + g \\ \dot{y}(t) = lx - \beta y + h \end{cases}$$

平衡点

$$x_0 = \frac{kh + \beta g}{\alpha\beta - kl}, \quad y_0 = \frac{lg + \alpha h}{\alpha\beta - kl}$$

稳定性判断

系数
矩阵

$$A = \begin{bmatrix} -\alpha & k \\ l & -\beta \end{bmatrix}$$

$$p = -(-\alpha - \beta) = \alpha + \beta > 0$$

$$q = \det A = \alpha\beta - kl$$

平衡点 (x_0, y_0) 稳定的条件

$$p > 0, q > 0$$



$$\alpha\beta > kl$$

模型的定性解释

模型

$$\begin{cases} \dot{x}(t) = -\alpha x + ky + g \\ \dot{y}(t) = lx - \beta y + h \end{cases}$$

平衡点 $x_0 = \frac{kh + \beta g}{\alpha\beta - kl}, \quad y_0 = \frac{lg + \alpha h}{\alpha\beta - kl}$

双方军备稳定(时间充分长后趋向有限值)的条件

$$\alpha\beta > kl$$

$\alpha, \beta \sim$ 本方经济实力的制约;

$k, l \sim$ 对方军备数量的刺激;

$g, h \sim$ 本方军备竞赛的潜力。

- 1) 双方经济制约大于双方军备刺激时, 军备竞赛才会稳定, 否则军备将无限扩张。
- 2) 若 $g=h=0$, 则 $x_0=y_0=0$, 在 $\alpha\beta > kl$ 下 $x(t), y(t) \rightarrow 0$, 即友好邻国通过裁军可达到永久和平。

模型的定性解释

模型

$$\begin{cases} \dot{x}(t) = -\alpha x + ky + g \\ \dot{y}(t) = lx - \beta y + h \end{cases}$$

$\alpha, \beta \sim$ 本方经济实力的制约;

$k, l \sim$ 对方军备数量的刺激;

$g, h \sim$ 本方军备竞赛的潜力。

3) 若 g, h 不为零, 即便双方一时和解, 使某时 $x(t), y(t)$ 很小, 但因 $\dot{x} > 0, \dot{y} > 0$, 也会重整军备。

4) 即使某时一方(由于战败或协议)军备大减, 如 $x(t)=0$, 也会因 $\dot{x} = ky + g$ 使该方重整军备,

即存在互不信任($k \neq 0$) 或固有争端($g \neq 0$) 的单方面裁军不会持久。

Discussion

3 种群的相互竞争

- 一个自然环境中有两个种群生存，它们之间的关系：相互竞争；相互依存；弱肉强食。
- 当两个种群为争夺同一食物来源和生存空间相互竞争时，常见的结局是，竞争力弱的灭绝，竞争力强的达到环境容许的最大容量。
- 建立数学模型描述两个种群相互竞争的过程，分析产生这种结局的条件。

模型假设

- 有甲乙两个种群，它们独自生存时数量变化均服从Logistic规律；

$$\dot{x}_1(t) = r_1 x_1 \left(1 - \frac{x_1}{N_1}\right) \quad \dot{x}_2(t) = r_2 x_2 \left(1 - \frac{x_2}{N_2}\right)$$

- 两种群在一起生存时，乙对甲增长的阻滞作用与乙的数量成正比；甲对乙有同样的作用。

模型

$$\dot{x}_1(t) = r_1 x_1 \left(1 - \frac{x_1}{N_1} - \sigma_1 \frac{x_2}{N_2}\right) \quad \dot{x}_2(t) = r_2 x_2 \left(1 - \sigma_2 \frac{x_1}{N_1} - \frac{x_2}{N_2}\right)$$

对于消耗甲的资源而言，
乙(相对于 N_2)是甲(相
对于 N_1)的 σ_1 倍。

$$\sigma_1 > 1$$



对甲增长的阻滞
作用，乙大于甲



乙的竞争力强

模型

$$\dot{x}_1(t) = r_1 x_1 \left(1 - \frac{x_1}{N_1} - \sigma_1 \frac{x_2}{N_2} \right)$$

$$\dot{x}_2(t) = r_2 x_2 \left(1 - \sigma_2 \frac{x_1}{N_1} - \frac{x_2}{N_2} \right)$$

模型分析

$t \rightarrow \infty$ 时 $x_1(t), x_2(t)$ 的趋向 (平衡点及其稳定性)

(二阶)非线性
(自治)方程 $\dot{x}_1(t) = f(x_1, x_2)$
 $\dot{x}_2(t) = g(x_1, x_2)$

的平衡点及其稳定性

平衡点 $P_0(x_1^0, x_2^0) \sim$ 代数方程

$f(x_1, x_2) = 0$
 $g(x_1, x_2) = 0$ 的根

若从 P_0 某邻域的任一初值出发, 都有 $\lim_{t \rightarrow \infty} x_1(t) = x_1^0$,

$\lim_{t \rightarrow \infty} x_2(t) = x_2^0$, 称 P_0 是微分方程的**稳定平衡点**

判断 $P_0(x_1^0, x_2^0)$ 稳定性的方法——直接法

(1)的近似线性方程

$$\begin{aligned}\dot{x}_1(t) &= f(x_1, x_2) \\ \dot{x}_2(t) &= g(x_1, x_2) \quad (1)\end{aligned}$$

$$\begin{aligned}\dot{x}_1(t) &= f_{x_1}(x_1^0, x_2^0)(x_1 - x_1^0) + f_{x_2}(x_1^0, x_2^0)(x_2 - x_2^0) \\ \dot{x}_2(t) &= g_{x_1}(x_1^0, x_2^0)(x_1 - x_1^0) + g_{x_2}(x_1^0, x_2^0)(x_2 - x_2^0) \quad (2)\end{aligned}$$

$$A = \begin{bmatrix} f_{x_1} & f_{x_2} \\ g_{x_1} & g_{x_2} \end{bmatrix} \Big|_{P_0} \quad \begin{cases} \lambda^2 + p\lambda + q = 0 \\ p = -(f_{x_1} + g_{x_2}) \Big|_{P_0} \\ q = \det A \end{cases}$$

$p > 0$ 且 $q > 0$ $p < 0$ 或 $q < 0$

平衡点 P_0 稳定(对2,1)

平衡点 P_0 不稳定(对2,1)

模型

$$\dot{x}_1(t) = r_1 x_1 \left(1 - \frac{x_1}{N_1} - \sigma_1 \frac{x_2}{N_2} \right)$$

$$\dot{x}_2(t) = r_2 x_2 \left(1 - \sigma_2 \frac{x_1}{N_1} - \frac{x_2}{N_2} \right)$$

$$\begin{cases} f(x_1, x_2) \equiv r_1 x_1 \left(1 - \frac{x_1}{N_1} - \sigma_1 \frac{x_2}{N_2} \right) = 0 \\ g(x_1, x_2) \equiv r_2 x_2 \left(1 - \sigma_2 \frac{x_1}{N_1} - \frac{x_2}{N_2} \right) = 0 \end{cases}$$

平衡点: $P_1(N_1, 0), P_2(0, N_2),$

$$P_3\left(\frac{N_1(1-\sigma_1)}{1-\sigma_1\sigma_2}, \frac{N_2(1-\sigma_2)}{1-\sigma_1\sigma_2}\right), P_4(0, 0)$$

仅当 $\sigma_1, \sigma_2 < 1$ 或 $\sigma_1, \sigma_2 > 1$ 时, P_3 才有意义

平衡点稳定性分析

$$\begin{cases} f(x_1, x_2) \equiv r_1 x_1 \left(1 - \frac{x_1}{N_1} - \sigma_1 \frac{x_2}{N_2} \right) \\ g(x_1, x_2) \equiv r_2 x_2 \left(1 - \sigma_2 \frac{x_1}{N_1} - \frac{x_2}{N_2} \right) \end{cases}$$

$$A = \begin{bmatrix} f_{x1} & f_{x2} \\ g_{x1} & g_{x2} \end{bmatrix} = \begin{bmatrix} r_1 \left(1 - \frac{2x_1}{N_1} - \frac{\sigma_1 x_2}{N_2} \right) & -\frac{r_1 \sigma_1 x_1}{N_2} \\ -\frac{r_2 \sigma_2 x_2}{N_1} & r_2 \left(1 - \frac{\sigma_2 x_1}{N_1} - \frac{2x_2}{N_2} \right) \end{bmatrix}$$

$$p = -(f_{x1} + g_{x2})|_{p_i}, \quad q = \det A|_{p_i}, \quad i = 1, 2, 3, 4$$

平衡点 P_i 稳定条件: $p > 0$ 且 $q > 0$

平衡点稳定性的相轨线分析

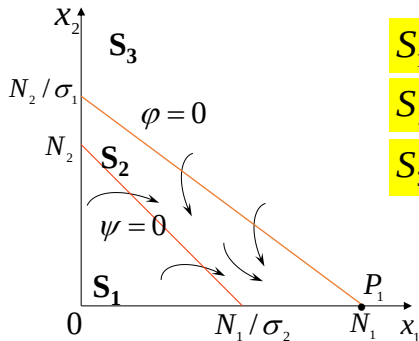
$$\dot{x}_1(t) = r_1 x_1 \left(1 - \frac{x_1}{N_1} - \sigma_1 \frac{x_2}{N_2} \right)$$

$$\dot{x}_2(t) = r_2 x_2 \left(1 - \sigma_2 \frac{x_1}{N_1} - \frac{x_2}{N_2} \right)$$

$$\varphi(x_1, x_2) = 1 - \frac{x_1}{N_1} - \sigma_1 \frac{x_2}{N_2}$$

$$\psi(x_1, x_2) = 1 - \sigma_2 \frac{x_1}{N_1} - \frac{x_2}{N_2}$$

(1) $\sigma_2 > 1$, $\sigma_1 < 1$



$S_1 : \varphi > 0, \psi > 0$

$S_1 : \dot{x}_1 > 0, \dot{x}_2 > 0$

$t \uparrow \rightarrow x_1, x_2 \uparrow$

$S_2 : \dot{x}_1 > 0, \dot{x}_2 < 0$

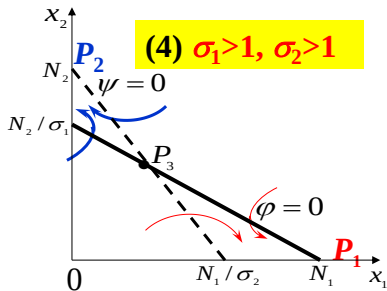
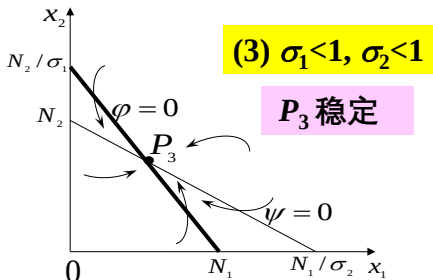
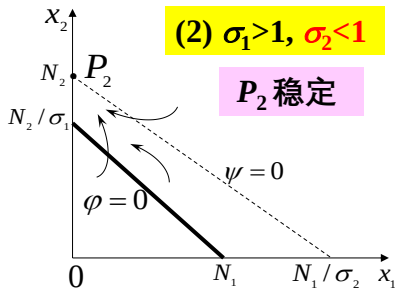
$t \uparrow \rightarrow x_1 \uparrow, x_2 \downarrow$

$S_3 : \dot{x}_1 < 0, \dot{x}_2 < 0$

$t \uparrow \rightarrow x_1, x_2 \downarrow$

从任意点出发($t=0$)的相轨线都趋向 $P_1(N_1, 0)$ ($t \rightarrow \infty$)

$P_1(N_1, 0)$ 是稳定平衡点



有相轨线趋向 P_1
有相轨线趋向 P_2

P_1, P_2 都不
(局部)稳定

P_1 稳定的条件: 直接法 $\sigma_2 > 1$

加上与(4)相区别的 $\sigma_1 < 1$

P_1 全局稳定

结果解释

• P_1 稳定的条件: $\sigma_1 < 1, \sigma_2 > 1$

对于消耗甲的资源而言,
乙(相对于 N_2)是甲(相对
于 N_1)的 σ_1 倍。

$$\sigma_1 < 1$$



对甲增长的阻滞
作用, 乙小于甲
 $\Rightarrow$ 乙的竞争力弱

$\sigma_2 > 1 \Rightarrow$ 甲的竞争力强

甲达到最大容量, 乙灭绝

• P_2 稳定的条件: $\sigma_1 > 1, \sigma_2 < 1$

• P_3 稳定的条件: $\sigma_1 < 1, \sigma_2 < 1$

通常 $\sigma_1 \approx 1/\sigma_2$, P_3 稳定条件不满足

Discussion

4 种群的相互依存

甲乙两种群的相互依存有三种形式

- 1) 甲可以独自生存，乙不能独自生存；甲乙一起生存时相互提供食物、促进增长。
- 2) 甲乙均可以独自生存；甲乙一起生存时相互提供食物、促进增长。
- 3) 甲乙均不能独自生存；甲乙一起生存时相互提供食物、促进增长。

模型假设

- 甲可以独自生存，数量变化服从Logistic规律；甲乙一起生存时乙为甲提供食物、促进增长。
- 乙不能独自生存；甲乙一起生存时甲为乙提供食物、促进增长；乙的增长又受到本身的阻滞作用 (服从Logistic规律)。

模型

$$\dot{x}_1(t_1) = r_1 x_1 \left(1 - \frac{x_1}{N_1} + \sigma_1 \frac{x_2}{N_2} \right)$$

乙为甲提供食物
是甲消耗的 σ_1 倍

$$\dot{x}_2(t) = r_2 x_2 \left(-1 + \sigma_2 \frac{x_1}{N_1} - \frac{x_2}{N_2} \right)$$

甲为乙提供食物
是乙消耗的 σ_2 倍

种群依存模型的平衡点及稳定性

平衡点	p	q	稳定条件
$P_1(N_1, 0)$	$r_1 - r_2(\sigma_2 - 1)$	$-r_1 r_2(\sigma_2 - 1)$	$\sigma_2 < 1, \sigma_1 \sigma_2 < 1$
$P_2\left(\frac{N_1(1-\sigma_1)}{1-\sigma_1\sigma_2}, \frac{N_2(\sigma_2-1)}{1-\sigma_1\sigma_2}\right)$	$\frac{r_1(1-\sigma_1)+r_2(\sigma_2-1)}{1-\sigma_1\sigma_2}$	$\frac{r_1 r_2(1-\sigma_1)(\sigma_2-1)}{1-\sigma_1\sigma_2}$	$\sigma_1 < 1, \sigma_2 > 1,$ $\sigma_1 \sigma_2 < 1$
$P_3(0, 0)$	$-r_1 + r_2$	$-r_1 r_2$	不稳定

P_2 是甲乙相互依存而共生的平衡点

平衡点 P_2 稳定性的相轨线

$$P_2 \left(\frac{N_1(1-\sigma_1)}{1-\sigma_1\sigma_2}, \frac{N_2(\sigma_2-1)}{1-\sigma_1\sigma_2} \right)$$

$$\dot{x}_1(t) = r_1 x_1 \left(1 - \frac{x_1}{N_1} + \sigma_1 \frac{x_2}{N_2} \right) = r_1 x_1 \varphi(x_1, x_2) \quad \dot{x}_2(t) = r_2 x_2 \left(-1 + \sigma_2 \frac{x_1}{N_1} - \frac{x_2}{N_2} \right) = r_2 x_2 \psi(x_1, x_2)$$

$$\sigma_1 < 1, \sigma_2 > 1, \sigma_1\sigma_2 < 1$$

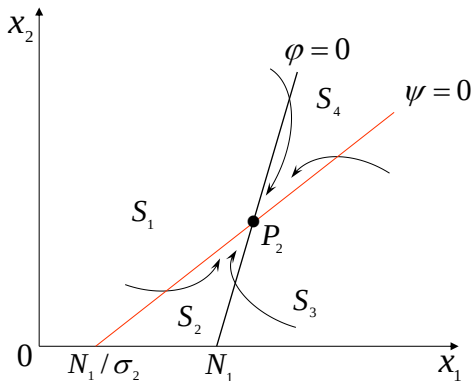
$$S_1 : \dot{x}_1 > 0, \dot{x}_2 < 0;$$

$$S_2 : \dot{x}_1 > 0, \dot{x}_2 > 0;$$

$$S_3 : \dot{x}_1 < 0, \dot{x}_2 > 0;$$

$$S_4 : \dot{x}_1 < 0, \dot{x}_2 < 0.$$

P_2 稳定



结果 解释

甲可以独立生存

$$\dot{x}_1(t_1) = r_1 x_1 \left(1 - \frac{x_1}{N_1} + \sigma_1 \frac{x_2}{N_2} \right)$$

乙不能独立生存

$$\dot{x}_2(t) = r_2 x_2 \left(-1 + \sigma_2 \frac{x_1}{N_1} - \frac{x_2}{N_2} \right)$$

$$P_2 \left(\frac{N_1(1-\sigma_1)}{1-\sigma_1\sigma_2}, \frac{N_2(\sigma_2-1)}{1-\sigma_1\sigma_2} \right)$$

P_2 稳定条件:
 $\sigma_1 < 1, \sigma_2 > 1, \sigma_1\sigma_2 < 1$

$\sigma_2 > 1$ ~ 甲必须为乙提供足够的食物——
甲为乙提供的食物是乙消耗的 σ_2 倍

$\sigma_1\sigma_2 < 1$ ~ $\sigma_2 > 1$ 前提下 P_2 存在的必要条件

$\sigma_1 < 1$ ~ $\sigma_2 > 1, \sigma_1\sigma_2 < 1$ 的需要, 且 σ_1 必须足
够小, 才能在 $\sigma_2 > 1$ 条件下使 $\sigma_1\sigma_2 < 1$ 成立

Discussion

5 种群的弱肉强食(食饵-捕食者模型)

- 种群甲靠丰富的天然资源生存，种群乙靠捕食甲为生，形成食饵-捕食者系统，如食用鱼和鲨鱼，美洲兔和山猫，害虫和益虫。
- **模型的历史背景**——一次世界大战期间地中海渔业的捕捞量下降(食用鱼和鲨鱼同时捕捞)，但是其中鲨鱼的比例却增加，为什么？

食饵-捕食者模型(Volterra)

食饵（甲）数量 $x(t)$, 捕食者（乙）数量 $y(t)$

甲独立生存的增长率 r $\dot{x} = rx$

乙使甲的增长率减小,
减小量与 y 成正比

$$\dot{x}(t) = (r - ay)x = rx - axy \quad (1)$$

乙独立生存的死亡率 d $\dot{y} = -dy$

甲使乙的死亡率减小,
减小量与 x 成正比

$$\dot{y}(t) = -(d - bx)y = -dy + bxy \quad (2)$$

$a \sim$ 捕食者掠取食饵能力 $b \sim$ 食饵供养捕食者能力

方程(1),(2) 无解析解

Volterra模型的平衡点及其稳定性

$$\dot{x}(t) = (r - ay)x = rx - axy$$

稳定性分析

$$\dot{y}(t) = -(d - bx)y = -dy + bxy$$

平衡点

$$P(d/b, r/a), P'(0,0)$$

$$A = \begin{bmatrix} r - ax & -ax \\ by & -d + bx \end{bmatrix}$$

$$A|_P = \begin{bmatrix} 0 & -ad/b \\ br/a & 0 \end{bmatrix}$$

$p = 0, q > 0$
 P : 临界状态

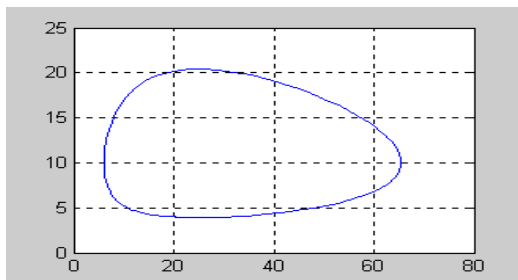
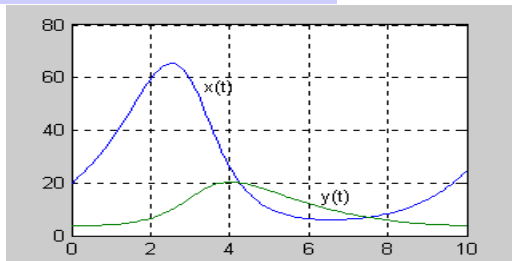
$$A|_{P'} = \begin{bmatrix} r & 0 \\ 0 & -d \end{bmatrix}$$

$q < 0$
 P' 不稳定

P 点稳定性不能用近似线性方程分析

用数学软件MATLAB求微分方程数值解

t	$x(t)$	$y(t)$
0	20.0000	4.0000
0.1000	21.2406	3.9651
0.2000	22.5649	3.9405
0.3000	23.9763	3.9269
...	...	...
5.1000	9.6162	16.7235
5.2000	9.0173	16.2064
...	...	...
9.5000	18.4750	4.0447
9.6000	19.6136	3.9968
9.7000	20.8311	3.9587



$x \sim y$ 平面上的相轨线

食饵-捕食者模型(Volterra)

$$\dot{x}(t) = (r - ay)x \quad \dot{y}(t) = -(d - bx)y$$

计算结果 (数值, 图形)



观察, 猜测

$x(t), y(t)$ 是周期函数, 相图 (x, y) 是封闭曲线

$x(t), y(t)$ 的周期约为 9.6

$$x_{\max} \approx 65.5, x_{\min} \approx 6, y_{\max} \approx 20.5, y_{\min} \approx 3.9$$

用数值积分可算出 $x(t), y(t)$ 一周期的平均值:
 $x(t)$ 的平均值约为 25, $y(t)$ 的平均值约为 10。

用相轨线分析 $P(d/b, r/a)$ 点稳定性

$$\begin{aligned}\dot{x}(t) &= (r - ay)x \\ \dot{y}(t) &= (-d + bx)y\end{aligned}\quad \begin{array}{c} \text{消去 } dt \\ \Rightarrow \end{array} \quad \frac{dx}{dy} = \frac{x(r - ay)}{y(-d + bx)}$$

$$\Rightarrow \frac{-d + bx}{x} dx = \frac{r - ay}{y} dy$$

$$\Rightarrow -d \ln x + bx = r \ln y - ay + c_1$$

$$\Rightarrow (x^d e^{-bx})(y^r e^{-ay}) = c$$

取指数

c 由初始条件确定

用相轨线分析 $P(d/b, r/a)$ 点稳定性

$$(x^d e^{-bx})(y^r e^{-ay}) = c$$


$$f(x)$$


$$g(y)$$

相轨线

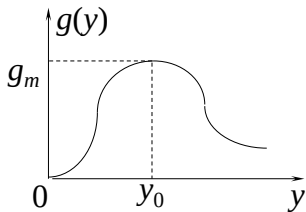
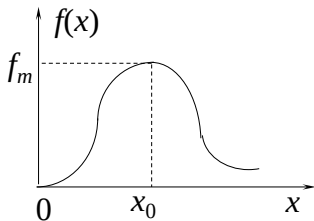
$$f(x)g(y) = c$$

在相平面上讨论相轨线的图形

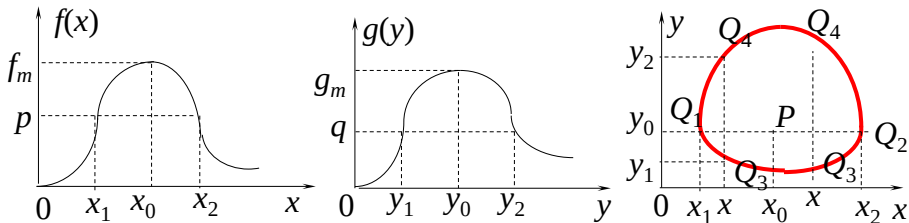
$$f(0) = f(\infty) = 0, \quad f(x_0) = f_m, \quad x_0 = d/b$$

$$g(0) = g(\infty) = 0, \quad g(y_0) = g_m, \quad y_0 = r/a$$

$c > f_m g_m$ 时无相轨线 以下设 $c \leq f_m g_m$



相轨线 $f(x)g(y) = c$



$c = f_m g_m \Rightarrow x = x_0, y = y_0 \Rightarrow$ 相轨线退化为 P 点 $P \sim$ 中心

$c < f_m g_m \Rightarrow$ 设 $c = p g_m$ 令 $y = y_0 \Rightarrow g(y) = g_m \quad f(x) = p < f_m$
 $\Rightarrow$ 存在 $x_1 < x_0 < x_2$, 使 $f(x_1) = f(x_2) = p \Rightarrow Q_1(x_1, y_0), Q_2(x_2, y_0)$

考察 $x \in [x_1, x_2] \quad f(x)g(y) = p g_m \quad f(x) > p \quad g(y) = q < g_m$

$\Rightarrow$ 存在 $y_1 < y_0 < y_2$, 使 $g(y_1) = g(y_2) = q \Rightarrow Q_3(x, y_1), Q_4(x, y_2)$

x 是 $[x_1, x_2]$ 内任意点 $\Rightarrow$ 相轨线是封闭曲线族

用相轨线分析 $P(d/b, r/a)$ 点稳定性

相轨线是封闭曲线 $\Leftrightarrow x(t), y(t)$ 是周期函数(周期记 T)

求 $x(t), y(t)$ 在一周期的平均值 $\bar{x}, \bar{y}$

$$\bar{x} = \frac{1}{T} \int_0^T x(t) dt = \frac{1}{T} \int_0^T \frac{1}{b} \left(\frac{\dot{y}}{y} + d \right) dt$$

$$= \frac{1}{T} \left(\frac{\ln y(T) - \ln y(0)}{b} + \frac{dT}{b} \right)$$

$$\dot{y}(t) = (-d + bx)y$$

$$x(t) = \frac{1}{b} \left(\frac{\dot{y}}{y} + d \right)$$

$$\Rightarrow \bar{x} = d/b$$

$$\dot{x}(t) = (r - ay)x$$

$\Rightarrow \dots \Rightarrow$

$$\Rightarrow \bar{y} = r/a$$

轨线中心 $P(x_0, y_0): x_0 = d/b, y_0 = r/a \Rightarrow \bar{x} = x_0, \bar{y} = y_0$

模型解释

$$\dot{x}(t) = (r - ay)x$$

$$\dot{y}(t) = -(d - bx)y$$

初值 $P_0(x'_0, y'_0)$

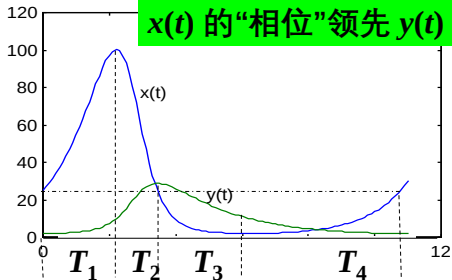
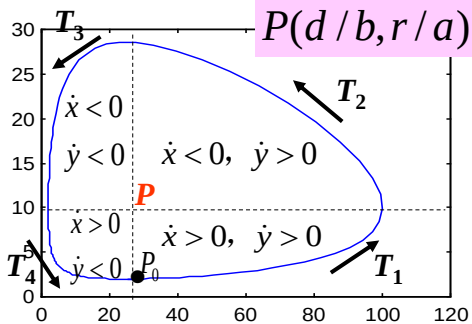
相轨线的方向

$T_1 : x(t) \uparrow y(t) \uparrow$

$T_2 : x(t) \downarrow y(t) \uparrow$

$T_3 : x(t) \downarrow y(t) \downarrow$

$T_4 : x(t) \uparrow y(t) \downarrow$



模型解释

捕食者
数量 $\bar{y} = \frac{r}{a}$

$r \sim$ 食饵增长率

$a \sim$ 捕食者掠取食饵能力

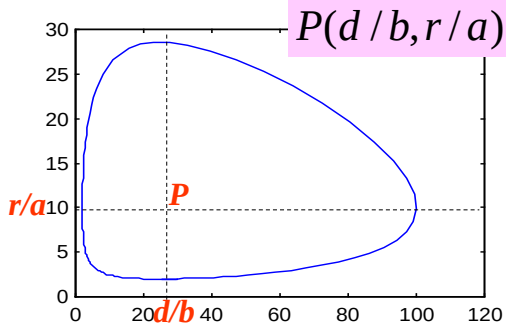
捕食者数量与 r 成正比，与 a 成反比

食饵
数量 $\bar{x} = \frac{d}{b}$

$d \sim$ 捕食者死亡率

$b \sim$ 食饵供养捕食者能力

食饵数量与 d 成正比，与 b 成反比



模型解释

一次大战期间地中海渔业的捕捞量下降，但是其中鲨鱼的比例却在增加，为什么？

自然环境

$$P(\bar{x}, \bar{y}) \quad \bar{x} = d/b, \bar{y} = r/a$$

捕捞

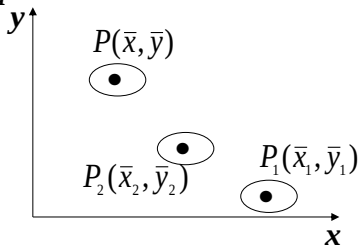
$$r \rightarrow r - \varepsilon_1, d \rightarrow d + \varepsilon_1$$

$$\Rightarrow \bar{x}_1 > \bar{x}, \bar{y}_1 < \bar{y} \quad P \rightarrow P_1$$

战时捕捞

$$r \rightarrow r - \varepsilon_2, d \rightarrow d + \varepsilon_2, \varepsilon_2 < \varepsilon_1$$

$$\Rightarrow \bar{x}_2 < \bar{x}_1, \bar{y}_2 > \bar{y}_1 \quad P_1 \rightarrow P_2$$



食饵(鱼)减少，
捕食者(鲨鱼)增加

$P \rightarrow P_1$ 还表明：对害虫(食饵)—益虫(捕食者)系统，使用灭两种虫的杀虫剂，会使害虫增加，益虫减少。

食饵-捕食者模型(Volterra)的缺点与改进

多数食饵—捕食者系统观察不到周期震荡，
而是趋向某个平衡状态，即存在稳定平衡点

Volterra模型 $\dot{x}(t) = (r - ay)x \quad \dot{y}(t) = -(d - bx)y$



改写

$$\dot{x}_1(t) = r_1 x_1 \left(1 - \sigma_1 \frac{x_2}{N_2} \right) \quad \dot{x}_2(t) = r_2 x_2 \left(-1 + \sigma_2 \frac{x_1}{N_1} \right)$$

加Logistic项



$$\dot{x}_1(t) = r_1 x_1 \left(1 - \frac{x_1}{N_1} - \sigma_1 \frac{x_2}{N_2} \right) \quad \dot{x}_2(t) = r_2 x_2 \left(-1 + \sigma_2 \frac{x_1}{N_1} - \frac{x_2}{N_2} \right)$$

有稳定平衡点

食饵-捕食者模型(Volterra)的缺点与改进

- 相轨线是封闭曲线，结构不稳定——一旦离开某一条闭轨线，就进入另一条闭轨线，不恢复原状。
- 自然界存在的周期性平衡生态系统是结构稳定的，即偏离周期轨道后，内部制约使系统恢复原状。

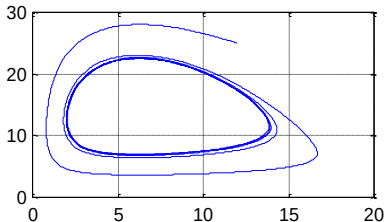
$$\dot{x}_1(t) = r_1 x_1 \left(1 - \frac{x_1}{N_1} - \sigma_1 \frac{x_2}{1 + w x_1} \right) \quad \dot{x}_2(t) = r_2 x_2 \left(-1 + \sigma_2 \frac{x_1}{1 + w x_1} \right)$$

$$r_1=1, N_1=20, \sigma_1=0.1, \\ w=0.2, r_2=0.5, \sigma_2=0.18$$

相轨线趋向极限环



结构稳定



两种群模型的几种形式

相互竞争

$$\dot{x}_1(t) = r_1 x_1 \left(1 - \frac{x_1}{N_1} - \sigma_1 \frac{x_2}{N_2} \right) \quad \dot{x}_2(t) = r_2 x_2 \left(1 - \sigma_2 \frac{x_1}{N_1} - \frac{x_2}{N_2} \right)$$

相互依存

$$\dot{x}_1(t) = r_1 x_1 \left(\pm 1 - \frac{x_1}{N_1} + \sigma_1 \frac{x_2}{N_2} \right) \quad \dot{x}_2(t) = r_2 x_2 \left(\pm 1 + \sigma_2 \frac{x_1}{N_1} - \frac{x_2}{N_2} \right)$$

弱肉强食

$$\dot{x}_1(t) = r_1 x_1 \left(1 - \frac{x_1}{N_1} - \sigma_1 \frac{x_2}{N_2} \right) \quad \dot{x}_2(t) = r_2 x_2 \left(-1 + \sigma_2 \frac{x_1}{N_1} - \frac{x_2}{N_2} \right)$$

Discussion



中国科学技术大学
University of Science and Technology of China

谢谢！



中国科学技术大学

University of Science and Technology of China

数学建模

Mathematical Modeling

刘利刚

中国科学技术大学

统计回归模型

回归分析

- 研究因变量对自变量的依赖关系的一种统计分析方法
- 通过自变量的给定值来估计或预测因变量的均值
- 可用于预测、时间序列建模以及发现各种变量之间的因果关系
- 本质：数据拟合（函数拟合）+ 统计分析（显著性检验、区间估计等）

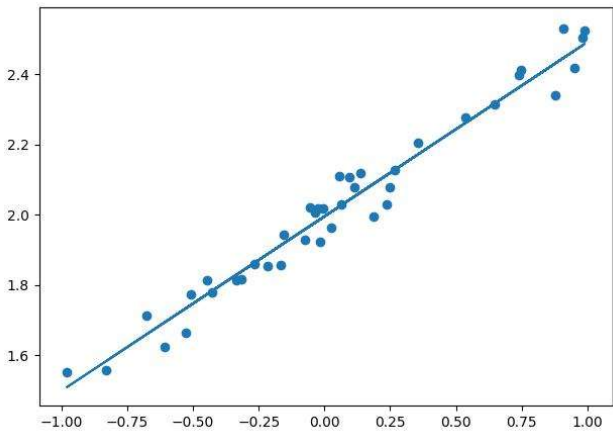
回归分析的作用

- 1) 挑选与因变量相关的自变量;
- 2) 描述因变量与自变量之间的关系强度;
- 3) 生成模型, 通过自变量来预测因变量;
- 4) 根据模型, 通过因变量, 来控制自变量。

回归分析方法



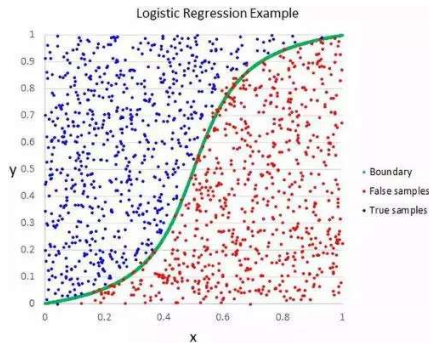
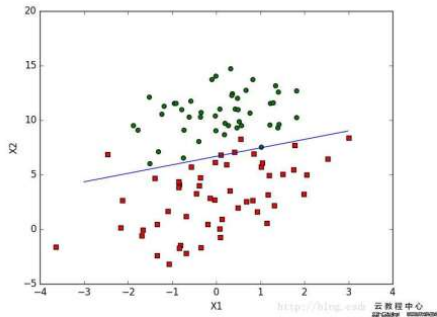
1. 线性回归



线性回归

- 1) 自变量与因变量之间必须有线性关系;
- 2) 多元回归存在多重共线性, 自相关性和异方差性;
- 3) 线性回归对异常值非常敏感。它会严重影响回归线, 最终影响预测值;
- 4) 多重共线性会增加系数估计值的方差, 使得估计值对于模型的轻微变化异常敏感, 结果就是系数估计值不稳定;
- 5) 在存在多个自变量的情况下, 我们可以使用向前选择法, 向后剔除法和逐步筛选法来选择最重要的自变量。

2. Logistic回归



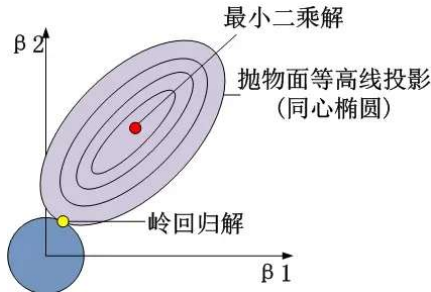
Logistic回归

- Logistic回归广泛用于分类问题;
- Logistic回归不要求自变量和因变量存在线性的关系。它可以处理多种类型的关系,因为它对预测的相对风险指数使用了一个非线性的log转换;
- Logistic回归需要较大的样本量,因为在样本数量较少的情况下,极大似然估计的效果比普通的最小二乘法差;
- 自变量之间应该互不相关,即不存在多重共线性。然而,在分析和建模中,我们可以选择包含分类变量相互作用的影晌;
- 如果因变量的值是定序变量,则称它为序Logistic回归。

3. 岭回归 (ridge regression)

- 又称脊回归、吉洪诺夫正则化 (Tikhonov regularization)
- 目的：给回归估计值添加一个偏差值，来降低标准误差（方差值）

$$= \operatorname{argmin}_{\beta \in \mathbb{R}^p} \underbrace{\|y - X\beta\|_2^2}_{\text{Loss}} + \lambda \underbrace{\|\beta\|_2^2}_{\text{Penalty}}$$



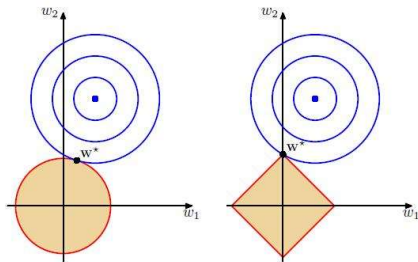
岭回归

- 1) 除常数项以外，岭回归的假设与最小二乘回归相同；
- 2) 它收缩了相关系数的值，但没有达到零，这表明它不具有特征选择功能；
- 3) 这是一个正则化方法，并且使用的是 L2 正则化。

4. 套索回归 (Lasso regression)

- 目标：使一些参数估计结果等于零。使用的惩罚值越大，估计值会越趋近于零。
- 从给定的n个变量之中选择变量（特征选择）

$$= \operatorname{argmin}_{\beta \in \mathbb{R}^p} \underbrace{\|y - X\beta\|_2^2}_{\text{Loss}} + \lambda \underbrace{\|\beta\|_1}_{\text{Penalty}}$$



套索回归

- 1) 除常数项以外，这种回归的假设与最小二乘回归类似；
- 2) 它将收缩系数缩减至零（等于零），这确实有助于特征选择；
- 3) 这是一个正则化方法，使用的是 L1 正则化；
- 4) 如果一组预测因子是高度相关的，套索回归会选出其中一个因子并且将其它因子收缩为零。

相关模型

- 稀疏优化
- 压缩感知

5. ElasticNet 回归

- ElasticNet 回归是套索回归和岭回归的组合物。它会事先使用 L1 和 L2 作为正则化矩阵进行训练
- 1) 在高度相关变量的情况下，它会产生群体效应；
- 2) 选择变量的数目没有限制；
- 3) 它可以承受双重收缩。

如何选择回归模型？

- 数据探索：识别变量的关系和影响
- 比较不同模型的拟合优点：分析不同的指标参数，如统计意义的参数，R-square，调整 R-square，AIC，BIC以及误差项，另一个是 Mallows' Cp 准则
- 交叉验证：评估预测模型最好的方法。需将数据集分成两份（一份用于训练，一份用于验证）。使用观测值和预测值之间的均方差即可快速衡量预测精度。
- 回归正则化方法（套索，岭和ElasticNet）在高维数据和数据集变量之间存在多重共线性的情况下运行良好。

诊断回归分析结果

- 1.自变量与因变量是否具有预期的关系
- 2.自变量对模型是否有帮助
- 3.残差是否有空间聚类
- 4.模型是否出现了倾向性
- 5.自变量中是否存在冗余
- 6.评估模型的性能

数学建模的基本方法

机理分析

测试分析

由于客观事物内部规律的复杂及人们认识程度的限制，无法分析实际对象内在的因果关系，建立合乎机理规律的数学模型。

通过对数据的统计分析，找出与数据拟合最好的模型
回归模型是用统计分析方法建立的最常用的一类模型

- 不涉及回归分析的数学原理和方法
- 通过实例讨论如何选择不同类型的模型
- 对软件得到的结果进行分析，对模型进行改进

统计回归模型

1 牙膏的销售量

2 软件开发人员的薪金

3 酶促反应

4 投资额与国民生产总值和
物价指数

1 牙膏的销售量

问题 建立牙膏销售量与价格、广告投入之间的模型
预测在不同价格和广告费用下的牙膏销售量
收集了30个销售周期本公司牙膏销售量、价格、
广告费用，及同期其它厂家同类牙膏的平均售价

销售周期	本公司价格(元)	其它厂家价格(元)	广告费用(百万元)	价格差(元)	销售量(百万支)
1	3.85	3.80	5.50	-0.05	7.38
2	3.75	4.00	6.75	0.25	8.51
...	...	...	...	...	...
29	3.80	3.85	5.80	0.05	7.93
30	3.70	4.25	6.80	0.55	9.26

基本模型

y ~ 公司牙膏销售量

x_1 ~ 其它厂家与本公司价格差

x_2 ~ 公司广告费用

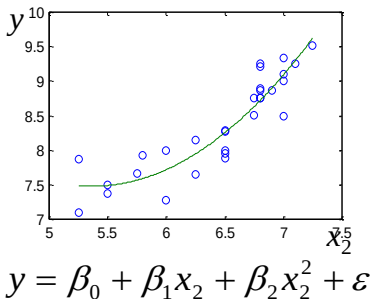
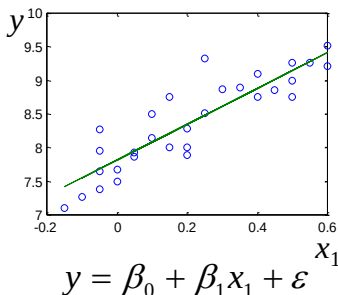
$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_2^2 + \varepsilon$$

y ~ 被解释变量 (因变量)

x_1, x_2 ~ 解释变量 (回归变量, 自变量)

$\beta_0, \beta_1, \beta_2, \beta_3$ ~ 回归系数

ε ~ 随机误差 (均值为零的正态分布随机变量)



模型求解

MATLAB 统计工具箱

$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_2^2 + \varepsilon$ 由数据 y, x_1, x_2 估计 β

`[b,bint,r,rint,stats]=regress(y,x,alpha)`

输入 $y \sim n$ 维数据向量

$x = [1 \ x_1 \ x_2 \ x_2^2] \sim n \times 4$ 数据矩阵, 第1列为全1向量

α (置信水平, 0.05)

输出 $b \sim \beta$ 的估计值

$bint \sim b$ 的置信区间

$r \sim$ 残差向量 $y - xb$

$rint \sim r$ 的置信区间

参数	参数估计值	置信区间
β_0	17.3244	[5.7282 28.9206]
β_1	1.3070	[0.6829 1.9311]
β_2	-3.6956	[-7.4989 0.1077]
β_3	0.3486	[0.0379 0.6594]
$R^2=0.9054 \quad F=82.9409 \quad p=0.0000$		

$Stats \sim$
检验统计量
 R^2, F, p

结果分析

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_2^2 + \varepsilon$$

参数	参数估计值	置信区间
β_0	17.3244	[5.7282 28.9206]
β_1	1.3070	[0.6829 1.9311]
β_2	-3.6956	[-7.4989 0.1077]
β_3	0.3486	[0.0379 0.6594]
$R^2=0.9054$ $F=82.9409$ $p=0.0000$		

y 的90.54%可由模型确定

F 远超过 F 检验的临界值

p 远小于 $\alpha=0.05$

模型从整体上看成立

β_2 的置信区间包含零点
(右端点距零点很近)

x_2 对因变量 y 的
影响不太显著

x_2^2 项显著

可将 x_2 保留在模型中



销售量预测

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \hat{\beta}_3 x_2^2$$

价格差 x_1 =其它厂家价格 x_3 -本公司价格 x_4

估计 x_3 调整 x_4 $\Rightarrow$ 控制 x_1 $\Rightarrow$ 通过 x_1, x_2 预测 y

控制价格差 $x_1=0.2$ 元，投入广告费 $x_2=650$ 万元

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \hat{\beta}_3 x_2^2 = 8.2933 \text{ (百万支)}$$

销售量预测区间为 [7.8230, 8.7636] (置信度95%)

上限用作库存管理的目标值 下限用来把握公司的现金流

若估计 $x_3=3.9$ ，设定 $x_4=3.7$ ，则可以95%的把握知道销售额在 $7.8320 \times 3.7 \approx 29$ (百万元) 以上

模型改进

x_1 和 x_2 对 y
的影响独立



x_1 和 x_2 对 y
的影响有
交互作用

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_2^2 + \varepsilon$$

参数	参数估计值	置信区间
β_0	17.3244	[5.7282 28.9206]
β_1	1.3070	[0.6829 1.9311]
β_2	-3.6956	[-7.4989 0.1077]
β_3	0.3486	[0.0379 0.6594]
$R^2=0.9054$ $F=82.9409$ $p=0.0000$		

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_2^2 + \beta_4 x_1 x_2 + \varepsilon$$

参数	参数估计值	置信区间
β_0	29.1133	[13.7013 44.5252]
β_1	11.1342	[1.9778 20.2906]
β_2	-7.6080	[-12.6932 -2.5228]
β_3	0.6712	[0.2538 1.0887]
β_4	-1.4777	[-2.8518 -0.1037]
$R^2=0.9209$ $F=72.7771$ $p=0.0000$		

两模型销售量预测比较

控制价格差 $x_1=0.2$ 元，投入广告费 $x_2=6.5$ 百万元

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \hat{\beta}_3 x_2^2$$

$$\hat{y} = 8.2933 \text{ (百万支)}$$

$$\text{区间 } [7.8230, 8.7636]$$

$$\hat{y} = \beta_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \hat{\beta}_3 x_2^2 + \hat{\beta}_4 x_1 x_2$$

$$\hat{y} = 8.3272 \text{ (百万支)}$$

$$\text{区间 } [7.8953, 8.7592]$$

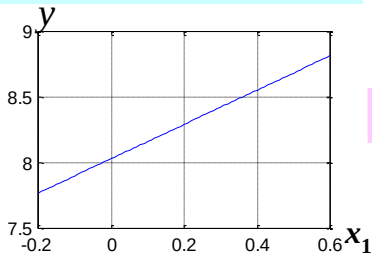
$\hat{y}$ 略有增加

预测区间长度更短

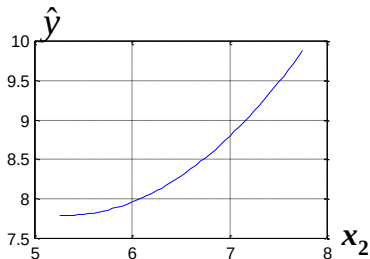
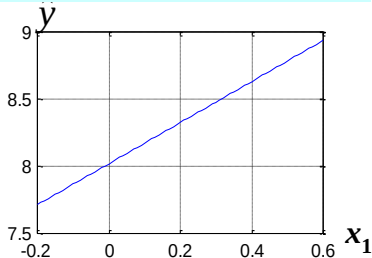
两模型 $\hat{y}$ 与 x_1, x_2 关系的比较

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \hat{\beta}_3 x_2^2$$

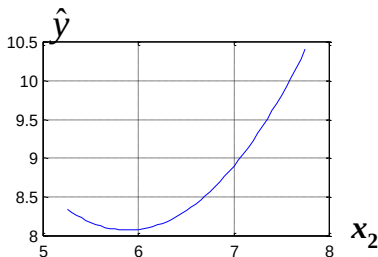
$$\hat{y} = \beta_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \hat{\beta}_3 x_2^2 + \hat{\beta}_4 x_1 x_2$$



$x_2 = 6.5$



$x_1 = 0.2$



交互作用影响的讨论

$$\hat{y} = \beta_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \hat{\beta}_3 x_2^2 + \hat{\beta}_4 x_1 x_2$$

价格差 $x_1=0.1$

$$\hat{y}|_{x_1=0.1} = 30.2267 - 7.7558x_2 + 0.6712x_2^2$$

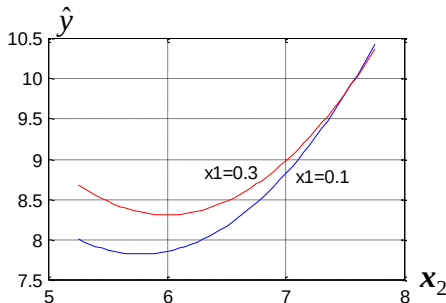
价格差 $x_1=0.3$

$$\hat{y}|_{x_1=0.3} = 32.4535 - 8.0513x_2 + 0.6712x_2^2$$

$$x_2 < 7.5357 \Rightarrow \hat{y}|_{x_1=0.3} > \hat{y}|_{x_1=0.1}$$

价格优势会使销售量增加

加大广告投入使销售量增加
(x_2 大于6百万元)



价格差较小时增加的
速率更大

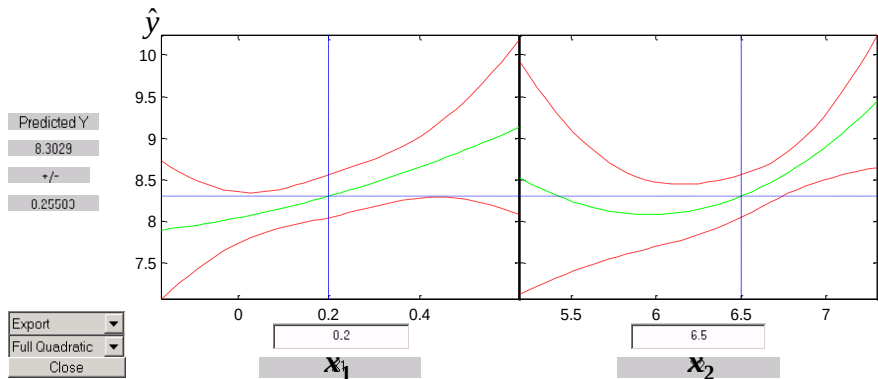


价格差较小时更需要靠广告
来吸引顾客的眼球

完全二次多项式模型

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1 x_2 + \beta_4 x_1^2 + \beta_5 x_2^2 + \varepsilon$$

MATLAB中有命令`rstool`直接求解



从输出 **Export** 可得 $\hat{\beta} = (\hat{\beta}_0, \hat{\beta}_1, \hat{\beta}_2, \hat{\beta}_3, \hat{\beta}_4, \hat{\beta}_5)$

2 软件开发人员的薪金

建立模型研究薪金与资历、管理责任、教育程度的关系

分析人事策略的合理性，作为新聘用人员薪金的参考

46名软件开发人员的档案资料

编号	薪金	资历	管理	教育	编号	薪金	资历	管理	教育
01	13876	1	1	1	42	27837	16	1	2
02	11608	1	0	3	43	18838	16	0	2
03	18701	1	1	3	44	17483	16	0	1
04	11283	1	0	2	45	19207	17	0	2
...	...	...	...	...	46	19346	20	0	1

资历~ 从事专业工作的年数；管理~ 1=管理人员，0=非管理人员；教育~ 1=中学，2=大学，3=更高程度

分析与假设

$y \sim$ 薪金, $x_1 \sim$ 资历 (年)

$x_2 = 1 \sim$ 管理人员, $x_2 = 0 \sim$ 非管理人员

教育

1=中学
2=大学
3=更高

$$x_3 = \begin{cases} 1, & \text{中学} \\ 0, & \text{其它} \end{cases}$$

$$x_4 = \begin{cases} 1, & \text{大学} \\ 0, & \text{其它} \end{cases}$$

中学: $x_3 = 1, x_4 = 0$;

大学: $x_3 = 0, x_4 = 1$;

更高: $x_3 = 0, x_4 = 0$

资历每加一年薪金的增长是常数;

管理、教育、资历之间无交互作用

线性回归模型

$$y = a_0 + a_1 x_1 + a_2 x_2 + a_3 x_3 + a_4 x_4 + \varepsilon$$

$a_0, a_1, \dots, a_4$ 是待估计的回归系数, ε 是随机误差

模型求解

$$y = a_0 + a_1x_1 + a_2x_2 + a_3x_3 + a_4x_4 + \varepsilon$$

参数	参数估计值	置信区间
a_0	11032	[10258 11807]
a_1	546	[484 608]
a_2	6883	[6248 7517]
a_3	-2994	[-3826 -2162]
a_4	148	[-636 931]
$R^2=0.957 \quad F=226 \quad p=0.000$		

资历增加1年薪
金增长546

管理人员薪金多
6883

中学程度薪金比更
高的少2994

大学程度薪金比更
高的多148

$R^2, F, p \rightarrow$ 模型整体上可用

$x_1 \sim$ 资历(年)

$x_2 = 1 \sim$ 管理, x_2
 $= 0 \sim$ 非管理

中学: $x_3 = 1, x_4 = 0$; 大

学: $x_3 = 0, x_4 = 1$; 更高:
 $x_3 = 0, x_4 = 0$.

a_4 置信区间包含零点,
解释不可靠!

结果分析

残差分析方法

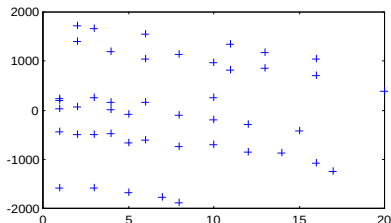
$$\hat{y} = \hat{a}_0 + \hat{a}_1 x_1 + \hat{a}_2 x_2 + \hat{a}_3 x_3 + \hat{a}_4 x_4$$

残差 $e = y - \hat{y}$

管理与教育的组合

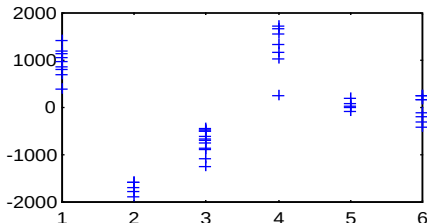
组合	1	2	3	4	5	6
管理	0	1	0	1	0	1
教育	1	1	2	2	3	3

e 与资历 x_1 的关系



残差大概分成3个水平，
6种管理—教育组合混在一起，未正确反映。

e 与管理—教育组合的关系



残差全为正，或全为负，管
理—教育组合处理不当

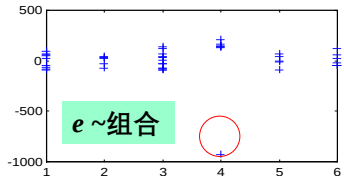
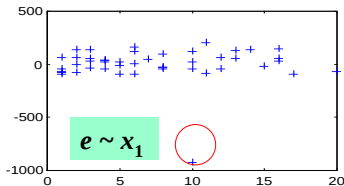
应在模型中增加管理 x_2 与教育 x_3, x_4 的交互项

进一步的模型

增加管理 x_2 与教育 x_3, x_4 的交互项

$$y = a_0 + a_1x_1 + a_2x_2 + a_3x_3 + a_4x_4 + a_5x_2x_3 + a_6x_2x_4 + \varepsilon$$

参数	参数估计值	置信区间
a_0	11204	[11044 11363]
a_1	497	[486 508]
a_2	7048	[6841 7255]
a_3	-1727	[-1939 -1514]
a_4	-348	[-545 -152]
a_5	-3071	[-3372 -2769]
a_6	1836	[1571 2101]
$R^2=0.999$ $F=554$ $p=0.000$		



R^2, F 有改进, 所有回归系数置信区间都不含零点, 模型完全可用

消除了不正常现象

异常数据(33号)应去掉

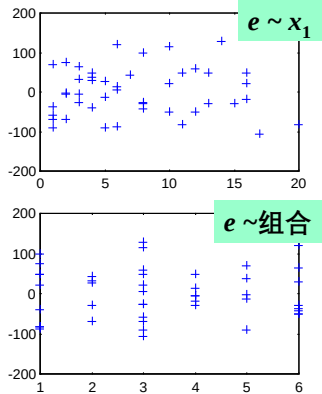
去掉异常数据后的结果

参数	参数估计值	置信区间
a_0	11200	[11139 11261]
a_1	498	[494 503]
a_2	7041	[6962 7120]
a_3	-1737	[-1818 -1656]
a_4	-356	[-431 -281]
a_5	-3056	[-3171 -2942]
a_6	1997	[1894 2100]
$R^2 = 0.9998$ $F=36701$ $p=0.0000$		

R^2 : 0.957 $\rightarrow$ 0.999 $\rightarrow$ 0.9998

F : 226 $\rightarrow$ 554 $\rightarrow$ 36701

置信区间长度更短



残差图十分正常

最终模型的结果可以应用

模型应用

$$\hat{y} = \hat{a}_0 + \hat{a}_1x_1 + \hat{a}_2x_2 + \hat{a}_3x_3 + \hat{a}_4x_4 + \hat{a}_5x_2x_3 + \hat{a}_6x_2x_4$$

制订6种管理—教育组合人员的“基础”薪金(资历为0)

$x_1=0$; $x_2=1\sim$ 管理, $x_2=0\sim$ 非管理

中学: $x_3=1, x_4=0$; 大学: $x_3=0, x_4=1$; 更高: $x_3=0, x_4=0$

组合	管理	教育	系数	“基础”薪金
1	0	1	a_0+a_3	9463
2	1	1	$a_0+a_2+a_3+a_5$	13448
3	0	2	a_0+a_4	10844
4	1	2	$a_0+a_2+a_4+a_6$	19882
5	0	3	a_0	11200
6	1	3	a_0+a_2	18241

大学程度管理人员比更高程度管理人员的薪金高

大学程度非管理人员比更高程度非管理人员的薪金略低

软件开发人员的薪金

对定性因素(如管理、教育)，可以引入0-1变量处理，0-1变量的个数应比定性因素的水平少1

残差分析方法可以发现模型的缺陷，引入交互作用项常常能够改善模型

剔除异常数据，有助于得到更好的结果

注：可以直接对6种管理—教育组合引入5个0-1变量

3 酶促反应

问题

研究酶促反应（酶催化反应）中嘌呤霉素对反应速度与底物（反应物）浓度之间关系的影响

建立数学模型，反映该酶促反应的速度与底物浓度以及经嘌呤霉素处理与否之间的关系

方案

设计了两个实验：酶经过嘌呤霉素处理；酶未经嘌呤霉素处理。实验数据见下表：

底物浓度(ppm)		0.02		0.06		0.11		0.22		0.56		1.10	
反应速度	处理	76	47	97	107	123	139	159	152	191	201	207	200
	未处理	67	51	84	86	98	115	131	124	144	158	160	/

线性化模型

$$y = \frac{\beta_1 x}{\beta_2 + x} \quad \Rightarrow \quad \frac{1}{y} = \frac{1}{\beta_1} + \frac{\beta_2}{\beta_1} \frac{1}{x} = \theta_1 + \theta_2 \frac{1}{x}$$

对 β_1, β_2 非线性



对 θ_1, θ_2 线性

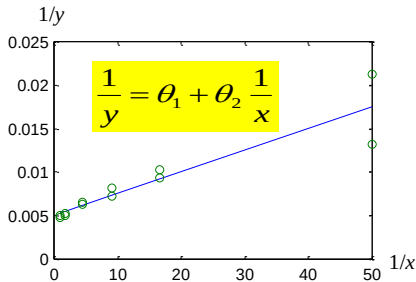
经嘌呤霉素处理后实验数据的估计结果

参数	参数估计值 ($\times 10^{-3}$)	置信区间 ($\times 10^{-3}$)
θ_1	5.107	[3.539 6.676]
θ_2	0.247	[0.176 0.319]
$R^2=0.8557$ $F=59.2975$ $p=0.0000$		

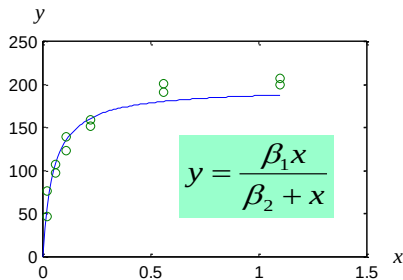
$$\hat{\beta}_1 = 1/\hat{\theta}_1 = 195.8027$$

$$\hat{\beta}_2 = \hat{\theta}_2/\hat{\theta}_1 = 0.04841$$

线性化模型结果分析



$1/x$ 较小时有很好的线性趋势， $1/x$ 较大时出现很大的起落



x 较大时， y 有较大偏差

- 参数估计时， x 较小（ $1/x$ 很大）的数据控制了回归参数的确定

非线性模型参数估计

MATLAB 统计工具箱

[beta,R,J] = nlinfit (x,y,'model',beta0)

输入

x~自变量数据矩阵
y ~因变量数据向量

$$y = \frac{\beta_1 x}{\beta_2 + x}$$

beta0~线性化
模型估计结果

model ~模型的函数M文件名

beta0 ~给定的参数初值

x= ; y= ;

beta0=[195.8027 0.04841];

[beta,R,J]=nlinfit(x,y,'f1',beta0);

betaci=nlparci(beta,R,J);

beta, betaci

输出

beta ~参数的估计值
R ~残差, J ~估计预测误差的Jacobi矩阵

beta的置信区间

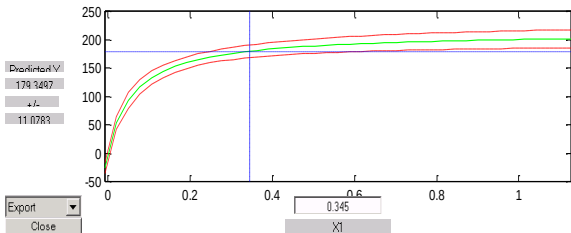
betaci =nlparci(beta,R,J)

function y=f1(beta, x)
y=beta(1)*x./(beta(2)+x);

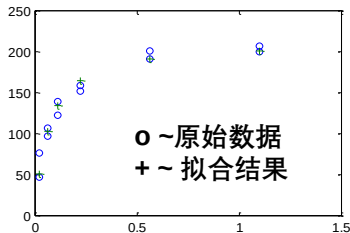
非线性模型结果分析

参数	参数估计值	置信区间
β_1	212.6819	[197.2029 228.1609]
β_2	0.0641	[0.0457 0.0826]

最终反应速度为 $\hat{\beta}_1 = 212.6831$
半速度点(达到最终速度一半时的x
值)为 $\hat{\beta}_2 = 0.0641$



$$y = \frac{\beta_1 x}{\beta_2 + x}$$



拖动画面的十字线，得
y的预测值和预测区间
画面左下方的Export
输出其它统计结果。

剩余标准差s= 10.9337

混合反应模型

在同一模型中考虑嘌呤霉素处理的影响

$$y = \frac{\beta_1 x}{\beta_2 + x} \quad \Rightarrow \quad y = \frac{(\beta_1 + \gamma_1 x_2) x_1}{(\beta_2 + \gamma_2 x_2) + x_1}$$

x_1 为底物浓度， x_2 为一示性变量

$x_2=1$ 表示经过处理， $x_2=0$ 表示未经处理

β_1 是未经处理的最终反应速度

γ_1 是经处理后最终反应速度的增长值

β_2 是未经处理的反应的半速度点

γ_2 是经处理后反应的半速度点的增长值

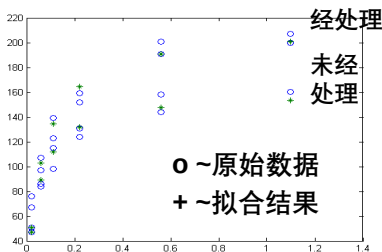
混合模型求解

$$y = \frac{(\beta_1 + \gamma_1 x_2) x_1}{(\beta_2 + \gamma_2 x_2) + x_1}$$

参数初值 (基于对数据的分析) $\beta_1^0 = 170, \gamma_1^0 = 60, \beta_2^0 = 0.05, \gamma_2^0 = 0.01$

估计结果和预测

参数	参数估计值	置信区间
β_1	160.2802	[145.8466 174.7137]
β_2	0.0477	[0.0304 0.0650]
γ_1	52.4035	[32.4130 72.3941]
γ_2	0.0164	[-0.0075 0.0403]



剩余标准差s= 10.4000

γ_2 置信区间包含零点，表明 γ_2 对因变量y的影响不显著



经嘌呤霉素处理的作用不影响半速度点参数

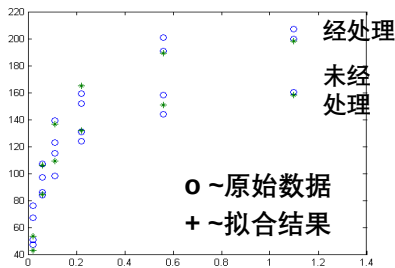
简化的混合模型

$$y = \frac{(\beta_1 + \gamma_1 x_2) x_1}{(\beta_2 + \gamma_2 x_2) + x_1}$$

$$y = \frac{(\beta_1 + \gamma_1 x_2) x_1}{\beta_2 + x_1}$$

估计结果和预测

参数	参数估计值	置信区间
β_1	166.6025	[154.4886 178.7164]
β_2	0.0580	[0.0456 0.0703]
γ_1	42.0252	[28.9419 55.1085]



简化的混合模型形式简单，参数置信区间不含零点

剩余标准差 $s = 10.5851$ ，比一般混合模型略大

一般混合模型与简化混合模型预测比较

$$y = \frac{(\beta_1 + \gamma_1 x_2) x_1}{(\beta_2 + \gamma_2 x_2) + x_1}$$

$$y = \frac{(\beta_1 + \gamma_1 x_2) x_1}{\beta_2 + x_1}$$

预测区间为
预测值 $\pm \Delta$

实际值	一般模型预测值	Δ (一般模型)	简化模型预测值	Δ (简化模型)
67	47.3443	9.2078	42.7358	5.4446
51	47.3443	9.2078	42.7358	5.4446
84	89.2856	9.5710	84.7356	7.0478
...	...	...	...	...
191	190.8329	9.1484	189.0574	8.8438
201	190.8329	9.1484	189.0574	8.8438
207	200.9688	11.0447	198.1837	10.1812
200	200.9688	11.0447	198.1837	10.1812

简化混合模型的预测区间较短，更为实用、有效

酶促反应

机理分析

反应速度与底物浓度的关系



非线性关系

求解线性模型



求解非线性模型

发现问题，
得参数初值

嘌呤霉素处理对反应速度与底物浓度关系的影响



混合模型



简化模型

引入0-1变量

检查参数置信区
间是否包含零点

注：非线性模型拟合程度的评价无法直接利用线性模型的方法，但 R^2 与 s 仍然有效。

4 投资额与国民生产总值和物价指数

问题

建立投资额模型，研究某地区实际投资额与国民生产总值 (GNP) 及物价指数 (PI) 的关系

根据对未来GNP及PI的估计，预测未来投资额

该地区连续20年的统计数据

年份 序号	投资额	国民生产 总值	物价 指数	年份 序号	投资额	国民生 产总值	物价 指数
1	90.9	596.7	0.7167	11	229.8	1326.4	1.0575
2	97.4	637.7	0.7277	12	228.7	1434.2	1.1508
3	113.5	691.1	0.7436	13	206.1	1549.2	1.2579
4	125.7	756.0	0.7676	14	257.9	1718.0	1.3234
5	122.8	799.0	0.7906	15	324.1	1918.3	1.4005
6	133.3	873.4	0.8254	16	386.6	2163.9	1.5042
7	149.3	944.0	0.8679	17	423.0	2417.8	1.6342
8	144.2	992.7	0.9145	18	401.9	2631.7	1.7842
9	166.4	1077.6	0.9601	19	474.9	2954.7	1.9514
10	195.0	1185.9	1.0000	20	424.5	3073.0	2.0688

投资额与国民生产总值和物价指数

分析

许多经济数据在时间上有一定的**滞后性**

以时间为序的数据，称为**时间序列**

时间序列中同一变量的顺序观测值之间存在**自相关**

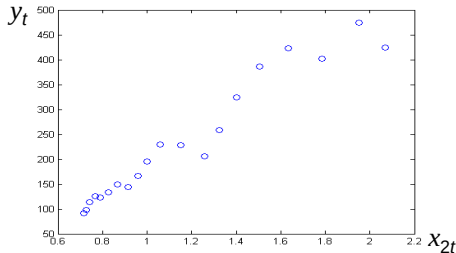
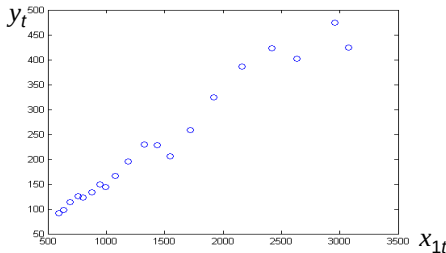
若采用普通回归模型直接处理，将会出现不良后果

需要诊断并消除数据的自相关性，建立新的模型

年份 序号	投资额	国民生产 总值	物价 指数	年份 序号	投资额	国民生 产总值	物价 指数
1	90.9	596.7	0.7167	11	229.8	1326.4	1.0575
2	97.4	637.7	0.7277	12	228.7	1434.2	1.1508
3	113.5	691.1	0.7436	13	206.1	1549.2	1.2579
4	125.7	756.0	0.7676	14	257.9	1718.0	1.3234
...	...	...	...	...	...	...	...

基本回归模型

$t \sim$ 年份, $y_t \sim$ 投资额, $x_{1t} \sim$ GNP, $x_{2t} \sim$ 物价指数



投资额与 GNP 及物价指数间均有很强的线性关系

$$y_t = \beta_0 + \beta_1 x_{1t} + \beta_2 x_{2t} + \varepsilon_t \quad \beta_0, \beta_1, \beta_2 \sim \text{回归系数}$$

$\varepsilon_t \sim$ 对 t 相互独立的零均值正态随机变量

基本回归模型的结果与分析

参数	参数估计值	置信区间
β_0	322.7250	[224.3386 421.1114]
β_1	0.6185	[0.4773 0.7596]
β_2	-859.4790	[-1121.4757 -597.4823]
$R^2= 0.9908$ $F= 919.8529$ $p=0.0000$		

$$\hat{y}_t = 322.725 + 0.6185x_{1t} - 859.479x_{2t}$$

剩余标准差
 $s=12.7164$

模型优点

$R^2 = 0.9908$ ，拟合度高

模型缺点

没有考虑时间序列数据的**滞后性影响**
可能忽视了随机误差存在**自相关**；如果存在自相关性，用此模型会有不良后果

自相关性的定性诊断

模型残差 $e_t = y_t - \hat{y}_t$

e_t 为随机误差 ε_t 的估计值

在MATLAB工作区中输出

作残差 $e_t \sim e_{t-1}$ 散点图

大部分点落在第1, 3象限

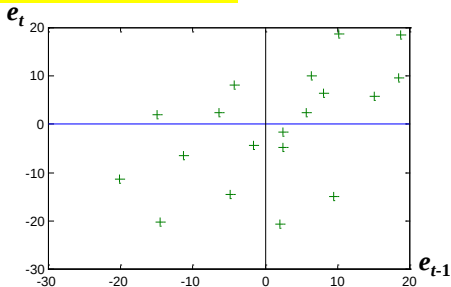
大部分点落在第2, 4象限

自相关性直观判断



基本回归模型的随机误差项 ε_t 存在正的自相关

残差诊断法



ε_t 存在正的自相关



ε_t 存在负的自相关

自回归性的定量诊断

D-W检验

自回归模型 $y_t = \beta_0 + \beta_1 x_{1t} + \beta_2 x_{2t} + \varepsilon_t$, $\varepsilon_t = \rho \varepsilon_{t-1} + u_t$

$\beta_0, \beta_1, \beta_2 \sim$ 回归系数

$\rho \sim$ 自相关系数

$|\rho| \leq 1$

$u_t \sim$ 对 t 相互独立的零均值正态随机变量

$\rho = 0$



无自相关性

$\rho > 0$



存在正自相关性

$\rho < 0$



存在负自相关性

如何估计 ρ



D-W统计量

如何消除自相关性



广义差分法

D-W统计量与D-W检验

$$DW = \frac{\sum_{t=2}^n (e_t - e_{t-1})^2}{\sum_{t=2}^n e_t^2}$$

$$\approx 2 \left[1 - \frac{\sum_{t=2}^n e_t e_{t-1}}{\sum_{t=2}^n e_t^2} \right] \quad n \text{较大}$$

$$\hat{\rho} = \sum_{t=2}^n e_t e_{t-1} / \sum_{t=2}^n e_t^2$$

$$= 2(1 - \hat{\rho})$$

$$-1 \leq \hat{\rho} \leq 1 \rightarrow 0 \leq DW \leq 4$$

$$\hat{\rho} = 1 \rightarrow DW = 0$$

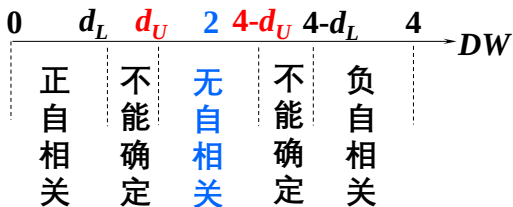
$$\hat{\rho} = -1 \rightarrow DW = 4$$

$$\hat{\rho} = 0 \rightarrow DW = 2$$

检验水平, 样本容量,
回归变量数目

D-W分布表 

检验临界值 d_L 和 d_U



由DW值的大小确定自相关性

广义差分变换

$$DW = 2(1 - \hat{\rho}) \quad \Leftrightarrow \quad \hat{\rho} = 1 - \frac{DW}{2}$$

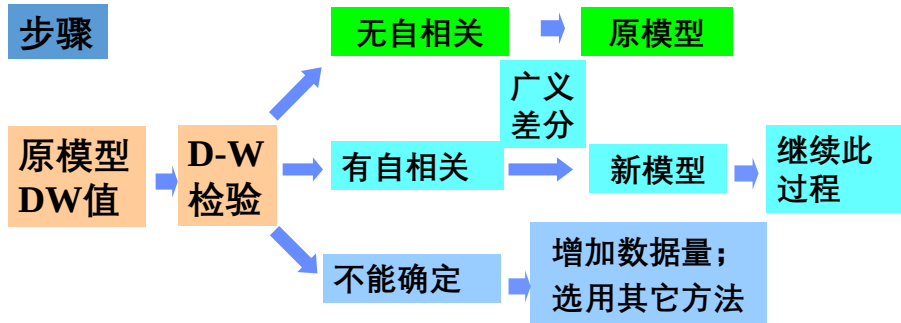
原模型 $y_t = \beta_0 + \beta_1 x_{1t} + \beta_2 x_{2t} + \varepsilon_t, \quad \varepsilon_t = \rho \varepsilon_{t-1} + u_t$

变换 $y_t^* = y_t - \rho y_{t-1}, \quad x_{it}^* = x_{it} - \rho x_{i,t-1}, \quad i = 1, 2$

新模型 $y_t^* = \beta_0^* + \beta_1 x_{1t}^* + \beta_2 x_{2t}^* + u_t \quad \beta_0^* = \beta_0(1 - \rho)$

以 $\beta_0^*, \beta_1, \beta_2$ 为回归系数的普通回归模型

步骤



投资额新模型的建立

原模型
残差 e_t $DW_{old} = 0.8754$

样本容量 $n=20$, 回归
变量数目 $k=3$, $\alpha=0.05$

查表 

临界值 $d_L=1.10$, $d_U=1.54$

作变换

$$y_t^* = y_t - 0.5623y_{t-1}$$

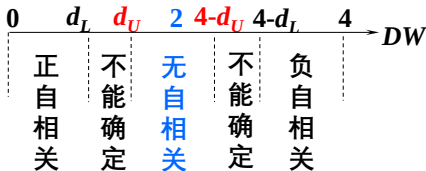
$$x_{it}^* = x_{it} - 0.5623x_{i,t-1}, \quad i=1,2$$

$$DW = \frac{\sum_{t=2}^n (e_t - e_{t-1})^2}{\sum_{t=2}^n e_t^2}$$

$$DW_{old} < d_L$$

原模型有
正自相关

$$\hat{\rho} = 1 - DW / 2 = 0.5623$$



投资额新模型的建立

$$y_t^* = y_t - 0.5623y_{t-1} \quad x_{it}^* = x_{it} - 0.5623x_{i,t-1}, \quad i=1,2$$

$$y_t^* = \beta_0^* + \beta_1 x_{1t}^* + \beta_2 x_{2t}^* + u_t$$

由数据 $y_t^*, x_{1t}^*, x_{2t}^*$ 估计系数 $\beta_0^*, \beta_1, \beta_2$

参数	参数估计值	置信区间
β_0^*	163.4905	[1265.4592 2005.2178]
β_1	0.6990	[0.5751 0.8247]
β_2	-1009.0333	[-1235.9392 -782.1274]
$R^2= 0.9772 \quad F=342.8988 \quad p=0.0000$		

总体效果良好

剩余标准差

$$s_{new} = 9.8277 < s_{old} = 12.7164$$

新模型的自相关性检验

新模型
残差 e_t



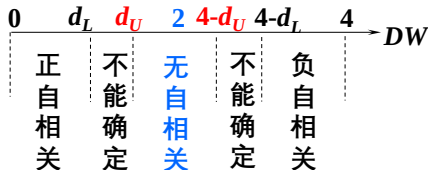
$$DW_{new} = 1.5751$$

样本容量 $n=19$ ，回归
变量数目 $k=3$ ， $\alpha=0.05$

查表



临界值 $d_L=1.08$ ， $d_U=1.53$



$$d_U < DW_{new} < 4 - d_U$$



新模型无自相关性

新模型 $\hat{y}_t^* = 163.4905 + 0.699x_{1t}^* - 1009.033x_{2t}^*$

还原为
原始变量

$$\hat{y}_t = 163.4905 + 0.5623y_{t-1} + 0.699x_{1,t} - 0.3930x_{1,t-1} - 1009.033x_{2,t} + 567.3794x_{2,t-1}$$

一阶自回归模型

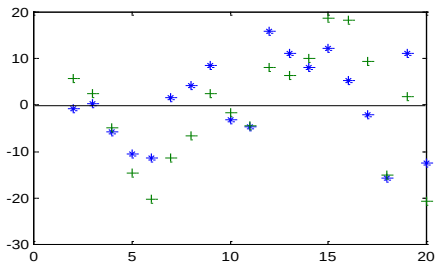
模型结果比较

基本回归模型 $\hat{y}_t = 322.725 + 0.6185x_{1t} - 859.479x_{2t}$

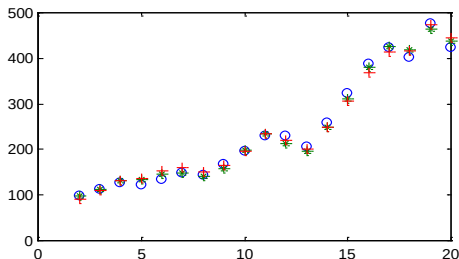
一阶自回归模型

$$\hat{y}_t = 163.4905 + 0.5623y_{t-1} + 0.699x_{1t} - 0.3930x_{1,t-1} - 1009.0333x_{2,t} + 567.3794x_{2,t-1}$$

残差图比较



拟合图比较



新模型 $e_t \sim *$, 原模型 $e_t \sim +$

新模型 $\hat{y}_t \sim *$, 新模型 $\hat{y}_t \sim +$

一阶自回归模型残差 e_t 比基本回归模型要小

投资额预测

对未来投资额 y_t 作预测，需先估计出未来的国民生产总值 x_{1t} 和物价指数 x_{2t}

年份 序号	投资额	国民生产 总值	物价 指数	年份 序号	投资额	国民生 产总值	物价 指数
1	90.9	596.7	0.7167	18	401.9	2631.7	1.7842
2	97.4	637.7	0.7277	19	474.9	2954.7	1.9514
3	113.5	691.1	0.7436	20	424.5	3073.0	2.0688

设已知 $t=21$ 时， $x_{1t}=3312$ ， $x_{2t}=2.1938$

基本回归模型 $\hat{y}_t = 485.6720$

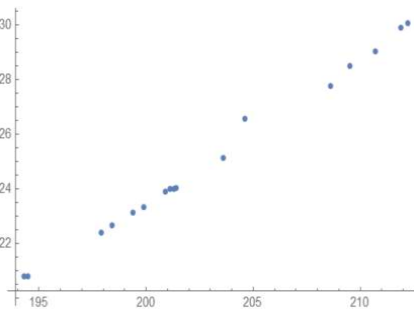
一阶自回归模型 $\hat{y}_t = 469.7638$

$\hat{y}_t$ 较小是由于 $y_{t-1}=424.5$ 过小所致

练习思考题

例1：（杨启帆《数学建模》§ 10.5）沸点与气压的实验分析。

#	沸点	气压	#	沸点	气压
1	194.5	20.79	10	201.3	24.01
2	194.3	20.79	11	203.6	25.14
3	197.9	22.40	12	204.6	26.57
4	198.4	22.67	13	209.5	28.49
5	199.4	23.15	14	208.6	27.76
6	199.9	23.35	15	210.7	29.04
7	200.9	23.89	16	211.9	29.88
8	201.1	23.99	17	212.2	30.06
9	201.4	24.02			

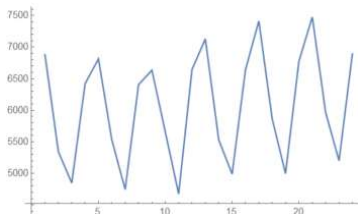


【模型建立】

- 分别作 $y_1 = ax + b$, $y_2 = ax^2 + bx + c$, $y_3 = \exp(ax + b)$, $y_4 = \exp(ax^2 + bx + c)$ 形式的拟合, 得 y_2, y_4 的效果差不多。
- 思考：哪个模型好？

例2：（杨启帆《数学建模》§ 10.5）城市居民用煤量预测。

年份	1季度	2季度	3季度	4季度
1991	6878.4	5343.7	4847.9	6421.9
1992	6815.4	5532.6	4745.6	6406.2
1993	6634.4	5658.5	4674.8	6645.5
1994	7130.2	5532.6	4989.6	6642.3
1995	7413.5	5863.1	4997.4	6776.1
1996	7476.5	5965.5	5202.1	6894.1



【模型建立】

- 居民用煤的主要用途：做饭、取暖。故存在季节性变化。
- 用煤量也与城市人口、经济发展、环境保护等因素有关。
- 模型1：对每个季度的用煤量分别作线性函数拟合。
- 模型2：作拟合 $y = ax + b + c_1 \cos\left(\frac{\pi x}{2}\right) + c_2 \sin\left(\frac{\pi x}{2}\right)$ 。
- 思考：比较模型1和模型2，哪个更好？

例3: (杨启帆《数学建模》§10.6) 木材体积的快速折算。

#	直径	高度	体积	#	直径	高度	体积	#	直径	高度	体积	#	直径	高度	体积
1	8.3	70	10.3	9	11.1	80	22.6	17	12.9	85	33.8	25	16.3	77	42.6
2	8.6	65	10.3	10	11.2	75	19.9	18	13.3	86	27.4	26	17.3	81	55.4
3	8.8	63	10.2	11	11.3	79	24.2	19	13.7	71	25.7	27	17.5	82	55.7
4	10.5	72	16.4	12	11.4	76	21.0	20	13.8	64	24.9	28	17.9	80	58.3
5	10.7	81	18.8	13	11.4	76	21.4	21	14.0	78	34.5	29	18.0	80	51.5
6	10.8	83	19.7	14	11.7	69	21.3	22	14.2	80	31.7	30	18.0	80	51.0
7	11.0	66	15.6	15	12.0	75	19.1	23	14.5	74	36.3	31	20.6	87	77.0
8	11.0	75	18.2	16	12.9	74	22.2	24	16.0	72	38.3				

- 模型1: 猜测 $V = \lambda d^2 h$ 。算得 $\lambda \approx 0.0021$, 误差 ≈ 13.47 。
- 模型2: 猜测 $\ln V = c_0 + c_1 \ln d + c_2 \ln h$ 。算得 $\exp(c_0) \approx 0.0013$, $c_1 \approx 1.98$, $c_2 \approx 1.12$, 误差 ≈ 13.45 。
- 思考: 模型2是否有合理解释?

例4：（杨启帆《数学建模》§ 10.8）Simpson悖论。
根据下列谋杀案的判决情况分析美国司法制度是否公正？

被告人	被害人	判处死刑	未判死刑
白	白	19	132
白	黑	0	9
黑	白	11	52
黑	黑	6	97

【模型建立】

- 假设是公正的，则每起谋杀案件判处死刑的概率 $p \approx \frac{36}{326} \approx 0.11$ ，应当与被告人和被害人的种族无关。
- 上述4种情形下的比例 $p_1 = \frac{19}{151} \approx 0.13$ ， $p_2 = 0$ ， $p_3 = \frac{11}{63} \approx 0.17$ ， $p_4 = \frac{6}{103} \approx 0.06$ 。显然， p_2, p_3, p_4 与 p 差别很大。
- 对于每种情形检验“判处死刑的案件数服从概率 p 的二项分布”。
- 注：“被告人/被害人是白人/黑人”不是随机事件，“是否判决死刑”是随机事件。不应作独立性检验，而应作参数检验。

例5：（CUMCM2012C）脑卒中发病环境因素分析及干预。

脑卒中（俗称脑中风）是目前威胁人类生命的严重疾病之一，它的发生是一个漫长的过程，一旦得病就很难逆转。这种疾病的诱发已经被证实与环境因素，包括气温和湿度之间存在密切的关系。对脑卒中的发病环境因素进行分析，其目的是为了进行疾病的风险评估，对脑卒中高危人群能够及时采取干预措施，也让尚未得病的健康人，或者亚健康人了解自己得脑卒中风险程度，进行自我保护。同时，通过数据模型的建立，掌握疾病发病率的规律，对于卫生行政部门和医疗机构合理调配医务力量、改善就诊治疗环境、配置床位和医疗药物等都具有实际的指导意义。

数据（见Appendix-C1）来源于中国某城市各家医院2007年1月至2010年12月的脑卒中发病病例信息以及相应期间当地的逐日气象资料（Appendix-C2）。请你们根据题目提供的数据，回答以下问题：

1. 根据病人基本信息，对发病人群进行统计描述。
2. 建立数学模型研究脑卒中发病率与气温、气压、相对湿度间的关系。
3. 查阅和搜集文献中有关脑卒中高危人群的重要特征和关键指标，结合1、2中所得结论，对高危人群提出预警和干预的建议方案。

【模型建立】

- 数据有许多遗漏和错误，格式也不统一。首先需要整理数据。
- 由数据可得，病人中男性占54%，女性占46%，男性比女性易发病。
- 结合各年龄的人口数，发病率（定义为 $\frac{\text{病人数}}{\text{人口数}}$ ）随年龄上升。
- 发病率与湿度的相关系数约-9.5%，空气干燥使发病率上升。
- 发病率与温差的相关系数约3.3%，天气忽冷忽热使发病率上升。

例6：（CUMCM2017C）颜色与物质浓度辨识。

比色法是目前常用的一种检测物质浓度的方法，即把待测物质制备成溶液后滴在特定的白色试纸表面，等其充分反应以后获得一张有颜色的试纸，再把该颜色试纸与一个标准比色卡进行对比，就可以确定待测物质的浓度档位了。由于每个人对颜色的敏感差异和观测误差，使得这一方法在精度上受到很大影响。随着照相技术和颜色分辨率的提高，希望建立颜色读数和物质浓度的数量关系，即只要输入照片中的颜色读数就能够获得待测物质的浓度。试根据附件所提供的有关颜色读数和物质浓度数据完成下列问题：

1. 附件Data1.xls中分别给出了5种物质在不同浓度下的颜色读数，讨论从这5组数据中能否确定颜色读数和物质浓度之间的关系，并给出一些准则来评价这5组数据的优劣。
2. 对附件Data2.xls中的数据，建立颜色读数和物质浓度的数学模型，并给出模型的误差分析。
3. 探讨数据量和颜色维度对模型的影响。

【模型建立】

- 观察数据，推测溶液颜色的 R, G, B, H, S 值读数是溶液浓度 ρ 的函数。
- 可选某个主成分指标 $X = c_1R + c_2G + c_3B + c_4H + c_5S$ ，把 ρ 表示成 X 的非线性函数，并把 $\rho(X)$ 的误差作为评价数据优劣的准则。

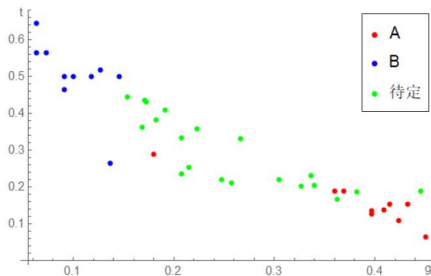
例7：（CUMCM2000A）DNA序列分类。

下面有20个已知类别的人工制造的序列，其中序列标号1~10为A类，11~20为B类。请从中提取特征，构造分类方法，并用这些已知类别的序列，衡量你的方法是否足够好。然后用你认为满意的方法，对另外20个未标明类别的人工序列（标号21~40）进行分类，把结果用序号标明它们的类别。

1. aggcacggaaaaacgggaataacggaggaggacttggcacggcattacacggaggacgaggta
aaggaggcttgtctacggccggaagtgaagggggatatgaccgcttgg
2. cggaggacaaacgggatggcggtattggaggtggcggactgttcggggaattattcggtttaaacg
ggacaaggaaggcggctggaacaaccggacggtggcagcaaagga
3.
11. gttagatttaacgtttttatggaatttatggaattataaaatttaaaatttatatttttaggtaagtaatcca
acgtttttattacttttttaaaattaaatatttatt
12. gtttaattactttatcatttaatttaggttttaattttaaaatttaatttaggtaagatgaatttggttttttaagg
tagttatttaattatcgtaaggaaggttaa
13.

【模型建立】

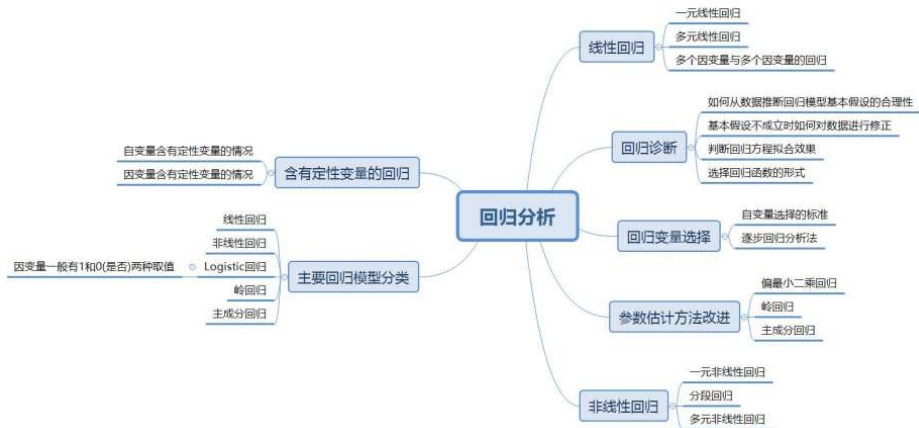
- 经观察后发现，在A类序列中g较多，在B类序列中t较多。
- 故以每个序列中四类碱基的比例(x_1, x_2, x_3, x_4)作为分类指标。
- 当 $x_3 \leq 0.16$ 时，属于A类；否则属于B类。
- 数据量太少，并且缺少生物学知识。
上述分类方法纯属猜测，无法检验。



统计分析方法

- 统计方法
 - 总体与样本、统计量
- 参数估计方法
- 方差分析法
- 相关分析法
- 显著性检验
- 模型选择

回归分析小结



参考书目

- 何晓群，刘文卿，《应用回归分析》，中国人民大学出版社，2019（第五版）
- 高惠璇，《应用多元统计分析》，北京大学出版社，2014
- 《试验设计与数据分析》



中国科学技术大学
University of Science and Technology of China

谢 谢 !



中国科学技术大学
University of Science and Technology of China

数学建模 Mathematical Modeling

刘利刚

中国科学技术大学

CT成像模型

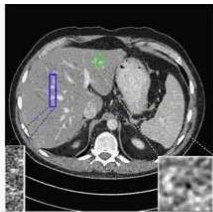
(课本第十一章)

多视点几何：测量物体表面信息

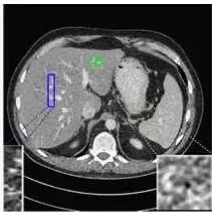


如何“看到”物体内部？

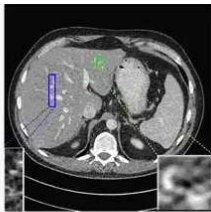




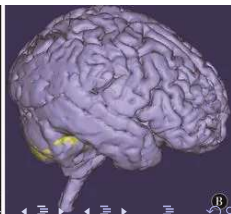
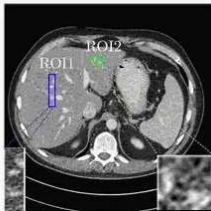
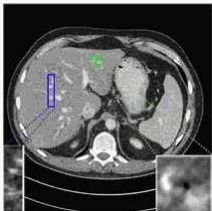
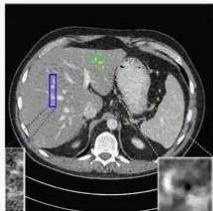
(a) LDCT



(b) BM3D



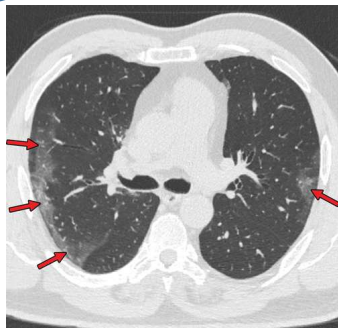
(c) RED-CNN



A simple question

Q: How to see inside the human body noninvasively?

Tomography
(断层/层析成像)



Computed Tomography (CT)

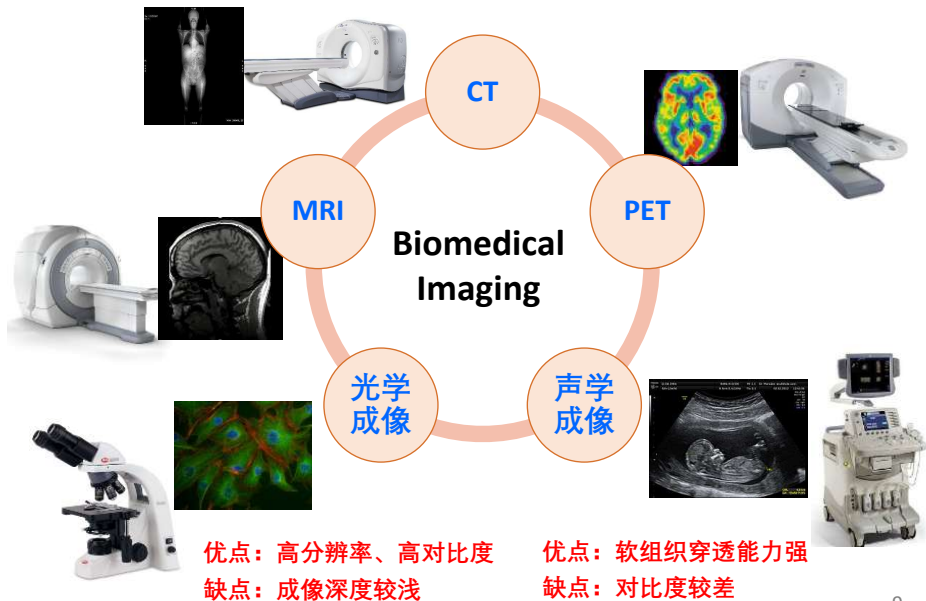
计算机体层摄影、计算机断层成像



CT: A. Cormack and N. Hounsfield

- 1972年，发明
- 1979年，获得了诺贝尔奖

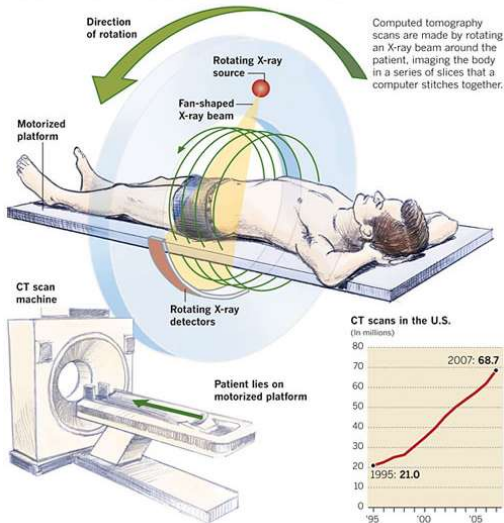
Medical imaging techniques



Principle

Anatomy of a CT scan

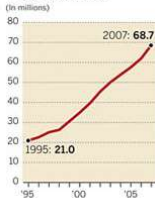
CT scanners give doctors a 3-D view of the body. The images are exquisitely detailed but require a dose of radiation that can be 100 times that of a standard X-ray.



Computed tomography scans are made by rotating an X-ray beam around the patient, imaging the body in a series of slices that a computer stitches together.

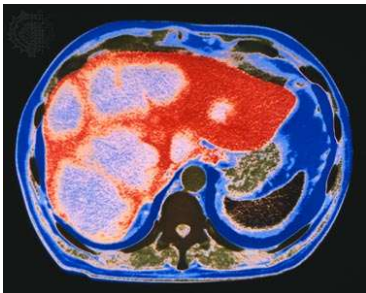


CT scans in the U.S.

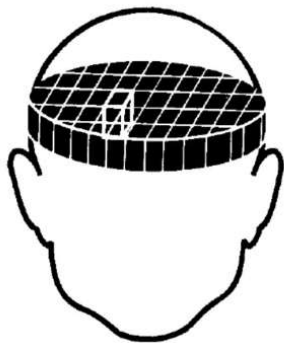
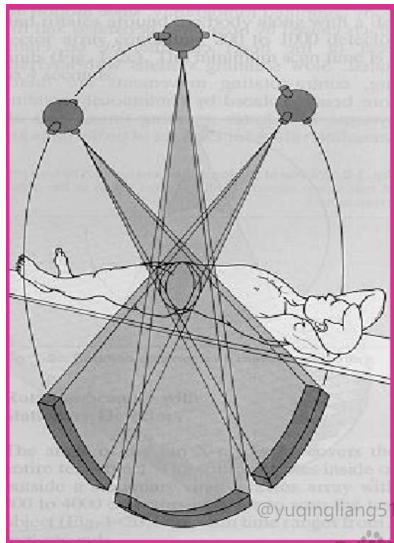


Solution

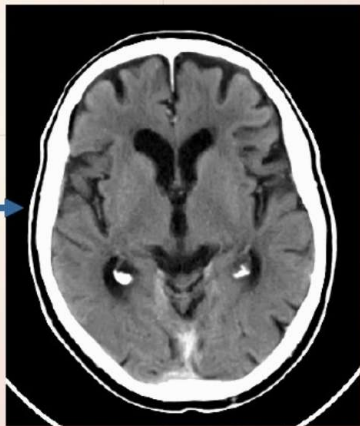
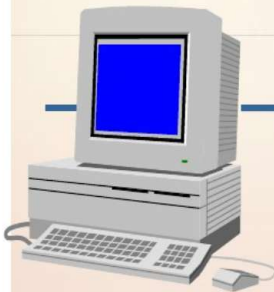
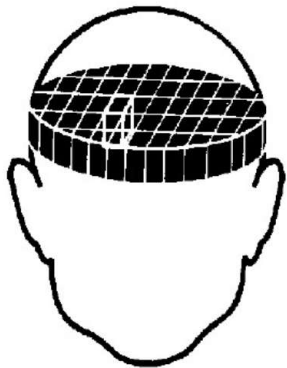
- Tomography
 - Reconstructs a function using line integrals
 - Goal: recover the interior structure of a body using exterior measurements
 - Routine for medicine, earth sciences



CT成像



CT成像



Necessary Mathematics

- Line integrals are integrals along a line
- Coordinate system: $(x,y) \rightarrow (\theta, t)$
- θ : Angle, t : distance along source
- Fourier Transform: $F(w) = \int f(t) e^{-j2\pi w t} dt$
- $F(u,v) = \iint f(x,y) e^{-j2\pi (ux + vy)} dx dy$

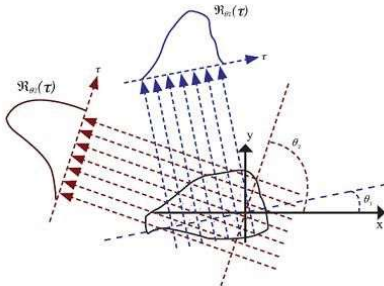


Image reconstruction through backprojection

Problem description: Reconstructing an image from a series of projections

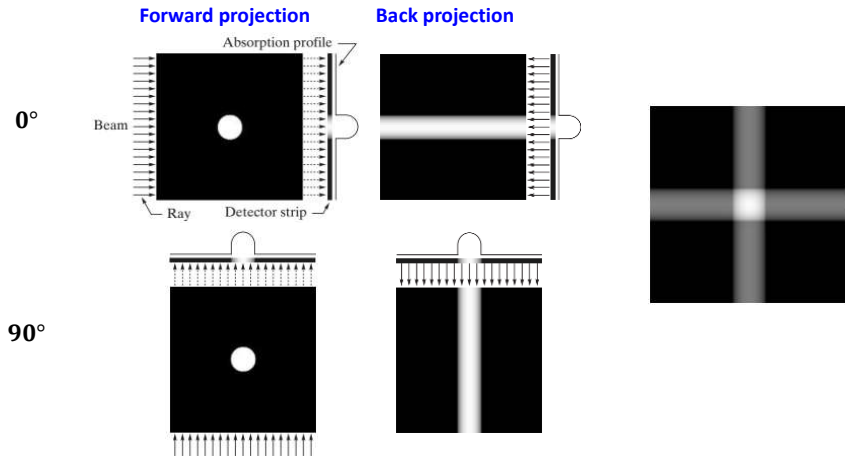
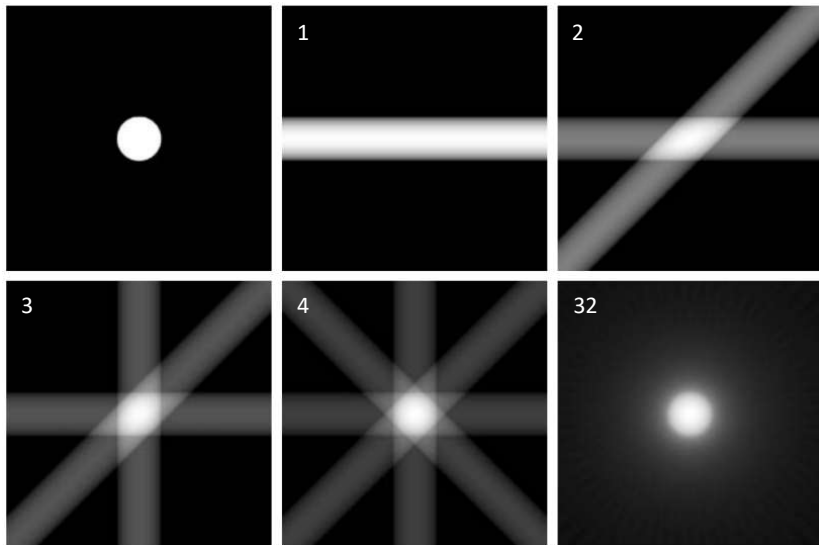


Image reconstruction through backprojection



1. The Radon Transform (雷登变换)

A. Forward projection

The normal form of a line

$$x \cos \theta + y \sin \theta = \rho$$

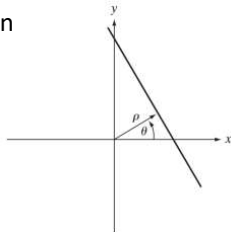
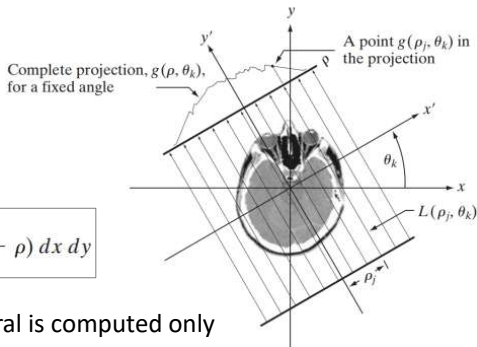
The line integral is the **Radon transform**

$$g(\rho, \theta) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \delta(x \cos \theta + y \sin \theta - \rho) dx dy$$

- The right side indicates that the integral is computed only along a line.
- The **Radon transform** is the cornerstone of reconstruction from projections.

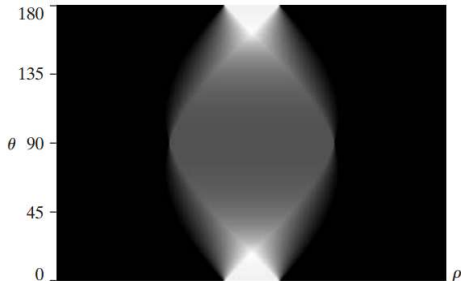
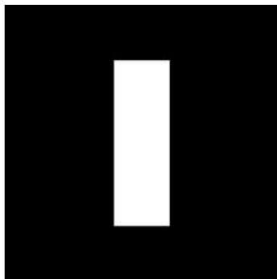
In the discrete case

$$g(\rho, \theta) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) \delta(x \cos \theta + y \sin \theta - \rho)$$

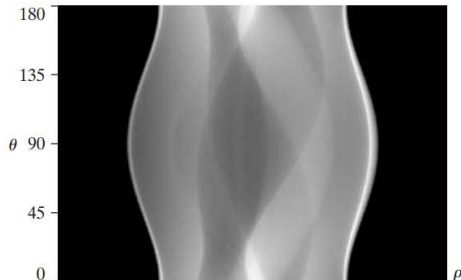


Example: Interpreting sinograms

**Rectangular
phantom**



**Shepp-Logan
phantom**



B. Back projection

The image from a **single** backprojection is

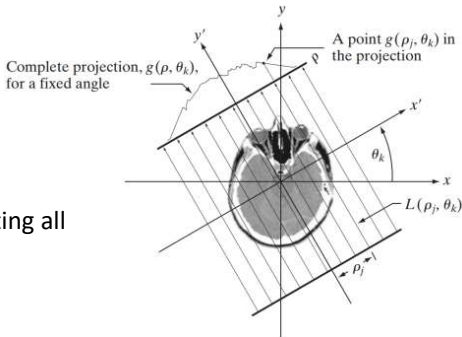
$$f_{\theta}(x, y) = g(x \cos \theta + y \sin \theta, \theta)$$

The final image can be formed by integrating all back-projected images

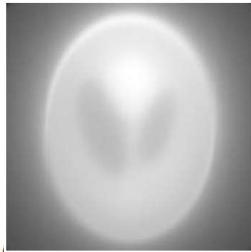
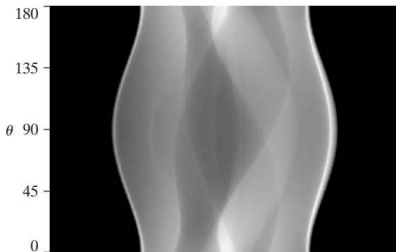
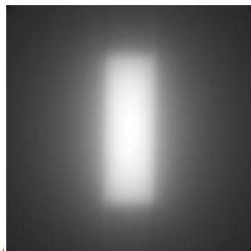
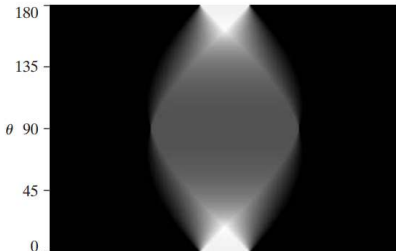
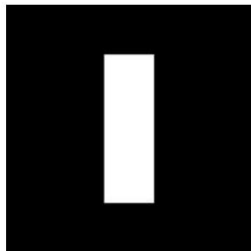
$$f(x, y) = \int_0^{\pi} f_{\theta}(x, y) d\theta$$

In the discrete case,

$$f(x, y) = \sum_{\theta=0}^{\pi} f_{\theta}(x, y)$$



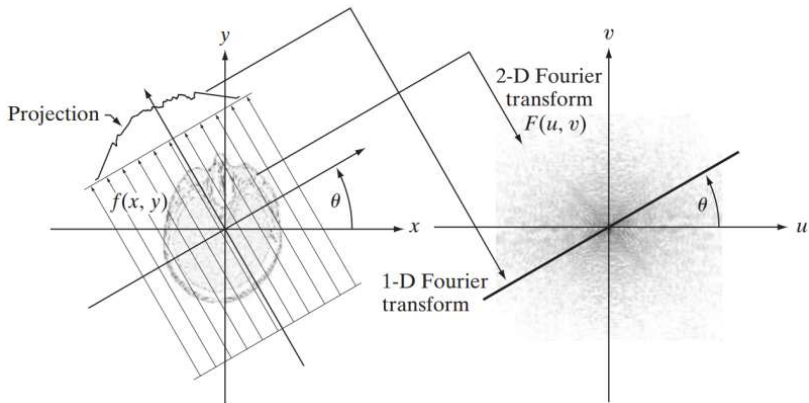
Examples: Back Projection



3. Image Reconstruction: Filtered Back Projections

A. The Fourier-slice theorem

$$\begin{aligned} G(\rho, \theta) &= [F(u, v)]_{u=\omega \cos \theta; v=\omega \sin \theta} \\ &= F(\omega \cos \theta, \omega \sin \theta) \end{aligned}$$



A. The Fourier-slice theorem

The 1-D Fourier transform of a projection with respect to ρ is

$$G(\omega, \theta) = \int_{-\infty}^{\infty} g(\rho, \theta) e^{-j2\pi\omega\rho} d\rho$$

$$g(\rho, \theta) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \delta(x \cos \theta + y \sin \theta - \rho) dx dy$$

$$\begin{aligned} G(\omega, \theta) &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \delta(x \cos \theta + y \sin \theta - \rho) e^{-j2\pi\omega\rho} dx dy d\rho \\ &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \left[\int_{-\infty}^{\infty} \delta(x \cos \theta + y \sin \theta - \rho) e^{-j2\pi\omega\rho} d\rho \right] dx dy \\ &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) e^{-j2\pi\omega(x \cos \theta + y \sin \theta)} dx dy \end{aligned}$$



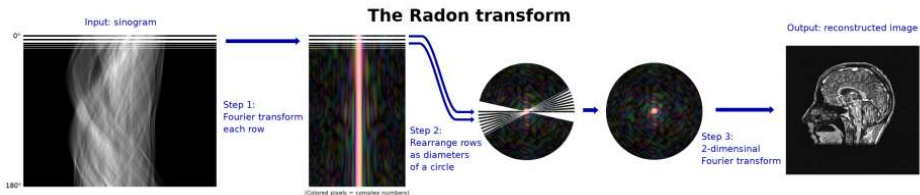
Let $u = \omega \cos \theta$ and $v = \omega \sin \theta$

$$G(\omega, \theta) = \left[\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) e^{-j2\pi(ux+vy)} dx dy \right]_{u=\omega \cos \theta; v=\omega \sin \theta}$$

$$\begin{aligned} G(\omega, \theta) &= [F(u, v)]_{u=\omega \cos \theta; v=\omega \sin \theta} \\ &= F(\omega \cos \theta, \omega \sin \theta) \end{aligned}$$

The 1-D Fourier transform of a projection is a slice of the 2-D Fourier transform of the object.

How to reconstruct an image?



https://en.wikipedia.org/wiki/Radon_transform

B. Filtered Back Projection (FBP)

The 2-D inverse Fourier transform of $F(u, v)$ is

$$f(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F(u, v) e^{j2\pi(ux+vy)} du dv$$



$$u = \omega \cos \theta, v = \omega \sin \theta \quad du dv = \omega d\omega d\theta$$

$$f(x, y) = \int_0^{2\pi} \int_0^{\infty} F(\omega \cos \theta, \omega \sin \theta) e^{j2\pi\omega(x \cos \theta + y \sin \theta)} \omega d\omega d\theta$$



The Fourier-slice theorem

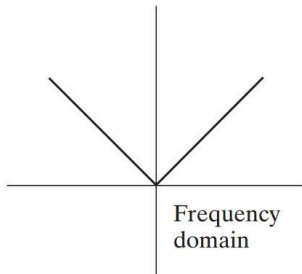
$$f(x, y) = \int_0^{2\pi} \int_0^{\infty} G(\omega, \theta) e^{j2\pi\omega(x \cos \theta + y \sin \theta)} \omega d\omega d\theta$$

$$G(\omega, \theta + 180^\circ) = G(-\omega, \theta)$$



$$f(x, y) = \int_0^{\pi} \int_{-\infty}^{\infty} |\omega| G(\omega, \theta) e^{j2\pi\omega(x \cos \theta + y \sin \theta)} d\omega d\theta$$

$$f(x, y) = \int_0^{\pi} \left[\int_{-\infty}^{\infty} |\omega| G(\omega, \theta) e^{j2\pi\omega\rho} d\omega \right]_{\rho=x \cos \theta + y \sin \theta} d\theta$$



- $|\omega|$ is a ramp filter (斜坡滤波器) .

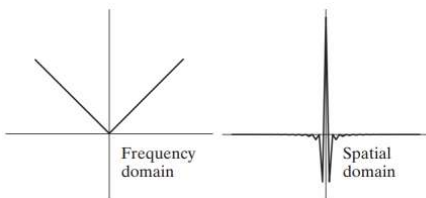
Filtering

Windowing with a **smooth** function helps decrease ringing.

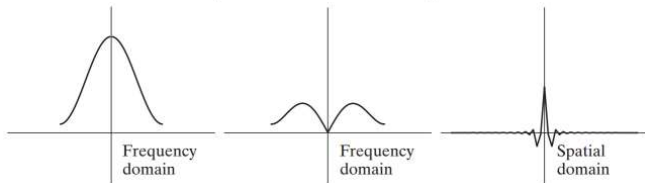
$$H(\omega) = \begin{cases} c + (c - 1) \cos \frac{2\pi\omega}{M - 1} & 0 \leq \omega \leq (M - 1) \\ 0 & \text{otherwise} \end{cases}$$

- $c = 0.54$, **Hamming window**;
- $c = 0.50$, **Hann window**.

Box filter



Hamming filter



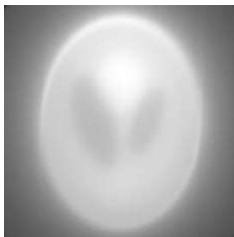
Example: Image reconstruction using FBP

Original

Direct
Backprojection

FBP (ramp)

FBP (Hamming)



Blurring present



**No blurring
Ringing present**



**No blurring
No ringing**

C. Iterative Reconstruction (IR)

Linear equations

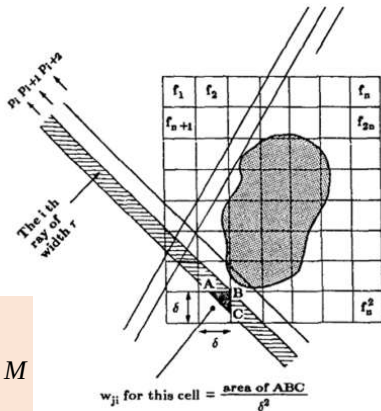
$$\omega_{11} f_1 + \omega_{12} f_2 + \omega_{13} f_3 + \cdots \omega_{1N} f_N = p_1$$

$$\omega_{21} f_1 + \omega_{22} f_2 + \omega_{23} f_3 + \cdots \omega_{2N} f_N = p_2$$

$\vdots$

$$\omega_{M1} f_1 + \omega_{M2} f_2 + \omega_{M3} f_3 + \cdots \omega_{MN} f_N = p_M$$

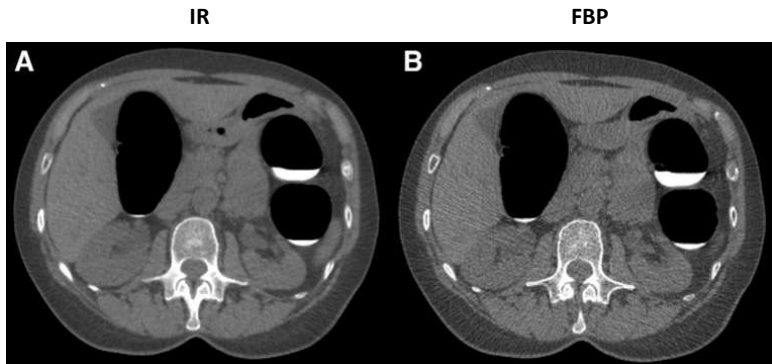
$$f^{(k+1)} = f^{(k)} + \alpha \frac{\left(p_i - \sum_{n=1}^N \omega_{in} f_n^{(k)} \right)}{\sum_{n=1}^N \omega_{in}^2} \omega_i, \quad i = 1, 2, \dots, M$$



- Algebraic reconstruction technique (ART)
- Simultaneous Iterative Reconstruction Technique (SIRT)
- Simultaneous Algebraic Reconstruction Technique (SART)
- Multiplicative algebraic reconstruction technique (MART)
-

FBP vs. IR

- FBP is much faster
- IR needs fewer projections than FBP and have improved image quality.



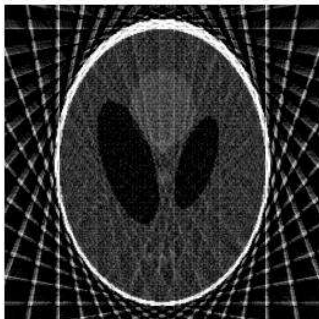
Geyer, Lucas L., et al. "State of the art: iterative CT reconstruction techniques." *Radiology* 276.2 (2015): 339-357.

Example

- The `imradon` function reconstructs images from projection data

Example

- Reconstruction with 36 projections



Example

- Reconstruction with 36 projections



Example

- Reconstruction with 90 projections



压缩感知

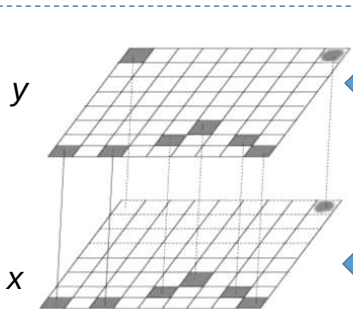
$$\begin{matrix} M \times 1 \\ \text{measurements} \end{matrix} y = \begin{matrix} \Phi \\ M \times N \end{matrix} x \quad M \ll N$$

- 已知 y 和 Φ ，有无穷多解 x
- 对于稀疏信号 x ，可通过优化能完全重建 x
 - 在一定条件下 (on Φ) [Candes and Tao 2005]

L_0 优化

$$\begin{aligned} \min & \|x\|_0 \\ \text{s.t. } & \Phi x = y \end{aligned}$$

感知采样



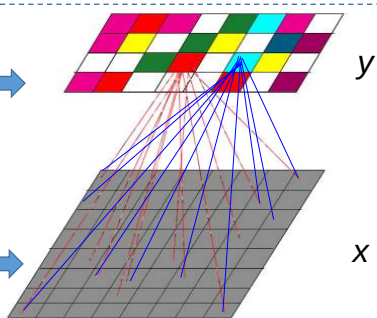
$$x \rightarrow y$$

传统采样：
采样**所有数据**

采样数据

VS.

原始数据

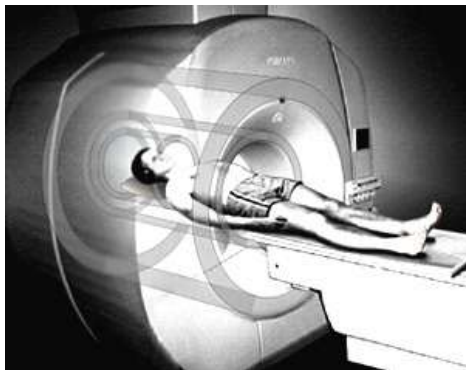


$$\begin{matrix} y \\ M \times 1 \\ \text{measurements} \end{matrix} = \begin{matrix} \Phi \\ M \times N \end{matrix} \begin{matrix} x \\ N \times 1 \end{matrix}$$

压缩感知采样：
采样**少量数据**

压缩感知方法的重大应用

- MR imaging time can be reduced (taking fewer frequency measurements)



(nobelprize.org)



中国科学技术大学
University of Science and Technology of China

谢谢！



中国科学技术大学
University of Science and Technology of China

数学建模 Mathematical Modeling

刘利刚

中国科学技术大学

多目标决策模型

例子

- 路线的选择
- 工作的选择
- 学校的选择
- 人才选拔
- 评优评奖
- ...

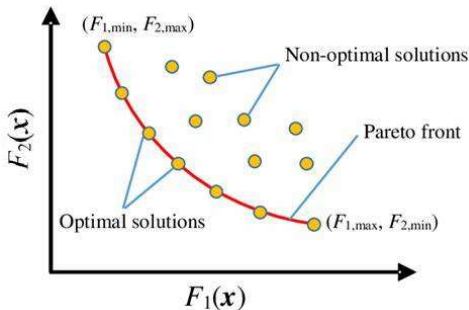
多目标决策问题

- 定义：具有多个目标的决策问题
- 特点：
 - 目标之间的不可公度性：多个目标之间没有一个统一的度量标准
 - 目标之间的矛盾性：某一目标的改善往往会损害其他目标的实现

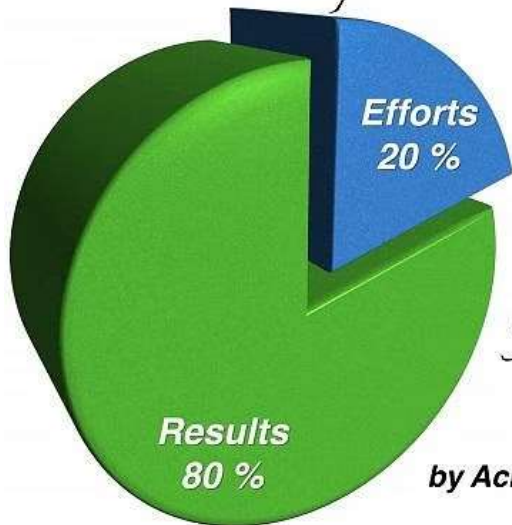
一般不存在通常意义下的最优解

多目标决策问题的解法

- 效用函数（价值函数）
- 非劣解：有效解、帕雷托(Pareto)解
- 求解方法
 - 加权法
 - 约束法

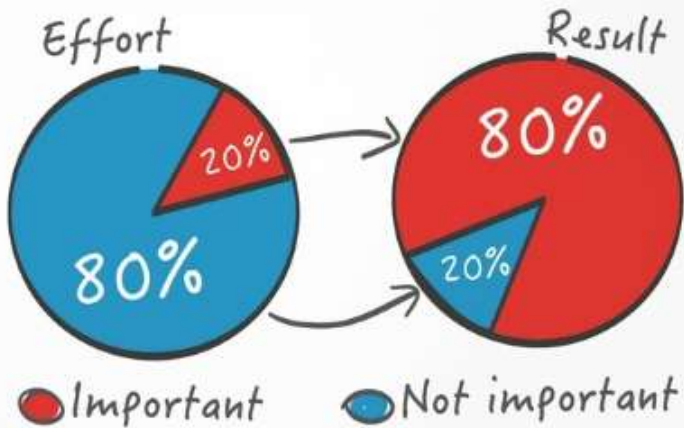


The Pareto Principle



by Vilfredo Pareto
1848-1923

**The Secret to Success
by Achieving More with Less**



多目标群组决策问题

- 群组决策问题：决策人有多个
- 数学模型

多目标群组决策问题的求解

- 两级优化

- 第一级优化：选择合适的方案使对每一个决策人都是非劣的，即不存在另一个方案使任何一个决策人的某一目标函数值得到改进，且又不损害其他的目标函数值
- 第二级优化：选择合适的方案对于决策群为帕雷托最优，即不存在另一方案对群中的某一个成员产生更大的作用，而不降低其他成员的效用

- 两级优化须同时求解

- 困难：要进行人与人之间的效用比较

多目标群组决策问题的求解2

- 子系统优化：首先将 n 个决策人的目标作为 n 个子系统，分别寻求决策方案使他们各自的目标达到最优，得到 n 个决策方案
- 协调优化：让总协调人按照某种优化原则来协调 n 个决策人之间的矛盾



中国科学技术大学
University of Science and Technology of China

谢谢！



中国科学技术大学
University of Science and Technology of China

数学建模 Mathematical Modeling

刘利刚

中国科学技术大学

层次分析法模型

层次分析模型

背景

- 日常工作、生活中的决策问题
- 涉及经济、社会等方面的因素
- 作比较判断时人的主观选择起相当大的作用，各因素的重要性难以量化
- Saaty于1970年代提出层次分析法
AHP (Analytic Hierarchy Process)
- AHP——一种**定性**与**定量**相结合的、**系统化**、**层次化**的分析方法

一. 层次分析法的基本步骤

例. 选择旅游地

如何在3个目的地中按照景色、费用、居住条件等因素选择.

目标层

O(选择旅游地)

准则层

C_1
景色

C_2
费用

C_3
居住

C_4
饮食

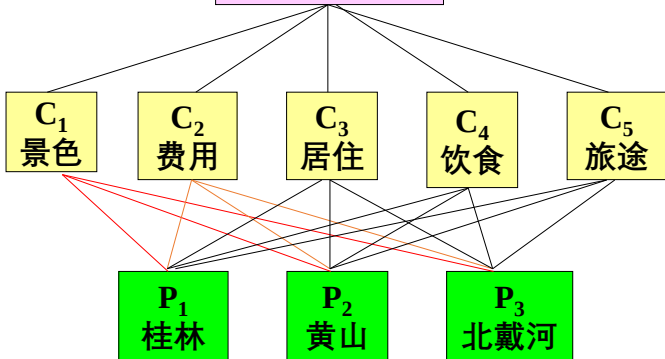
C_5
旅途

方案层

P_1
桂林

P_2
黄山

P_3
北戴河



“选择旅游地”思维过程的归纳

- 将决策问题分为3个层次：目标层O，准则层C，方案层P；每层有若干元素，各层元素间的关系用相连的直线表示。
- 通过相互比较确定各准则对目标的权重，及各方案对每一准则的权重。
- 将上述两组权重进行综合，确定各方案对目标的权重。

层次分析法将定性分析与定量分析结合起来完成以上步骤，给出决策问题的定量结果。

层次分析法的基本步骤

成对比较阵 和权向量

元素之间两两对比，对比采用相对尺度

设要比较各准则 $C_1, C_2, \dots, C_n$ 对目标 O 的重要性

$$C_i : C_j \Rightarrow a_{ij} \quad A = (a_{ij})_{n \times n}, a_{ij} > 0, a_{ji} = \frac{1}{a_{ij}}$$

选择
旅游地

$$A = \begin{bmatrix} 1 & 1/2 & 4 & 3 & 3 \\ 2 & 1 & 7 & 5 & 5 \\ 1/4 & 1/7 & 1 & 1/2 & 1/3 \\ 1/3 & 1/5 & 2 & 1 & 1 \\ 1/3 & 1/5 & 3 & 1 & 1 \end{bmatrix}$$

$A \sim$ 成对比较阵

A 是正互反阵

要由 A 确定 $C_1, \dots, C_n$ 对 O 的权向量

成对比较阵和权向量

成对比较的不一致情况

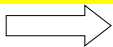
$$A = \begin{bmatrix} 1 & 1/2 & 4 & \dots \\ 2 & 1 & 7 & \dots \\ \dots & \dots & \dots & \dots \end{bmatrix}$$

不一致

$$a_{12} = 1/2 (C_1 : C_2)$$

一致比较

$$a_{13} = 4 (C_1 : C_3)$$



$$a_{23} = 8 (C_2 : C_3)$$

允许不一致，但要确定不一致的允许范围

考察完全一致的情况

$$W(=1) \Rightarrow w_1, w_2, \dots, w_n$$

$$\text{令 } a_{ij} = w_i / w_j$$

$$w = (w_1, w_2, \dots, w_n)^T \sim \text{权向量}$$

$$A = \begin{bmatrix} \frac{w_1}{w_1} & \frac{w_1}{w_2} & \dots & \frac{w_1}{w_n} \\ \frac{w_2}{w_1} & \frac{w_2}{w_2} & \dots & \frac{w_2}{w_n} \\ \dots & \dots & \dots & \dots \\ \frac{w_n}{w_1} & \frac{w_n}{w_2} & \dots & \frac{w_n}{w_n} \end{bmatrix}$$

成对比较阵和权向量

成对比较完全一致的情况

满足 $a_{ij} \cdot a_{jk} = a_{ik}$, $i, j, k = 1, 2, \dots, n$
的正互反阵 A 称**一致阵**，如

$$A = \begin{bmatrix} \frac{w_1}{w_1} & \frac{w_1}{w_2} & \dots & \frac{w_1}{w_n} \\ \frac{w_2}{w_1} & \frac{w_2}{w_2} & \dots & \frac{w_2}{w_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{w_n}{w_1} & \frac{w_n}{w_2} & \dots & \frac{w_n}{w_n} \end{bmatrix}$$

一致阵 性质

- A 的秩为 1, A 的唯一非零特征根为 n
- A 的任一行向量是对应于 n 的特征向量
- A 的归一化特征向量可作为权向量

对于不一致(但在允许范围内)的成对比较阵 A , 建议用对应于最大特征根 λ 的特征向量作为权向量 w , 即

$$Aw = \lambda w$$

成对比较阵和权向量

比较尺度 a_{ij}

Saaty等人提出1~9尺度—— a_{ij} 取值1,2,...,9及其互反数1,1/2,...,1/9

• 便于定性到定量的转化:

尺度 a_{ij}	1	2	3	4	5	6	7	8	9
$C_i : C_j$ 的重要性	相同		稍强		强		明显强		绝对强

$a_{ij} = 1, 1/2, \dots, 1/9 \sim C_i : C_j$ 的重要性与上面相反

• 心理学家认为成对比较的因素不宜超过9个

• 用1~3, 1~5, ..., 1~17, ..., $1^p \sim 9^p$ ($p=2,3,4,5$), $d+0.1 \sim d+0.9$ ($d=1,2,3,4$)等27种比较尺度对若干实例构造成对比较阵, 算出权向量, 与实际对比发现, 1~9尺度较优。

一致性检验

对A确定不一致的允许范围

已知： n 阶一致阵的唯一非零特征根为 n

可证： n 阶正互反阵最大特征根 $\lambda \geq n$ ，且 $\lambda = n$ 时为一致阵

定义一致性指标： $CI = \frac{\lambda - n}{n - 1}$ CI 越大，不一致越严重

为衡量 CI 的大小，引入**随机一致性指标 RI** ——随机模拟得到 a_{ij} ，形成 A ，计算 CI 即得 RI 。

Saaty的结果如下

n	1	2	3	4	5	6	7	8	9	10	11
RI	0	0	0.58	0.90	1.12	1.24	1.32	1.41	1.45	1.49	1.51

定义一致性比率 $CR = CI/RI$

当 $CR < 0.1$ 时，通过一致性检验

**“选择旅游地”中
准则层对目标的权
向量及一致性检验**

最大特征根 $\lambda=5.073$

准则层对目标的成对比较阵

$$A = \begin{bmatrix} 1 & 1/2 & 4 & 3 & 3 \\ 2 & 1 & 7 & 5 & 5 \\ 1/4 & 1/7 & 1 & 1/2 & 1/3 \\ 1/3 & 1/5 & 2 & 1 & 1 \\ 1/3 & 1/5 & 3 & 1 & 1 \end{bmatrix}$$

权向量(特征向量) $w=(0.263,0.475,0.055,0.090,0.110)^T$

$$\text{一致性指标 } CI = \frac{5.073 - 5}{5 - 1} = 0.018$$

随机一致性指标 $RI=1.12$ (查表)

一致性比率 $CR=0.018/1.12=0.016<0.1$

通过一致
性检验

组合权向量

记第2层（准则）对第1层（目标）

的权向量为 $w^{(2)} = (w_1^{(2)}, \dots, w_n^{(2)})^T$

同样求第3层(方案)对第2层每一元素(准则)的权向量

方案层对 C_1 (景色)
的成对比较阵

$$B_1 = \begin{bmatrix} 1 & 2 & 5 \\ 1/2 & 1 & 2 \\ 1/5 & 1/2 & 1 \end{bmatrix}$$

方案层对 C_2 (费用)
的成对比较阵

$$B_2 = \begin{bmatrix} 1 & 1/3 & 1/8 \\ 3 & 1 & 1/3 \\ 8 & 3 & 1 \end{bmatrix}$$

$\dots C_n$

$\dots B_n$

最大特征根 λ_1

λ_2

$\dots \lambda_n$

权向量

$w_1^{(3)}$

$w_2^{(3)}$

$\dots w_n^{(3)}$

组合权向量

第3层对第2层的计算结果

k	1	2	3	4	5	$w^{(2)}$
$w_k^{(3)}$	0.595	0.082	0.429	0.633	0.166	0.263
	0.277	0.236	0.429	0.193	0.166	0.475
	0.129	0.682	0.142	0.175	0.668	0.055
						0.090
λ_k	3.005	3.002	3	3.009	3	0.110
CI_k	0.003	0.001	0	0.005	0	

$RI=0.58$ ($n=3$), CI_k 均可通过一致性检验

方案 P_1 对目标的组合权重为 $0.595 \times 0.263 + \dots = 0.300$

方案层对目标的组合权向量为 $(0.300, 0.246, 0.456)^T$

组合 权向量

第2层对第1层的权向量

$$w^{(2)} = (w_1^{(2)}, \dots, w_n^{(2)})^T$$

第3层对第2层各元素的权向量

$$w_k^{(3)} = (w_{k1}^{(3)}, \dots, w_{km}^{(3)})^T, k = 1, 2, \dots, n$$

构造矩阵 $W^{(3)} = [w_1^{(3)}, \dots, w_n^{(3)}]$

则第3层对第1层的组合权向量 $w^{(3)} = W^{(3)} w^{(2)}$

第s层对第1层的组合权向量

$$w^{(s)} = W^{(s)} W^{(s-1)} \dots W^{(3)} w^{(2)}$$

第1层O

第2层 $C_1, \dots, C_n$

第3层 $P_1, \dots, P_m$

其中 $W^{(p)}$ 是由第p层对第p-1层权向量组成的矩阵

层次分析法的基本步骤

1) 建立层次分析结构模型

深入分析实际问题，将有关因素自上而下分层（目标—准则或指标—方案或对象），上层受下层影响，而层内各因素基本上相对独立。

2) 构造成对比较阵

用成对比较法和1~9尺度，构造各层对上一层每一因素的成对比较阵。

3) 计算权向量并作一致性检验

对每一成对比较阵计算最大特征根和特征向量，作一致性检验，若通过，则特征向量为权向量。

4) 计算组合权向量（作组合一致性检验*）

组合权向量可作为决策的定量依据。

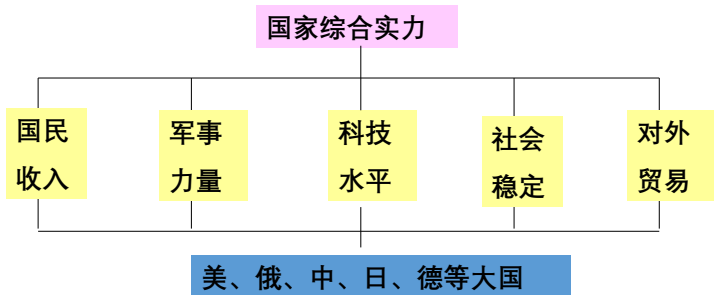
思考

- 残缺矩阵
- 多目标
- 群组决策

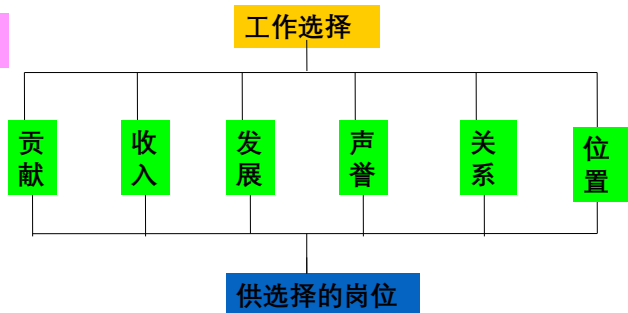
二. 层次分析法的广泛应用

- **应用领域：**经济计划和管理，能源政策和分配，人才选拔和评价，生产决策，交通运输，科研选题，产业结构，教育，医疗，环境，军事等。
- **处理问题类型：**决策、评价、分析、预测等。
- **建立层次分析结构模型是关键一步，要有主要决策层参与。**
- **构造成对比较阵是数量依据，应由经验丰富、判断力强的专家给出。**

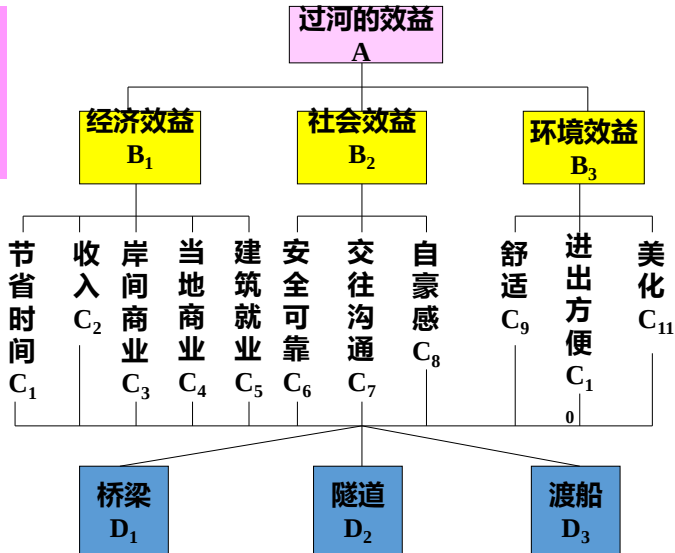
例1 国家实力分析



例2 工作选择

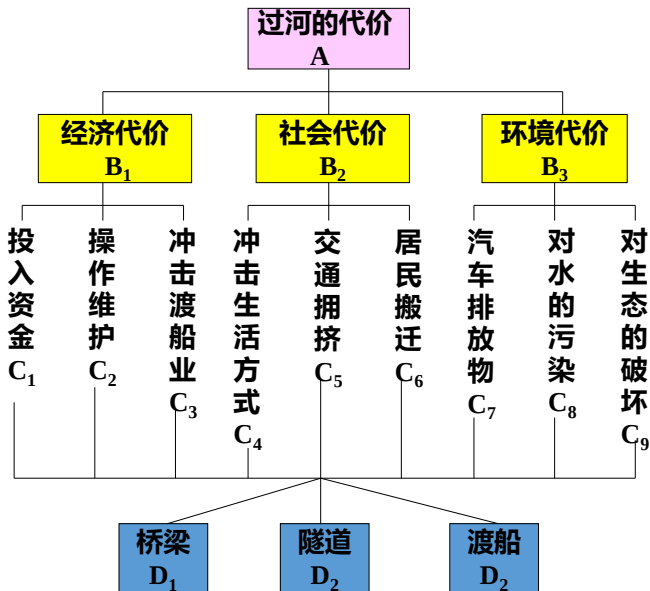


例3 横渡江河、海峡方案的抉择



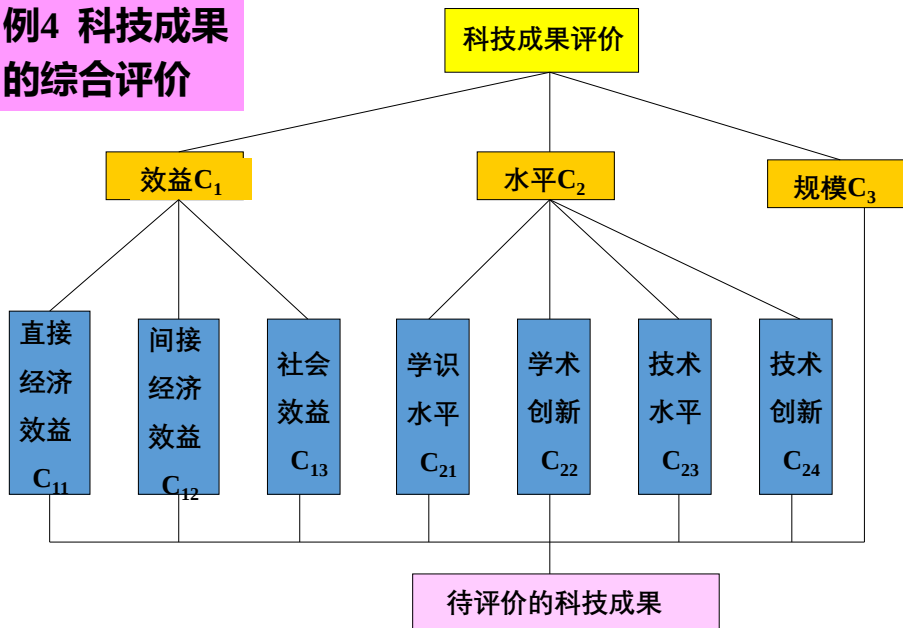
(1) 过河效益层次结构

例3 横渡江河、海峡方案的抉择



(2) 过河代价层次结构

例4 科技成果 的综合评价



三. 层次分析法的若干问题

- 正互反阵的最大特征根是否为正数？特征向量是否为正向量？一致性指标能否反映正互反阵接近一致阵的程度？
- 怎样简化计算正互反阵的最大特征根和特征向量？
- 为什么用特征向量作为权向量？
- 当层次结构不完全或成对比较阵有空缺时怎样用层次分析法？

1. 正互反阵的最大特征根和特征向量的性质

定理1 正矩阵 A 的最大特征根 λ 是正单根，对应正特征向量 w ，且 $\lim_{k \rightarrow \infty} \frac{A^k e}{e^T A^k e} = w$, $e = (1, 1, \dots, 1)^T$

 正互反阵的最大特征根是正数，特征向量是正向量。

定理2 n 阶正互反阵 A 的最大特征根 $\lambda = n$ ， $\lambda = n$ 是 A 为一致阵的充要条件。

 一致性指标 $CI = \frac{\lambda - n}{n - 1}$ 定义合理

2. 正互反阵最大特征根和特征向量的简化计算

- 精确计算的复杂和不必要
- 简化计算的思路——一致阵的任一列向量都是特征向量，一致性尚好的正互反阵的列向量都应近似特征向量，可取其某种意义下的平均。

和法——取列向量的算术平均

例 $A = \begin{bmatrix} 1 & 2 & 6 \\ 1/2 & 1 & 4 \\ 1/6 & 1/4 & 1 \end{bmatrix}$ $\xrightarrow{\text{列向量归一化}}$ $\begin{bmatrix} 0.6 & 0.615 & 0.545 \\ 0.3 & 0.308 & 0.364 \\ 0.1 & 0.077 & 0.091 \end{bmatrix}$ $\xrightarrow{\text{算术平均}}$ $\begin{bmatrix} 0.587 \\ 0.324 \\ 0.089 \end{bmatrix} = w$

$$Aw = \begin{bmatrix} 1.769 \\ 0.974 \\ 0.286 \end{bmatrix} \quad Aw = \lambda w \quad \Rightarrow \quad \lambda = \frac{1}{3} \left(\frac{1.769}{0.587} + \frac{0.974}{0.324} + \frac{0.268}{0.089} \right) = 3.009$$

精确结果: $w = (0.588, 0.322, 0.090)^T$, $\lambda = 3.010$

简化 计算

根法——取列向量的几何平均

幂法——迭代算法

1) 任取初始向量 $w^{(0)}$, $k:=0$, 设置精度 ε

2) 计算 $\tilde{w}^{(k+1)} = Aw^{(k)}$

3) 归一化 $w^{(k+1)} = \tilde{w}^{(k+1)} / \sum_{i=1}^n \tilde{w}_i^{(k+1)}$

4) 若 $\max_i |w_i^{(k+1)} - w_i^{(k)}| < \varepsilon$, 停止;

否则, $k:=k+1$, 转2

5) 计算 $\lambda = \frac{1}{n} \sum_{i=1}^n \frac{\tilde{w}_i^{(k+1)}}{w_i^{(k)}}$

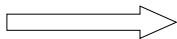
3. 特征向量作为权向量——成对比较的多步累积效应

问题

一致阵 A , 权向量 $w=(w_1, \dots, w_n)^T$, $a_{ij}=w_i/w_j$

A 不一致, 应选权向量 w 使 w_i/w_j 与 a_{ij} 相差尽量小 (对所有 i, j)。

用拟合方法确定 w



$$\min_{w_i (i=1, \dots, n)} \sum_{i=1}^n \sum_{j=1}^n \left(a_{ij} - \frac{w_i}{w_j} \right)^2$$

非线性
最小二乘

线性化——
对数最小二乘

$$\min_{w_i (i=1, \dots, n)} \sum_{i=1}^n \sum_{j=1}^n \left(\ln a_{ij} - \ln \frac{w_i}{w_j} \right)^2$$

结果与根法相同

多步累积效应

• 按不同准则确定的权向量不同，特征向量有什么优点。

成对比较

$C_i:C_j$ (直接比较)

$a_{ij} \sim 1$ 步强度

$$A^2 = (a_{ij}^{(2)}) \quad a_{ij}^{(2)} = \sum_{s=1}^n a_{is} a_{sj}$$

$a_{ij}^{(2)} \sim 2$ 步强度

$a_{is} a_{sj} \sim C_i$ 通过 C_s 与 C_j 的比较

更能反映 C_i 对 C_j 的强度

$$A^k = (a_{ij}^{(k)}), \quad a_{ij}^{(k)} \sim k \text{ 步强度}$$

体现多步累积效应

$$\forall i, j, \exists k_0, k > k_0, a_{is}^{(k)} \geq a_{js}^{(k)} \text{ 或 } a_{is}^{(k)} \leq a_{js}^{(k)} (s = 1, \dots, n)$$

$\Rightarrow$ 当 k 足够大, A^k 第 i 行元素反映 C_i 的权重 $\Rightarrow$ 求 A^k 的行和

定理1 $\lim_{k \rightarrow \infty} \frac{A^k e}{e^T A^k e} = w$

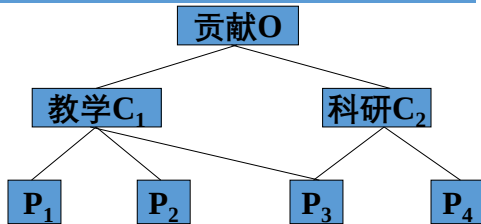
特征向量体现多步累积效应

4.不完全层次结构中组合权向量的计算

完全层次结构：上层每一元素与下层所有元素相关联

不完全层次结构

例：评价教师贡献的层次结构



设第2层对第1层权向量

$w^{(2)}=(w_1^{(2)}, w_2^{(2)})^T$ 已定

第3层对第2层权向量

$w_1^{(3)}=(w_{11}^{(3)}, w_{12}^{(3)}, w_{13}^{(3)}, 0)^T$

$w_2^{(3)}=(0, 0, w_{23}^{(3)}, w_{24}^{(3)})^T$ 已得

讨论由 $w^{(2)}, W^{(3)}=(w_1^{(3)}, w_2^{(3)})$

计算第3层对第1层权向量

$w^{(3)}$ 的方法

P_1, P_2 只作教学, P_4 只作科研,
 P_3 兼作教学、科研。

C_1, C_2 支配元素的数目不等

考察一个特例：

若 C_1, C_2 重要性相同, $w^{(2)} = (1/2, 1/2)^T$,

$P_1 \sim P_4$ 能力相同, $w_1^{(3)} = (1/3, 1/3, 1/3, 0)^T, w_2^{(3)} = (0, 0, 1/2, 1/2)^T$

公正的评价应为： $P_1:P_2:P_3:P_4=1:1:2:1$

• 不考虑支配元素数目不等的影响

仍用 $w^{(3)} = W^{(3)} w^{(2)}$ 计算



$$w^{(3)} = (1/6, 1/6, 5/12, 1/4)^T$$

• 支配元素越多权重越大

教学、科研任务由上级安排

用支配元素数目 n_1, n_2 对 $w^{(2)}$ 加权修正

$$n_1 = 3, n_2 = 2,$$

$$\tilde{w}^{(2)} = (n_1 w_1^{(2)}, n_2 w_2^{(2)})^T / (n_1 w_1^{(2)} + n_2 w_2^{(2)})$$

$$\tilde{w}^{(2)} = (3/5, 2/5)^T$$

再用 $w^{(3)} = W^{(3)} \tilde{w}^{(2)}$ 计算



$$w^{(3)} = (1/5, 1/5, 2/5, 1/5)^T$$

• 支配元素越多权重越小

教学、科研靠个人积极性

5. 残缺成对比较阵的处理

例 $A = \begin{bmatrix} 1 & 2 & \theta \\ 1/2 & 1 & 2 \\ \theta & 1/2 & 1 \end{bmatrix} \xrightarrow{\text{辅助矩阵}} C = \begin{bmatrix} 1 & 2 & w_1/w_3 \\ 1/2 & 1 & 2 \\ w_3/w_1 & 1/2 & 1 \end{bmatrix}$

θ 为残缺元素

$$Cw = \lambda w \Rightarrow \lambda = 3, w = (0.5714, 0.2857, 0.1429)^T$$



$$\bar{A}w = \lambda w$$

$$\bar{A} = \begin{bmatrix} 2 & 2 & 0 \\ 1/2 & 1 & 2 \\ 0 & 1/2 & 2 \end{bmatrix}$$

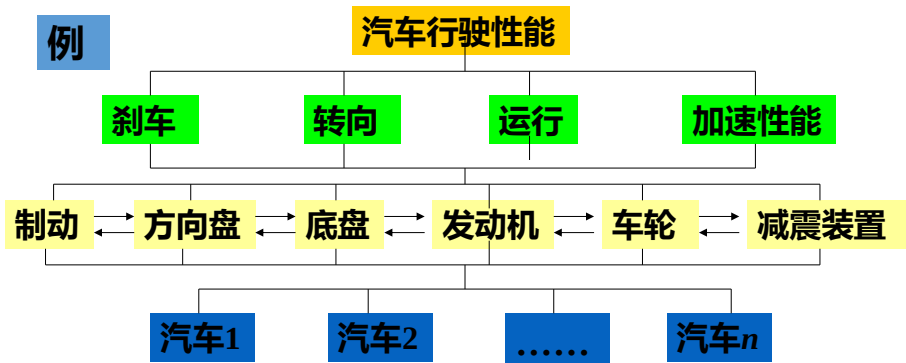
$$\bar{a}_{ij} = \begin{cases} a_{ij}, & i \neq j, a_{ij} \neq \theta \\ 0, & i \neq j, a_{ij} = \theta \\ m_i + 1, & i = j \end{cases}$$

$m_i \sim A$ 第*i*行
中 θ 的个数

6. 更复杂的层次结构

- 递阶层次结构：层内各元素独立，无相互影响和支配；层间自上而下、逐层传递，无反馈和循环。
- 更复杂的层次结构：层内各元素间存在相互影响或支配；层间存在反馈或循环。

例



层次分析法的优点

- **系统性**——将对象视作系统，按照分解、比较、判断、综合的思维方式进行决策——系统分析（与机理分析、测试分析并列）；
- **实用性**——定性与定量相结合，能处理传统的优化方法不能解决的问题；
- **简洁性**——计算简便，结果明确，便于决策者直接了解和掌握。

层次分析法的局限

- **囿旧**——只能从原方案中选优，不能产生新方案；
- **粗略**——定性化为定量，结果粗糙；
- **主观**——主观因素作用大，结果可能难以服人。



中国科学技术大学
University of Science and Technology of China

谢谢！



中国科学技术大学

University of Science and Technology of China

数学建模

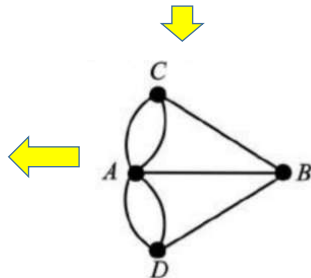
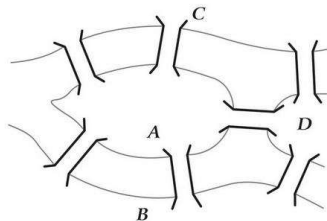
Mathematical Modeling

刘利刚

中国科学技术大学

图与网络模型

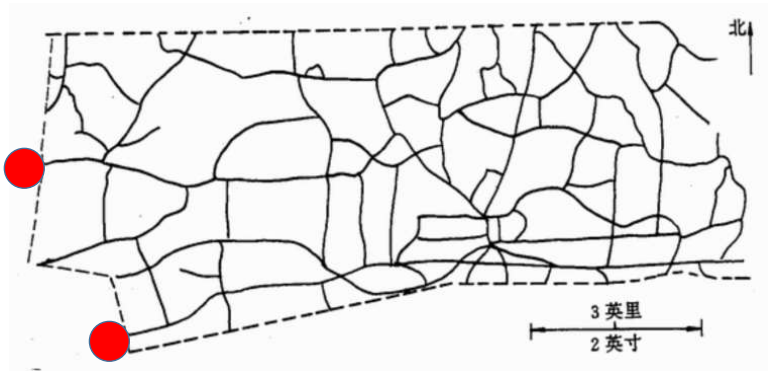
欧拉七桥问题



问题1：扫雪问题

课本第十三章（MCM1990B）

- 安排两辆扫雪车分别从指定位置(红点处)出发清除路面上的积雪。给出清扫积雪的最佳方案。



【模型建立】

- 假设扫雪车扫雪时的行驶速度 u ，不扫雪时的行驶速度 $v \gg u$ 。
- 假设扫雪车可在州高速公路上快速行驶，无需扫雪，速度 $w > v$ 。
- 以道路交叉口为顶点、县道为边建立无向图 $G = (V, E)$ 。
- 所有位于高速公路上的顶点合并为1个顶点。
- 任意两个相邻顶点之间恰有2条边。
- 设每条边 $e \in E$ 的长度为 $L(e)$ 。

【数学问题】

- 求 G 上两条指定起点的路 P_1, P_2 ，使得 $P_1 \cup P_2 \supset E$ ，并且

$$T = \max \left(\sum_{e \in P_1} L(e), \sum_{e \in P_2} L(e) \right) \text{ 尽可能小。}$$

在一个有向图上找到使行驶路程最短的最佳方案

【问题求解】

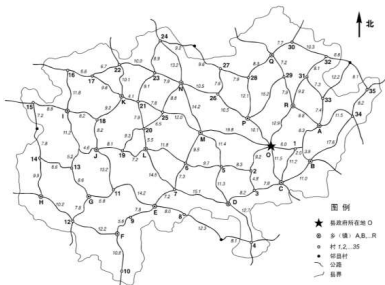
- 把 G 分解成一些两两无公共边的圈。
- 把这些圈分成长度之和近似相等的两部分，每部分是 G 的连通子图。
- 若某部分不连通，可尝试枚举调整1个圈、2个圈、...
- 把每部分的圈拼接成一个回路。

问题2：灾情巡视路线

课本第十三章思考题1（CUMCM1998B）

下图为某县的乡镇村公路网示意图，公路边的数字为该路段的公里数。今年夏天该县遭受水灾。为考察灾情、组织自救，县领导决定，带领有关部门负责人到全县各乡镇村巡视。巡视路线指从县政府所在地出发，走遍各乡镇村，又回到县政府所在地的路线。

1. 若分三组（路）巡视，试设计总路程最短且各组尽可能均衡的巡视路线。
2. 假定巡视人员在各乡（镇）停留时间 $T = 2$ 小时，在各村停留时间 $t = 1$ 小时，汽车行驶速度 $V = 35$ 公里/小时。要在24小时内完成巡视，至少应分几组；给出这种分组下你认为最佳的巡视路线。
3. 在上述关于 T, t 和 V 的假定下，如果巡视人员足够多，完成巡视的最短时间是多少；给出在这种最短时间完成巡视的要求下，你认为最佳的巡视路线。
4. 若巡视组数已定（比如三组），要求尽快完成巡视，讨论 T, t 和 V 改变对最佳巡视路线的影响。



【模型建立】

- 以乡(镇)、村为顶点，连接相邻乡(镇)、村的道路为边建立无向图 $G = (V, E)$ 。
- 计算任意两个顶点 u, v 之间的最短路长度 $d(u, v)$ 。
- 每组巡视线路是图 G 上从顶点 O 出发的回路。

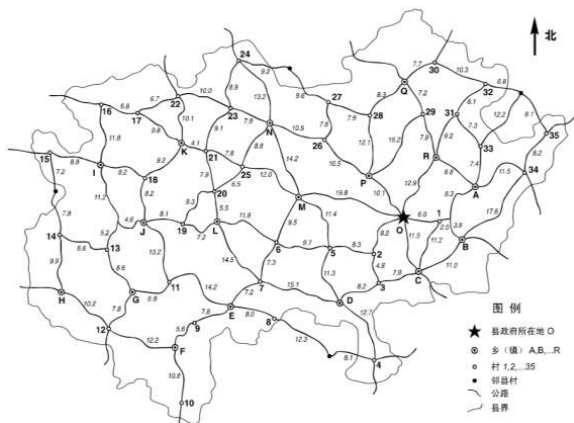
【思路】

- 把 $V \setminus \{O\}$ 中所有顶点分成互不相交的三组 X_1, X_2, X_3 ，并且最小化 $f(X_1) + f(X_2) + f(X_3)$ ，
其中 $f(X_i)$ 是经过 $X_i \cup \{O\}$ 中所有顶点的最短回路长度。
- 使用下述算法求以上问题的局部极小值解。
- 初始时， $X_1 = X_2 = X_3 = \emptyset$ ， $S = V - \{O\}$ 。
- 每次把某个 $s \in S$ 移至某个 X_i ，使得 $f(X_i \cup \{s\})$ 最小，直至 $S = \emptyset$ 。
- 设 $f(X_i) < f(X_j)$ ，尝试把 X_j 中的点移至 X_i 。

问题2与问题1的目标函数不同，其它相同

什么是图/网络?

- 离散点集及其相互关联的“抽象”
- 学科：图论
- 例子：
 - 地图
 - 互联网/物联网
 - 人际网络
 - 社交网络
 - 论文引用
 - ...



图论简介

图的定义

- 图是由顶点集合(vertex)及顶点间的关系集合组成的一种数据结构:

$$\text{Graph} = (V, E)$$

其中 $V = \{x \mid x \in \text{某个数据对象}\}$
是顶点的有穷非空集合;

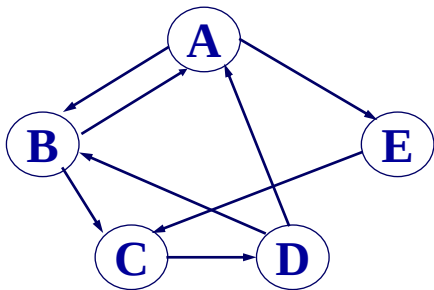
$E = \{(x, y) \mid x, y \in V\}$
是顶点之间关系的有穷集合, 也叫做边(edge)集合。

有向图与无向图

- 在有向图中，顶点对 $\langle x, y \rangle$ 是有序的。
- 在无向图中，顶点对 (x, y) 是无序的。
 - 若 $\langle v, w \rangle \in E$ 必有 $\langle w, v \rangle \in E$

有向图的例子

$$G_1 = (V_1, E_1)$$



其中

$$V_1 = \{A, B, C, D, E\}$$

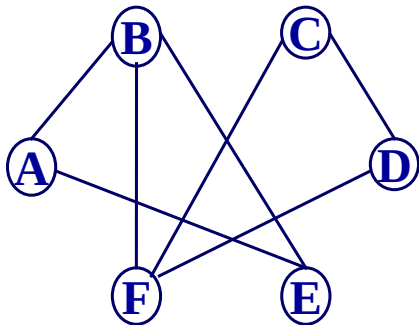
$$E_1 = \{ \langle A, B \rangle, \langle B, A \rangle, \langle A, E \rangle, \langle B, C \rangle, \langle C, D \rangle, \langle D, B \rangle, \langle D, A \rangle, \langle E, C \rangle \}$$

无向图的例子

$$G_2=(V_2,E_2)$$

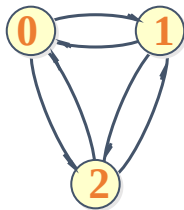
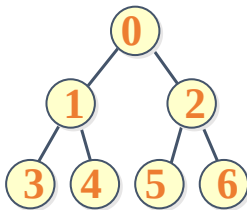
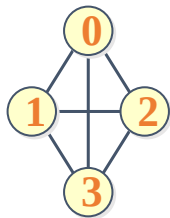
$$V_2=\{A, B, C, D, E, F\}$$

$$E_2=\{(A, B), (A, E), \\ (B, E), (C, D), (D, F), \\ (B, F), (C, F) \}$$

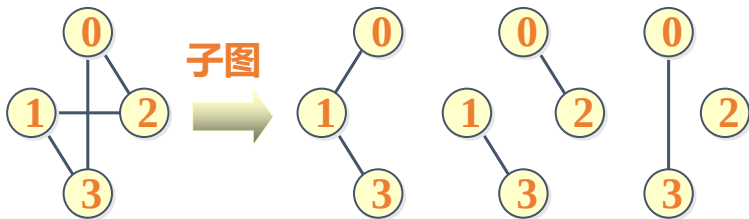


完全图

- 若有 n 个顶点的无向图有 $n(n-1)/2$ 条边, 则此图为完全无向图。有 n 个顶点的有向图有 $n(n-1)$ 条边, 则此图为完全有向图。

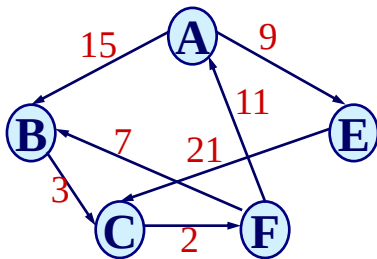


- 邻接顶点 如果 (u, v) 是 $E(G)$ 中的一条边, 则称 u 与 v 互为邻接顶点。
- 子图 设有两个图 $G = (V, E)$ 和 $G' = (V', E')$ 。若 $V' \subseteq V$ 且 $E' \subseteq E$, 则称 图 G' 是 图 G 的子图。



网络

- 某些图的边具有与它相关的数, 称之为权。这种带权图叫做网络。

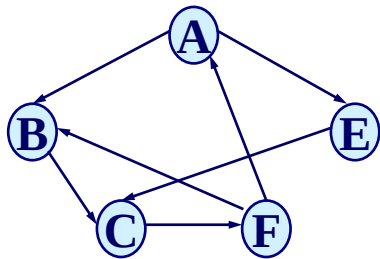


顶点的度

- 顶点 v 的入度是以 v 为终点的有向边的条数, 记作 $ID(v)$
- 顶点 v 的出度是以 v 为始点的有向边的条数, 记作 $OD(v)$
- 顶点的度 一个顶点 v 的度是与它相关联的边的条数, 记作 $TD(v)$ 。
 - 在有向图中, 顶点的度等于该顶点的入度与出度之和:
 $TD(v) = ID(v) + OD(v)$

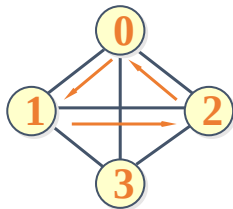
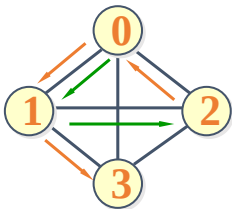
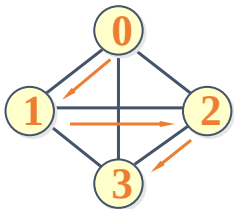
路径

- 在图 $G = (V, E)$ 中, 若从顶点 v_i 出发, 沿一些边经过一些顶点 $v_{p1}, v_{p2}, \dots, v_{pm}$, 到达顶点 v_j 。则称顶点序列 $(v_i, v_{p1}, v_{p2}, \dots, v_{pm}, v_j)$ 为从顶点 v_i 到顶点 v_j 的路径。它经过的边 (v_i, v_{p1}) 、 (v_{p1}, v_{p2}) 、 $\dots$ 、 (v_{pm}, v_j) 应是属于 E 的边。



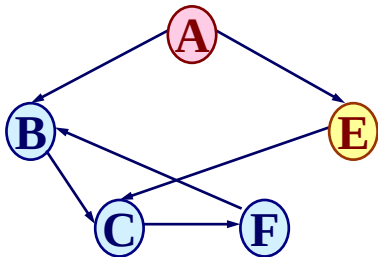
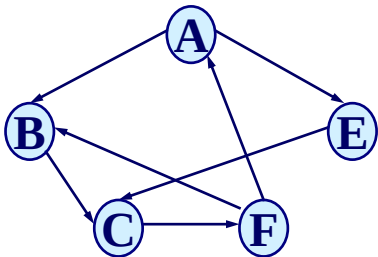
如:从A到F长度为 3
的路径{A,B,C,F}

- **路径长度** 非带权图的路径长度是指此路径上边的条数。带权图的路径长度是指路径上各边的权之和。
- **简单路径** 若路径上各顶点 $v_1, v_2, \dots, v_m$ 均不互相重复, 则称这样的路径为简单路径。
- **回路** 若路径上第一个顶点 v_1 与最后一个顶点 v_m 重合, 则称这样的路径为回路或环。



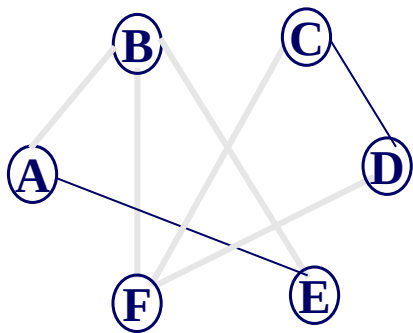
强连通图

- 强连通图** 在有向图中, 若对于每一对顶点 v_i 和 v_j , 都存在一条从 v_i 到 v_j 和从 v_j 到 v_i 的路径, 则称此图是强连通图。
- 强连通分量** 非强连通图的极大强连通子图叫做强连通分量。



生成树

- **生成树** 一个连通图的生成树是其极小连通子图，在 n 个顶点的情形下，有 $n-1$ 条边。



对非连通图，则称由各个连通分量的生成树的集合为此非连通图的**生成森林**。

图的基本操作

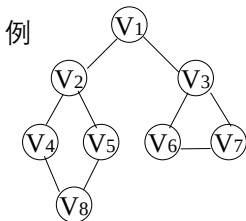
- 图结构的建立和销毁
- 对顶点的访问操作
- 插入或删除顶点
- 插入和删除弧
- 对邻接点的操作
- 遍历

图的抽象数据类型

```
class Graph
{
public:
    Graph ( );
    void InsertVertex ( Type & vertex );
    void InsertEdge ( int v1, int v2, int weight );
    void RemoveVertex ( int v );
    void RemoveEdge ( int v1, int v2 );
    int IsEmpty ( );
    Type GetWeight ( int v1, int v2 );
    int GetFirstNeighbor ( int v );
    int GetNextNeighbor ( int v1, int v2 );
}
```

图的遍历(Graph Traversal)

- 从图中某个顶点出发游历图，访遍图中其余顶点，并且使图中的每个顶点仅被访问一次的过程



遍历1: $V1 \Rightarrow V2 \Rightarrow V4 \Rightarrow V8 \Rightarrow V5 \Rightarrow V3 \Rightarrow V6 \Rightarrow V7$

遍历2: $V1 \Rightarrow V2 \Rightarrow V3 \Rightarrow V4 \Rightarrow V5 \Rightarrow V6 \Rightarrow V7 \Rightarrow V8$

图的遍历分类

- ◆ 深度优先搜索

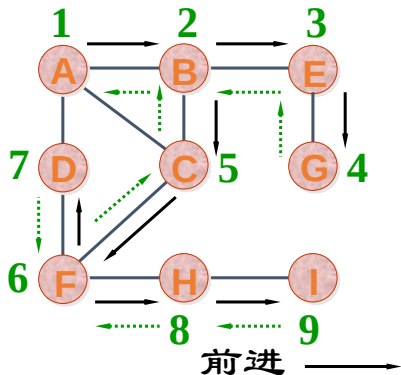
DFS (Depth First Search)

- ◆ 广度优先搜索

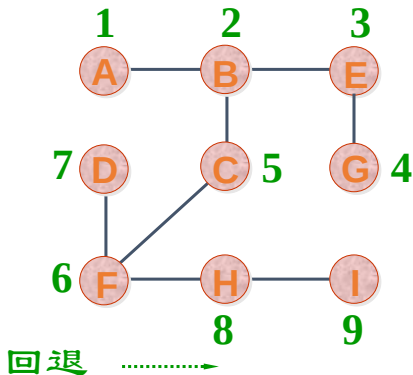
BFS (Breadth First Search)

深度优先搜索DFS

深度优先搜索的示例



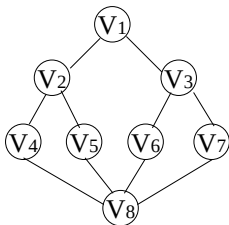
深度优先搜索过程



深度优先生成树

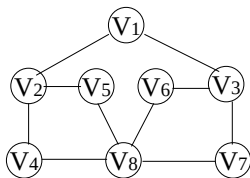
- **DFS** 在访问图中某一起始顶点 v 后, 由 v 出发, 访问它的任一邻接顶点 w_1 ; 再从 w_1 出发, 访问与 w_1 邻接但还没有访问过的顶点 w_2 ; 然后再从 w_2 出发, 进行类似的访问, ... 如此进行下去, 直至到达所有的邻接顶点都被访问过的顶点 v 为止。接着, 退回一步, 退到前一次刚访问过的顶点, 看是否还有其它没有被访问的邻接顶点。如果有, 则访问此顶点, 之后再从此顶点出发, 进行与前述类似的访问; 如果没有, 就再退回一步进行搜索。重复上述过程, 直到连通图中所有顶点都被访问过为止。

例子



深度遍历: $V1 \Rightarrow V2 \Rightarrow V4 \Rightarrow V8 \Rightarrow V5 \Rightarrow V6 \Rightarrow V3 \Rightarrow V7$

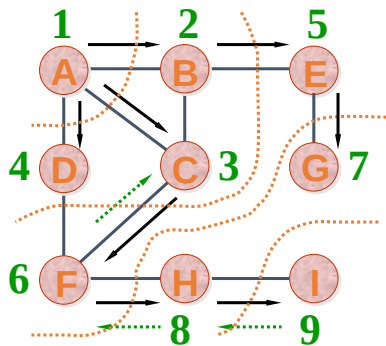
例



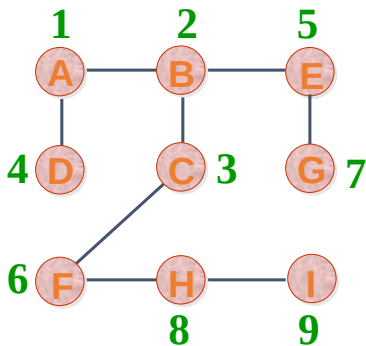
深度遍历: $V1 \Rightarrow V2 \Rightarrow V4 \Rightarrow V8 \Rightarrow V5 \Rightarrow V6 \Rightarrow V3 \Rightarrow V7$

广度优先搜索BFS

广度优先搜索的示例



广度优先搜索过程

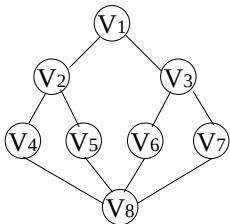


广度优先生成树

算法描述

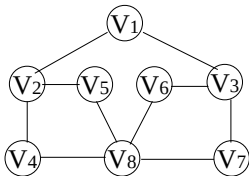
- **BFS**在访问了起始顶点 v 之后, 由 v 出发, 依次访问 v 的各个未被访问过的邻接顶点 $w_1, w_2, \dots, w_t$, 然后再顺序访问 $w_1, w_2, \dots, w_t$ 的所有还未被访问过的邻接顶点。再从这些访问过的顶点出发, 再访问它们的所有还未被访问过的邻接顶点, ... 如此做下去, 直到图中所有顶点都被访问到为止。

例子



广度遍历: $V1 \Rightarrow V2 \Rightarrow V3 \Rightarrow V4 \Rightarrow V5 \Rightarrow V6 \Rightarrow V7 \Rightarrow V8$

例



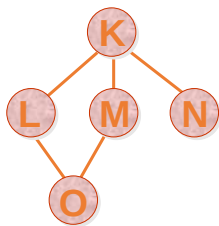
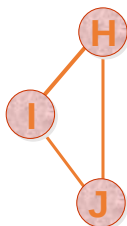
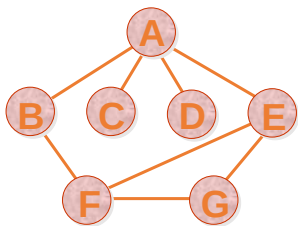
广度遍历: $V1 \Rightarrow V2 \Rightarrow V3 \Rightarrow V4 \Rightarrow V5 \Rightarrow V6 \Rightarrow V7 \Rightarrow V8$

非连通图的连通分量

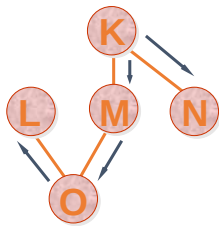
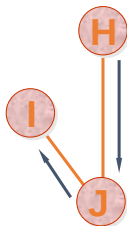
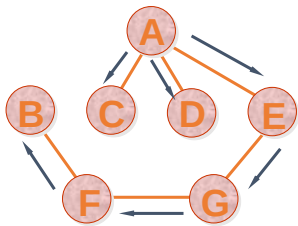
- 当无向图为非连通图时, 从图中某一顶点出发, 利用深度优先搜索算法或广度优先搜索算法不可能遍历到图中的所有顶点, 只能访问到该顶点所在的最大连通子图(连通分量)的所有顶点。

连通分量

- 若从无向图的每一个连通分量中的一个顶点出发进行遍历，可求得无向图的所有连通分量。
- 在算法中，需要对图的每一个顶点进行检测：若已被访问过，则该顶点一定是落在图中已求得的连通分量上；若还未被访问，则从该顶点出发遍历图，可求得图的另一个连通分量。
- 对于非连通的无向图，所有连通分量的生成树组成了非连通图的生成森林。



非连通无向图



非连通图的连通分量

最小生成树

minimum cost spanning tree

最小生成树

- 使用不同的遍历图的方法，可以得到不同的生成树；从不同的顶点出发，也可能得到不同的生成树。
- 按照生成树的定义， n 个顶点的连通网络的生成树有 n 个顶点、 $n-1$ 条边。

构造最小生成树的准则

- 必须使用且仅使用该网络中的 $n-1$ 条边来联结网络中的 n 个顶点;
- 不能使用产生回路的边;
- 各边上的权值的总和达到最小

最短路径

Shortest path

最短路径问题

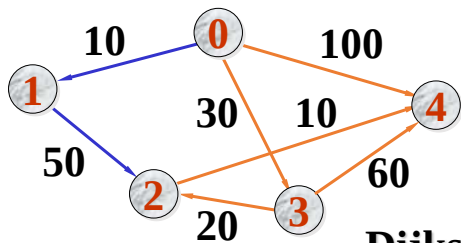
- 如果从图中某一顶点(称为源点)到达另一顶点(称为终点)的路径可能不止一条, 如何找到一条路径使得沿此路径上各边上的权值总和达到最小

算法

- ◆ **边上权值非负情形的单源最短路径问题**
 - Dijkstra算法
- ◆ **边上权值为任意值的单源最短路径问题**
 - Bellman和Ford算法
- ◆ **所有顶点之间的最短路径**
 - Floyd算法

边上权值非负情形的单源最短路径问题

- **问题的提法：** 给定一个带权有向图 D 与源点 v ，求从 v 到 D 中其它顶点的最短路径。限定各边上的权值大于或等于0。
- 为求得这些最短路径，Dijkstra提出按路径长度的递增次序，逐步产生最短路径的算法。首先求出长度最短的一条最短路径，再参照它求出长度次短的一条最短路径，依次类推，直到从顶点 v 到其它各顶点的最短路径全部求出为止。



Dijkstra逐步求解的过程

源点	终点	最短路径	路径长度
v_0	v_1	(v_0, v_1)	10
v_2	—	(v_0, v_1, v_2) (v_0, v_3, v_2)	$\infty, 60, 50$
v_3		(v_0, v_3)	30
v_4		(v_0, v_4) (v_0, v_3, v_4) (v_0, v_3, v_2, v_4)	100, 90, 60

- 引入辅助数组dist。它的每一个分量dist[i]表示当前找到的从源点 v_0 到终点 v_i 的最短路径的长度。初始状态:
 - 若从源点 v_0 到顶点 v_i 有边, 则dist[i]为该边上的权值;
 - 若从源点 v_0 到顶点 v_i 无边, 则dist[i]为 ∞ 。
- 假设S是已求得的最短路径的终点的集合, 则可证明: 下一条最短路径必然是从 v_0 出发, 中间只经过S中的顶点便可到达的那些顶点 v_x ($v_x \in V-S$) 的路径中的一条。
- 每次求得一条最短路径后, 其终点 v_k 加入集合S, 然后对所有的 $v_i \in V-S$, 修改其dist[i]值。

Dijkstra算法可描述如下:

- ① 初始化: $S \leftarrow \{v_0\};$
 $\text{dist}[j] \leftarrow \text{Edge}[0][j], j = 1, 2, \dots, n-1;$
// n 为图中顶点个数
- ② 求出最短路径的长度:
 $\text{dist}[k] \leftarrow \min \{ \text{dist}[i] \}, i \in V-S;$
 $S \leftarrow S \cup \{k\};$
- ③ 修改:
 $\text{dist}[i] \leftarrow \min \{ \text{dist}[i], \text{dist}[k] + \text{Edge}[k][i] \},$
对于每一个 $i \in V-S;$
- ④ 判断: 若 $S = V$, 则算法结束, 否则转 ②。

Sensor Network Localization

传感器网络定位

问题

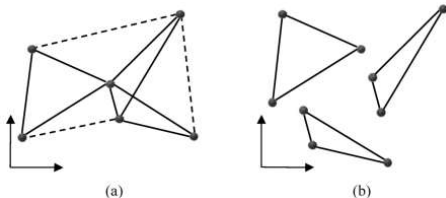
- 背景：
 - 无线传感器网络（军事、工业、环境、医疗等）
 - 分子结构预测
 - 互联网拓扑关系
 - 地图重建
 - 数据可视化
- 问题：
 - 每个传感器能感知到其周边一定范围内的其他传感器的距离，求整体的传感器分布结构
 - 从局部距离出发，推断整体结构

数学模型

- 优化模型

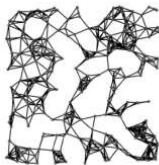
$$Stress(p_1, \dots, p_n) = \sum_{(i,j) \in E} (\|p_i - p_j\| - d_{ij})^2$$

- 几何模型

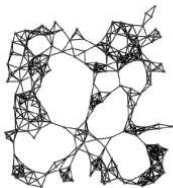


不同方法的比较

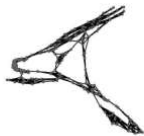
Ground truth



SMACOF



LRE



MDS-MAP(P, R)



ARAP



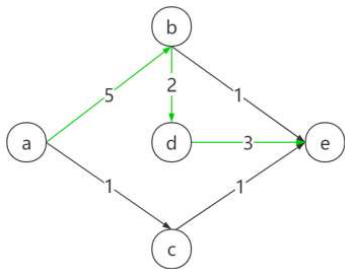
最大流最小割

最大流最小割定理

- 在一个网络流中，能够从源点到达汇点的最大流量等于如果从网络中移除就能够导致网络流中断的边的集合的最小容量和。即在任何网络中，最大流的值等于最小割的容量。
- 背景：
 - 现实生活中，人们经常见到一些网络，如铁路网、公路网、通信网、运输网等等。这些网络有一个共同的特点，就是在网络中都有物资、人或信息等某种量从一个地方流向另一个地方，如何安排这些量的流动以便取得最大效益是一个很有意义的实际问题。

最大流和最小割

- 最大流
 - 给定指定的一个有向图，其中有两个特殊的点源S(Sources)和汇T(Sinks)，每条边有指定的容量(Capacity)，求满足条件的从S到T的最大流(MaxFlow)。
- 最小割
 - 割是网络中定点的一个划分，它把网络中的所有顶点划分成两个顶点集合S和T，其中源点 $s \in S$ ，汇点 $t \in T$ 。记为 $CUT(S, T)$ ，满足条件的从S到T的最小割 (Min cut) 。



Graphcut

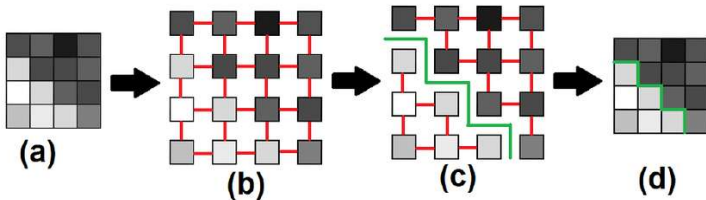
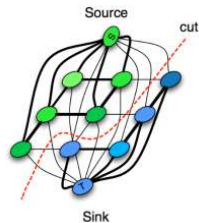
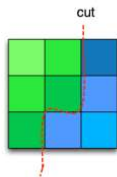
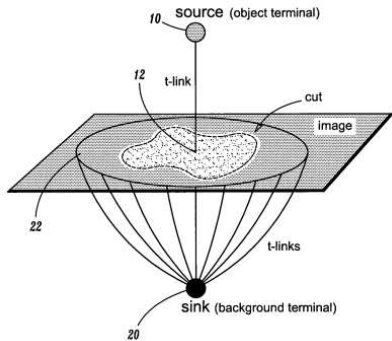
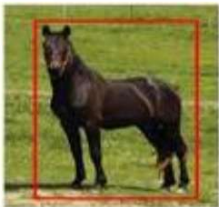


Image Segmentation



Lazy Snapping



更多例子

例3：（CUMCM2007B）乘公交看奥运。

我国人民翘首企盼的第29届奥运会明年8月将在北京举行，届时有大量观众到现场观看奥运比赛，其中大部分人将会乘坐公共交通工具（简称公交，包括公汽、地铁等）出行。这些年来，城市的公交系统有了很大发展，北京市的公交线路已达800条以上，使得公众的出行更加通畅、便利，但同时也面临多条线路的选择问题。针对市场需求，某公司准备研制开发一个解决公交线路选择问题的自主查询计算机系统。

为了设计这样一个系统，其核心是线路选择的模型与算法，应该从实际情况出发考虑，满足查询者的各种不同需求。请你们解决如下问题：

1、仅考虑公汽线路，给出任意两公汽站点之间线路选择问题的一般数学模型与算法。并根据附录数据，利用你们的模型与算法，求出以下6对起始站→终到站之间的最佳路线（要有清晰的评价说明）。

(1) S3359→S1828 (2) S1557→S0481 (3) S0971→S0485

(4) S0008→S0073 (5) S0148→S0485 (6) S0087→S3676

2、同时考虑公汽与地铁线路，解决以上问题。

3、假设又知道所有站点之间的步行时间，请你给出任意两站点之间线路选择问题的数学模型。

【附录1】基本参数设定

相邻公汽车站平均行驶时间(包括停站时间): 3分钟

相邻地铁站平均行驶时间(包括停站时间): 2.5分钟

公汽换乘公汽平均耗时: 5分钟(其中步行时间2分钟)

地铁换乘地铁平均耗时: 4分钟(其中步行时间2分钟)

地铁换乘公汽平均耗时: 7分钟(其中步行时间4分钟)

公汽换乘地铁平均耗时: 6分钟(其中步行时间4分钟)

公汽票价: 分为单一票价与分段计价两种, 标记于线路后; 其中分段计价的票价为: 0~20站1元; 21~40站2元; 40站以上3元

地铁票价: 3元 (无论地铁线路间是否换乘)

注: 以上参数均为简化问题而作的假设, 未必与实际数据完全吻合。

【附录2】公交线路及相关信息 (见数据文件B2007data.rar)

【模型建立】

- 以公汽和地铁站点为顶点构造有向图 $G = (V, E)$ ，其中 $\overrightarrow{uv} \in E \Leftrightarrow$ 可乘某一条线路从 u 到 v 或者从 u 换乘到 v 。
- $|V| = 3996$ ， $|E| = 547050$ ，两个顶点之间可能有多条边。
- 对于每条边 $e \in E$ ，定义 $L(e) = (l_1, l_2, l_3)$ ，分别代表总时间、步行时间、票价。
- 数学问题：求任意两个顶点之间以 $\sum_{e \in P} L(e)$ 为优化目标的最优路径 P 。

例4：（CUMCM2017D）巡检线路的排班。

某化工厂有26个点需要进行巡检以保证正常生产，各个点的巡检周期、巡检耗时、两点之间的连通关系及行走所需时间在附件中给出。

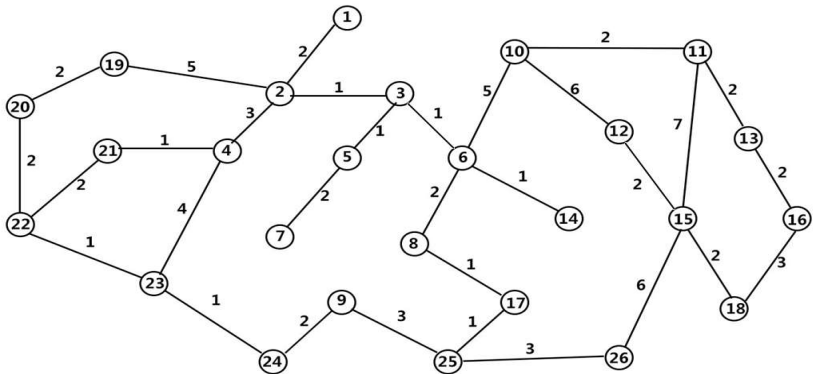
每个点每次巡检需要一名工人，巡检工人的巡检起始地点在巡检调度中心（XJ0022），工人可以按固定时间上班，也可以错时上班，在调度中心得到巡检任务后开始巡检。现需要建立模型来安排巡检人数和巡检路线，使得所有点都能按要求完成巡检，并且耗费的人力资源尽可能少，同时还应考虑每名工人在一时间段内（如一周或一月等）的工作量尽量平衡。

问题1. 如果采用固定上班时间，不考虑巡检人员的休息时间，采用每天三班倒，每班工作8小时左右，每班需要多少人，巡检线路如何安排，并给出巡检人员的巡检线路和巡检的时间表。

问题2. 如果巡检人员每巡检2小时左右需要休息一次，休息时间大约是5到10分钟，在中午12时和下午6时左右需要进餐一次，每次进餐时间为30分钟，仍采用每天三班倒，每班需要多少人，巡检线路如何安排，并给出巡检人员的巡检线路和巡检的时间表。

问题3. 如果采用错时上班，重新讨论问题1和问题2，试分析错时上班是否更节省人力。

位号	1	2	3	4	5	6	7	8	9	10	11	12
周期	35	50	35	35	720	35	80	35	35	120	35	35
耗时	3	2	3	2	2	3	2	3	4	2	3	2
位号	13	14	15	16	17	18	19	20	21	22	23	24
周期	80	35	35	35	480	35	35	35	80	35	35	35
耗时	5	3	2	3	2	2	2	3	3	2	3	2



【模型建立】

- 建立附件所示的无向图 $G = (V, E)$, $|V| = 26$ 。
- 固定上班时间与错时上班并无区别。关键是上一班的工人何时下班。
- 设所有人的巡检路线都是 $x_0 \rightarrow x_1 \rightarrow \cdots \rightarrow x_{26}$, 其中 $x_0 = x_{26} = 22$ 。
- 巡检时间 $T = \sum_{i=1}^{26} \text{巡检耗时}(x_i) + \text{行走时间}(x_{i-1}, x_i)$ 。
- 问题化为求经过所有顶点的回路, 使得 T 最小。
- 工人只需间隔35分钟从调度中心出发即可满足题设。
- 若考虑休息时间 r , 则间隔时间改为 $35 - r$ 。
- 不可能所有工人同时集中进餐30分钟。
- 经计算, 存在巡检时间139分钟的路线 (非最优), 每班4人即可。

例5：（CUMCM2013B）碎纸片的拼接复原。

破碎文件的拼接在司法物证复原、历史文献修复以及军事情报获取等领域都有着重要的应用。传统上，拼接复原工作需由人工完成，准确率较高，但效率很低。特别是当碎片数量巨大，人工拼接很难在短时间内完成任务。随着计算机技术的发展，人们试图开发碎纸片的自动拼接技术，以提高拼接复原效率。请讨论以下问题：

1. 对于给定的来自同一页印刷文字文件的碎纸机破碎纸片（仅纵切），建立碎纸片拼接复原模型和算法，并针对附件1、附件2给出的中、英文各一页文件的碎片数据进行拼接复原。如果复原过程需要人工干预，请写出干预方式及干预的时间节点。复原结果以图片形式及表格形式表达（见【结果表达格式说明】）。

2. 对于碎纸机既纵切又横切的情形，请设计碎纸片拼接复原模型和算法，并针对附件3、附件4给出的中、英文各一页文件的碎片数据进行拼接复原。如果复原过程需要人工干预，请写出干预方式及干预的时间节点。复原结果表达要求同上。

3. 上述所给碎片数据均为单面打印文件，从现实情形出发，还可能有两面打印文件的碎纸片拼接复原问题需要解决。附件5给出的是一页英文印刷文字双面打印文件的碎片数据。请尝试设计相应的碎纸片拼接复原模型与算法，并就附件5的碎片数据给出拼接复原结果，结果表达要求同上。

【数据文件说明】

- (1) 每一附件为同一页纸的碎片数据。
- (2) 附件1、附件2为纵切碎片数据，每页纸被切为19条碎片。
- (3) 附件3、附件4为纵横切碎片数据，每页纸被切为 11×19 个碎片。
- (4) 附件5为纵横切碎片数据，每页纸被切为 11×19 个碎片，每个碎片有正反两面。该附件中每一碎片对应两个文件，共有 $2 \times 11 \times 19$ 个文件，例如，第一个碎片的两面分别对应文件000a、000b。

【结果表达格式说明】

复原图片放入附录中，表格表达格式如下：

- (1) 附件1、附件2的结果：将碎片序号按复原后顺序填入 1×19 的表格；
- (2) 附件3、附件4的结果：将碎片序号按复原后顺序填入 11×19 的表格；
- (3) 附件5的结果：将碎片序号按复原后顺序填入两个 11×19 的表格；
- (4) 不能确定复原位置的碎片，可不填入上述表格，单独列表。

【模型建立】

- 需要解决的实际问题：求碎片的排列（对于双面打印的碎片可能还需要翻转），使得相邻的碎片可以拼接起来。
- 对于仅纵切情形。以碎片为顶点构造有向图 $G = (V, E)$ ，其中 $\overrightarrow{uv} \in E \Leftrightarrow$ 碎片 u 的右边界与碎片 v 的左边界可以拼接。
- 对于任意两块碎片 x, y ，可用一个函数 $f(x, y)$ 描述它们的拼接吻合程度。问题转化为求图 G 中的一条经过所有顶点的有向Path，使得 $\sum_{e \in P} f(e)$ 最大。
- 对于既纵切又横切的情形。定义函数 $g(x, y)$ 判断两块碎片 x, y 是否属于相同行，并描述它们的拼接吻合程度。据此把 $m \times n$ 个碎片平均分成 m 部分。问题转化为纵切和横切情形。
- 对于双面情形。定义两块碎片 x, y 的拼接吻合程度函数
$$h(x, y) = \max(g(x, y), g(x, \tilde{y}), g(\tilde{x}, y), g(\tilde{x}, \tilde{y}))$$



中国科学技术大学
University of Science and Technology of China

谢谢！



中国科学技术大学

University of Science and Technology of China

数学建模

Mathematical Modeling

刘利刚

中国科学技术大学

课程总结

我们学习了什么？

日期	课程内容		
第1讲 2023.03.10.	1-课程介绍 2-简化与抽象	第7讲 2023.04.21.	9-神经网络模型
第2讲 2023.03.17.	3-高维数据分析模型 4-高维数据变换模型	第8讲 2023.04.28.	10-各种网络模型 11-深度学习的理论基础
第3讲 2023.03.24.	5-矩阵分解模型	第9讲 2023.05.05.	12-高阶拟合模型 13-微分方程模型
第4讲 2023.03.31.	6-最优化模型	第10讲 2023.05.12.	14-数码相机标定 15-平衡与稳定性模型 16-统计回归模型
第5讲 2023.04.07.	7-聚类分析模型	第11讲 2023.05.19.	17-CT成像模型 18-多目标决策模型 19-层次分析法模型
第6讲 2023.04.14.	8-拟合模型	第12讲 2023.05.26.	20-图与网络模型 22-课程总结

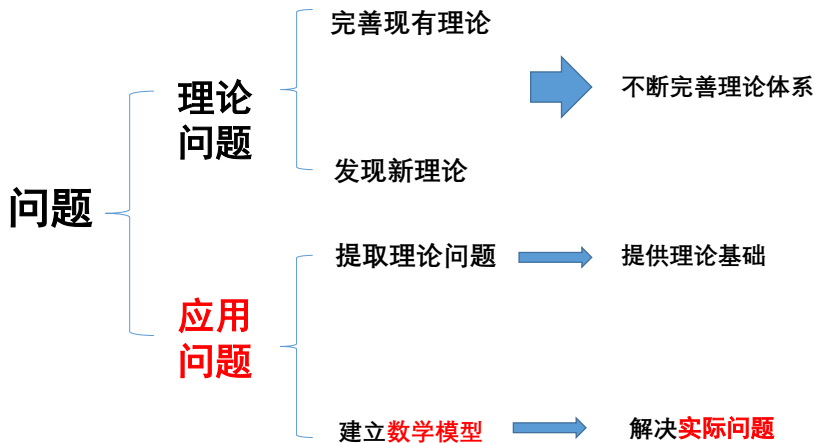
数学建模的“定义”

- 定义1: 关于部分现实世界为一定目的而作的抽象、简化的数学结构, 即, 用数学术语对部分现实世界的描述
- 定义2: 用数学的语言和方法, 通过抽象、简化建立能近似刻画并“解决”实际问题的一种数学工具

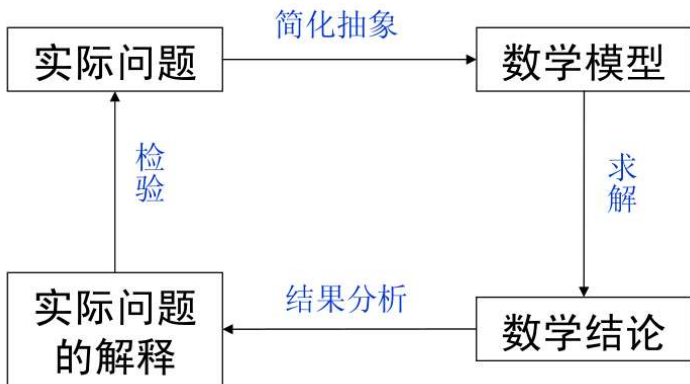
数学建模的核心能力： “抽象”能力

1. 透过表象看本质：第一性原理
2. 逻辑推理形递归：演绎法

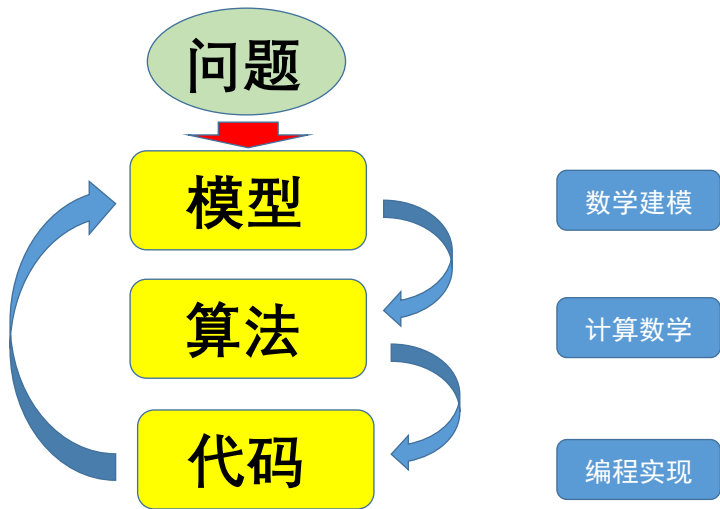
工作（学习、研究）范式



数学建模的过程



解决问题的关键在于**数学建模**能力



即：科学研究的过程

方法论

- 抽象
 - 简化非本质的因素
 - 透过现象看本质
- 迭代
 - 将问题转化为已有问题
 - 组合、递归
- 执行方法
 - 20-80原则
 - Divide-and-conquer
 - Coarse-to-fine

科研工作：建模

- 锤子 + 钉子
- 模型的创新
- 求解的创新

学习、生活、工作中的模型

- 思维模型（模式）
- 不断成长、成长为自己想象中的那个自己

祝各位同学学习顺利！
生活愉快！



中国科学技术大学
University of Science and Technology of China

谢谢！