法律声明

- 本课件包括演示文稿、示例、代码、题库、视频和声音等内容，小象学院和主讲老师拥有完全知识产权的权利；只限于善意学习者在本课程使用，不得在课程范围外向任何第三方散播。任何其他人或机构不得盗版、复制、仿造其中的创意及内容，我们保留一切通过法律手段追究违反者的权利。

- 课程详情请咨询
  - 微信公众号：小象
  - 新浪微博：ChinaHadoop

机器学习与数学分析

邹博

主要内容

- 机器学习与本课程示例概述
- 机器学习的角度看数学：
  - 数学分析
    - 导数与梯度
    - Taylor展式的应用
  - 概率论基础
    - 古典概型
    - 频率学派与贝叶斯学派
    - 常见概率分布
    - Sigmoid/Logistic函数的引入

什么是机器学习

- 对于某给定的任务T，在合理的性能度量方案P的前提下，某计算机程序可以自主学习任务T的经验E；随着提供合适、优质、大量的经验E，该程序对于任务T的性能逐步提高。
- 这里最重要的是机器学习的对象：
  - 任务Task,T，一个或者多个
  - 经验Experience,E
  - 性能Performance,P
- 即：随着任务的不断执行，经验的累积会带来计算机性能的提升。
  - Tom Michael Mitchell, 1997

换个表述

- 机器学习是人工智能的一个分支。我们使用计算机设计一个系统，使它能够根据提供的训练数据按照一定的方式来学习；随着训练次数的增加，该系统可以在性能上不断学习和改进；通过参数优化的学习模型，能够用于预测相关问题的输出。
- 思考：
  - 如何设计无人驾驶机动车？

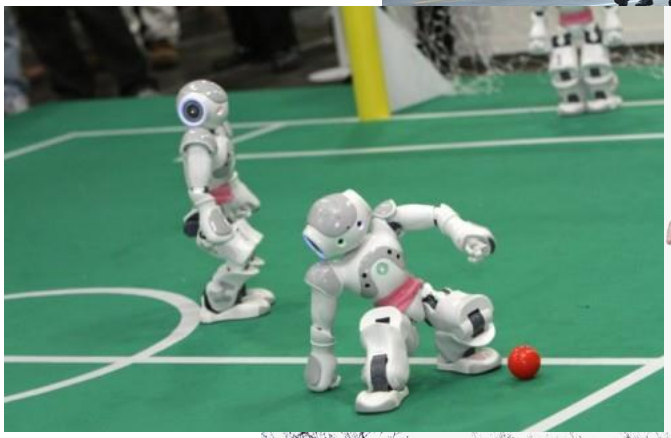
无人驾驶汽车

- 汽车的无人汽车模块已经成熟：全自动公共交通工具已经出现在了世界上的多个城市。
  - Lutz探路者/CYCAB/Google
- 问题：如何设计自动驾驶系统？



人类的学习？

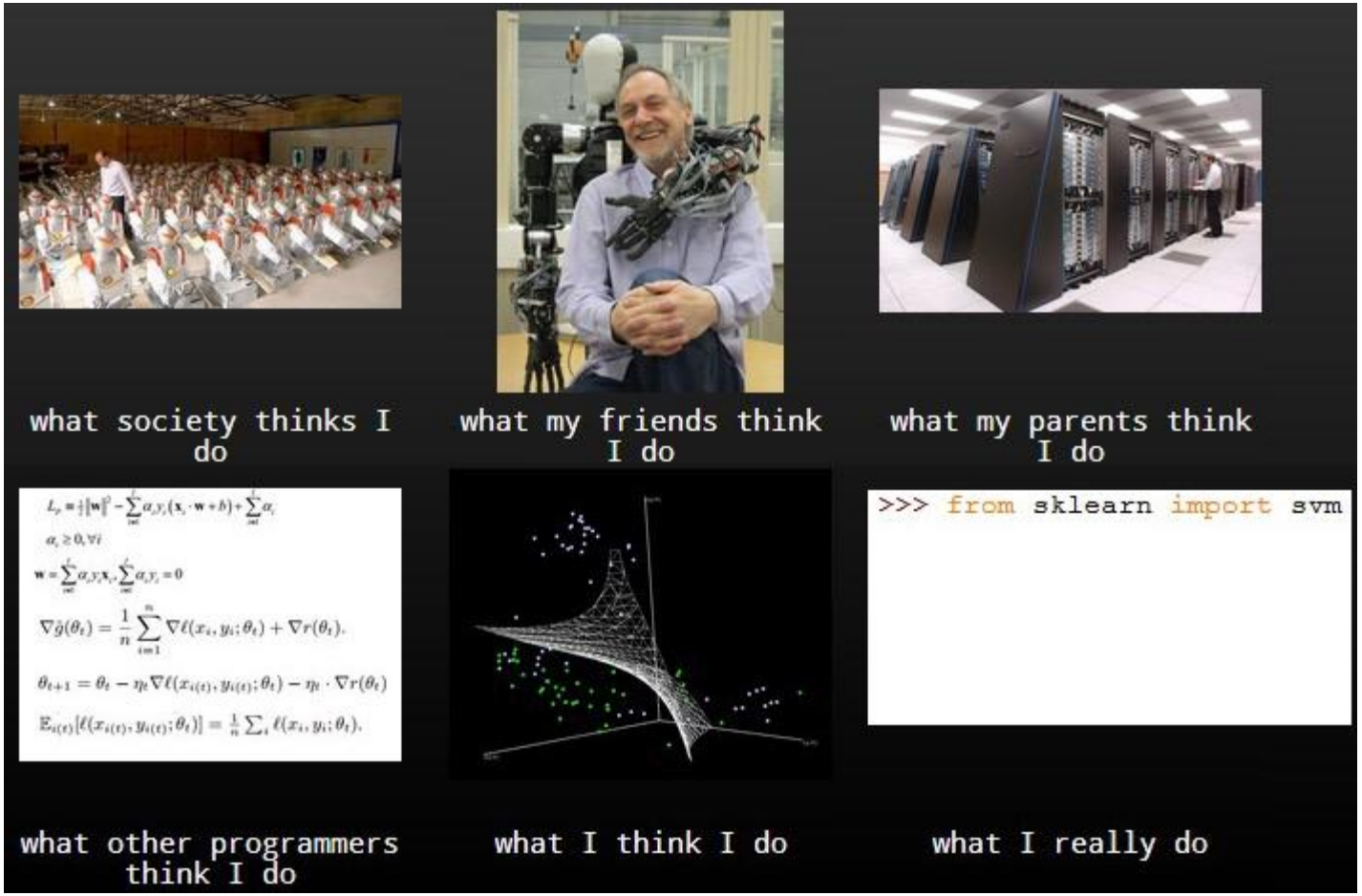
- 如何从完全“无知”到掌握知识？
  - 语言/颜色/形状等特征统计
- 有监督学习
  - 月亮
- 无监督学习
  - 9月3日：“阅兵”
- 增强学习
  - 走路、踢球



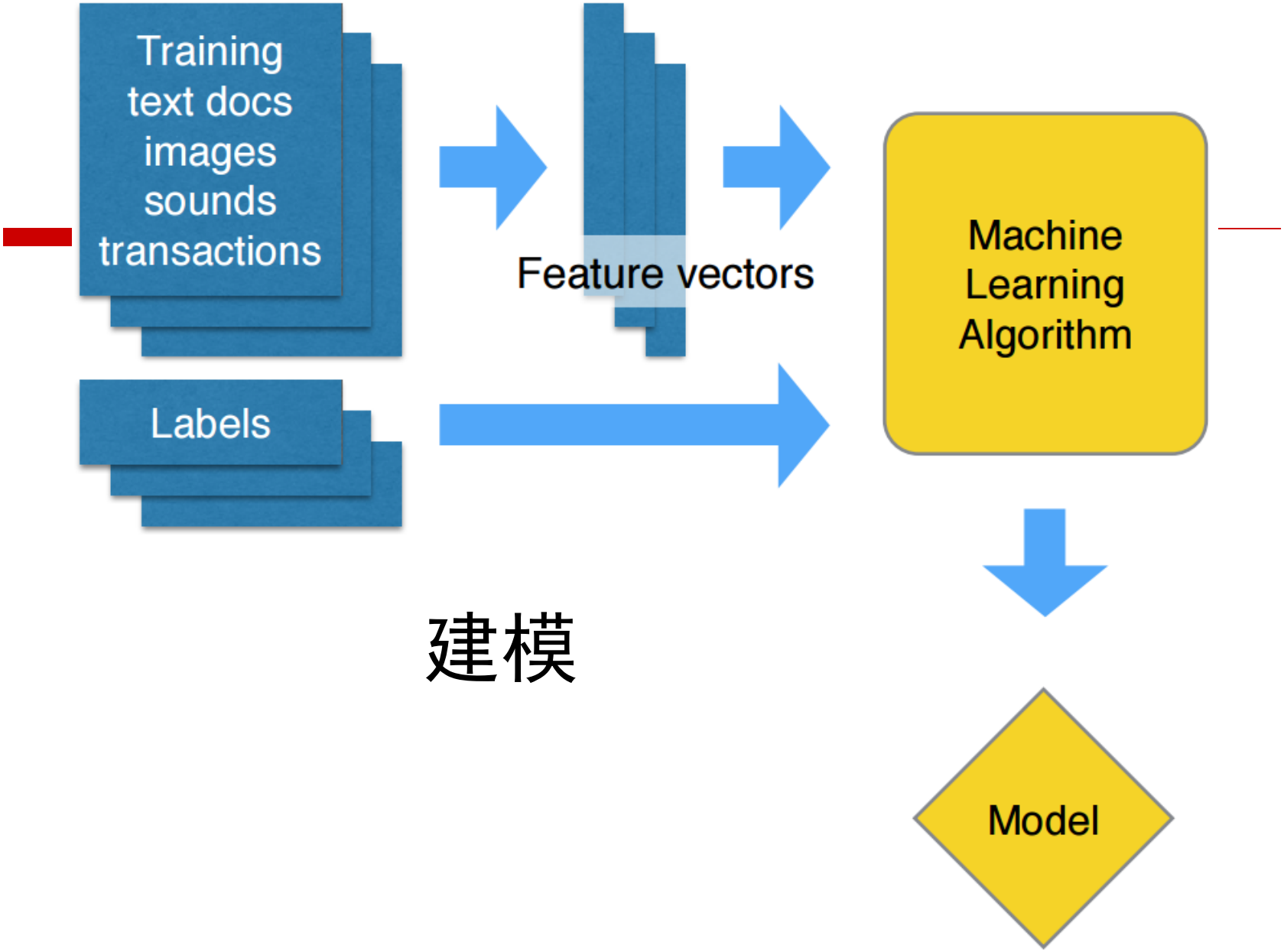
机器学习的内涵与外延

- 机器学习可以解决什么
  - 给定数据的预测问题
    - 数据清洗/特征选择
    - 确定算法模型/参数优化
    - 结果预测
- 不能解决什么
  - 大数据存储/并行计算
  - 做一个机器人
- 举例：
  - 机器学习：“盯住2号位，她很容易起快球”
  - 传统算法：排球规则。

机器学习

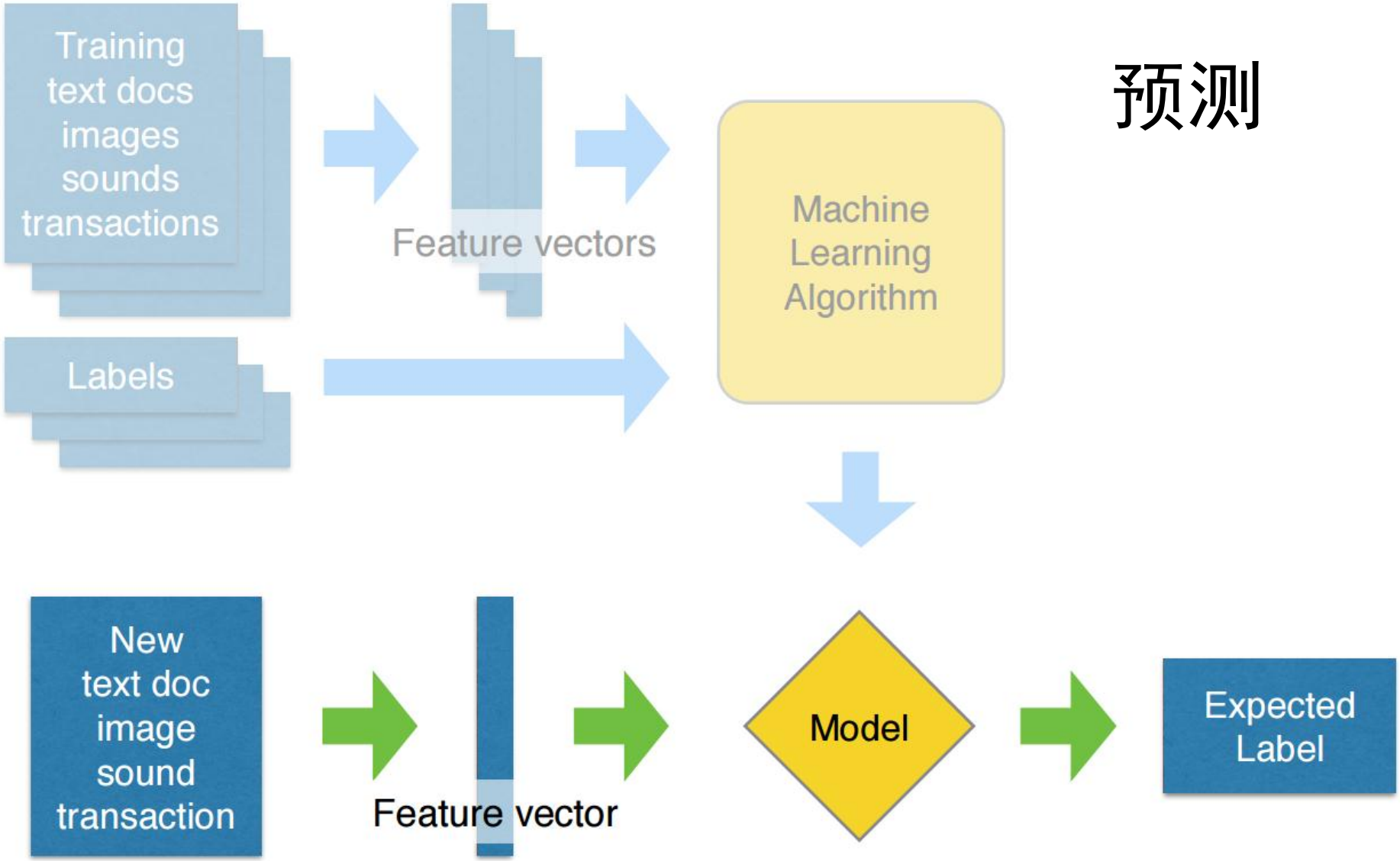

what society thinks I do
what my friends think I do
what my parents think I do
what other programmers think I do
what I think I do
>>> from sklearn import svm
what I really do

	features			target
	type (category)	# rooms (int)	surface (float m2)	public trans (boolean)	sold (float k€)
samples (train)	Apartment	3	50	TRUE	450
	House	5	254	FALSE	430
	Duplex	4	68	TRUE	712
	Apartment	2	32	TRUE	234
samples (test)	Apartment	2	33	TRUE	?
	House	4	210	TRUE	?


Training
text docs
images
sounds
transactions
Feature vectors
Machine
Learning
Algorithm
Labels
Model

建模

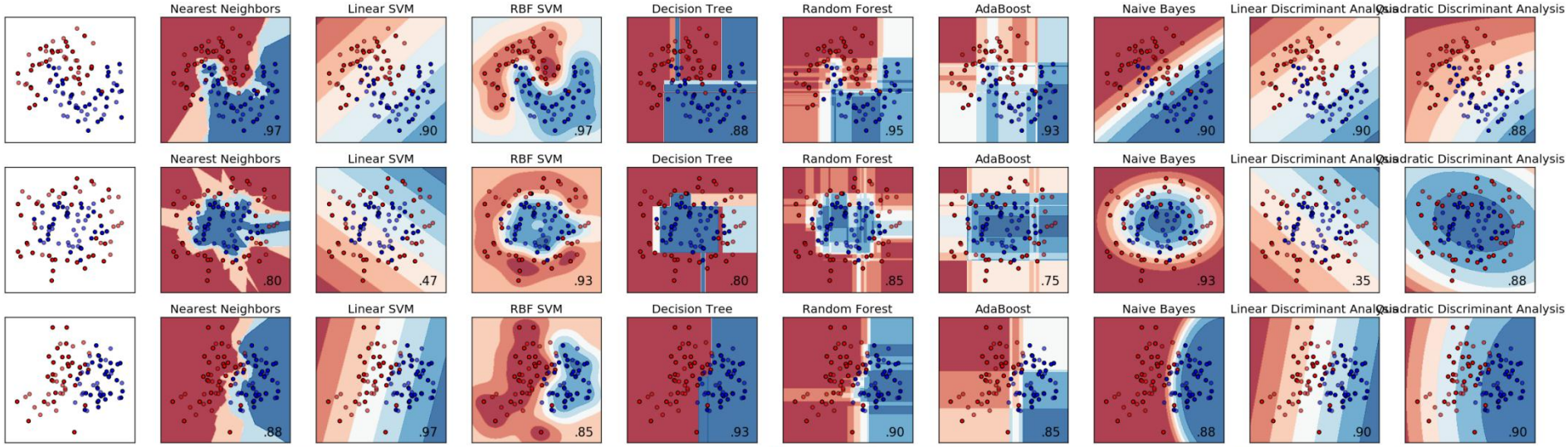
预测


Training
text docs
images
sounds
transactions
Feature vectors
Machine
Learning
Algorithm
Labels
New
text doc
image
sound
transaction
Feature vector
Model
Expected
Label

机器学习的一般流程

- 数据挖掘/机器学习的流程与西红柿炒鸡蛋

机器学习方法



- different assumptions on data
- different scalability profiles at training time
- different latencies at prediction time
- different model sizes (embedability in mobile devices)

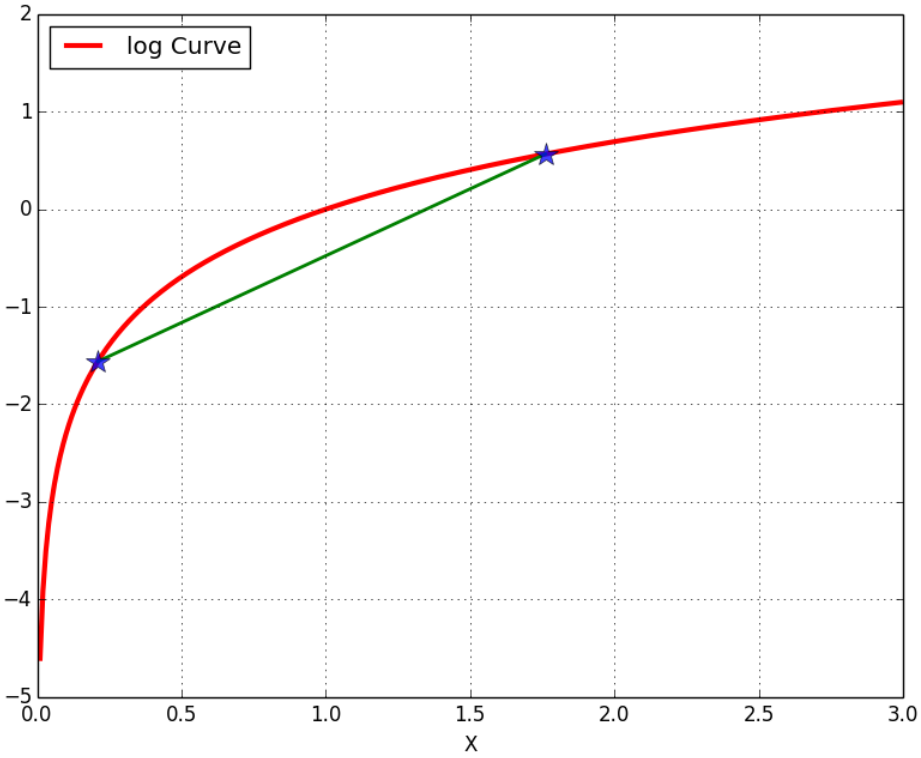
思考：机器如何发现新词

- 频数：Count(X)
- 凝固程度
  - X = A.B
  - P(A)P(B) vs P(X)
- 自由程度
  - aXb
  - 信息熵H(a)、H(b)
- 凝固程度和自由程度缺一不可。只考虑凝固程度，会找出“巧克”、“俄罗”、“颜六色”、“柴可夫”等“半个词”；只考虑自由程度，会把“吃了一顿”、“看了一遍”、“睡了一晚”、“去了一趟”中的“了一”提取出来，因为它的左右邻字都太丰富了。
  - 调参
- 问题：给定某长文本，如何利用上述参数设计可行算法？

Python Code示例1

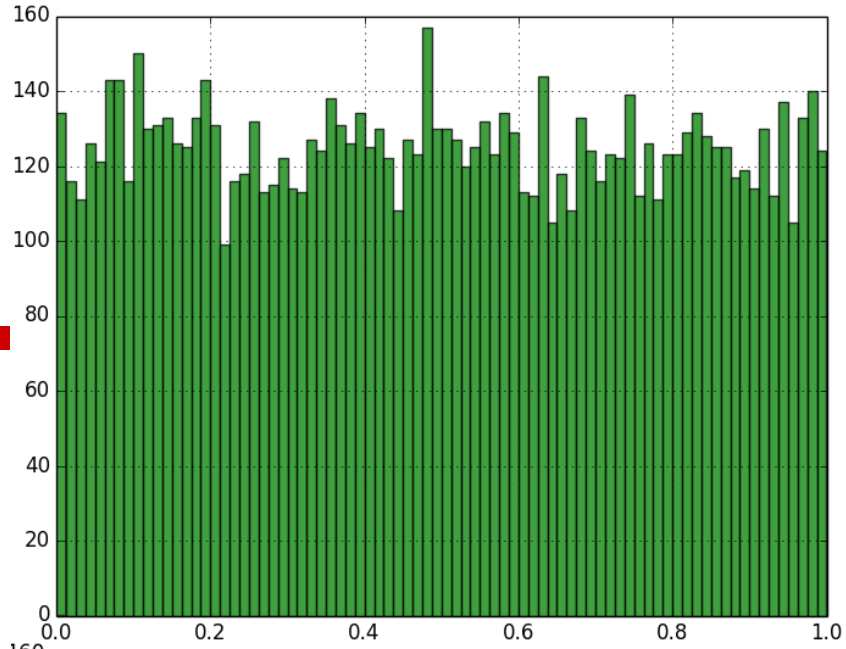
TestLog.py

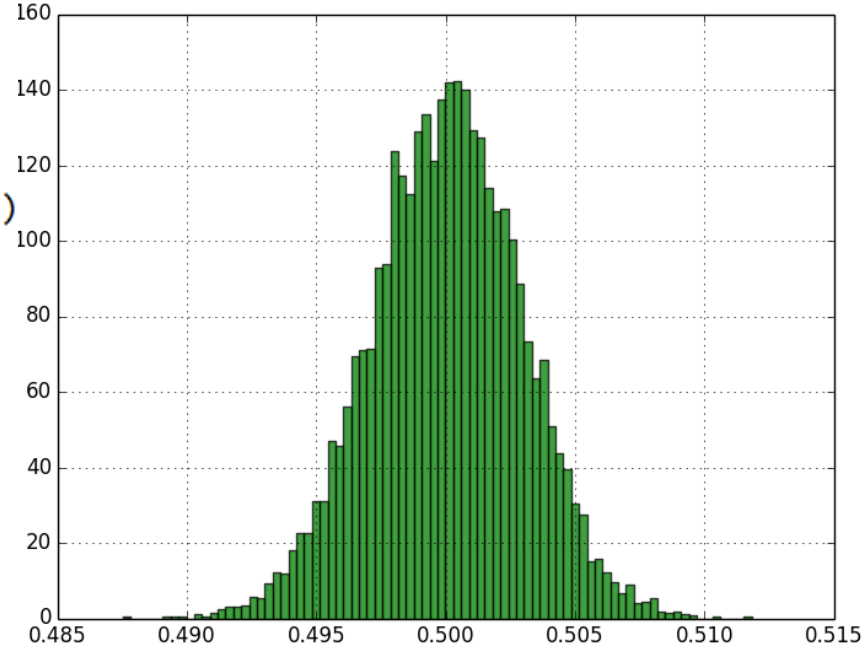
```
# -*- coding:utf8 -*-

import math
import matplotlib.pyplot as plt

if __name__ == "__main__":
    x = [float(i)/100.0 for i in range(1,300)]
    y = [math.log(i) for i in x]
    plt.plot(x, y, 'r-', linewidth=3, label='log Curve')
    a = [x[20], x[175]]
    b = [y[20], y[175]]
    plt.plot(a, b, 'g-', linewidth=2)
    plt.plot(a, b, 'b*', markersize=15, alpha=0.75)
    plt.legend(loc='upper left')
    plt.grid(True)
    plt.xlabel('X')
    plt.ylabel('log(X)')
    plt.show()
```



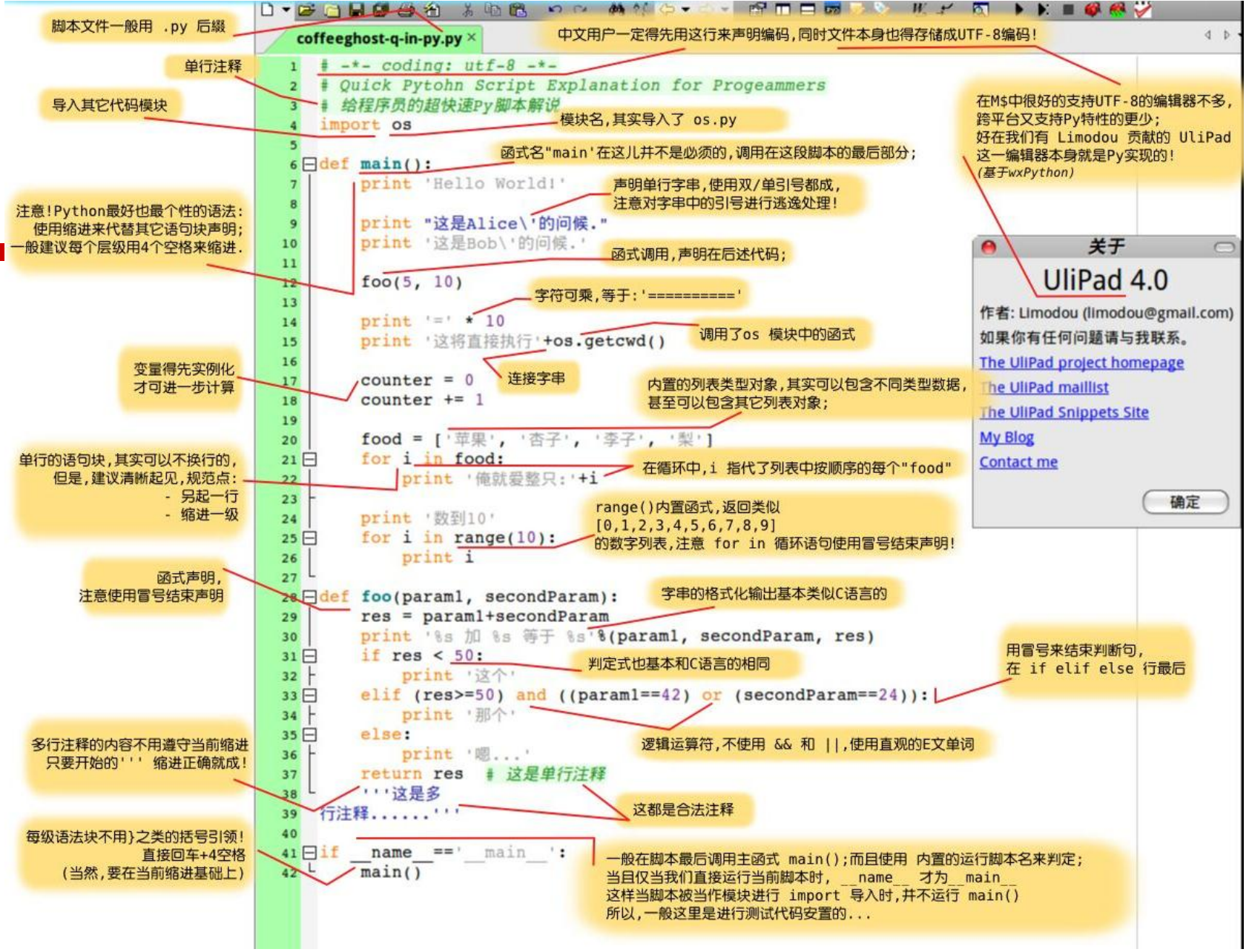
Python Code示例2

```
if __name__ == "__main__":
    u = numpy.random.uniform(0.0, 1.0, 10000)
    plt.hist(u, 80, facecolor='g', alpha=0.75)
    plt.grid(True)
    plt.show()

    times = 10000
    for time in range(times):
        u += numpy.random.uniform(0.0, 1.0, 10000)
    print len(u)
    u /= times
    print len(u)
    plt.hist(u, 80, facecolor='g', alpha=0.75)
    plt.grid(True)
    plt.show()
```



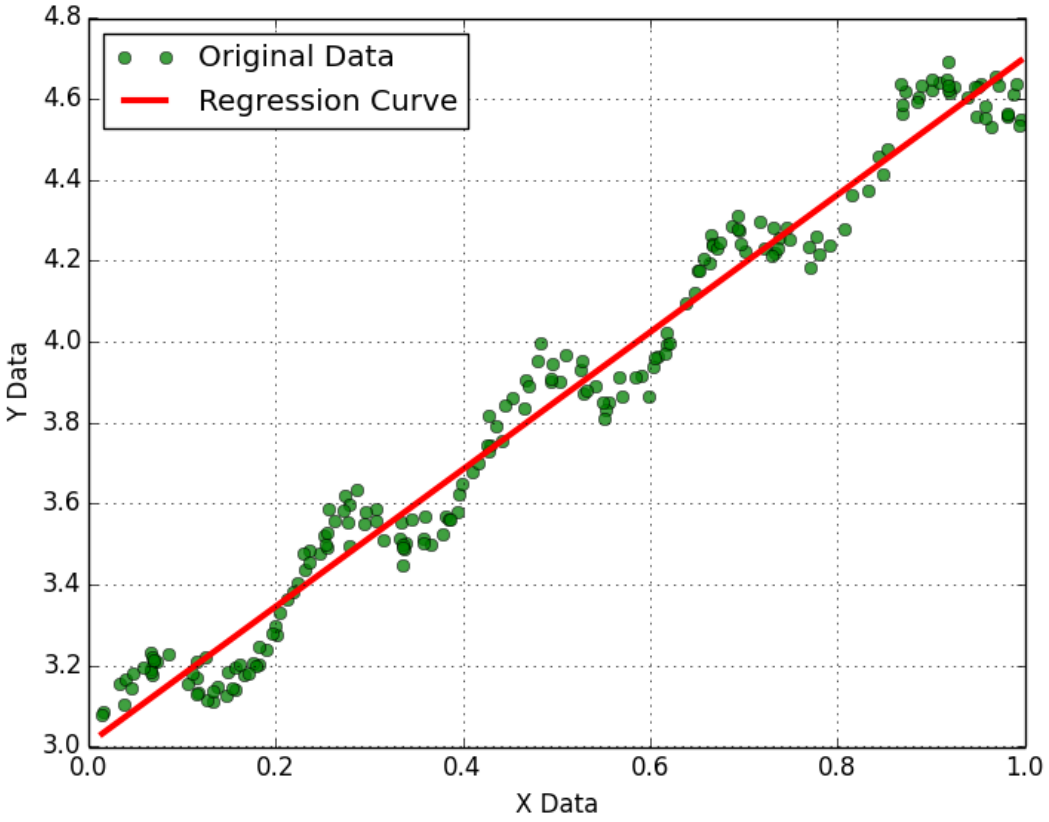


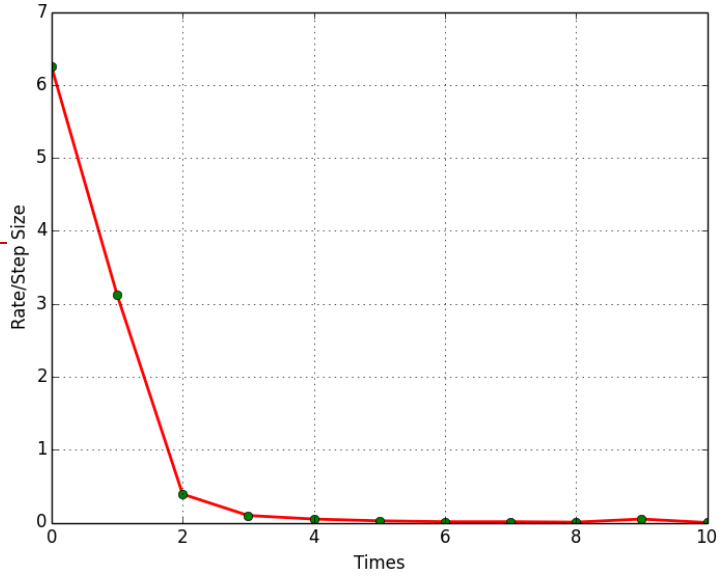
Py

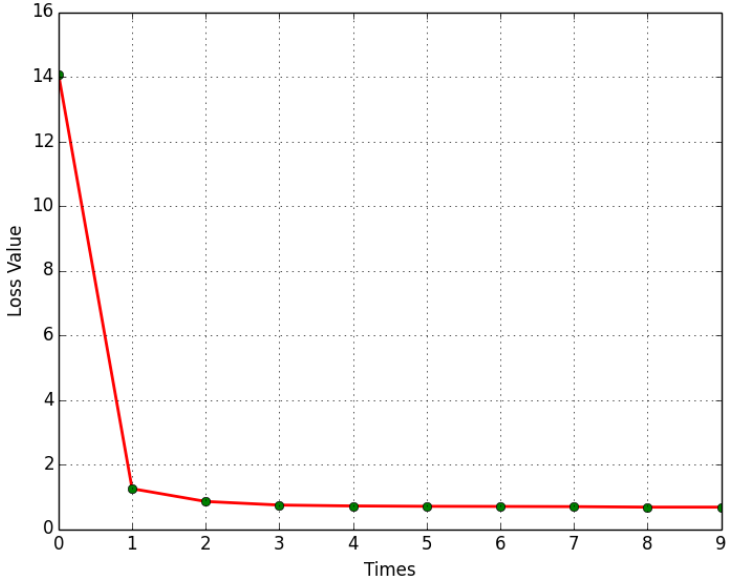



```
1  # -*- coding: utf-8 -*-
2  # Quick Pytohn Script Explanation for Progeammers
3  # 给程序员的超快速Py脚本解说
4  import os
5
6  def main():
7      print 'Hello World!'
8
9      print "这是Alice\'的问候."
10     print '这是Bob\'的问候.'
11
12     foo(5, 10)
13
14     print '=' * 10
15     print '这将直接执行'+os.getcwd()
16
17     counter = 0
18     counter += 1
19
20     food = ['苹果', '杏子', '李子', '梨']
21     for i in food:
22         print '俺就爱整只:'+i
23
24     print '数到10'
25     for i in range(10):
26         print i
27
28 def foo(param1, secondParam):
29     res = param1+secondParam
30     print '%s 加 %s 等于 %s'%(param1, secondParam, res)
31     if res < 50:
32         print '这个'
33     elif (res>=50) and ((param1==42) or (secondParam==24)):
34         print '那个'
35     else:
36         print '嗯...'
37     return res  # 这是单行注释
38     '''这是多
39 行注释.....'''
40
41 if __name__=='__main__':
42     main()
```

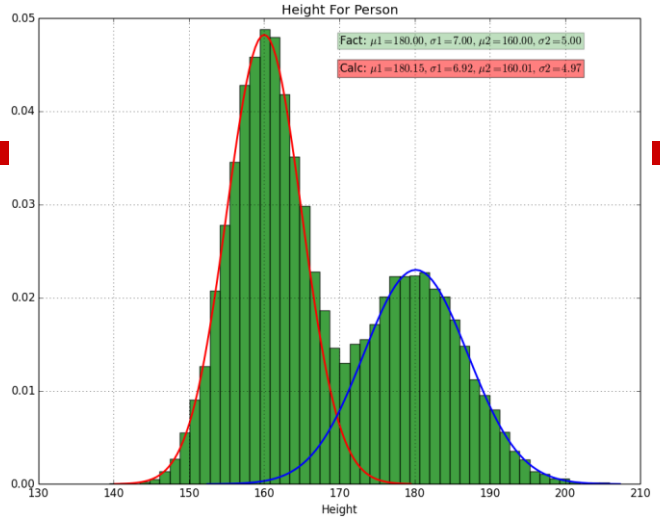
线性回归、rate、Loss

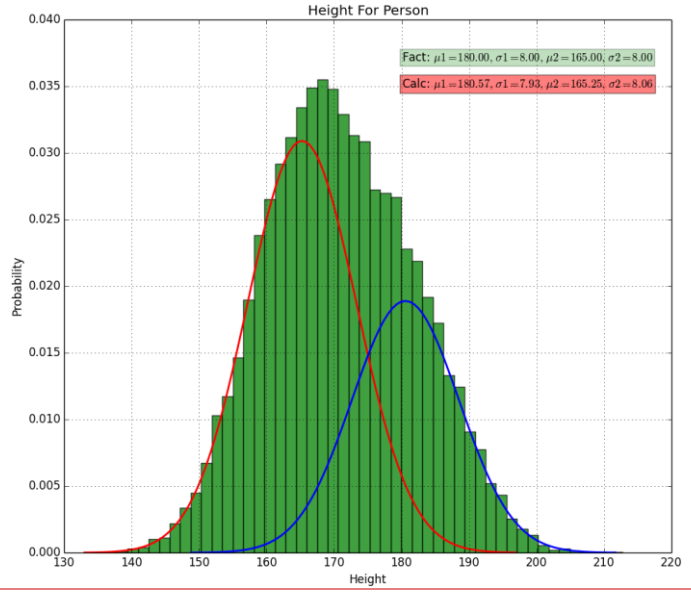






EM Code



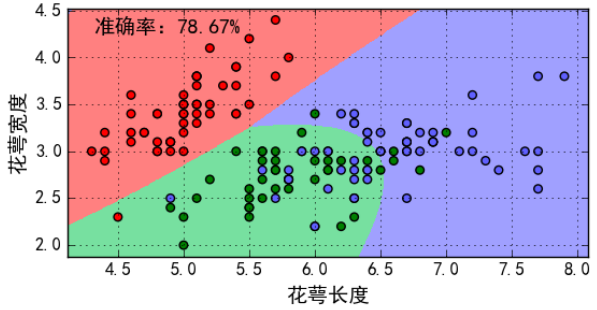


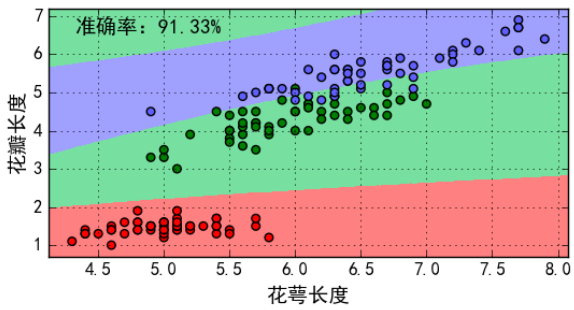
em3.py

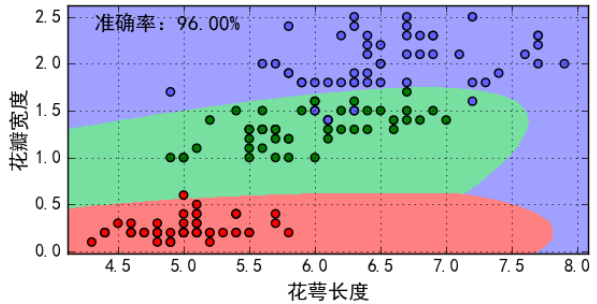
```
def calcEM(height):
    N = len(height)
    gp = 0.5    #girl probability
    bp = 0.5    #boy probability
    gmu,gsigma = min(height),1  #先验：直接取最大和最小值
    bmu,bsigma = max(height),1
    ggamma = range(N)
    bgamma = range(N)
    cur = [gp, bp, gmu, gsigma, bmu, bsigma]
    now = []

    times = 0
    while times < 100:
        i = 0
        for x in height:
            ggamma[i] = gp * gauss(x, gmu, gsigma)
            bgamma[i] = bp * gauss(x, bmu, bsigma)
            s = ggamma[i] + bgamma[i]
            ggamma[i] /= s
            bgamma[i] /= s
            i += 1

        gn = sum(ggamma)
        gp = float(gn) / float(N)
        bn = sum(bgamma)
        bp = float(bn) / float(N)
        gmu = averageWeight(height, ggamma, gn)
        gsigma = varianceWeight(height, ggamma, gmu, gn)
        bmu = averageWeight(height, bgamma, bn)
        bsigma = varianceWeight(height, bgamma, bmu, bn)

        now = [gp, bp,gmu,gsigma,bmu,bsigma]
        if isSame(cur, now):
            break
        cur = now
        print "Times:\t", times
        print "Girl mean/gsigma:\t", gmu,gsigma
        print "Boy mean/bsigma:\t", bmu,bsigma
        print "Boy/Girl:\t", bn, gn, bn+gn
        print "\n\n"
        times += 1
    return now
```

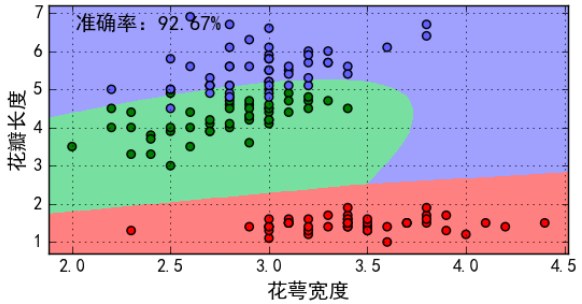
EM算法

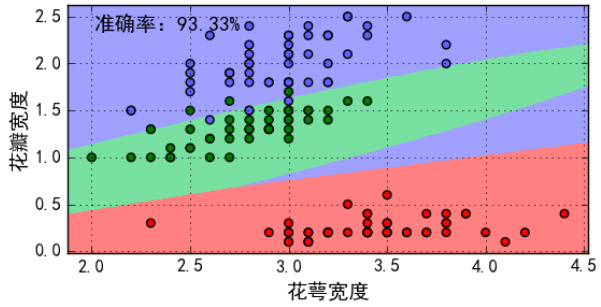
EM算法无监督分类鸢尾花数据

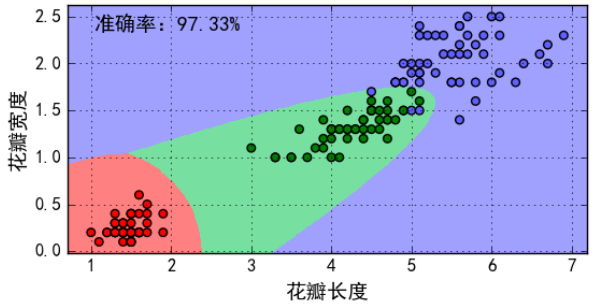




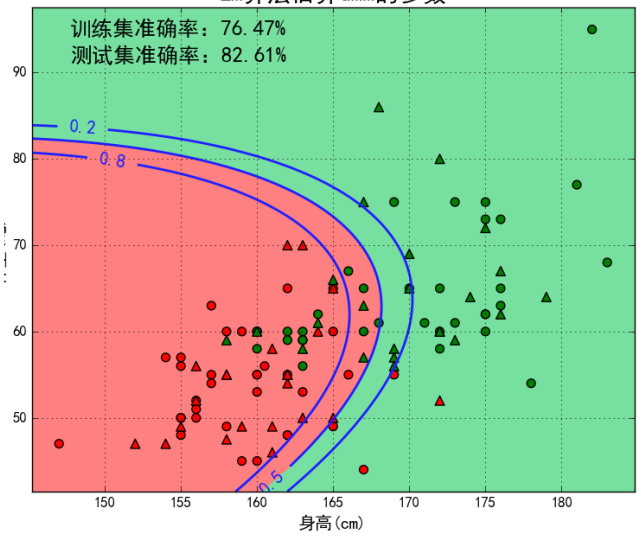




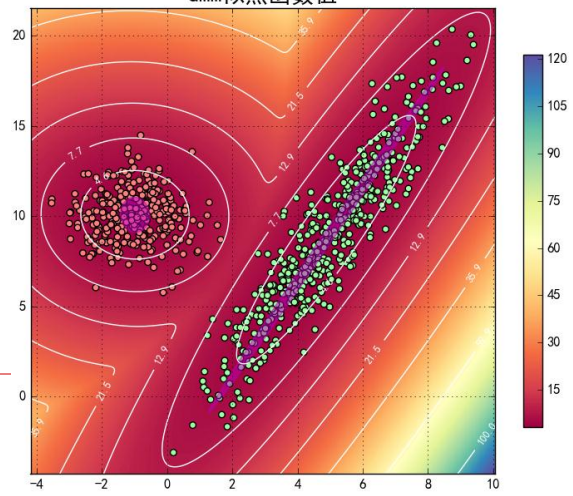




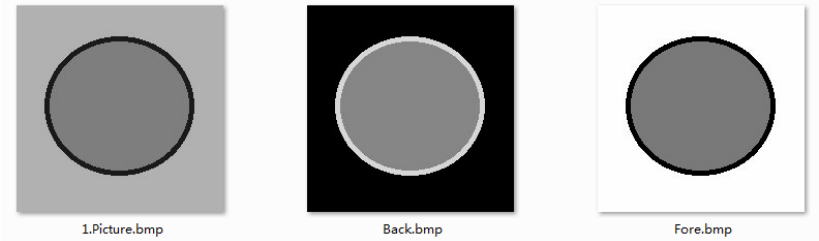
EM算法估算GMM的参数

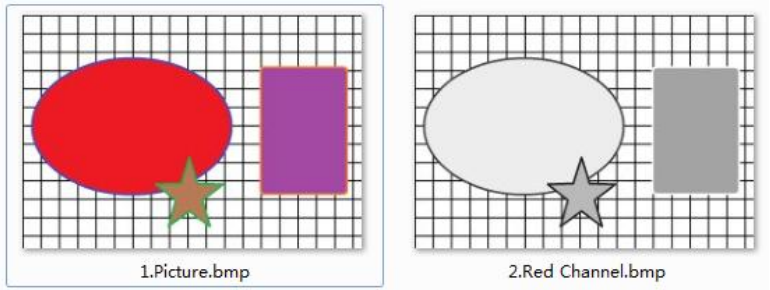


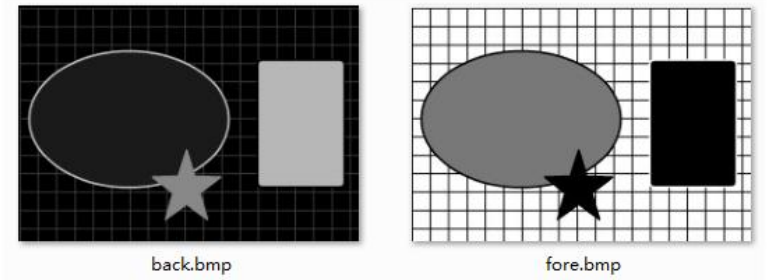
GMM似然函数值



GMM与图像








```
def composite(band, parameter):
    c1 = parameter[0]
    mu1 = parameter[2]
    sigma1 = parameter[3]
    c2 = parameter[1]
    mu2 = parameter[4]
    sigma2 = parameter[5]

    p1 = []
    p2 = []
    for pixel in band:
        p1.append(c1 * gauss(pixel, mu1, sigma1))
        p2.append(c2 * gauss(pixel, mu2, sigma2))

    scale(p1)    #灰度均衡
    scale(p2)
    return [p1, p2]

if __name__ == "__main__":
    im = Image.open('.\\Pic\\test.bmp')
    print im.format, im.size, im.mode

    im = im.split()[0]  #只处理第一个通道
    nb = []             #处理后的新通道
    data = list(im.getdata())
    parameter = GMM(data)
    t = composite(data, parameter)

    im1 = Image.new('L', im.size)
    im1.putdata(t[0])
```

图像的卷积

laplacian.png

laplacian2.png

prewitt.png

prewitt_x.png

prewitt_y.png

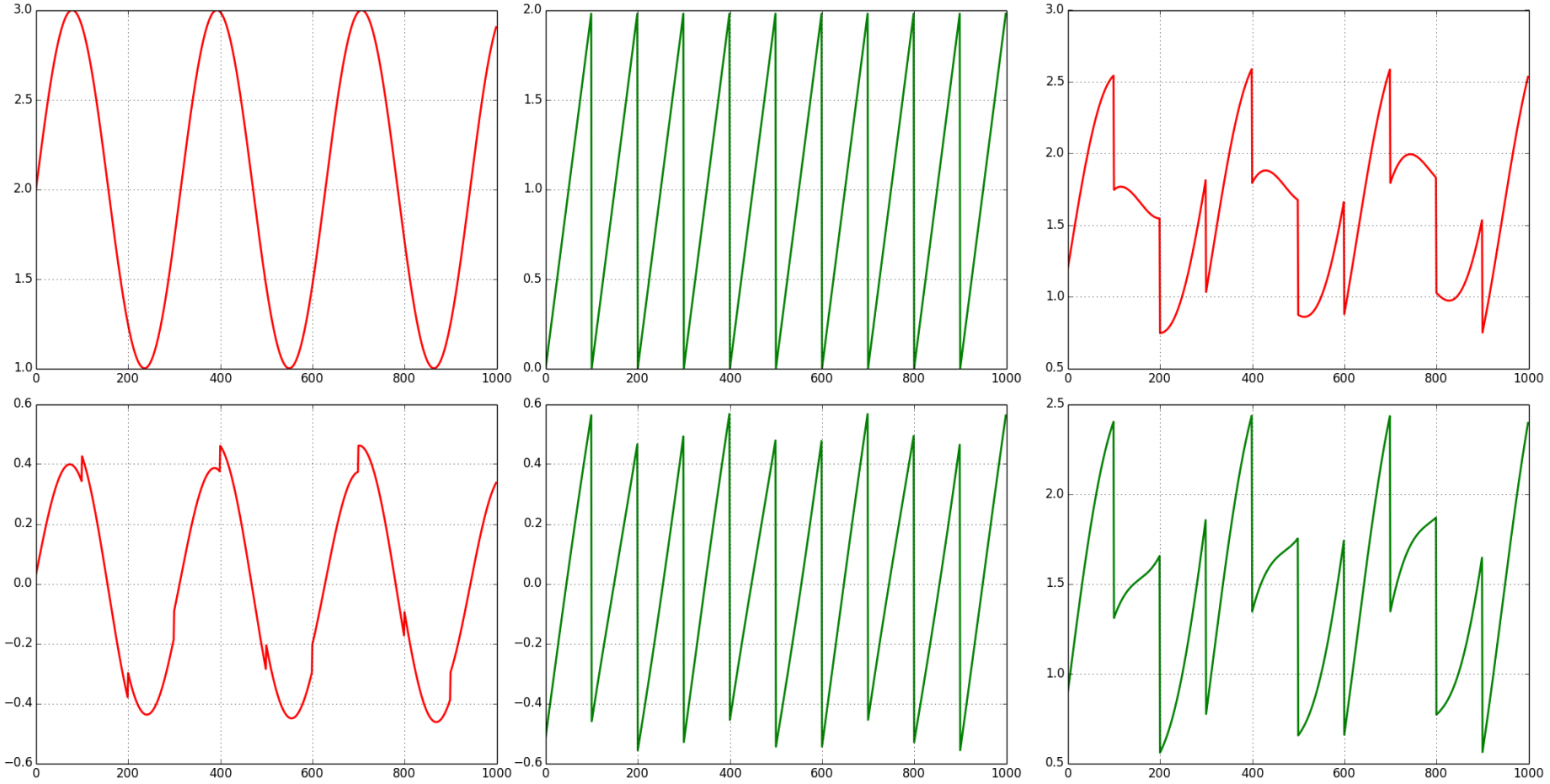
soble.png

soble_x.png

soble_y.png

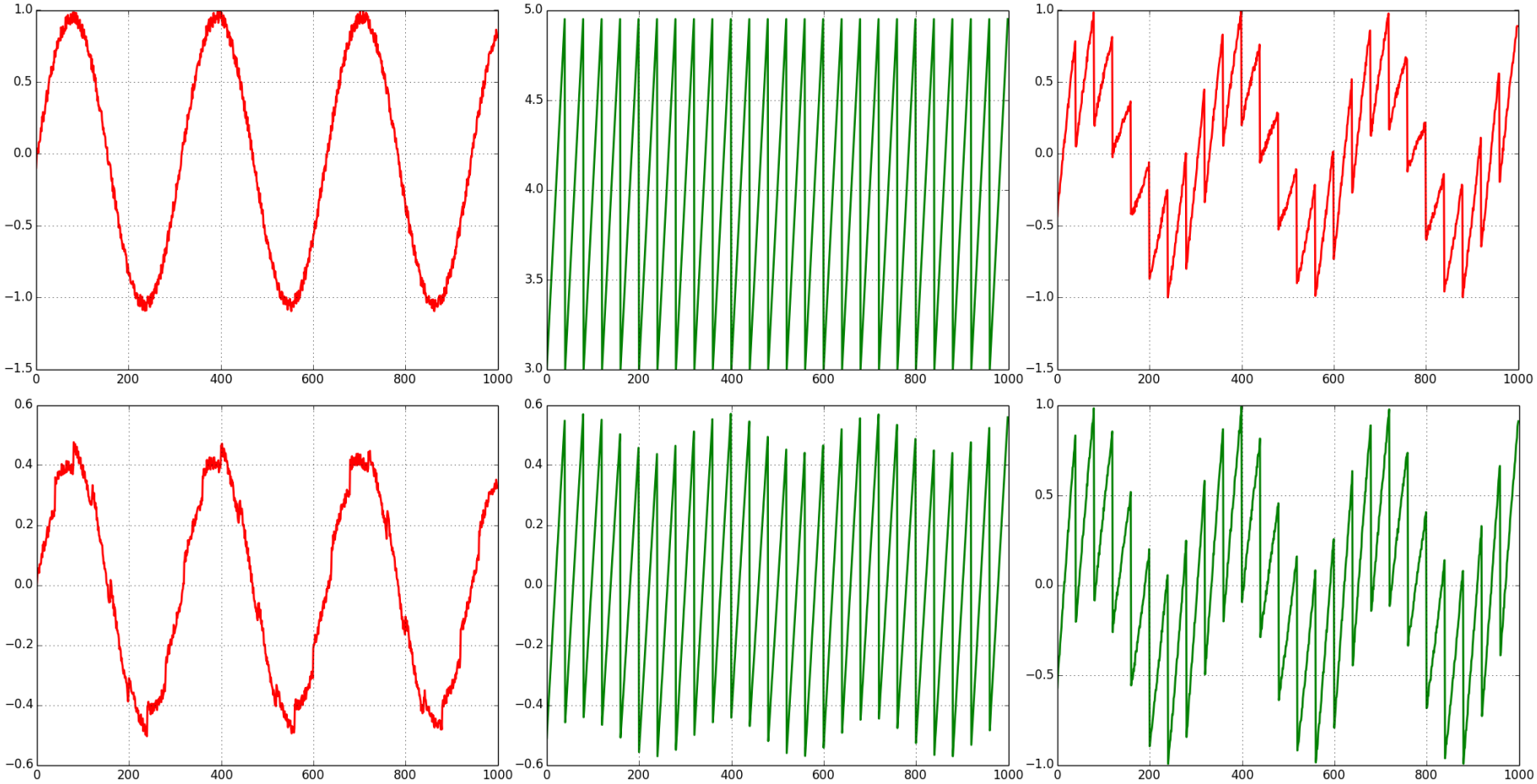
去均值ICA分离

源信号1	源信号2	混合信号1
独立成分1	独立成分2	混合信号2

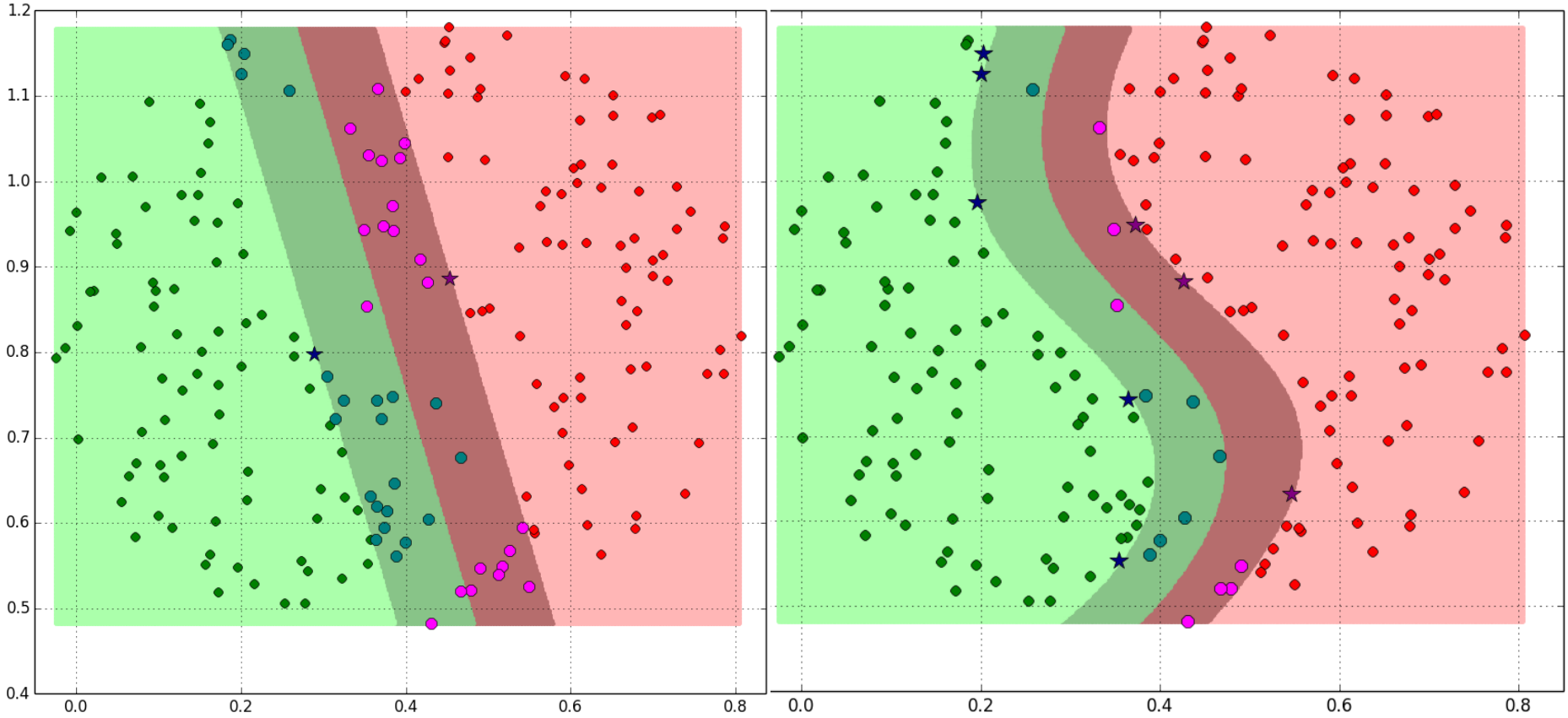


带噪声的信号分离

源信号1	源信号2	混合信号1
独立成分1	独立成分2	混合信号2

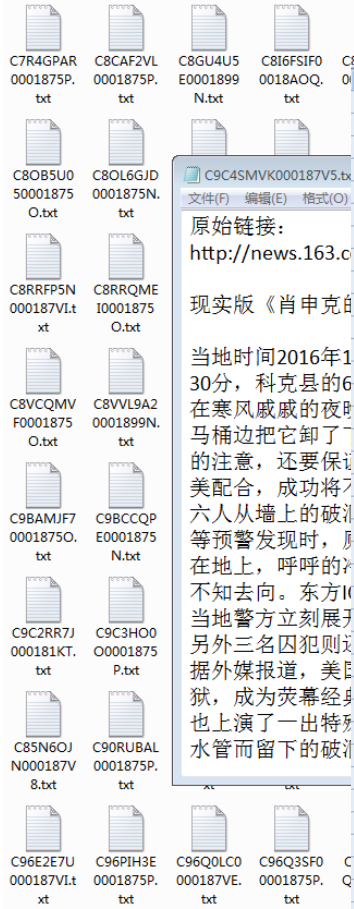


SVM：高斯核函数的影响



Crawler爬取数据

```
writer = csv.writer(f)
writer.writerow(['区域', '小区名称', '户型', '面积', '价格(万)', '单价(元/平米)',
                 '性质', '朝向', '装修', '是否有电梯', '楼层', '建筑年代', '楼型'])
res = requests.get('http://bj.lianjia.com/ershoufang')
res = res.text.encode(res.encoding).decode('utf-8')
soup = BeautifulSoup(res, 'html.parser')
# print soup.prettify()
districts = soup.find(name='div', attrs={'data-role':'ershoufang'}) # <div data-role="ers
for district in districts.find_all(name='a'):
    print district['title']
    district_name = district.text   # '东城', '西城', '朝阳', '海淀'.....
    url = '%s%s' % (url_main, district['href'])
    print url
    res = requests.get(url)
    res = res.text.encode(res.encoding).decode('utf-8')
```



	A	B	C	D	E	F	G	H	I	J	K	L	M
1	区域	小区名称	户型	面积	价格(万)	单价(元/平米)	性质	朝向	装修	是否有电梯	楼层	建筑年代	楼型
2	东城	桃杨路北里12号院	3室1厅	98.17	638	64990	房本满五年	东西	简装	无电梯	中楼层(共6层)	1999	板楼
3	东城	民安小区东扬威街	3室1厅	103.29	1000	96815	房本满五年	西南北	简装	有电梯	低楼层(共16层)	2002	塔楼
4	东城	民安小区东直门内北小街	2室1厅	74.54	732	98203	房本满五年	南北	简装	有电梯	高楼层(共16层)	2003	板楼
5	东城	海运仓小区	2室1厅	64.82	700	107992	房本满五年	南北	精装	无电梯	中楼层(共7层)	2003	板楼
6	东城	新景家园西区	1室1厅	57.18	560	97937	房本满五年	西南	精装	有电梯	中楼层(共17层)	2004	板塔结合
7	东城	小羊宜宾胡同	4室1厅	93.81	910	97005	房本满五年	南北	简装	有电梯	低楼层(共16层)	1992	塔楼
8	东城	禾风相府	3室2厅	163.83	1801	109932	房本满五年	南北	其他	有电梯	低楼层(共14层)	2008	板塔结合
9	东城	西革新里110号院	2室1厅	61.4	400	65147	房本满五年	南北	简装	无电梯	高楼层(共6层)	1992	板楼
10	东城	东中街	4室1厅	200.77	1770	88161	房本满五年	南北	精装	有电梯	中楼层(共9层)	1999	板塔结合
11	东城	东花市北里中区	3室2厅	98.65	990	100355	房本满五年	南北	精装	无电梯	中楼层(共5层)	1994	板楼
12	东城	民安小区东直门北中街	2室1厅	75.11	722	96126	房本满五年	东西	精装	有电梯	中楼层(共11层)	2002	板塔结合
13	东城	中景濠庭	1室1厅	74.21	630	84895	房本满五年	北	精装	有电梯	低楼层(共22层)	2001	板塔结合
14	东城	华府景园	3室2厅	149.97	1350	90019	房本满五年	东西	简装	有电梯	中楼层(共12层)	2001	板塔结合
15	东城	朝阳门南小街	2室1厅	69.39	650	93674	房本满五年	东西	简装	有电梯	低楼层(共8层)	2004	板塔结合
16	东城	兴隆都市馨园	3室2厅	130.7	980	74981	房本满五年	南北	精装	无电梯	高楼层(共6层)	2003	板塔结合
17	东城	保利蔷薇	2室2厅	88.63	960	108316	房本满五年	西北	精装	有电梯	中楼层(共16层)	2008	板塔结合
18	东城	小羊宜宾胡同	2室1厅	61.77	650	105230	房本满五年	南	简装	有电梯	低楼层(共16层)	1992	塔楼
19	东城	安化北里	3室1厅	80.34	730	90864	房本满五年	西南	简装	有电梯	中楼层(共17层)	1989	塔楼
20	东城	香饵胡同	2室1厅	67.92	815	119995	房本满五年	南北	精装	无电梯	低楼层(共6层)	2002	板楼
21	东城	什锦花园胡同21号院	2室1厅	48	650	135417	房本满五年	南北	精装	无电梯	高楼层(共5层)	1985	板楼
22	东城	富贵园一区	3室2厅	144.45	1650	114227	房本满五年	南北	精装	有电梯	低楼层(共11层)	2003	板楼
23	东城	富莱茵花园	3室1厅	98.38	570	57939	房本满五年	南西北	简装	有电梯	中楼层(共21层)	1995	塔楼
24	东城	丽水湾畔家园	2室1厅	130	1100	84616	房本满五年	西	简装	有电梯	高楼层(共21层)	2002	塔楼
25	东城	国瑞城中区	3室1厅	90.47	880	97270	房本满五年	西南	精装	有电梯	低楼层(共16层)	2006	板塔结合
26	东城	金鱼池中区	4室1厅	139.34	850	61002	房本满五年	南北	精装	无电梯	低楼层(共6层)	2002	板楼
27	东城	新景家园东区	3室1厅	92.82	1030	110968	房本满五年	南	精装	有电梯	高楼层(共13层)	2002	板塔结合
28	东城	本家润园三期	3室1厅	97.45	930	95434	房本满五年	南西	精装	有电梯	高楼层(共18层)	2005	板塔结合
29	东城	新景家园东区	1室1厅	59.68	595	99699	房本满五年	南	简装	有电梯	中楼层(共17层)	2002	板楼
30	东城	北京上舍	1室0厅	61.2	580	94772	房本满五年	北	简装	有电梯	高楼层(共15层)	2008	塔楼
31	东城	和平里七区	2室1厅	41.82	450	107605	房本满五年	南	简装	无电梯	高楼层(共5层)	1983	板楼
32	东城	东土城路13号院	1室0厅	51.15	372	72728	房本满两年	南	精装	无电梯	中楼层(共5层)	1994	板楼
33	东城	金世纪嘉园	2室1厅	116.89	950	81273	房本满五年	东南	精装	有电梯	高楼层(共21层)	2003	塔楼
34	东城	民安小区民安街	2室1厅	74.29	868	116840	房本满五年	南北	精装	无	低楼层(共6层)	2002	板楼
35	东城	金鱼池中区	4室2厅	135.91	950	69900	房本满五年	南北	简装	无电梯	高楼层(共6层)	2002	板塔结合
36	东城	海晟名苑	2室1厅	140.53	1550	110297	房本满五年	东西	简装	有电梯	低楼层(共20层)	2003	板塔结合
37	东城	京城仁合	2室2厅	114.51	1010	88202	房本满五年	西南北	精装	有电梯	高楼层(共16层)	2002	板塔结合
38	东城	西水井胡同	2室1厅	74.95	680	90728	房本满五年	东西	精装	有电梯	中楼层(共16层)	2003	板塔结合
39	东城	新奥洋房	2室1厅	93.76	698	74446	房本满五年	东	精装	有电梯	中楼层(共11层)	2005	板塔结合
40	东城	西水井胡同	2室1厅	74.95	690	92062	房本满五年	东西	简装	有电梯	中楼层(共16层)	2003	板塔结合
41	东城	仓南胡同	2室1厅	77.8	920	118252	房本满五年	东西	简装	无电梯	中楼层(共6层)	2003	板楼
42	东城	新奥洋房	2室1厅	123.21	910	73858	房本满五年	东南西	精装	有电梯	高楼层(共13层)	2006	板塔结合
43	东城	北河沿大街	1室1厅	41.47	490	118158	房本满两年	南	简装	无电梯	高楼层(共6层)	1987	板楼
44	东城	东直门内大街	2室2厅	80.52	880	109290	房本满五年	东南	精装	有电梯	低楼层(共16层)	2003	塔楼

HMM分词：MLE

```
if __name__ == "__main__":
    pi, A, B = load_train()
    f = file(".\\text\\novel.txt")
    data = f.read()[3:].decode('utf-8')
    f.close()
    decode = viterbi(pi, A, B, data)
    segment(data, decode)
```

bug HMM

Console | Frames | Variables | Watches

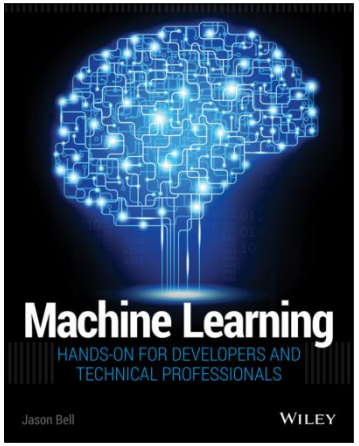
Connected to pydev debugger (build 139.1001)

我 | 与 | 地坛 |
史 | 铁生 |
一 |

我 | 在 | 好 | 几 | 篇 | 小说 | 中 | 都 | 提到 | 过 | 一 | 座 | 废弃 | 的 | 古园 | , | 实际 | 就 | 是 | 地坛 | 。 | 许多 | 年前 | 旅游业 | 还 | 没有 | 开
地坛 | 离 | 我家 | 很 | 近 | 。 | 或者 | 说 | 我家 | 离 | 地坛 | 很 | 近 | 。 | 总之 | , | 只好 | 认为 | 这 | 是 | 缘分 | 。 | 地坛 | 在 | 我出 | 生前 |
它 | 等待 | 我 | 出生 | , | 然后 | 又 | 等待 | 我活 | 到 | 最狂 | 妄 | 的 | 年龄 | 上 | 忽地 | 残废 | 了 | 双腿 | 。 | 四百 | 多 | 年里 | , | 它 | 一面
自 | 从 | 那个 | 下午 | 我 | 无意 | 中进 | 了 | 这 | 园子 | , | 就 | 再 | 没长 | 久 | 地离 | 开过 | 它 | 。 | 我 | 一 | 下子 | 就 | 理解 | 了 | 它 | 的 |
两 | 条 | 腿 | 残废 | 后 | 的 | 最初 | 几年 | , | 我找 | 不 | 到 | 工作 | , | 找 | 不到 | 去路 | , | 忽然 | 间 | 几乎 | 什么 | 都 | 找 | 不到 | 了 | , |
除 | 去 | 几 | 座 | 殿堂 | 我 | 无法 | 进去 | , | 除去 | 那 | 座 | 祭坛 | 我 | 不 | 能 | 上去 | 而 | 只能 | 从 | 各个 | 角度 | 张望 | 它 | , | 地坛 | 的
剩 | 下 | 的 | 就 | 是 | 怎样活 | 的 | 问题 | 了 | , | 这 | 却 | 不 | 是 | 在 | 某 | 一个 | 瞬间 | 就 | 能 | 完全 | 想透 | 的 | 、 | 不 | 是 | 一次性 |
二 |

我 | 才 | 想到 | , | 当年 | 我 | 总是 | 独自 | 跑 | 到 | 地坛 | 去 | , | 曾 | 经给 | 母亲 | 出 | 了 | 一个 | 怎样 | 的 | 难题 | 。 |
她 | 不 | 是 | 那种 | 光会 | 疼爱 | 儿子 | 而 | 不 | 懂得 | 理解 | 儿子 | 的 | 母亲 | 。 | 她 | 知道 | 我心 | 里 | 的 | 苦闷 | , | 知道 | 不 | 该 | 阻止
有 | 一 | 回 | 我摇 | 车出 | 了 | 小院 | ; | 想 | 起 | 一件 | 什么 | 事 | 又 | 返身 | 回来 | , | 看见 | 母亲 | 仍站 | 在 | 原地 | , | 还 | 是 | 送 | 我
有 | 一次 | 与 | 一个 | 作家 | 朋友 | 聊天 | , | 我问 | 他学 | 写作 | 的 | 最 | 初动 | 机是 | 什么 | ? | 他想 | 了 | 一会 | 说 | : | "为 | 我 | 母亲 | 。
在 | 我 | 的 | 头 | 一 | 篇 | 小 | 说 | 发表 | 的 | 时候 | , | 在 | 我 | 的 | 小 | 说 | 第一 | 次 | 获奖 | 的 | 那些 | 日子 | 里 | , | 我真 | 是 | 多么
只 | 是 | 到 | 了 | 这 | 时候 | , | 纷纭 | 的 | 往事 | 才 | 在 | 我 | 眼前 | 幻现 | 得 | 清晰 | , | 母亲 | 的 | 苦难 | 与 | 伟大 | 才 | 在 | 我 | 心 |
摇 | 着 | 轮椅 | 在 | 园中 | 慢慢 | 走 | , | 又 | 是 | 雾霭 | 的 | 清晨 | , | 又 | 是 | 骄阳 | 高 | 悬 | 的 | 白昼 | , | 我 | 只想 | 着 | 一件 | 事 | :
曾 | 有 | 过好 | 多 | 回 | , | 我 | 在 | 这园 | 子里 | 呆得 | 太久 | 了 | , | 母亲 | 就 | 来找 | 我 | 。 | 她 | 来找 | 我 | 又 | 不想 | 让 | 我 | 发觉 |

前言 |
数据 | , | 数据 | , | 数据 | ! | 想 | 必在 | 新闻 | 、 | 报刊 | 等 | 媒介 | 的 | 持续 | 冲击 | 下 | , | 人们 | 无法 | 摆 脱| 大 | 的 | 洗礼 | 。 | 现实 | 需求 | 推动 | 了 | 对 | 数据 | 的 | 学习 | 这些 | 数据 | 来 | 自于 | 社交 | 媒体 | 、 | 智能 | 手机 | 、 | “ | 物联网 | ”| ） | 、 | 传感器 | 等 | 任何 | 可以 | 产生 | 数 | 大 | 多 | 数数 | 据 | 挖掘 | 的 | 宣传 | 着 | 重于 | 数据 | 规模 | 数据 | 洪 | 水(| data flood) | 的 | 预言 | 告诉 | 人们 | 我们 | 无法 | 实时 | 处理 | 这些 | 数据 | , | 硬件 | 推销 | 人员 | 会进 | 一步 | 卖 | 给 | 我们 | 需要 | 的 | 服务 | , | 以期 | 能够 | 满足 | 处理 | 速度 | 的 | 要求 | 。 | 从 | 某种 | 程度 | 上来 | 说 | , | 他们 | 是 | 对 | 的 | , | 但 | 是 | 我们 | 值得 | 停下 | 来 | 思考 | 片刻 | , | 并 | 对 | 手边 | 的 | 任务 | 进行 | 适当 | 的 | 再 | 认识 | 。 |
近 | 年来 | , | 数据 | 挖掘 | 和 | 机器 | 学习 | 在 | 我们 | 周围 | 持续 | 火爆 | , | 各种 | 媒体 | 也 | 不断 | 推送 | 着 | 海量 | 的 | 数据 | 。 | 仔细 | 观察 | 就 | 能 | 发现 | , | 实际 | 应用 | 中 | 的 | 那些 | 机器 | 学习 | 算法 | 与 | 多 | 年前 | 并 | 没有 | 什么 | 两样 | ; | 它们 | 只 | 是 | 在 | 应用 | 的 | 数据 | 规模 | 上 | 有些 | 不同 | 。 | 历数 | 一 | 下 | 产生 | 数据 | 的 | 组织 | , | 至少 | 在 | 我 | 看来 | , | 数目 | 其实 | 并 | 不 | 多 | 。 | 无非 | 是 | Google | 、 | Facebook | 、 | Twitter | 、 | NetFlix | 以及 | 其 | 他 | 为数 | 不 | 多 | 的 | 机构 | 在 | 使用 | 若 | 干学 | 习算法 | 和 | 工具 | , | 这些 | 算法 | 和 | 工具 | 使 | 得 | 他们 | 能够 | 对 | 数据 | 进行 | 测试 | 分析 | 。 | 那么 | , | 真正 | 的 | 问题 | 是 | : | “ | 对于 | 其 | 他人 | , | 大数 | 据 | 框架 | 下 | 的 | 算法 | 和 | 工具 | 的 | 作用 | 是 | 什么 | 呢 | ? | ”|
我承认 | 本书 | 将 | 多 | 次 | 提及 | 大 | 数据 | 和 | 机器 | 学习 | 之间 | 的 | 关系 | , | 这 | 是 | 我 | 无法 | 忽视 | 的 | 一个 | 客观 | 问题 | ; | 但 | 是 | 它 | 只 | 是 | 一个 | 很 | 小 | 的 | 因素 | , | 终极 | 目标 | 是 | 如何 | 利用 | 可用 | 数据 | 获取 | 数据 | 的 | 本质



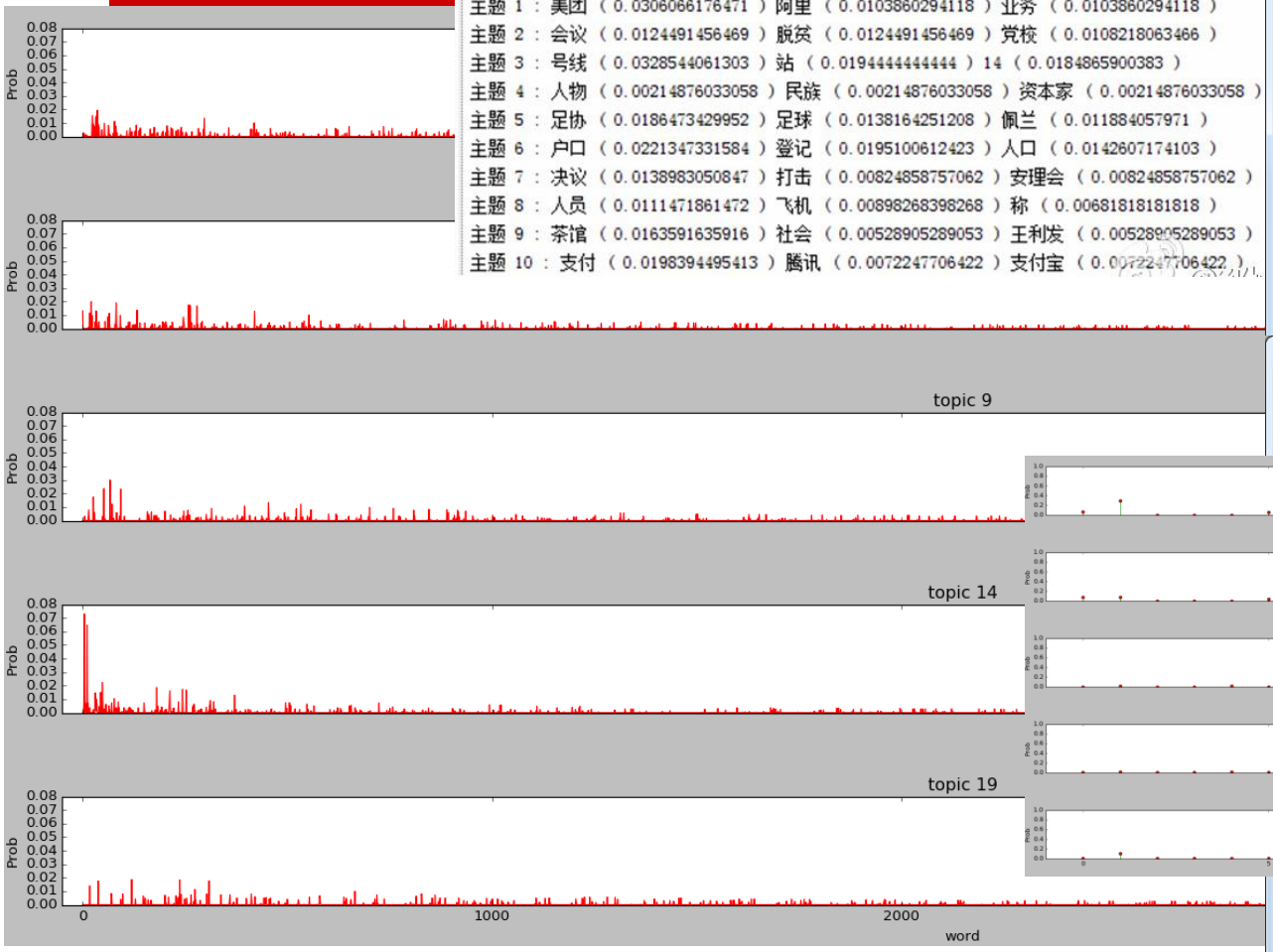
Jason Bell. *Machine Learning: Hands-On for Developers and Technical Professionals*. Wiley.2015

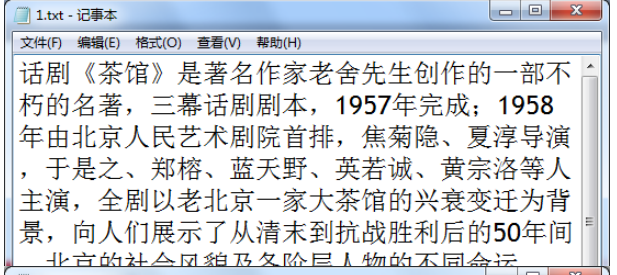
LDA

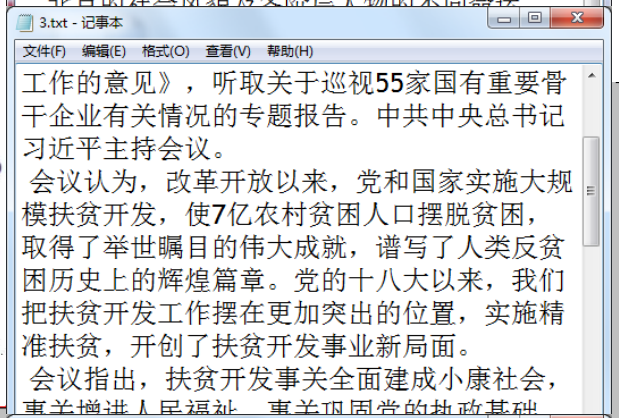
文档 1 ： 茶馆 （ 0.0163591635916 ） 社会 （ 0.00528905289053 ） 王利发 （ 0.00528905289053 ）
文档 2 ： 决议 （ 0.0138983050847 ） 打击 （ 0.00824858757062 ） 安理会 （ 0.00824858757062 ）
文档 3 ： 会议 （ 0.0124491456469 ） 脱贫 （ 0.0124491456469 ） 党校 （ 0.0108218063466 ）
文档 4 ： 美团 （ 0.0306066176471 ） 阿里 （ 0.0103860294118 ） 业务 （ 0.0103860294118 ）
文档 5 ： 户口 （ 0.0221347331584 ） 登记 （ 0.0195100612423 ） 人口 （ 0.0142607174103 ）
文档 6 ： 人员 （ 0.0111471861472 ） 飞机 （ 0.00898268398268 ） 称 （ 0.00681818181818 ）
文档 7 ： 号线 （ 0.0328544061303 ） 站 （ 0.0194444444444 ） 14 （ 0.0184865900383 ）
文档 8 ： 支付 （ 0.0198394495413 ） 腾讯 （ 0.0072247706422 ） 支付宝 （ 0.0072247706422 ）
文档 9 ： 决议 （ 0.0138983050847 ） 打击 （ 0.00824858757062 ） 安理会 （ 0.00824858757062 ）
文档 10 ： 足协 （ 0.0186473429952 ） 足球 （ 0.0138164251208 ） 佩兰 （ 0.011884057971 ）

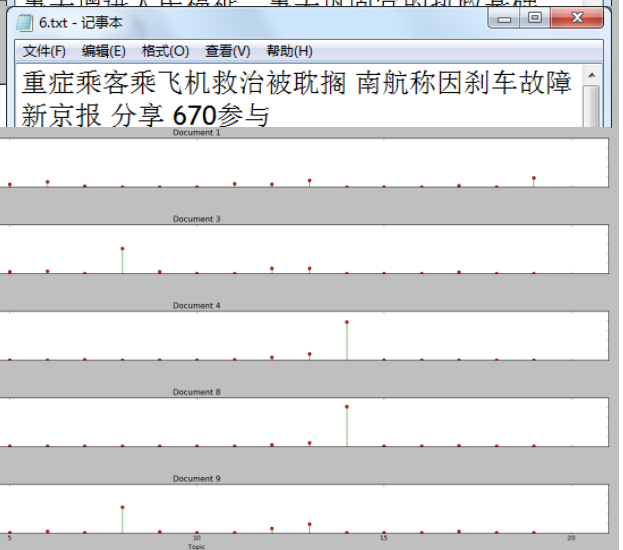
==========

主题 1 ： 美团 （ 0.0306066176471 ） 阿里 （ 0.0103860294118 ） 业务 （ 0.0103860294118 ）
主题 2 ： 会议 （ 0.0124491456469 ） 脱贫 （ 0.0124491456469 ） 党校 （ 0.0108218063466 ）
主题 3 ： 号线 （ 0.0328544061303 ） 站 （ 0.0194444444444 ） 14 （ 0.0184865900383 ）
主题 4 ： 人物 （ 0.00214876033058 ） 民族 （ 0.00214876033058 ） 资本家 （ 0.00214876033058 ）
主题 5 ： 足协 （ 0.0186473429952 ） 足球 （ 0.0138164251208 ） 佩兰 （ 0.011884057971 ）
主题 6 ： 户口 （ 0.0221347331584 ） 登记 （ 0.0195100612423 ） 人口 （ 0.0142607174103 ）
主题 7 ： 决议 （ 0.0138983050847 ） 打击 （ 0.00824858757062 ） 安理会 （ 0.00824858757062 ）
主题 8 ： 人员 （ 0.0111471861472 ） 飞机 （ 0.00898268398268 ） 称 （ 0.00681818181818 ）
主题 9 ： 茶馆 （ 0.0163591635916 ） 社会 （ 0.00528905289053 ） 王利发 （ 0.00528905289053 ）
主题 10 ： 支付 （ 0.0198394495413 ） 腾讯 （ 0.0072247706422 ） 支付宝 （ 0.0072247706422 ）



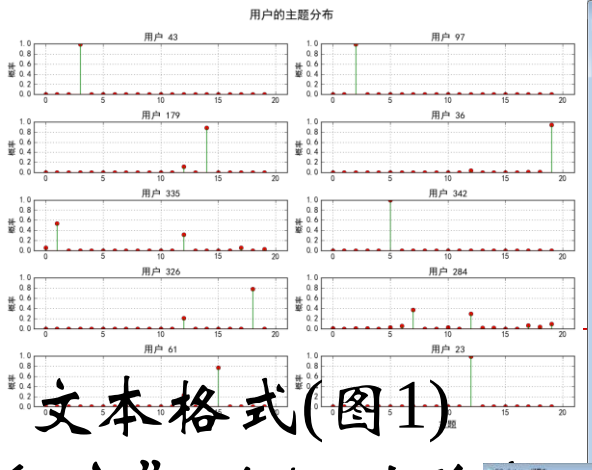


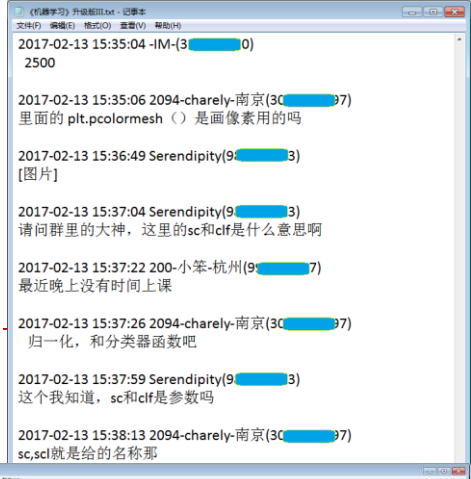


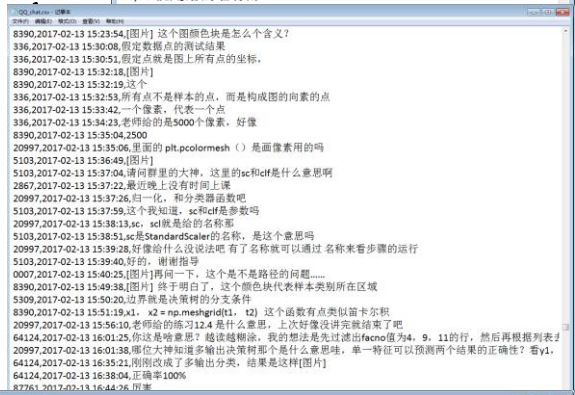


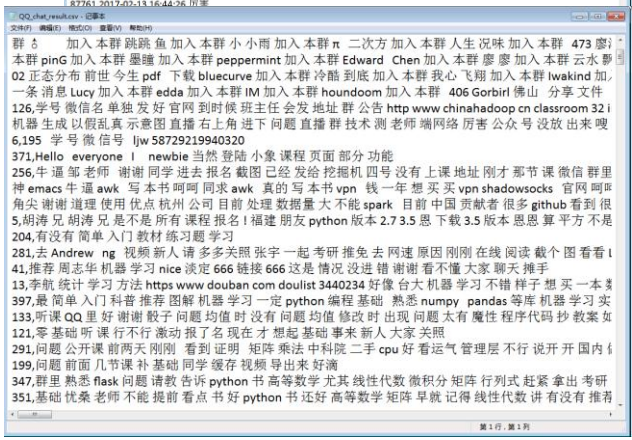
舆情

- 获取QQ群聊天记录：文本格式(图1)
- 整理成“QQ号/时间/留言”的规则形
  - 正则表达式
  - 清洗特定词：表情、@XX
  - 使用停止词库
  - 获得csv表格数据(图2)
- 合并相同QQ号的留言
  - 长文档利于计算每人感兴趣话题(图3)
- LDA模型计算主题
  - 调参与可视化
- 计算每个QQ号及众人感兴趣话题



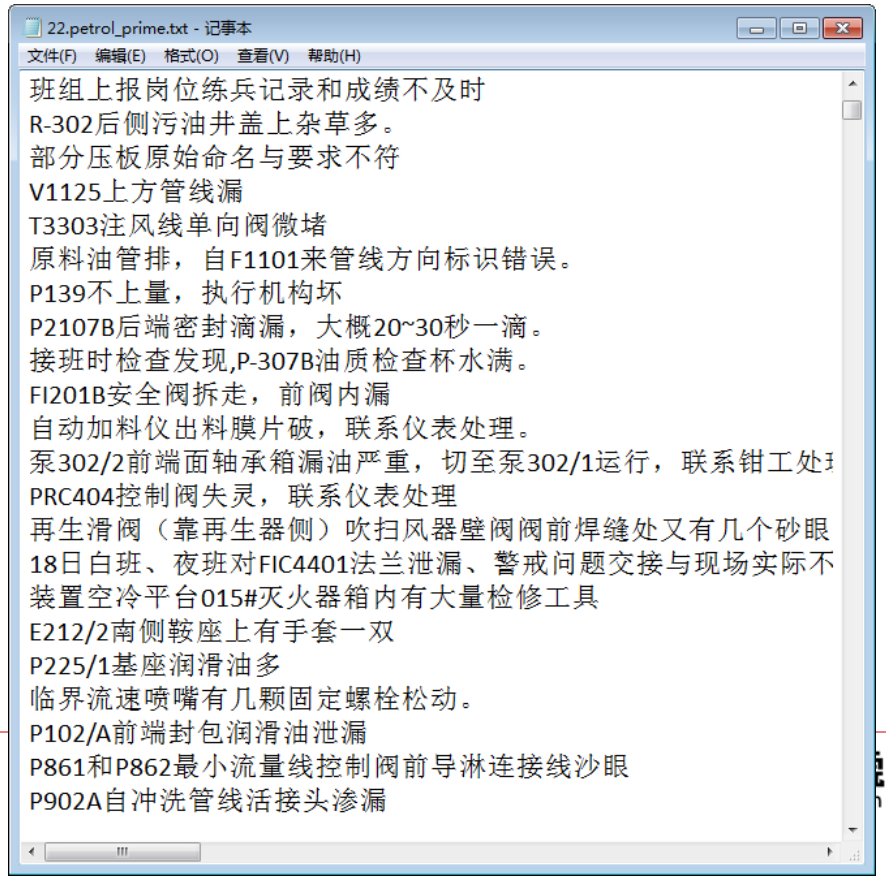




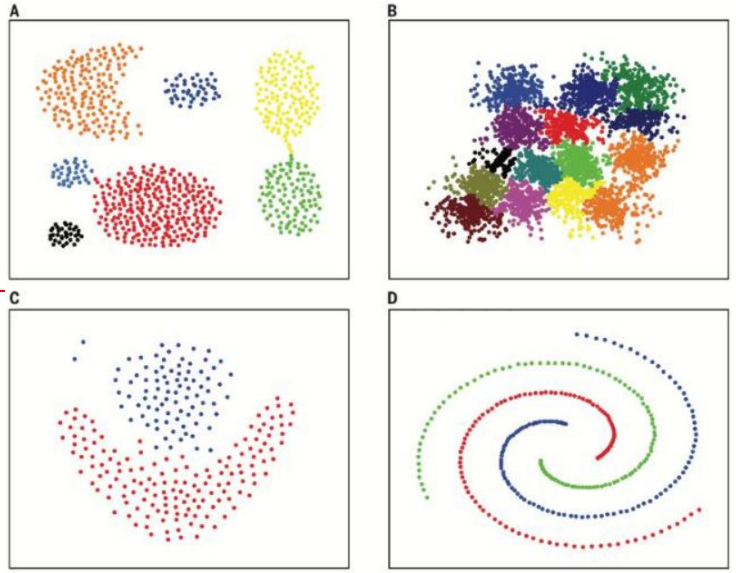


例：石油例检结果处理

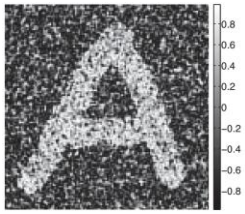
- 针对国内某石油企业的例行检查处理结果，试通过主题模型方案，分析例检结果中最突出的问题是什么？
  - 文本共4700个，
  - 单个文档十数字

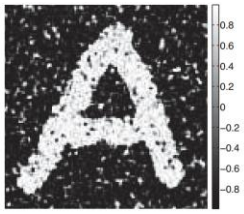


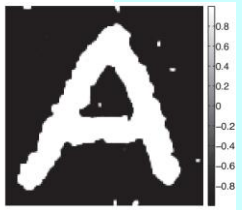
其他内容



- 最大熵模型
  - 自然语言处理解决标记问题
- 聚类
  - K-means/K-Mediods/密度聚类/谱聚类
- 降维
  - PCA/SVD/ICA
- SVM
  - 与核技术相结合
- 主题模型pLSA/LDA
  - 与聚类、标签传递算法相结合
- 条件随机场
  - 无向图模型，链式条件随机场解决标记问题
- 变分推导Variation Inference
  - 与EM、贝叶斯相结合，参数、隐变量的学习
- 深度学习
  - 大规模人工神经网络







本课程参考文献(部分)

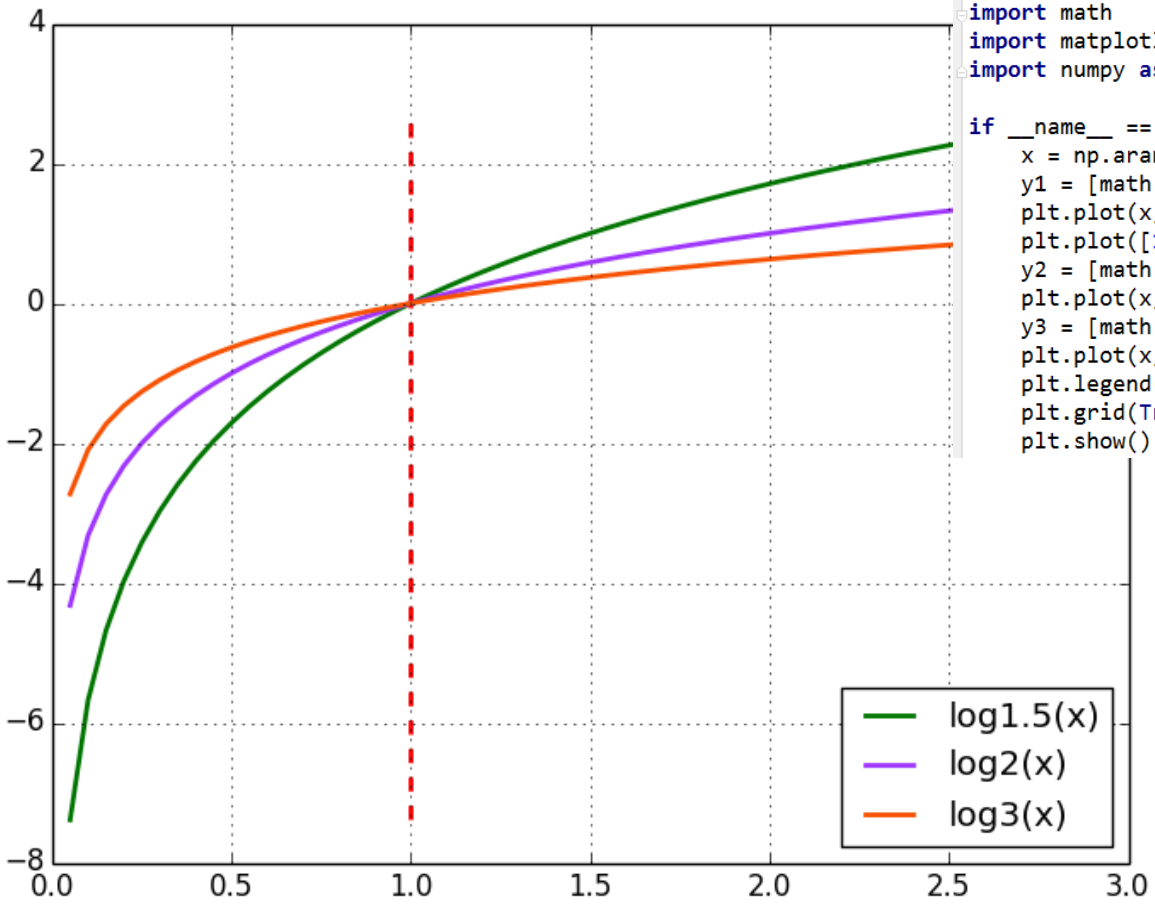
- Christopher M. Bishop, Pattern Recognition and Machine Learning, Springer-Verlag, 2006
- Kevin P. Murphy, Machine Learning:A Probabilistic Perspective, The MIT Press, 2012
- 李航，统计学习方法，清华大学出版社，2012
- Stephen Boyd,Lieven Vandenberghe, Convex Optimization, Cambridge University Press, 2004
- Thomas M. Cover, Joy A. Thomas, Elements of Information Theory,2006
- 各章节特定的经典论文，如：
  - Alex Rodriguez, Alessandro Laio, Clustering by fast search and find of density peak, Science 344.6191(2014)
  - David M. Blei, Andrew Y. Ng, Michael I. Jordan, Latent Dirichlet Allocation, 2003

回忆知识

- 求S的值：

$$S = \frac{1}{0!} + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \frac{1}{4!} + \cdots + \frac{1}{n!} + \cdots$$

对数函数的上升速度




```python
import math
import matplotlib.pyplot as plt
import numpy as np

if __name__ == "__main__":
    x = np.arange(0.05, 3, 0.05)
    y1 = [math.log(a, 1.5) for a in x]
    plt.plot(x, y1, linewidth=2, color='#007500', label='log1.5(x)')
    plt.plot([1, 1], [y1[0], y1[-1]], "r--", linewidth=2)
    y2 = [math.log(a, 2) for a in x]
    plt.plot(x, y2, linewidth=2, color='#9F35FF', label='log2(x)')
    y3 = [math.log(a, 3) for a in x]
    plt.plot(x, y3, linewidth=2, color='#F75000', label='log3(x)')
    plt.legend(loc='lower right')
    plt.grid(True)
    plt.show()
```

问题分析

- 令 $f(x)=\log_a x$
- 则：

$$\frac{f(x+\Delta x)-f(x)}{\Delta x}=\frac{\log_a(x+\Delta x)-\log_a x}{\Delta x}=\frac{\log_a\left(\frac{x+\Delta x}{x}\right)}{\Delta x}=\log_a\left(\frac{x+\Delta x}{x}\right)^{\frac{1}{\Delta x}}$$

$$\xrightarrow{\because x=1}\log_a(1+\Delta x)^{\frac{1}{\Delta x}}\overset{令}{==}1\Rightarrow\lim_{\Delta x\to 0}(1+\Delta x)^{\frac{1}{\Delta x}}=a$$

- 问：$\lim_{n\to\infty}\left(1+\frac{1}{n}\right)^n=?$

附：构造数列{x_n}

$$x_n = \left(1+\frac{1}{n}\right)^n$$

$$= 1 + C_n^1 \frac{1}{n} + C_n^2 \frac{1}{n^2} + C_n^3 \frac{1}{n^3} + \cdots + C_n^n \frac{1}{n^n}$$

$$= 1 + n\cdot\frac{1}{n} + \frac{n(n-1)}{2!}\cdot\frac{1}{n^2} + \frac{n(n-1)(n-2)}{3!}\cdot\frac{1}{n^3} + \cdots + \frac{n(n-1)(n-2)\cdots 1}{n!}\cdot\frac{1}{n^n}$$

$$= 1+1+\frac{1}{2!}\cdot\left(1-\frac{1}{n}\right) + \frac{1}{3!}\cdot\left(1-\frac{1}{n}\right)\left(1-\frac{2}{n}\right) + \cdots + \frac{1}{n!}\cdot\left(1-\frac{1}{n}\right)\left(1-\frac{2}{n}\right)\cdots\left(1-\frac{n-1}{n}\right)$$

$$< 1+1+\frac{1}{2!}+\frac{1}{3!}+\cdots+\frac{1}{n!}$$

$$< 1+1+\frac{1}{2}+\frac{1}{2^2}+\cdots+\frac{1}{2^{n-1}}$$

$$= 3-\frac{1}{2^{n-1}}$$

$$< 3$$

附：自然常数 $\lim\limits_{x\to\infty}\left(1+\frac{1}{x}\right)^x=e$

- 根据前文中 $a_n=\left(1+\frac{1}{n}\right)^n$ 的二项展开式，已经证明数组{a_n}单增有上界，因此，必有极限，记做e。
- 同时：

$$\left(1+\frac{1}{n+1}\right)^n<\left(1+\frac{1}{x}\right)^x<\left(1+\frac{1}{n}\right)^{n+1}$$

$$\lim_{n\to\infty}\left(1+\frac{1}{n+1}\right)^n=\lim_{n\to\infty}\frac{\left(1+\frac{1}{n+1}\right)^{n+1}}{1+\frac{1}{n+1}}=\frac{\lim\limits_{n\to\infty}\left(1+\frac{1}{n+1}\right)^{n+1}}{\lim\limits_{n\to\infty}\left(1+\frac{1}{n+1}\right)}=\frac{e}{1+0}=e$$

$$\lim_{n\to\infty}\left(1+\frac{1}{n}\right)^{n+1}=\lim_{n\to\infty}\left(\left(1+\frac{1}{n}\right)^n\left(1+\frac{1}{n}\right)\right)=\lim_{n\to\infty}\left(1+\frac{1}{n}\right)^n\cdot\lim_{n\to\infty}\left(1+\frac{1}{n}\right)=e\cdot(1+0)=e$$

- 根据两边夹定理，函数 $f(x)=\left(1+\frac{1}{x}\right)^x$ 的极限存在，为e.

导数

- 简单的说，导数就是曲线的斜率，是曲线变化快慢的反应
- 二阶导数是斜率变化快慢的反应，表征曲线凸凹性
  - 二阶导数连续的曲线，往往称之为“光顺”的。
  - 还记得高中物理老师时常念叨的吗：加速度的方向总是指向轨迹曲线凹的一侧。
- 根据 $\lim_{x\to\infty}\left(1+\frac{1}{x}\right)^{x}=e$ 可以得到函数 $f(x)=\ln x$ 的导数，进一步根据换底公式、反函数求导等，得到其他初等函数的导数。

常用函数的导数

$$C' = 0 \qquad (x^n)' = nx^{n-1}$$

$$(\sin x)' = \cos x \qquad (\cos x)' = -\sin x$$

$$(a^x)' = a^x \ln a \qquad (e^x)' = e^x$$

$$(\log_a x)' = \frac{1}{x}\log_a e \qquad (\ln x)' = \frac{1}{x}$$

$$(u+v)' = u' + v' \qquad (uv)' = u'v + uv'$$

应用1

- 已知函数 $f(x)=x^x, x>0$
- 求f(x)的最小值
  - 领会幂指函数的一般处理套路
- 附：$N^{\frac{1}{\log_2 N}}=?$
  - 在计算机算法跳跃表Skip List的分析中，用到了该常数。
  - 背景：跳表是支持增删改查的动态数据结构，能够达到与平衡二叉树、红黑树近似的效率，而代码实现简单。

求解x^x

$$t = x^x$$

$$\rightarrow \ln t = x \ln x$$

$$\xrightarrow{\text{两边对x求导}} \frac{1}{t} t' = \ln x + 1$$

$$\xrightarrow{\text{令}t'=0} \ln x + 1 = 0$$

$$\rightarrow x = e^{-1}$$

$$\rightarrow t = e^{-\frac{1}{e}}$$

积分应用2：$N \to \infty \Rightarrow \ln N! \to N(\ln N - 1)$

$$\ln N! = \sum_{i=1}^{N} \ln i \approx \int_1^N \ln x dx$$

$$= x \ln x \Big|_1^N - \int_1^N x d \ln x$$

$$= N \ln N - \int_1^N x \cdot \frac{1}{x} dx$$

$$= N \ln N - x \Big|_1^N$$

$$= N \ln N - N + 1$$

$$\to N \ln N - N$$

Taylor公式 – Maclaurin公式

$$f(x)=f(x_0)+f'(x_0)(x-x_0)+\frac{f''(x_0)}{2!}(x-x_0)^2+\cdots+\frac{f^{(n)}(x_0)}{n!}(x-x_0)^n+R_n(x)$$

$$f(x)=f(0)+f'(0)x+\frac{f''(0)}{2!}x^2+\cdots+\frac{f^{(n)}(0)}{n!}x^n+o(x^n)$$

Taylor公式的应用1

- 数值计算：初等函数值的计算(在原点展开)

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!} + \cdots + (-1)^{m-1}\frac{x^{2m-1}}{(2m-1)!} + R_{2m}$$

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots + \frac{x^n}{n!} + R_n$$

- 在实践中，往往需要做一定程度的变换。

Taylor公式的应用1：计算e^x

- 给定正实数x，计算e^x=?
- 一种可行的思路：
- 求整数k和小数r，使得
  - $x = k*\ln 2 + r,\ |r| \le 0.5*\ln 2$
- 从而：

$$e^x = e^{k\cdot\ln 2 + r}$$

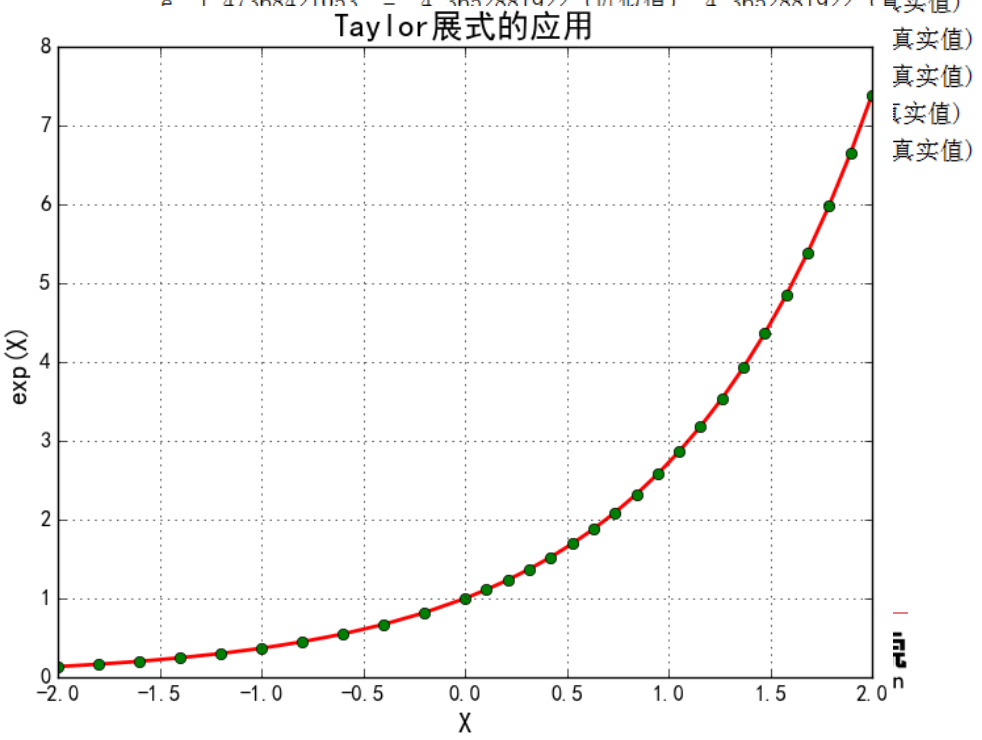
$$= e^{k\cdot\ln 2} \cdot e^r$$

$$= 2^k \cdot e^r$$

Taylor展式的应用

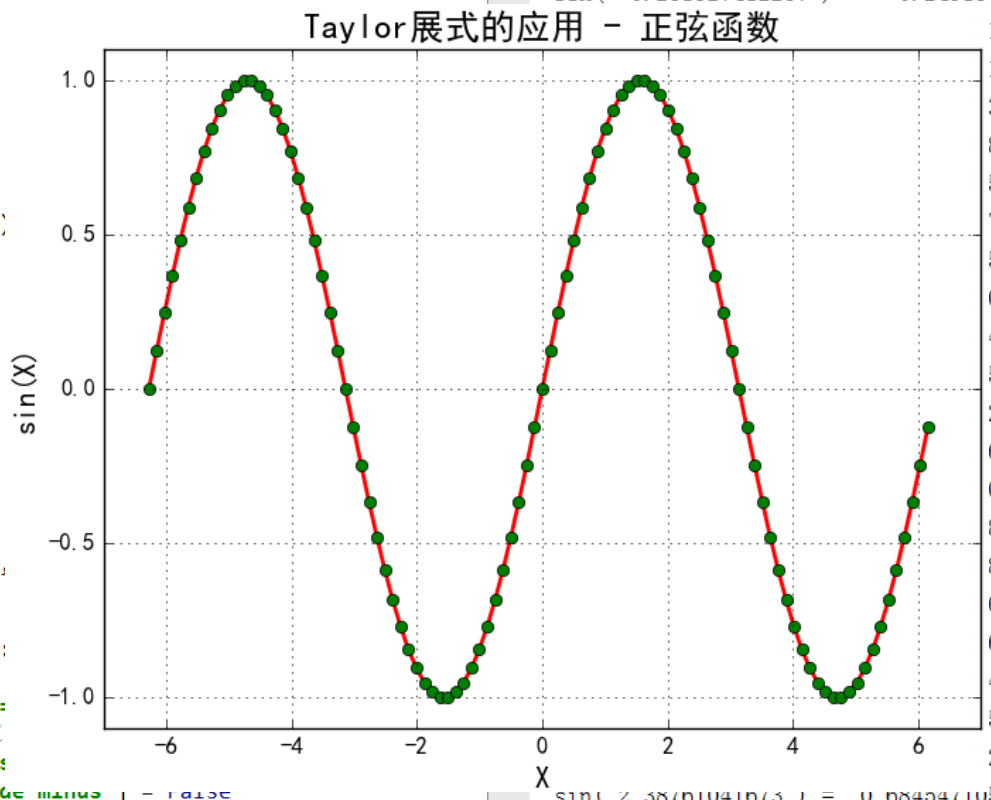
```
e^ -0.8  =  0.449328964117 (近似值) 0.449328964117 (真实值)
e^ -0.6  =  0.548811636094 (近似值) 0.548811636094 (真实值)
e^ -0.4  =  0.670320046036 (近似值) 0.670320046036 (真实值)
e^ -0.2  =  0.818730753078 (近似值) 0.818730753078 (真实值)
e^ 0.0  =  1.0 (近似值) 1.0 (真实值)
e^ 0.105263157895  =  1.11100294108 (近似值)  1.11100294108 (真实值)
e^ 0.210526315789  =  1.2343275351 (近似值) 1.2343275351 (真实值)
e^ 0.315789473684  =  1.37134152176 (近似值)  1.37134152176 (真实值)
e^ 0.421052631579  =  1.5235644639 (近似值) 1.5235644639 (真实值)
e^ 0.526315789474  =  1.69268460033 (近似值)  1.69268460033 (真实值)
e^ 0.631578947368  =  1.88057756929 (近似值)  1.88057756929 (真实值)
e^ 0.736842105263  =  2.08932721042 (近似值)  2.08932721042 (真实值)
e^ 0.842105263158  =  2.32124867566 (近似值)  2.32124867566 (真实值)
e^ 0.947368421053  =  2.57891410565 (近似值)  2.57891410565 (真实值)
e^ 1.05263157895  =  2.86518115618 (近似值) 2.86518115618 (真实值)
e^ 1.15789473684  =  3.18322469126 (近似值) 3.18322469126 (真实值)
e^ 1.26315789474  =  3.53657199412 (近似值) 3.53657199412 (真实值)
e^ 1.36842105263  =  3.92914188683 (近似值) 3.92914188683 (真实值)
e^ 1.47368421053  =  4.3652881922 (近似值)  4.3652881922 (真实值)
```

```python
def calc_e_small(x):
    n = 10
    f = np.arange(1, n+1).cumprod()
    b = np.array([x]*n).cumprod()
    return np.sum(b / f) + 1


def calc_e(x):
    reverse = False
    if x < 0:   # 处理负数
        x = -x
        reverse = True
    ln2 = 0.69314718055994530941723212145818
    c = x / ln2
    a = int(c+0.5)
    b = x - a*ln2
    y = (2 ** a) * calc_e_small(b)
    if reverse:
        return 1/y
    return y


if __name__ == "__main__":
    t1 = np.linspace(-2, 0, 10, endpoint=False)
    t2 = np.linspace(0, 2, 20)
    t = np.concatenate((t1, t2))
    print t     # 横轴数据
    y = np.empty_like(t)
    for i, x in enumerate(t):
        y[i] = calc_e(x)
        print 'e^', x, ' = ', y[i], '(近似值)\t', math.exp(x)
        # print '误差: ', y[i] - math.exp(x)
    mpl.rcParams['font.sans-serif'] = [u'SimHei']
    mpl.rcParams['axes.unicode_minus'] = False
    plt.plot(t, y, 'r-', t, y, 'go', linewidth=2)
    plt.title(u'Taylor展式的应用', fontsize=18)
    plt.xlabel('X', fontsize=15)
    plt.ylabel('exp(X)', fontsize=15)
    plt.grid(True)
    plt.show()
```



Taylor展式的应用

```
def calc_sin_small(x):
    x2 = -x ** 2
    t = x
    f = 1
    sum = 0
    for i in range(10):
        sum += t / f
        t *= x2
        f *= ((2*i+2)*(2*i+3)
    return sum


def calc_sin(x):
    a = x / (2*np.pi)
    k = np.floor(a)
    a = x - k*2*np.pi
    return calc_sin_small(a)


if __name__ == "__main__":
    t = np.linspace(-2*np.pi,
    print t      # 横轴数据
    y = np.empty_like(t)
    for i, x in enumerate(t):
        y[i] = calc_sin(x)
        print 'sin(', x, ') =
        # print '误差: ', y[i
    mpl.rcParams['font.sans-s
    mpl.rcParams['axes.unicode_minus'] = False
    plt.figure(facecolor='w')
    plt.plot(t, y, 'r-', t, y, 'go', linewidth=2)
    plt.title(u'Taylor展式的应用 - 正弦函数', fontsize=18)
    plt.xlabel('X', fontsize=15)
    plt.ylabel('sin(X)', fontsize=15)
    plt.xlim((-7, 7))
    plt.ylim((-1.1, 1.1))
    plt.grid(True)
    plt.show()
```



5.2.calc_sin

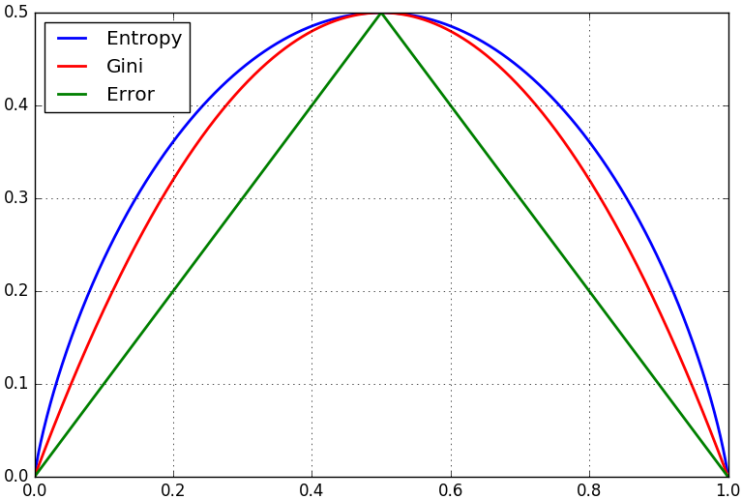
```
sin( -1.25663706144 ) =  -0.951066447544 (近似值)   -0.951056516295 (真实值)
sin( -1.13097335529 ) =  -0.904843692145 (近似值)   -0.904827052466 (真实值)
sin( -1.00530964915 ) =  -0.844355457307 (近似值)   -0.844327925502 (真实值)
sin( -0.879645943005 ) =  -0.770558254809 (近似值)  -0.770513242776 (真实值)
sin( -0.753982236862 ) =  -0.684619861245 (近似值)  -0.684547105929 (真实值)
sin( -0.628318530718 ) =  -0.587901575106 (近似值)  -0.587785252292 (真实值)
sin( -0.502654824574 ) =  -0.481937724358 (近似值)  -0.481753674102 (真实值)
sin( -0.376991118431 ) =  -0.368412871668 (近似值)  -0.368124552685 (真实值)
sin( -0.251327412287 ) =  -0.249137247033 (近似值)  -0.248689887165 (真实值)
                          1027336 (近似值)  -0.125333233564 (真实值)
                          197e-16 (近似值)  8.881784197e-16 (真实值)
                          33564 (近似值)  0.125333233564 (真实值)
                          87165 (近似值)  0.248689887165 (真实值)
                          52685 (近似值)  0.368124552685 (真实值)
                          74102 (近似值)  0.481753674102 (真实值)
                          52292 (近似值)  0.587785252292 (真实值)
                          05929 (近似值)  0.684547105929 (真实值)
                          42776 (近似值)  0.770513242776 (真实值)
                          5502 (近似值)  0.844327925502 (真实值)
                          2466 (近似值)  0.904827052466 (真实值)
                          6295 (近似值)  0.951056516295 (真实值)
                          0729 (近似值)  0.982287250729 (真实值)
                          8428 (近似值)  0.998026728428 (真实值)
                          8428 (近似值)  0.998026728428 (真实值)
                          0729 (近似值)  0.982287250729 (真实值)
                          6295 (近似值)  0.951056516295 (真实值)
                          466 (近似值)  0.904827052466 (真实值)
                          5502 (近似值)  0.844327925502 (真实值)
                          2775 (近似值)  0.770513242776 (真实值)
sin( 2.38761041673 ) =  0.684547105927 (近似值)  0.684547105929 (真实值)
sin( 2.51327412287 ) =  0.587785252288 (近似值)  0.587785252292 (真实值)
sin( 2.63893782902 ) =  0.481753674088 (近似值)  0.481753674102 (真实值)
sin( 2.76460153516 ) =  0.368124552648 (近似值)  0.368124552685 (真实值)
```

Taylor公式的应用2

- 考察Gini系数的图像、熵、分类误差率三者之间的关系
  - 将f(x)=-lnx在x=1处一阶展开，忽略高阶无穷小，得到f(x)≈1-x

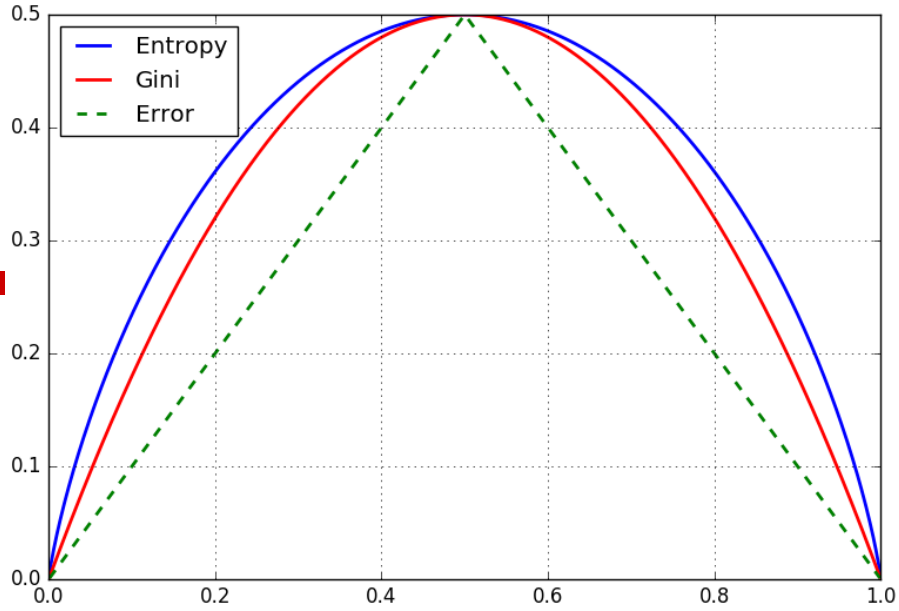
$$H(X)=-\sum_{k=1}^{K} p_k \ln p_k$$

$$\approx \sum_{k=1}^{K} p_k (1-p_k)$$



  - 上述结论，在决策树章节中会进一步讨论

Gini系数的生成




```
import numpy as np
import matplotlib.pyplot as plt


if __name__ == "__main__":
    p = np.arange(0.001, 1, 0.001, dtype=np.float)
    gini = 2 * p * (1-p)
    h = -(p * np.log2(p) + (1-p) * np.log2(1-p))/2
    err = 1 - np.max(np.vstack((p, 1-p)), 0)
    plt.plot(p, h, 'b-', linewidth=2, label='Entropy')
    plt.plot(p, gini, 'r-', linewidth=2, label='Gini')
    plt.plot(p, err, 'g-', linewidth=2, label='Error')
    plt.grid(True)
    plt.legend(loc='upper left')
    plt.show()
```

方向导数

- 如果函数z=f(x,y)在点P(x,y)是可微分的，那么，函数在该点沿任一方向L的方向导数都存在，且有：

$$\frac{\partial f}{\partial l}=\frac{\partial f}{\partial x}\cos\varphi+\frac{\partial f}{\partial y}\sin\varphi$$

- 其中，ψ为x轴到方向L的转角。

梯度

- 设函数z=f(x,y)在平面区域D内具有一阶连续偏导数，则对于每一个点P(x,y)∈D，向量

$$\left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right)$$

为函数z=f(x,y)在点P的梯度，记做**grad**f(x,y)

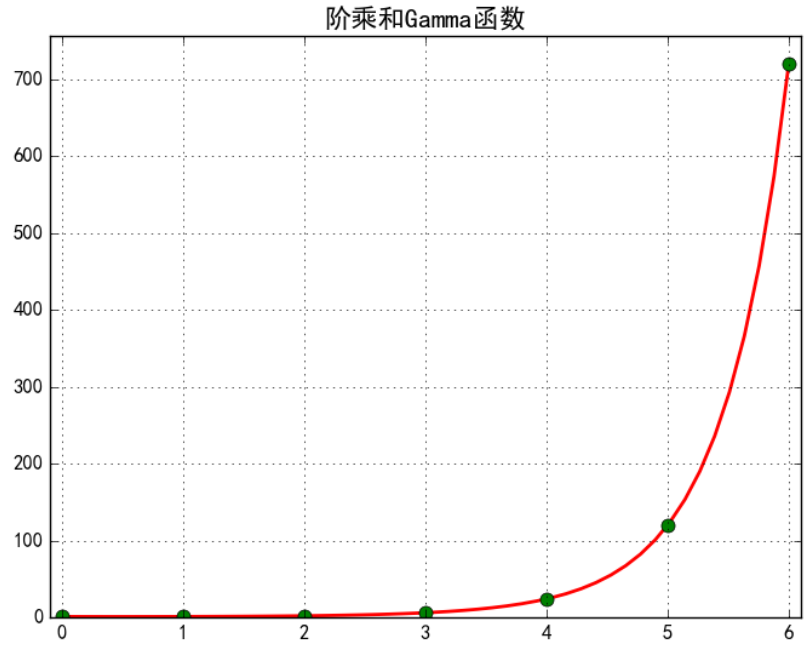
- 梯度的方向是函数在该点变化最快的方向
  - 考虑一座解析式为z=H(x,y)的山，在(x_0,y_0)的梯度是在该点坡度变化最快的方向。
- 梯度下降法
  - 思考：若下山方向和梯度呈θ夹角，下降速度是多少？

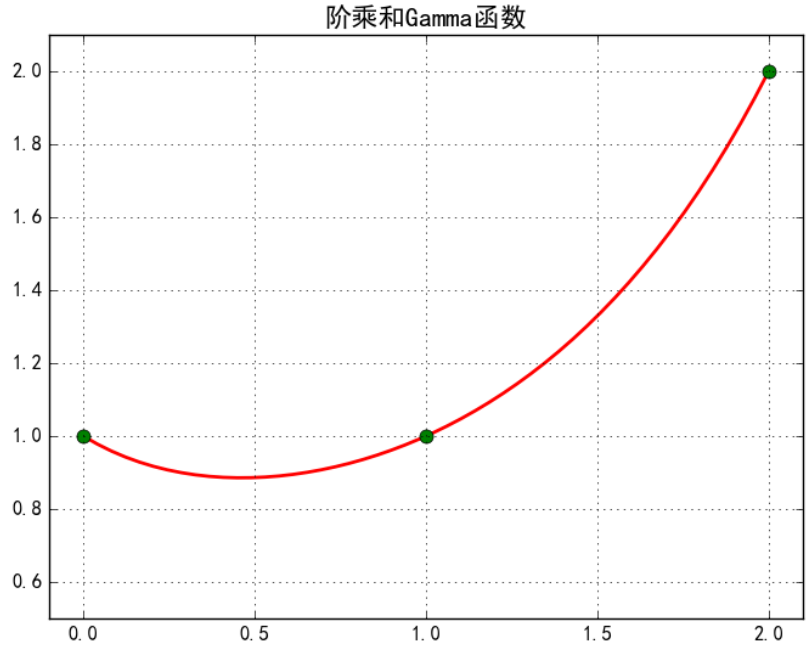
Γ函数

$$\Gamma(x)=(x-1)\cdot\Gamma(x-1)\Rightarrow\frac{\Gamma(x)}{\Gamma(x-1)}=x-1$$

- Γ函数是阶乘在实数上的推广

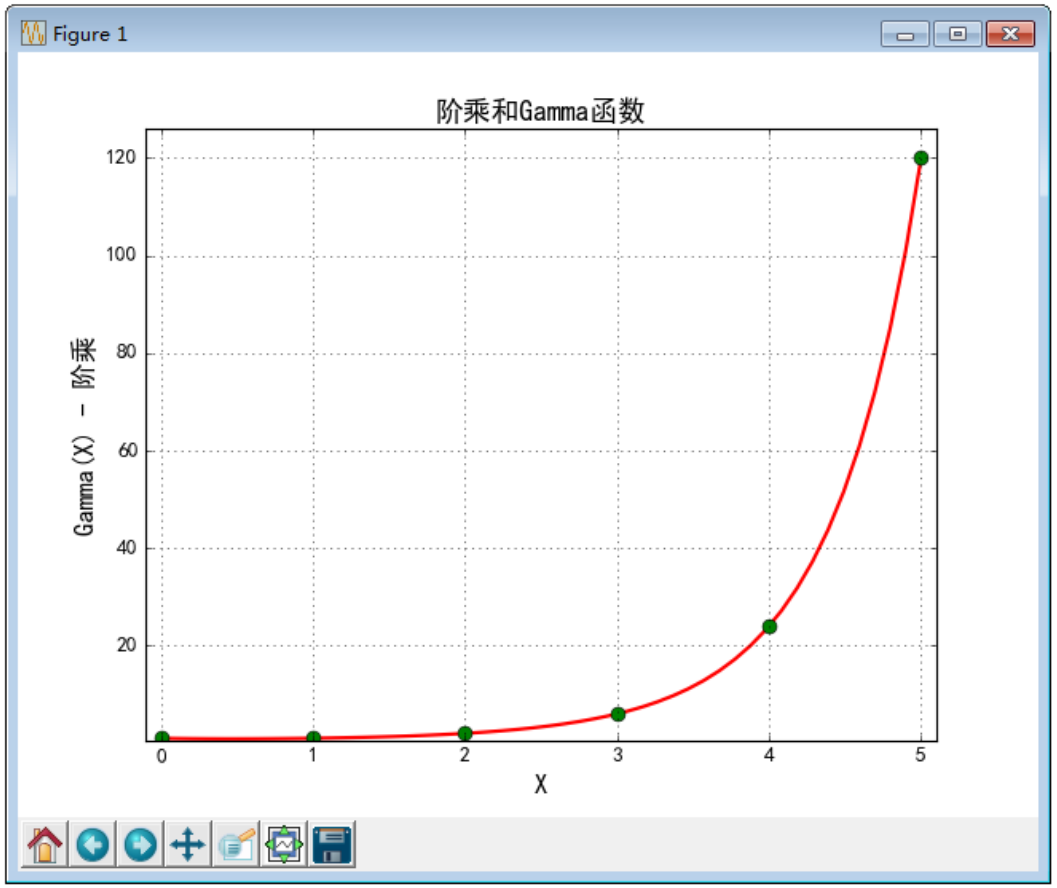
$$\Gamma(x)=\int_0^{+\infty}t^{x-1}e^{-t}dt=(x-1)!$$





Code

```python
import numpy as np
import matplotlib as mpl
import matplotlib.pyplot as plt
from scipy.special import gamma
from scipy.special import factorial

mpl.rcParams['axes.unicode_minus'] = False
mpl.rcParams['font.sans-serif'] = 'SimHei'


if __name__ == '__main__':
    N = 5
    x = np.linspace(0, N, 50)
    y = gamma(x+1)
    plt.figure(facecolor='w')
    plt.plot(x, y, 'r-', lw=2)
    z = np.arange(0, N+1)
    f = factorial(z, exact=True)    # 阶乘
    print f
    plt.plot(z, f, 'go', markersize=8)
    plt.grid(b=True)
    plt.xlim(-0.1,N+0.1)
    plt.ylim(0.5, np.max(y)*1.05)
    plt.xlabel(u'X', fontsize=15)
    plt.ylabel(u'Gamma(X) - 阶乘', fontsize=15)
    plt.title(u'阶乘和Gamma函数', fontsize=16)
    plt.show()
```



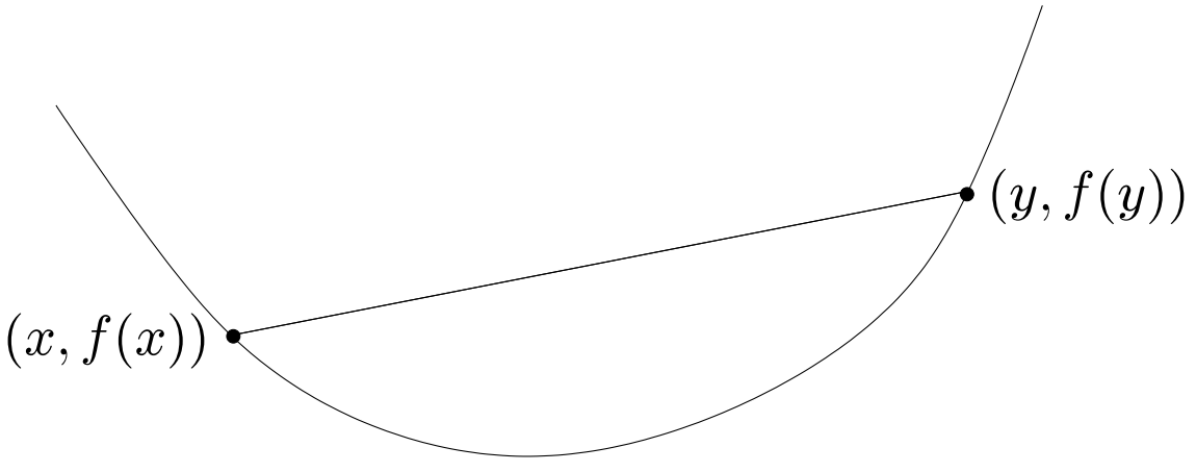

```
5.7.gamma
C:\Python27\python.exe D:/Python/ML/5.Python/5.7.gamma.py
[  1   1   2   6  24 120]
```

凸函数

- 若函数f的定义域domf为凸集，且满足

$$\forall x, y \in \mathrm{dom}\, f, 0 \le \theta \le 1, \text{ 有}$$

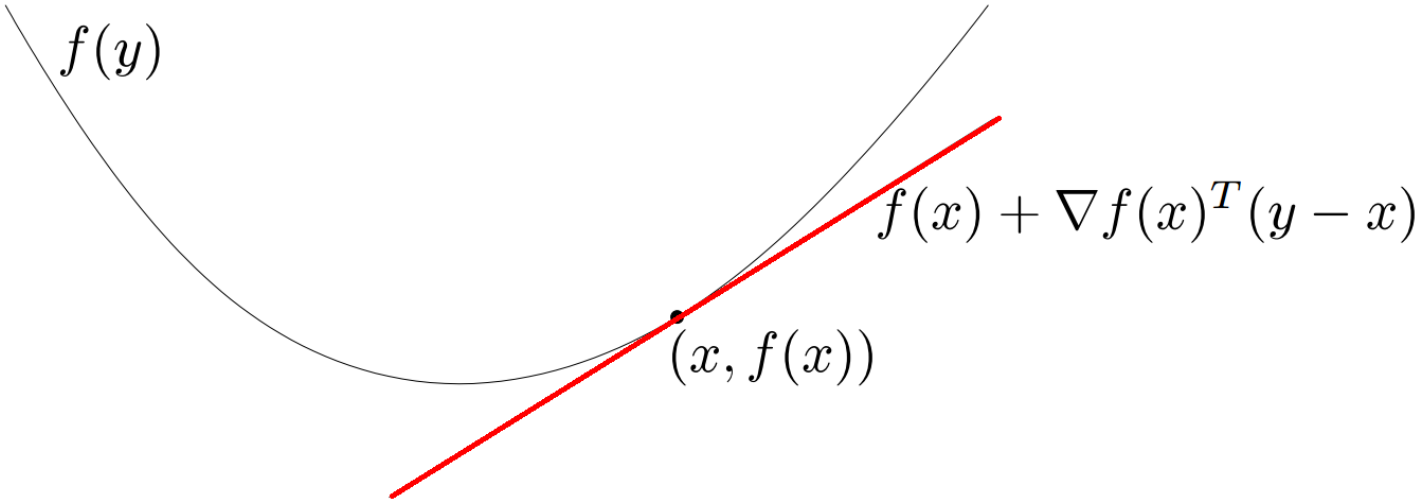
$$f(\theta x + (1-\theta) y) \le \theta f(x) + (1-\theta) f(y)$$



一阶可微

- 若f一阶可微，则函数f为凸函数当前仅当f的定义域domf为凸集，且

$$\forall x, y \in \mathrm{dom} f,\ f(y) \geq f(x) + \nabla f(x)^T (y - x)$$



二阶可微

- 若函数f二阶可微，则函数f为凸函数当前仅当dom为凸集，且

$$\nabla^2 f(x) \succ= 0$$

- 若f是一元函数，上式表示二阶导大于等于0
- 若f是多元函数，上式表示二阶导Hessian矩阵半正定。

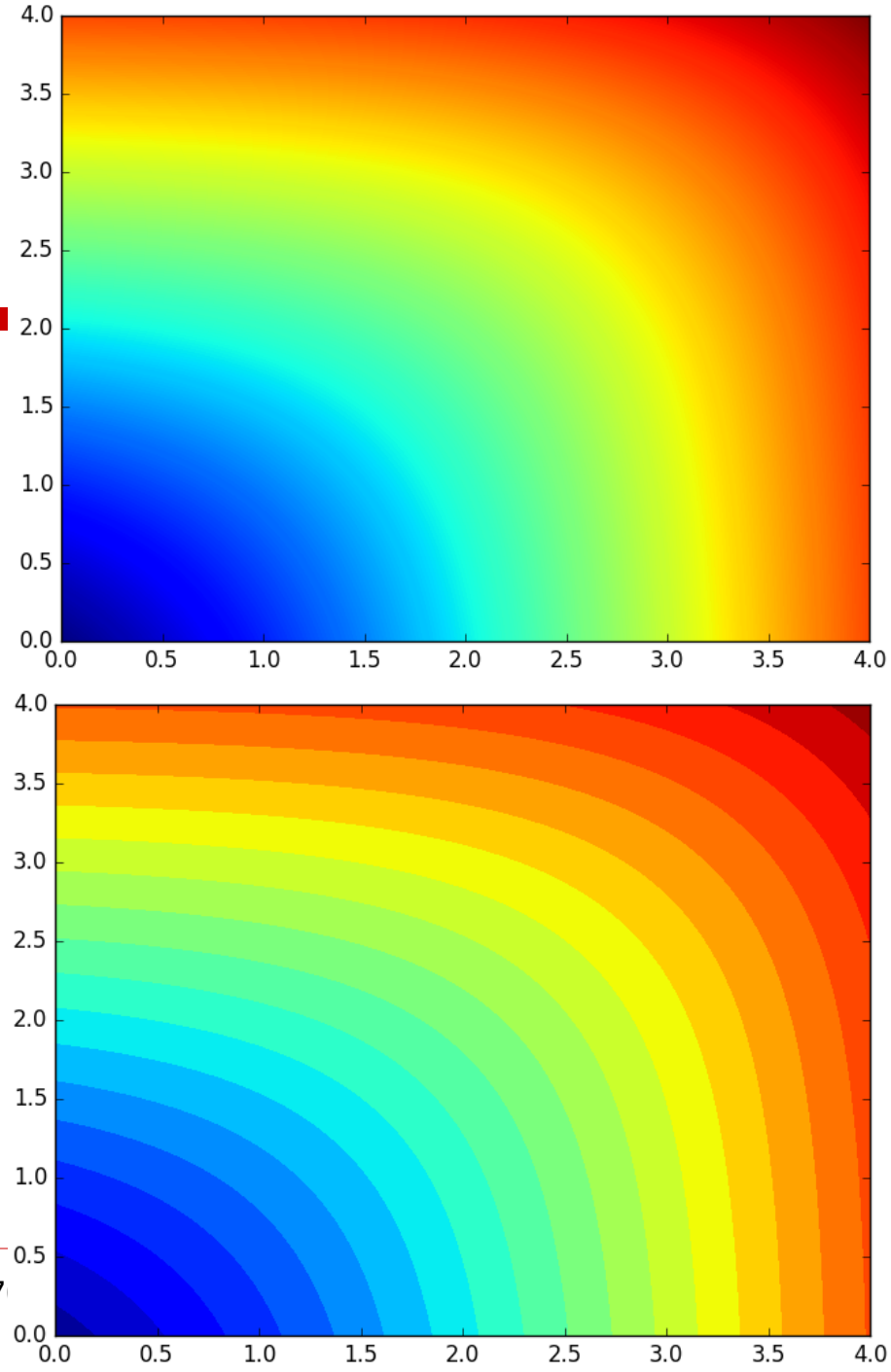
凸函数举例

- 指数函数 $f(x)=e^{ax}$
- 幂函数 $f(x)=x^a,\ x\in R^+, a\geq 1$或$a\leq 0$
- 负对数函数 $f(x)=-\ln x$
- 负熵函数 $f(x)=x\ln x$
- 范数函数 $f(\vec{x})=\|x\|$
- 最大值函数 $f(\vec{x})=\max(x_1,x_2,\cdots x_n)$
- 指数线性函数 $f(\vec{x})=\log\left(e^{x_1}+e^{x_2}+\cdots+e^{x_n}\right)$
  - Log Sum Exp→max(x1,x2…xn)

Code

$$f(x_1, x_2) = \log\left(e^{x_1} + e^{x_2}\right)$$

```
if __name__ == "__main__":
    fig = plt.figure()
    ax = fig.add_subplot(111)
    u = np.linspace(0, 4, 1000)
    x, y = np.meshgrid(u, u)
    z = np.log(np.exp(x) + np.exp(y))
    ax.contourf(x, y, z, 20)
    plt.show()
```



概率论

- 对概率的认识：$P(x) \in [0,1]$
  - P=0：事件出现的概率为0→事件不会发生？
  - 若x为离散/连续变量，则P(x=x0)表示x0发生的概率/概率密度
- 累计分布函数：$\Phi(x)=P(x \leq x0)$
  - $\Phi(x)$一定为单增函数
  - $\min(\Phi(x))=0$， $\max(\Phi(x))=1$
- 思考：将值域为[0,1]的某单增函数y=F(x)看成X事件的累积概率函数
  - 若y=F(x)可导，则f(x)=F'(x)为某概率密度函数
- P.S.
  - cumulative distribution function,CDF/Probability Density Function,pdf

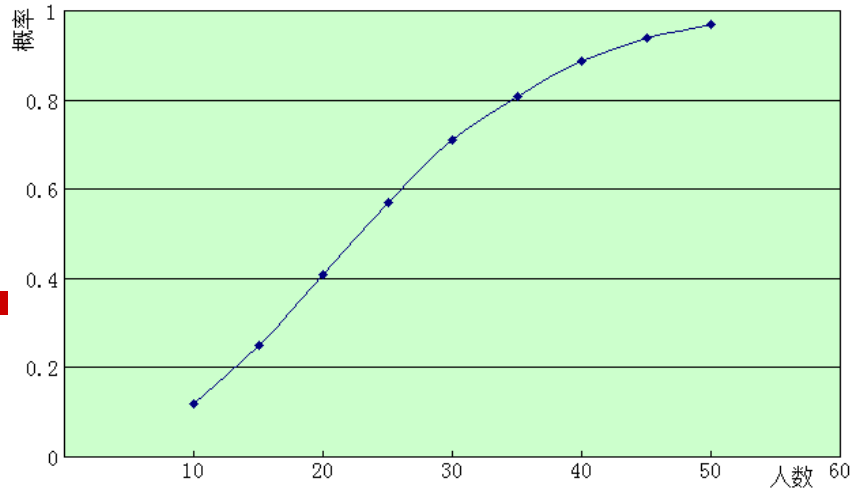
古典概型

- 举例：将n个不同的球放入N(N≥n)个盒子中，假设盒子容量无限，求事件A={每个盒子至多有1个球}的概率。

解 $P(A)=\frac{P_N^n}{N^n}$

- 基本事件总数：
  - 第1个球，有N种放法；
  - 第2个球，有N种放法；
  - ……
  - 共：N^n种放法。
- 每个盒子至多放1个球的事件数：
  - 第1个球，有N种放法；
  - 第2个球，有N-1种放法；
  - 第3个球，有N-2种放法；
  - ……
  - 共：$N(N-1)(N-2)\cdots(N-n+1)=P_N^n$

生日悖论



- 假定小象学院《机器学习》班有50位同学，则至少有2人生日相同的概率是多少？

n	10	15	20	25	30	35	40	45	50
P	0.12	0.25	0.41	0.57	0.71	0.81	0.89	0.94	0.97

装箱问题

- 将12件正品和3件次品随机装在3个箱子中，每箱装5件，则每箱中恰有1件次品的概率是多少？

解

- 将15件产品装入3个箱子，每箱装5件，共有15!/(5!5!5!)种装法；
- 先把3件次品放入3个箱子，有3!种装法。对于这样的每一种装法，把其余12件产品装入3个箱子，每箱装4件，共有12!/(4!4!4!)种装法；
- P(A)= (3!*12!/(4!4!4!)) / (15!/(5!5!5!)) = 25/91

与组合数的关系

- 把n个物品分成k组，使得每组物品的个数分别为n1,n2…nk，(n=n1+n2+…+nk)，则不同的分组方法有 $\frac{n!}{n_1!n_2!n_3!\cdots n_k!}$ 种。

- 上述问题的简化版本，即n个物品分成2组，第一组m个，第二组n-m个，则分组方法有 $\frac{n!}{m!(n-m)!}$，即：C_n^m 。

组合数背后的秘密 $N \to \infty \Rightarrow \ln N! \to N(\ln N - 1)$

$$H = \frac{1}{N} \ln \frac{N!}{\prod_{i=1}^{k} n_i!} = \frac{1}{N} \ln(N!) - \frac{1}{N} \sum_{i=1}^{k} \ln(n_i!)$$

$$\to (\ln N - 1) - \frac{1}{N} \sum_{i=1}^{k} n_i (\ln n_i - 1)$$

$$= \ln N - \frac{1}{N} \sum_{i=1}^{k} n_i \ln n_i = -\frac{1}{N} \left(\left(\sum_{i=1}^{k} n_i \ln n_i \right) - N \ln N \right)$$

$$= -\frac{1}{N} \sum_{i=1}^{k} (n_i \ln n_i - n_i \ln N) = -\frac{1}{N} \sum_{i=1}^{k} \left(n_i \ln \frac{n_i}{N} \right)$$

$$= -\sum_{i=1}^{k} \left(\frac{n_i}{N} \ln \frac{n_i}{N} \right) \to -\sum_{i=1}^{k} (p_i \ln p_i)$$

参考文献

- Prof. Andrew Ng, *Machine Learning*, Stanford University
- 同济大学数学教研室，高等数学，高等教育出版社，1996
- 王松桂、程维虎、高旅端，概率论与数理统计，科学出版社，2000

我们在这里

- http://wenda.ChinaHadoop.cn
  - 视频/课程/社区
- 微博
  - @ChinaHadoop
  - @邹博_机器学习
- 微信公众号
  - 小象
  - 大数据分析挖掘

感谢大家！

恳请大家批评指正！