2023春季学期
信息内容安全期末复习讲座

主讲人：Skyrim

2023-5-17

目录/Contents

串匹配

School of CS
Academic achievement

1. 串匹配算法分类和评价指标有哪些?
    1. 匹配的模式数目：单模式、多模式
    2. 匹配方式：精确匹配、近似匹配
    3. 匹配的具体内容：单词匹配、字符类匹配、正则表达匹配
    4. 文本的角度：实时(on-line)文本:文本可以动态地更新(例如:网络入侵监测)、非实时(off-line)文本:被查找文本是静态的(例如:搜索引擎中查找的数据)

    评价指标 时间复杂度/回溯次数

2.串匹配算法优化的主要思路?

- “以史为鉴”：从历史过程吸取经验，找相同之处
- “关注当前”：着眼现在，发现不同之处
- “预测未来”：失配概率高，优化空间
- “早做准备”：提前总结知识，确保运行最快

充分利用已经比较过的字符信息(kmp)
利用匹配失败时获得的信息(bm)
本质是利用了串的结构化信息

串匹配

School of CS
Academic achievement

3. 试比较不同串匹配算法的优缺点。

KMP和BM的优点: 对正文扫描一遍，不回溯。且每一次比较后，有明确的信息记录下一次比较的位置。

- BF算法时间复杂度为O(n*m)
- KMP算法时间复杂度为O(n+m)
- BM算法时间复杂度为O(n*m)，但实际运行效果最快。最好情况下O(n/m)

试比较不同串匹配算法的优缺点 BF算法思想简单，实际使用效果尚可，但时间复杂度很差/KMP算法时间复杂度良好，扫描文本不需要回溯/BM算法实际效果常常最快，可以实现跳跃查询，但时间复杂度不是最好。

AC自动机复杂度：O(n)，扫描时不需要回溯，时间复杂度与关键字的数目和长度无关。

8. 如何进一步优化 AC 算法？（内存空间优化）

因为失配的概率更高，所以试图提高 goto函数的查找效率，引入位图 bitmap。

9. 如何进一步优化 WM 算法？（模式长度敏感）

（模式长度敏感）该算法对最短模式长度敏感。如果最短模式长度很短，则移位的值不可能很大，因此对匹配过程的加速有限。

10. 针对千万量级的模式集，如何优化串匹配算法?

并行搜索，分布式匹配，预处理转换成状态机表示数据分片，分块匹配，GPU 加速。

串匹配

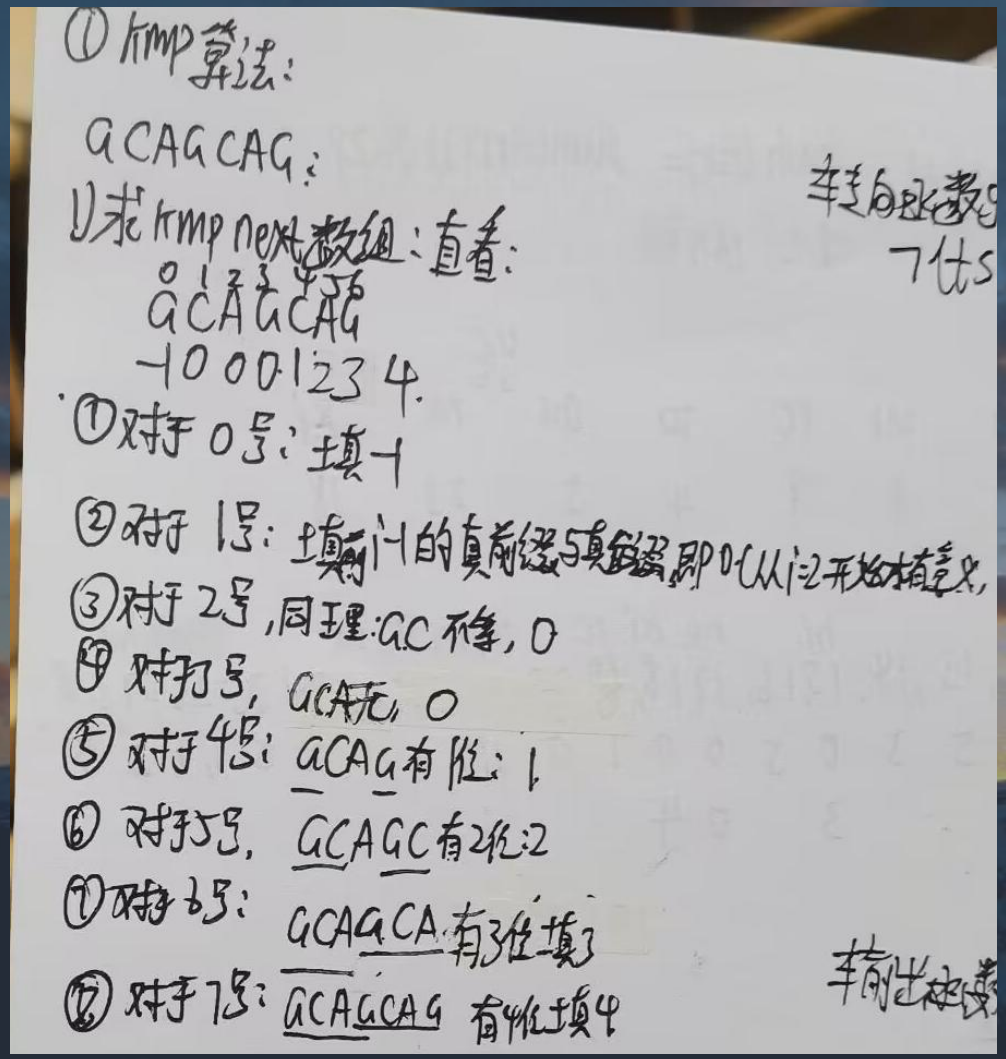
School of CS
Academic achievement

针对特定的 IP 地址或 URL 地址构成的模式集，如何优化串匹配算法?

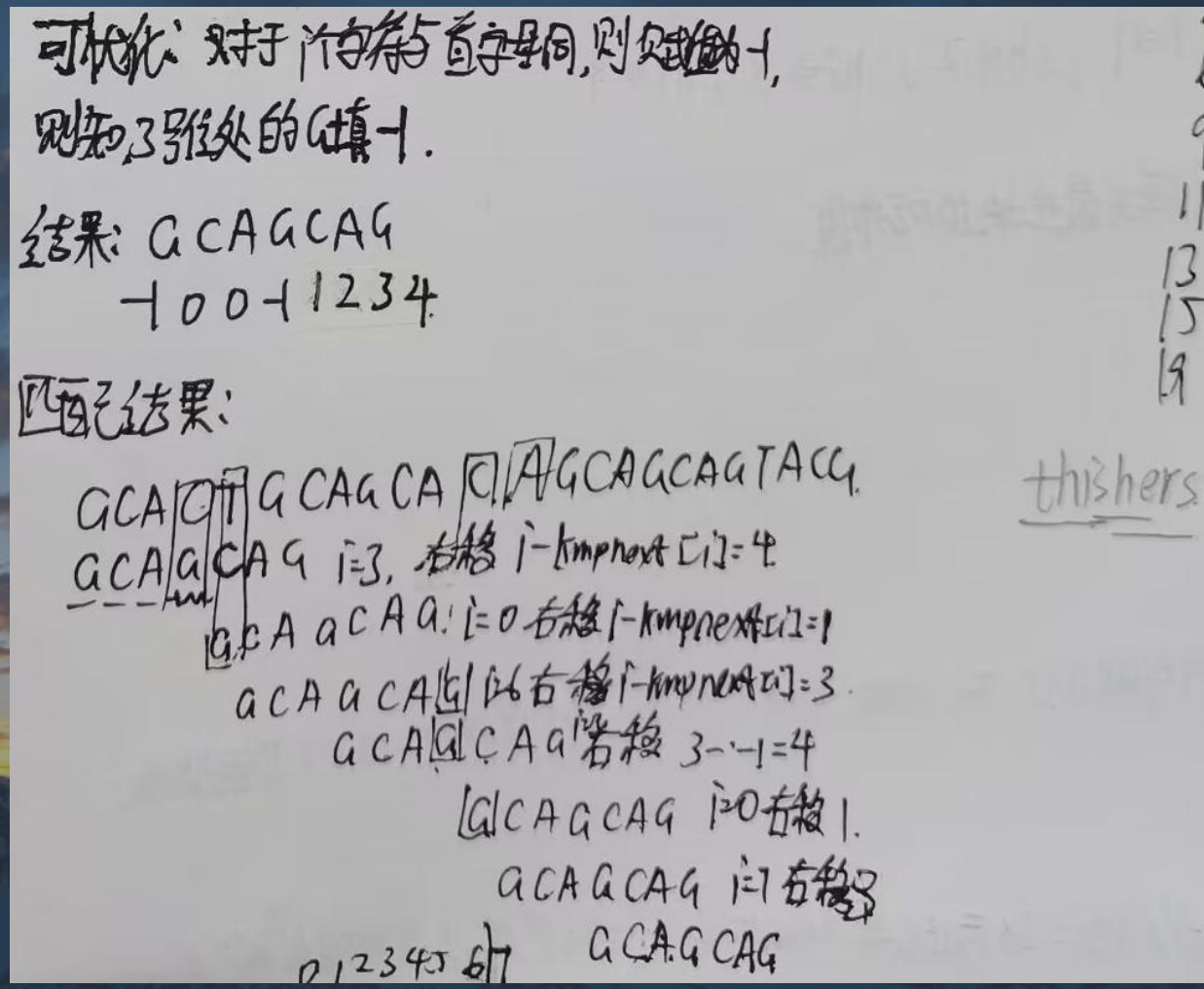
URL：现有URL过滤方法大多是利用通用模式匹配算法，而通用模式匹配方法没有充分利用URL的结构特征。1.改进传统模式匹配算法。针对URL长度普通较长的特点，对已有的传统模式匹配算法进行改进，使其尽可能适合于处理URL查找；2.对URL进行编码处理，将URL模式串的查找转化为数字或其它易处理的字符，从而简化匹配过程。3.借助于高性能硬件处理器，提高URL的查找效率。4.针对 URL块状结构，利用多层哈希设计实现一个 URL过滤算法

IP：基于PAT树的按位匹配的思想来设计算法，通常是将ip按位转换成字符串，再进行匹配 针对 URL块状结构，利用多层哈希设计实现一个 URL过滤算法

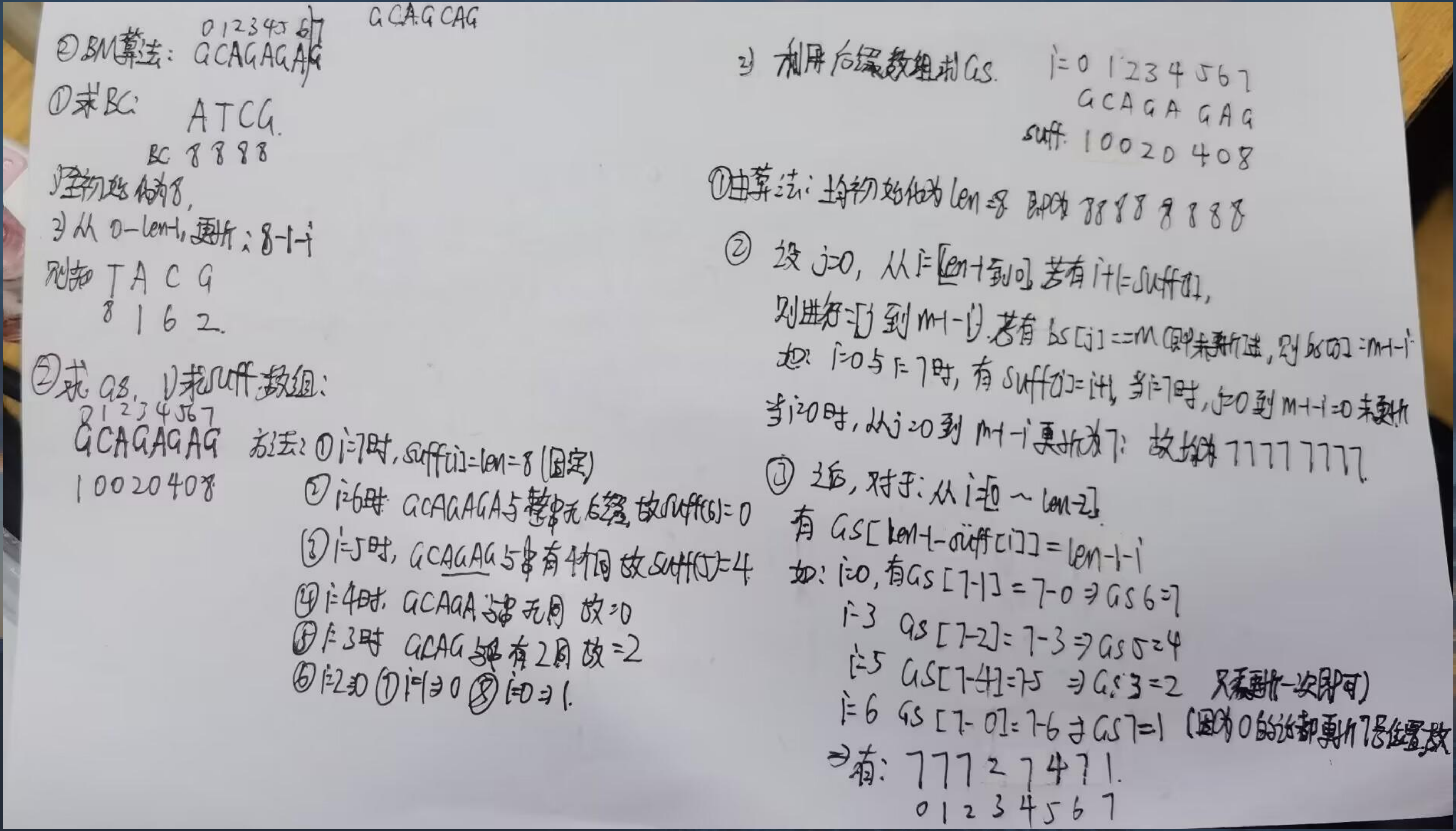
4. KMP算法流程与例题：



实际上这里的优化不彻底，一般当x[next[i]]=x[i],把next[i]更新成next[next[i]]即可。（但是考试还是按这种简单的优化来吧）



5. BM算法流程与例题



GCAGCAG

②BM算法:
0 1 2 3 4 5 6 7
GCAGAGAG

①求BC: ATCG
BC 8 8 8 8

1)全初始化为8,
2)从 0–len-1, 更新: 8-1-i
则有 T A C G
8 1 6 2.

②求GS. 1)求suff数组:
0 1 2 3 4 5 6 7
GCAGAGAG
1 0 0 2 0 4 0 8

方法: ① i=7时, suff[i]=len=8 (固定)
② i=6时 GCAGAGA与整串无后缀, 故suff[6]=0
③ i=5时, GCAGAG与串有4个同 故suff[5]=4
④ i=4时, GCAGA与串无同 故=0
⑤ i=3时 GCAG与串有2同 故=2
⑥ i=2→0 ⑦ i=1→0 ⑧ i=0→1.

2) 利用后缀数组求GS. i=0 1 2 3 4 5 6 7
G C A G A G A G
suff 1 0 0 2 0 4 0 8

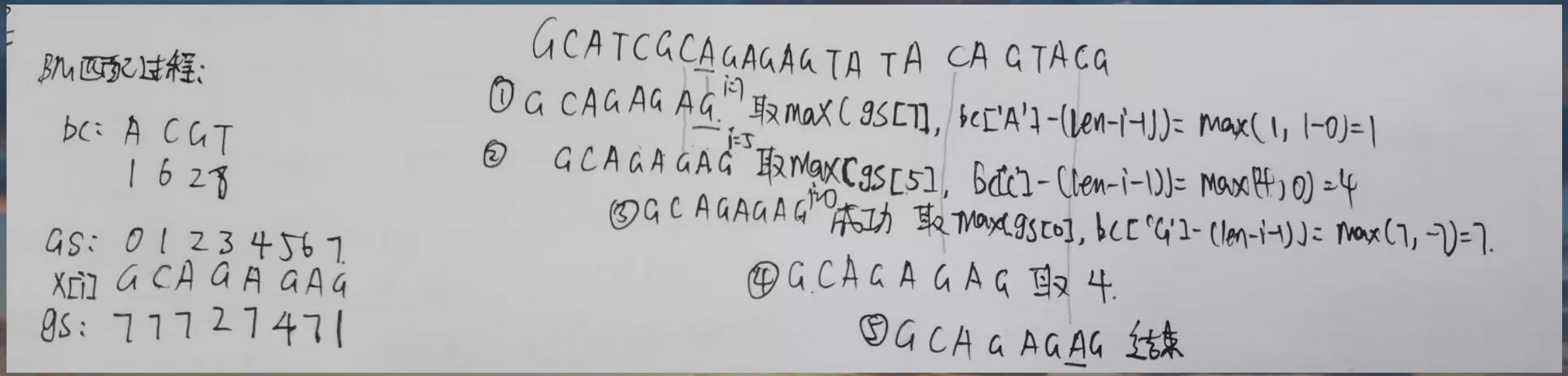
①由算法: 全初始化为 len=8 即GS 8 8 8 8 8 8 8 8

② 设j=0, 从i=[len-1到0] 若有 i+1=suff[i], 则更新[j]到m-1-i. 若有 bs[j]==m (即未更新过), 则bs[j]=m-1-i
如: i=0与i=7时, 有suff[i]=i+1, 当i=7时, j=0到m-1-i=0未更新
当i=0时, 从j=0到m-1-i更新为7; 故此时 77777777.

③ 之后, 对于: 从 i=[0~len-2]
有 GS[len-1-suff[i]] = len-1-i
如: i=0, 有GS[7-1]=7-0 ⇒ GS6=7
i=3 GS[7-2]=7-3 ⇒ GS5=4
i=5 GS[7-4]=7-5 ⇒ GS3=2 只需更新一次即可)
i=6 GS[7-0]=7-6 ⇒ GS7=1 (因为0的位都更新7号位置数

⇒有: 7 7 7 2 7 4 7 1.
0 1 2 3 4 5 6 7

5. BM算法流程与例题


BM匹配过程:
bc: A C G T
1 6 2 8
GS: 0 1 2 3 4 5 6 7
X[i] G C A G A G A G
gs: 7 7 7 2 7 4 7 1
GCATCGCAGAGAGTATACAGTACG
① G CAGAGAG i=7 取max(gs[7], bc['A']-(len-i-1))= max(1, 1-0)=1
② GCAGAGAG j=5 取max(gs[5], bc[C]-(len-i-1))= max(4, 0)=4
③ GCAGAGAG j=0 成功 取max(gs[0], bc['G']-(len-i-1))= max(7, -)=7.
④ GCAGAGAG 取4.
⑤ GCAGAGAG 结束

KMP算法与BM算法注意点：

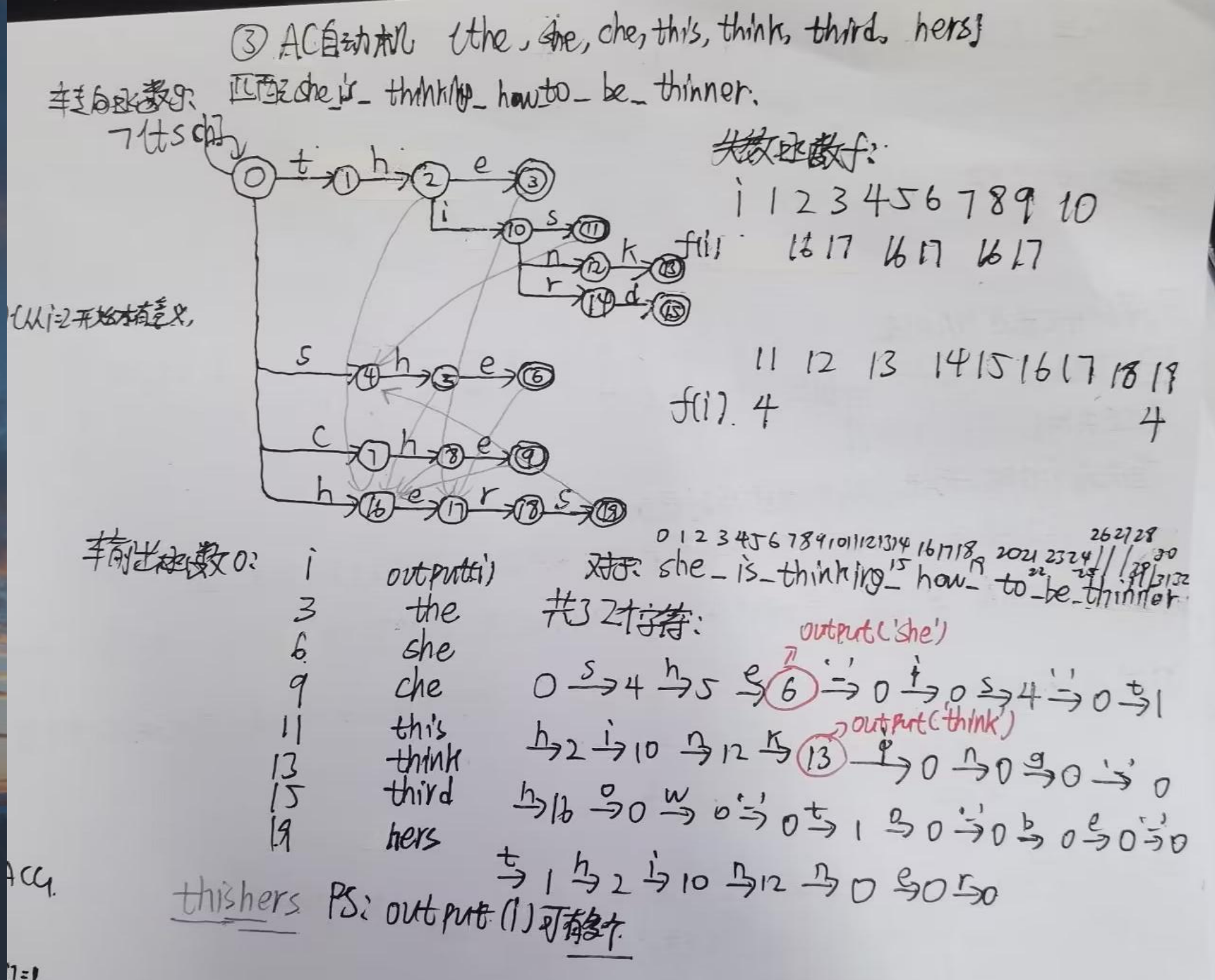
1.记录比较次数时，KMP不回溯。
记录比较次数时，即使是重复的，BM算法也需要多次比较。

2.当比较到最后一位（KMP模式串的最右一位，BM模式串的最左一位），并且成功匹配时，本次更新，KMP使用len-kmpnext[len]进行，而BM算法当做第0位失配的情况进行，即使用max(gs[0],bc[0]-(len-1-0))更新。

3.记录对齐次数时，移出的那一次不计入。

6. AC算法流程与例题

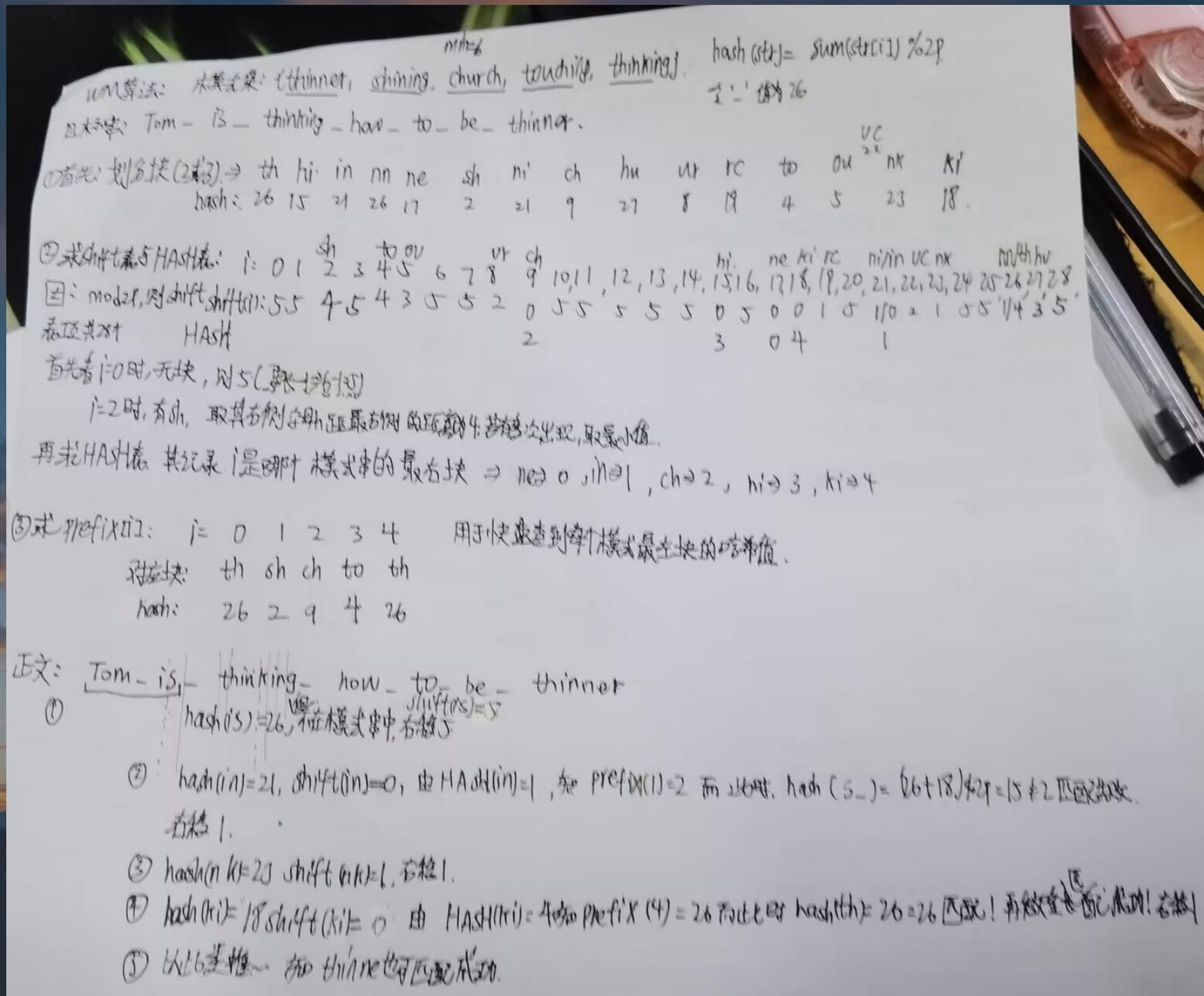
AC算法如果下一个输入不符合转向函数，那就跳到fail指针所指的状态，继续匹配。



7. WM算法流程与例题

如果有多个字符串截取最短长度后结果相等，我们保留一个即可。

比如sky和skyrim，我们保留一个sky即可。

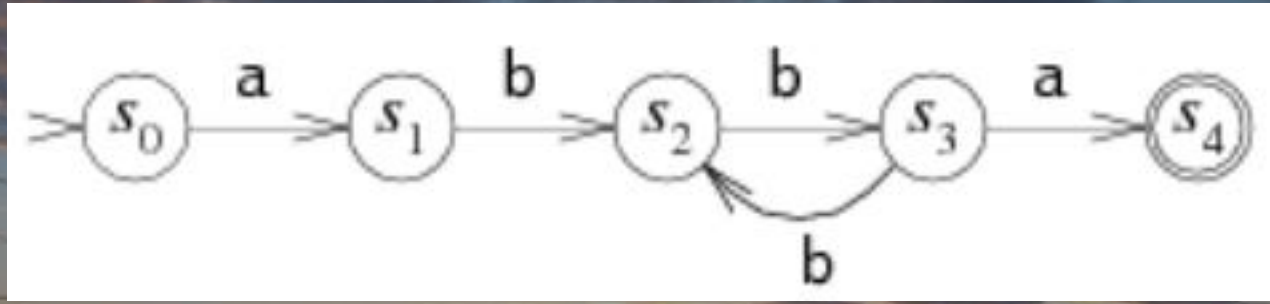


串匹配

School of CS
Academic achievement

正则表达式，Regular Expression，使用单个字符串来描述、匹配一系列满足某种句法规则的字符串。在很多文本编辑器里，正则表达式通常被用来检索、替换那些匹配某个模式的文本。最常见的，比如“.”，其中“.”表示匹配除“\n”之外的任何单个字符，“”表示匹配前面的子表达式零次或多次。

给定正则表达式a(bb)+a,其中+表示匹配前面的子表达式一次或多次，所以字符串abba或abbbba都能被这个模式所匹配。

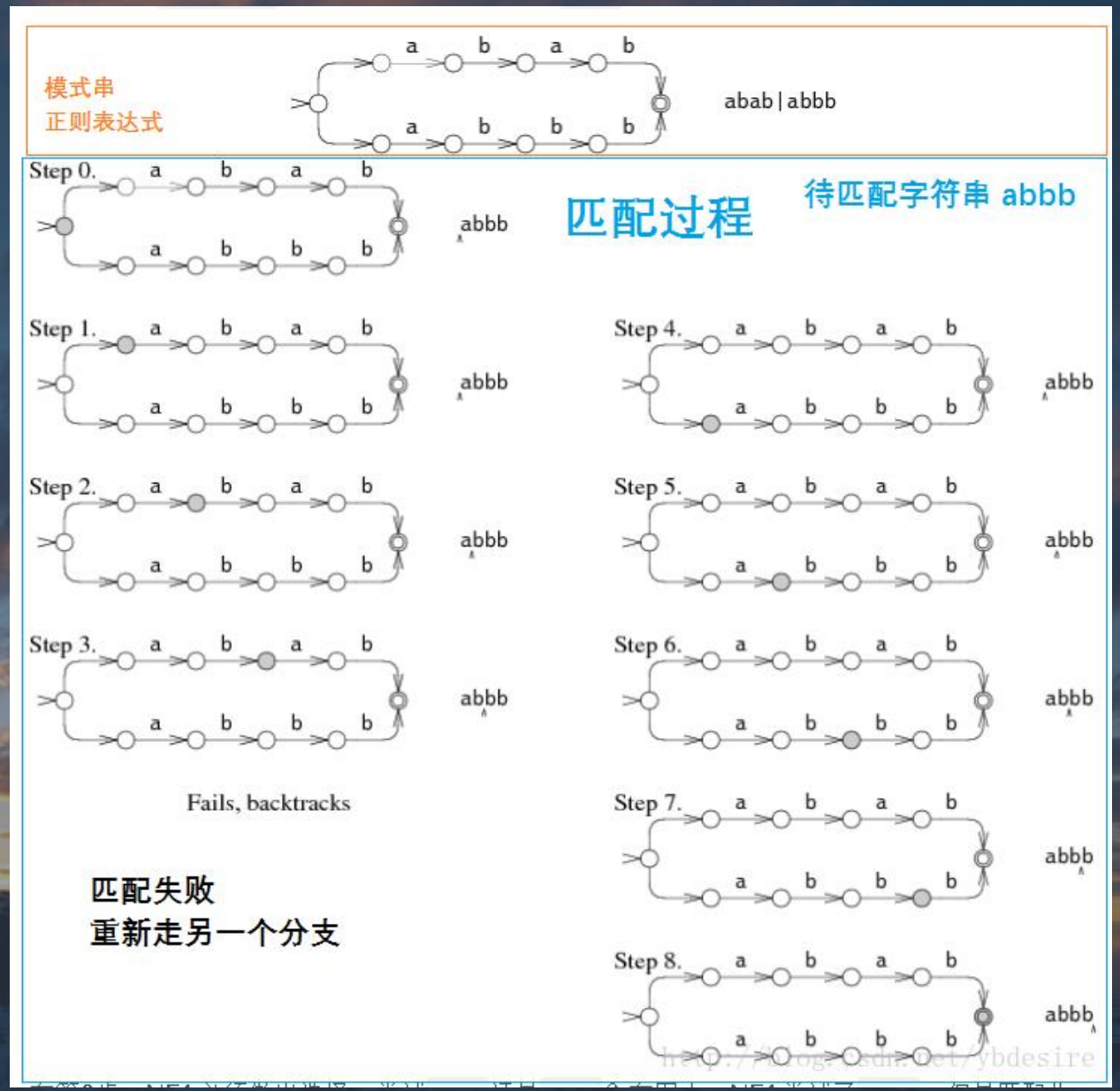


串匹配

School of CS
Academic achievement

匹配时，可能有多条路径，遇到分支时，可以采用试错法，一条走不通，再尝试另一条。但这种做法效率较低。

所以另一种更优的做法，是在分支处同时匹配多条分支，同时保持多个状态，这样避免了很多不必要的尝试。



文本分类

School of CS
Academic achievement

1. 文本的量化表示？文本分类系统的主要流程？

1. 收集训练集测试集并预处理，比如分词（中文分词：串匹配/统计学习方法），去掉停用词和特殊符号； 2. 文本表示及特征选择、进行人工标注等；***常见的特征选择算法***：1.TFIDF每个属于文档d的词项t的权重如下计算：TFIDF(t)= TF(t)*IDF(t)。TF-IDF与一个词在文档中的出现次数成正比，与该词在整个语言中的出现次数成反比。自动提取关键词的算法就是计算出文档的每个词的TF-IDF值，然后按降序排列，取排在最前面的几个词。2.主成分分析(PCA)用于提取数据的主要特征分量3.潜在语义分析(LSA)本质上是把高维的词频矩阵进行降维，降维方法是用奇异值分解4.word2vec将由one-hot编码获得的高维向量转换为低维的连续值向量，也就是稠密向量，又称分布式表示，可以很好的度量词与词之间的相似性。特征提取：高维的特征可能会大大增加机器的学习时间而仅产生与小得多的特征子集相关的学习分类结果。评价函数：构造一个评价函数，对特征集中的每一个特征进行独立的评估。3. 分类器构造； 4. 分类器根据文本的特征进行分类，训练学习； 5. 分类结果的评价（精确率、召回率、F1测试(2*精*召)/(精*召)、微平均(计算每一类的精/召/F1)、宏平均(全部类的精/召/F1)）。

量化表示：One-hot(离散向量表示，每个字/词编码为一个索引，由索引赋值)，词袋（也称Count Vector每个字/词出现的次数），N-gram（优化的词袋，N指N个相邻字组词，可取不同的值），TF-IDF（词语频率TF(t)=该词语在当前文档中出现的次数/当前文档中词语的总数 和 逆文档频率IDF(t)=loge(文档总数/含该词的文档总数)）

2. 主要的分类方法有哪些？比较各自的优缺点。

基于规则的分类器：适合于规模小的数据集、具有良好的表达能力、易于构造、分类效率高、与决策树的性能相当

决策树：:1)不需要任何领域知识或参数假设。2）适合高维数据。3）简单易于理解。4）短时间内处理大量数据，得到可行且效果较好的结果。5)能够同时处理数据型和常规性属性。缺点:1)对于各类别样本数量不一致数据，信息增益偏向于那些具有更多数值的特征。2）易于过拟合。3）忽略属性之间的相关性。4不支持在线学习。

贝叶斯分类法优点:1)所需估计的参数少，对于缺失数据不敏感。2)有着坚实的数学基础，以及稳定的分类效率。缺点:1）假设属性之间相互独立，这往往并不成立。2)需要知道先验概率。3)分类决策存在错误率

K-NN算法1）思想简单，理论成熟，既可以用来做分类也可以用来做回归;2)可用于非线性分类;3)训练时间复杂度为O(n);4)准确度高，对数据没有假设，对outlier不敏感;缺点:1)计算量太大2）对于样本分类不均衡的问题，会产生误判。3）需要大量的内存。4)输出的可解释性不强。

支持向量机1)可以解决小样本下机器学习的问题。2）提高泛化性能。3)可以解决高维、非线性问题。超高维文本分类仍受欢迎。4)避免神经网络结构选择和局部极小的问题；缺点：（1）核函数的选定非常关键，它的选择好坏直接影响到算法的实现与效果。（2）SVM 分类器的设计依赖于少量的支持向量，使得分类效果受噪声的影响非常大。（3）SVM 的计算速度较慢，尤其是训练样本很大时，传统的求解优化方法难以满足实时性要求；尽管如此，支持向量机较强的分类性能和适应性能使得这种方法能够在实际工程中发挥重要作用

3. 基于规则归纳的分类方法？能够针对实际数据给出求解步骤。
构造分类规则直接方法：数据中提取 间接方法：从其他分类模型中提取
基本的分类器：(condition)->y。
直接方法-顺序覆盖：对每个给定的类 Ci 希望规则可以覆盖该类的大多数元组，但不包括其它类的元组(或很少) (1)初始为空规则集(2)使用Learn-One-Rule函数得到一条新规则(3)从训练集中删去被新产生的规则所覆盖的实例(4)重复步骤(2)和步骤(3)直到满足停止标准为止 【可以归纳为：产生规则/消除实例(不消,总产生同一条,后续规则的正确度过低/过高)/规则评价(准确率（正确分类个数nc/被该规则覆盖个数n） 覆盖度((nc+1)/(n+类别总数k))M估计((nc+kp)/(n+k))适合于规模小的数据集)/停止标准(计算增益,增益不显著则停止)/规则的剪枝(与决策树后剪枝类似:删一支,计算新错误率,若降低则减掉)】
间接方法-从决策树提取规则：从根到树的叶节点的每条路径创建一个规则，沿每个划分准则的逻辑AND形成规则的前提，存放类预测的叶节点形成规则后件。规则间是互斥或穷举的

冲突的解决：规模序：最严格的规则赋予最高优先级(如最多属性测试)。基于类的序：按照类的频繁性或错分代价的降序排列。基于规则的序(决策表)：根据规则的质量度量/专家意见形成优先级列表。如有序规则集，若满足多条用排在前面的分类，若不满足任何规则使用默认类别。

4. 基于决策树的分类方法（C4.5）？能够针对实际数据给出求解步骤。
1)计算类别信息熵：如有14个样例9个正的5个负的，则所有样本中各种类别出现的不确定性之和：Info(D)=-9/14*log2(9/14)-5/14*log2(5/14)=0.940
2)计算每个属性的信息熵：一种条件熵，表示的是在某种属性的条件下，各种类别出现的不确定性之和。属性的信息熵越大，表示这个属性中拥有的样本类别越“不纯”。如14个样例中‘天气’属性上有晴（2正3负）、阴（4正）、雨（3正2负）；则Info(天气)=5/14*(-2/5*log2(2/5)-3/5*log2(3/5))+4/14*(-4/4*log2(4/4))+5/14*(-3/5*log2(3/5)-2/5*log2(2/5))=0.694
3)计算信息增益 信息增益=熵-条件熵 即类别信息熵-属性信息熵，表示信息不确定的减少程度。增益越大就越好。如此处Gain(天气)=Info(D)-Info(天气)-0.940-0.694=0.246
4)计算属性分裂信息度量（属性的内在信息，越大越倾向于不选） H(天气)=-5/14*log2(5/14)-5/14*log2(5/14)-4/14*log2(4/14)=1.577
5)计算每个属性的信息增益率 IGR(天气)=Gain(天气)/H(天气)=0.246/1.577=0.155比较可得天气增益率最高，选择天气为分裂属性。

分裂后，天气为阴时类别{进行/取消}唯一，将之定义为叶节点，选择不纯的晴天和雨天继续分裂。以雨天为例，在内部继续计算Info(D)（此时只有五个样本）Info(风速)等 Gain(风速)等 H(风速)等 IGR(风速)等 判断得风速的信息增益率最高所以选风速为分裂节点，分裂后发现均纯因此均为叶子节点，分类结束。

4. 基于决策树的分类方法（C4.5）？能够针对实际数据给出求解步骤。

天气	温度	湿度	风速	活动
晴	炎热	高	弱	取消
晴	炎热	高	强	取消
阴	炎热	高	弱	进行
雨	适中	高	弱	进行
雨	寒冷	正常	弱	进行
雨	寒冷	正常	强	取消
阴	寒冷	正常	强	进行
晴	适中	高	弱	取消
晴	寒冷	正常	弱	进行
雨	适中	正常	弱	进行
晴	适中	正常	强	进行
阴	适中	高	强	进行
阴	炎热	正常	弱	进行
雨	适中	高	强	取消

天气	温度	湿度	风速	活动
晴	寒冷	正常	弱	进行
晴	适中	正常	强	进行
晴	炎热	高	弱	取消
晴	炎热	高	强	取消
晴	适中	高	弱	取消
阴	炎热	高	弱	进行
阴	寒冷	正常	强	进行
阴	适中	高	强	进行
阴	炎热	正常	弱	进行
雨	适中	高	弱	进行
雨	寒冷	正常	弱	进行
雨	适中	正常	弱	进行
雨	寒冷	正常	强	取消
雨	适中	高	强	取消

C4.5

1. 计算类别信息熵： 类别信息熵表示的是所有样本中各种类别出现的不确定性之和。

$$Info(D) = -9/14 * log2(9/14) - 5/14 * log2(5/14) = 0.940$$

2. 计算每个属性的信息熵

每个属性的信息熵相当于一种条件熵，表示的是在某种属性的条件下，各种类别出现的不确定性之和。属性的信息熵越大，表示这个属性中拥有的样本类别越“不纯”。

$$Info(天气) = 5/14 * [-2/5 * log2(2/5) - 3/5 * log2(3/5)] + 4/14 * [-4/4 * log2(4/4)] + 5/14 * [-3/5 * log2(3/5) - 2/5 * log2(2/5)] = 0.694$$

$$Info(温度) = 4/14 * [-2/4 * log2(2/4) - 2/4 * log2(2/4)] + 6/14 * [-4/6 * log2(4/6) - 2/6 * log2(2/6)] + 4/14 * [-3/4 * log2(3/4) - 1/4 * log2(1/4)] = 0.911$$

$$Info(湿度) = 7/14 * [- 3/7 * log2(3/7) - 4/7 * log2(4/7)] + 7/14 * [- 6/7 * log2(6/7) - 1/7 * log2(1/7)] = 0.789$$

$$Info(风速) = 6/14 * [- 3/6 * log2(3/6) - 3/6 * log2(3/6)] + 8/14 * [- 6/8 * log2(6/8) - 2/8 * log2(2/8)] = 0.892$$

C4.5

3. 计算信息增益

- 信息增益的 = 熵 - 条件熵，在这里就是 类别信息熵 - 属性信息熵，它表示的是信息不确定性减少的程度。如果一个属性的信息增益越大，就表示用这个属性进行样本划分可以更好的减少划分后样本的不确定性，当然，选择该属性就可以更快更好地完成我们的分类目标。

```
Gain(天气) = Info(D) - Info(天气) = 0.940 - 0.694 = 0.246
Gain(温度) = Info(D) - Info(温度) = 0.940 - 0.911 = 0.029
Gain(湿度) = Info(D) - Info(湿度) = 0.940 - 0.789 = 0.15
Gain(风速) = Info(D) - Info(风速) = 0.940 - 0.892 = 0.048
```

4.计算属性分裂信息度量

- 用分裂信息度量来考虑某种属性进行分裂时分支的数量信息和尺寸信息（属性的内在信息）。
- 信息增益率用信息增益 / 内在信息，会导致属性的重要性随着内在信息的增大而减小。这个属性不确定性越大，就越不倾向于选取它。

```
H(天气) = - 5/14 * log2(5/14) - 5/14 * log2(5/14) - 4/14 * log2(4/14) = 1.577
H(温度) = - 4/14 * log2(4/14) - 6/14 * log2(6/14) - 4/14 * log2(4/14) = 1.556
H(湿度) = - 7/14 * log2(7/14) - 7/14 * log2(7/14) = 1.0  //湿度有两个取值
H(风速) = - 6/14 * log2(6/14) - 8/14 * log2(8/14) = 0.985   //风速有两个取值
```

C4.5

• 5. 计算信息增益率

```
IGR(天气) = Gain(天气) / H(天气) = 0.246/1.577 = 0.155
IGR(温度) = Gain(温度) / H(温度) = 0.029 / 1.556 = 0.0186
IGR(湿度) = Gain(湿度) / H(湿度) = 0.151/1.0 = 0.151
IGR(风速) = Info(风速) / H(风速) = 0.048/0.985 = 0.048
```

- 天气的信息增益率最高，选择天气为分裂属性。
- 分裂之后，天气为“阴”的条件下，类别唯一，则它定义为叶子节点
- 然后，选择不“纯”的结点继续分裂。

阴

温度	湿度	风速	活动
炎热	高	弱	进行
寒冷	正常	强	进行
适中	高	强	进行
炎热	正常	弱	进行

晴

温度	湿度	风速	活动
寒冷	正常	弱	进行
适中	正常	强	进行
炎热	高	弱	取消
炎热	高	强	取消
适中	高	弱	取消

正常

温度	风速	活动
寒冷	弱	进行
适中	强	进行

高

温度	风速	活动
炎热	弱	取消
炎热	强	取消
适中	弱	取消

雨

温度	湿度	风速	活动
适中	高	弱	进行
寒冷	正常	弱	进行
适中	正常	弱	进行
寒冷	正常	强	取消
适中	高	强	取消

弱

温度	湿度	活动
寒冷	正常	取消
适中	高	取消

强

温度	湿度	活动
炎热	高	进行
炎热	正常	进行
适中	正常	进行

C4.5

接着，再以天气=“雨”的子结点为例：

这里应该是2/2*log2(2/2)

$$Info(D) = -3/5 * log2(3/5) - 2/5 * log2(2/5) = 0.970$$

$$Info(风速) = 3/5 * [- 3/3 * log2(3/3)] + 2/5 * [-1/2 * log2(1/2) -1/2 * log2(1/2)] = 0.4$$
$$Info(湿度) = 3/5 * [-2/3 * log2(2/3) - 1/3 * log2(1/3)] + 2/5 * [-1/2 * log2(1/2) - 1/2 * log2(1/2)] = 0.789$$
$$Info(温度) = 3/5 * [-2/3 * log2(2/3) - 1/3 * log2(1/3)] + 2/5 * [-1/2 * log2(1/2) - 1/2 * log2(1/2)] =0.789$$

$$Gain(温度) = Info(D) - Info(温度) = 0.970 - 0.789 = 0.181$$
$$Gain(湿度) = Info(D) - Info(湿度) = 0.970 - 0.789 = 0.181$$
$$Gain(风速) = Info(D) - Info(风速) = 0.970 - 0.892 = 0.57$$

$$IGR(风速) = Gain(风速) / H(风速)$$ 最高

- 风速属性的信息增益率最高，所以选择风速作为分裂结点，分裂之后，发现子结点都是纯的，因此子节点均为叶子节点，分裂结束。

5. 贝叶斯分类方法？能够针对实际数据给出求解步骤。
每个数据样本用n维特征向量X={x1，x2，...，xn}表示，分别描述n个性质A1-An的n个度量。假定有m个类C1-Cm。2)给定一个未知的样本X（无类标号），朴贝将X分配给Ci，最大化P(Ci|X)，其中令P最大的类Ci称为最大后验假定。3)P(X)对所有类为常数(每个属性在不同类别下独立)，只需P(X|Ci)P(Ci)最大即可。若类先验概率未知则常假定等概率，即P(C1)=...=P(Cm)。并据此对P(X|Ci)最大化。类的先验概率可以用P(Ci)=si/s求(s是训练样本总数)。对于PPT上的例题可以这样做：

1)求P(Ci)的先验概率P(买电脑=’Yes’)=9/14 P(买电脑=’No’)=5/14

2)根据样本的每个属性值计算条件概率：P(age<30&&买电脑=’Yes’)=2/9...3)计算最终概率P(X|买电脑=’Yes’)=2/9*(还有另外三项,因为总共四个属性都要乘起来)=0.044 P(X|买电脑=’Yes’)P(买电脑=’Yes’)=0.044*0.643=0.028大于不买电脑的，所以预测为Yes.

练习：age=60, income=medium, student=no ， credit rating=fair

RID	age	income	student	credit_rating	Class:buys_computer
1	youth	high	no	fair	no
2	youth	high	no	excellent	no
3	middle_aged	high	no	fair	yes
4	senior	medium	no	fair	yes
5	senior	Low	yes	fair	yes
6	senior	Low	yes	excellent	no
7	middle_aged	low	yes	excellent	yes
8	youth	medium	no	fair	no
9	youth	low	yes	fair	yes
10	senior	medium	yes	fair	yes
11	youth	medium	yes	excellent	yes
12	middle_aged	medium	no	excellent	yes
13	middle_aged	high	yes	fair	yes
14	senior	medium	no	excellent	no

(C)， i = 1， 2。每个类的先验概率 P(C i) 可以根据训练样本计算：

$$P(buys\_computer = \text{“}yes\text{”}) = 9/14 = 0.643$$
$$P(buys\_computer = \text{“}no\text{”}) = 5/14 = 0.357$$

1， 2，我们计算下面的条件概率：

$$P(age = \text{“}<30\text{”} \mid buys\_computer = \text{“}yes\text{”}) = 2/9 = 0.222$$
$$P(age = \text{“}<30\text{”} \mid buys\_computer = \text{“no”}) = 3/5 = 0.600$$
$$P(income = \text{“}medium\text{”} \mid buys\_computer = \text{“}yes\text{”}) = 4/9 = 0.444$$
$$P(income = \text{“}medium\text{”} \mid buys\_computer = \text{“}no\text{”}) = 2/5 = 0.400$$
$$P(student = \text{“}yes\text{”} \mid buys\_computer = \text{“}yes\text{”}) = 6/9 = 0.667$$
$$P(student = \text{“}yes\text{”} \mid buys\_computer = \text{“}no\text{”}) = 1/5 = 0.200$$
$$P(credit\_rating = \text{“}fair\text{”} \mid buys\_computer = \text{“}yes\text{”}) = 6/9 = 0.667$$
$$P(credit\_rating = \text{“}fair\text{”} \mid buys\_computer = \text{“}no\text{”}) = 2/5 = 0.400$$

$$P(X \mid buys\_computer = \text{“}yes\text{”}) = 0.222 \times 0.444 \times 0.667 \times 0.667 = 0.044$$
$$P(X \mid buys\_computer = \text{“}no\text{”}) = 0.600 \times 0.400 \times 0.200 \times 0.400 = 0.019$$

$$P(X \mid buys\_computer = \text{“}yes\text{”})\, P(buys\_computer = \text{“}yes\text{”}) = 0.044 \times 0.643 = 0.028$$
$$P(X \mid buys\_computer = \text{“}no\text{”})\, P(buys\_computer = \text{“}no\text{”}) = 0.019 \times 0.357 = 0.007$$

叶斯分类预测 buys-computer = “ yes ”。

6. KNN分类方法？能够针对实际数据给出求解步骤。

1. 以向量空间模型的形式描述各训练样本。

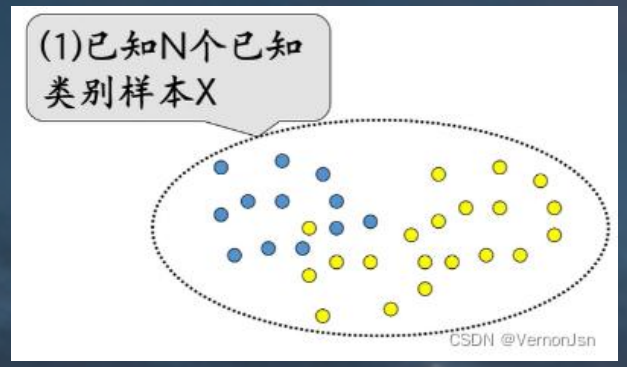
2. 在全部训练样本集中选出与待分类样本距离最近的K个样本(距离可选各种)。K值的确定目前没有很好的方法，一般采用先定一个100左右的初始值，然后再调整。

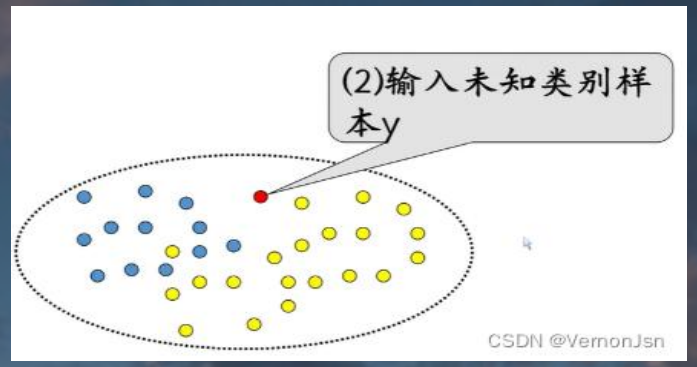
3. 将待分类样本标记为其K个邻居中所属最多的那个类别中.KNN算法最简单粗暴的就是将预测点与所有点距离进行计算，然后保存并排序，选出前面K个值看看哪些类别比较多。它只关心哪类样本的数量最多，而不去把距离远近考虑在内，因此，我们可以采用权值的方法来改进。

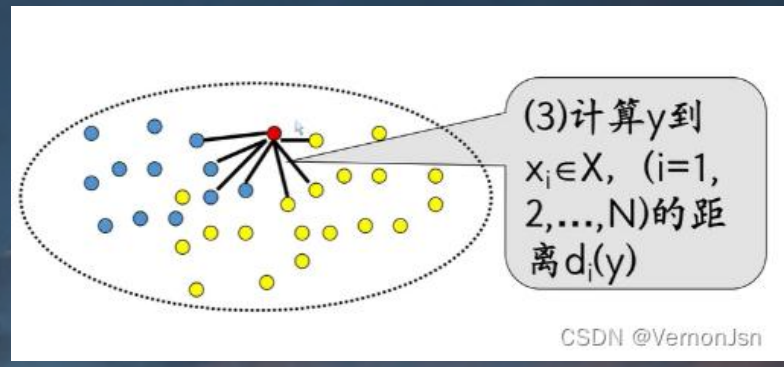
文本分类

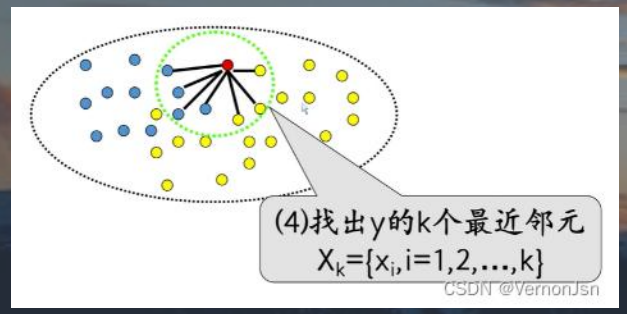
School of CS

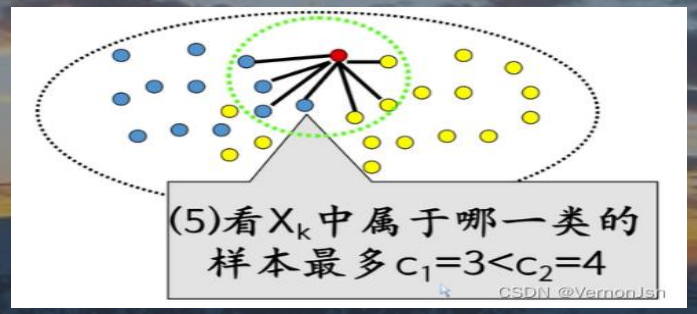
Academic achievement

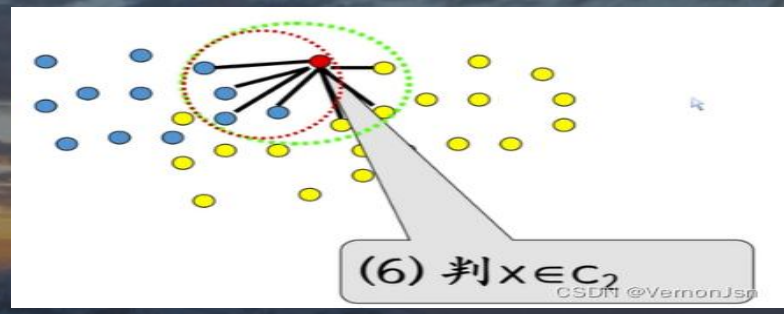












7. SVM 分类方法?

SVM一类按监督学习方式对数据进行二元分类的广义线性分类器,其决策边界是对学习样本求解的最大边距超平面。SVM使用损失函数计算经验风险并加入了正则化项以优化结构风险，是一个具有稀疏性和稳健性的分类器。在分开数据的超平面的两边建有两个互相平行的超平面，分隔超平面使两个平行超平面的距离最大化。线性分类器的间隔：到超平面最近的样本与此超平面之间的距离。具有最大间隔的线性分类器叫做最大间隔线性分类器。其就是一种最简单的SVM(称为线性支持向量机，即LSVM)支持向量:是那些距离超平面最近的点。
实际风险由两部分组成:训练样本的经验风险和置信范围。

7. SVM 分类方法?

与两个量有关 一是样本数量，显然给定的样本数量越大，学习结果越有可能正确，此时置信风险越小；二是分类函数的VC维，显然VC维越小，推广能力越好，置信风险会变小。N维实数空间中线性分类器和线性实函数的VC维是n+1。VC维的通俗定义是：对于一个指示函数集，如果存在H个样本能够被函数集中的函数按所有可能的 2^H 种形式分开，则称函数集能够将 H个样本打散，函数集的VC维就是它能够打散的最大样本数目Hmax。Plus-plane={x:w.x+b=+1}Minus-plane={x:w.x+b=-1} x正 = x负 +λw 把x正代入正平面，得到λ=2/w · W，可以得到M=|x正-x负|=|λw|=λ|w|=λ根号w · w=2/根号w · w。需要通过一种算法求出w与b使得间隔M最大。如梯度下降、退火算法等。可以表示成约束优化问题：最小化上式即可 核函数：支持向量机通过某非线性变换φ(x)，将输入空间映射到高维特征空间。特征空间的维数可能非常高。如果支持向量机的求解只用到内积运算，而在低维输入空间又存在某个函数 K(x, x′)，它恰好等于在高维空间中这个内积，那么支持向量机就不用计算复杂的非线性变换，而由这个函数 K(x, x′) 直接得到非线性变换的内积使大大简化了计算。这样的函数 K(x, x’) 称为核函数。用的核函数有：线性核函数，多项式核函数，径向基核函数，Sigmoid核函数和复合核函数，傅立叶级数核，B样条核函数和张量积核函数

8. 在样本集频繁更新的情况下，哪种分类方法更适合?

KNN

9. 主要的聚类方法有哪些？各有什么特点？

基于密度的方法：只要临近区域的密度超过某个阈值，就继续聚类。避免仅生成球状聚类。

基于层次的方法：层次的方法对给定数据集合进行层次的分解。根据层次的分解如何形成，层次的方法可以被分为凝聚或分裂方法。层次聚类方法的优点在于可以在不同粒度水平上对数据进行探测，而且容易实现相似度量或距离度量。缺点：单纯的层次聚类算法终止条件含糊，而且执行合并或分裂簇的操作后不可修正，这很可能导致聚类结果质量很低。由于需要检查和估算大量的对象或类才能决定类的合并或分裂，所以这种方法的可扩展性较差。

基于网格的方法：基于网格的方法把对象空间量化为有限数目的单元，所有的聚类操作都在这个量化的空间上进行。这种方法的主要优点是它的处理速度很快。

基于模型的方法：为每个簇假设一个模型，发现数据对模型的最好匹配。

基于划分的方法：在中小规模的数据库中发现球状簇很适用

11. 基于密度的聚类方法？能够针对实际数据给出求解步骤。

基于密度聚类：
{2，4，8，6，5，3，14，18，16，20，25，26}
已知r=2，miniptS=3，根据基于密度聚类算法给出计算过程
聚类结果：{2,3,4,5,6,8}{18,16,20} 噪音：14,25,26（这三个周边邻点数少于3）
不需要已知类的个数。能够很好地找到任意大小和任意形状的簇。簇的密度不同时，它的表现不如其他聚类算法。这个缺点也会在非常高维度的数据中出现，因为距离阈值ε再次变得难以估计。

步骤	当前对象	邻域范围	N（等条件延展）	新类C	未标记对象
0					2，4，8，6，5，3，14，18，16，20，25，26
1	2	[0,4]	{2,3,4,5,6,8}	{2,3,4,5,6,8}	14，18，16，20，25，26
2	14	[12,16]	没有足够的点	噪音	18，16，20，25，26
3	18	[16,20]	{18,16,20,14}	{18,16,20}	25，26
4	25	[23,27]		噪音	26
5	26	[24,28]		噪音	结束

12. 层次聚类方法？基于分裂和聚合的聚类方法主要异同？能够针对实际数据给出求解步骤。

聚集/分裂的层次聚类。聚合：将样本集中的所有的样本点都当做一个独立的类簇；repeat：2.计算两两类簇之间的距离，找到距离最小的两个类簇c1和c2；3.合并类簇c1和c2为一个类簇 util： 达到聚类的数目或者达到设定的条件将样本集中的所有的样本归为一个类簇。分裂：1.归为一个类簇，repeat：2.同一个类簇中找出最远的a,b，3.将其分配到不同的类簇c1,c2中，4.计算距离，若dis(a)<dis(b)，则归到c1否则c2，until达到聚类的数目或者达到设定的条件。

在于可以在不同粒度水平上对数据进行探测，而且容易实现相似度量或距离度量。
优点：可解释性好，缺点：单纯的层次聚类算法终止条件含糊，而且执行合并或分裂簇的操作后不可修正，这很可能导致聚类结果质量很低。由于需要检查和估算大量的对象或类才能决定类的合并或分裂，所以这种方法的可扩展性较差。K-Means 线性复杂度O(n)，层次聚类为 $O(n^3)$

举例：

分裂的层次聚类：

步骤	类别	最远的样本
1	{2，3，4，10，11，12，20，25}	2--25
2	{2，3，4，10，11，12}，{20，25}	2--12,20--25
3	{2，3，4,} {10，11，12} {20}，{25}	2-4,10-12
4	{2，3}{4} {10，11} {12} {20} {25}	2--3,10-11
5	{2} {3} {4} {10} {11} {12} {20} {25}	

聚合的层次聚类：

PPT上没有说具体的距离计算函数我们暂且把质心的距离当做聚类之间的距离。

步骤	类别
1	{2} {3} {4} {10} {11} {12} {20} {25}
2	{2，3} {4} {10,11} {12} {20} {25}
3	{2，3，4} {10，11，12} {20} {25}
4	{2，3，4} {10，11，12} {20，25}
5	{2，3，4，10，11，12} {20，25}
6	{2，3，4，10，11，12，20，25}

2. BPF 捕包原理是什么？

伯克利数据包过滤器（BPF，Berkeley Packet Filter）是一个高效的数据包捕获机制，工作在操作系统的内核层，主要由网络转发部分和数据包过滤两部分组成。

网络转发部分是从链路层捕获数据包并把它们转发给数据过滤部分，数据包过滤部分是从接收到的数据包中接收过滤规则决定的网络数据包，其他数据包被丢弃。在操作系统的内核中完成，效率很高。使用了数据缓存机制，使捕获数据包缓存在内核中，达到一定数量再传递给应用程序。（如libpcap）

3. 被动捕包程序的主要流程是什么？

被动捕包程序的主要流程：

1. 把网卡等同于文件进行I/O
2. 从网卡中读取数据
3. 处理获取的数据
4. 释放I/O资源

①查找网卡：Find all devices()②打开网卡：open ()
③监听：loop()④数据回传给用户变量
⑤转用户程序执行：Handler()

4. IP 首部主要包含哪些信息？在数据包分析时如何提取？TCP 首部主要包含哪些信息？在数据包分析时如何提取？

解析IP/TCP数据包：在pcap_loop中自定义回调函数，callback->IP处理(物理帧后14字节)->TCP处理(物理帧后20+14字节)

IP处理函数中执行ip_protocol =(struct ip_header*) (packet_content + 14); ip_protocol->ip_version（ip版本）/ip_header_length（头部长度）…；TCP处理函数中执行tcp_protocol=(struct tcp_header*)(packet_content + 14 + 20); tcp_protocol->sport(源端口)/dport(目的端口)等属性…UDP包有源/目的端口，数据包长度，校验值和数据组成。

版　本	头部长度	服务类型	总　长
标　识			分段偏移
生存期		协议	头部校验和
源 IP 地址			
目的 IP 地址			
选　项			

5. 大流量网络环境下如何提高捕包系统的性能?

旁路监测技术1. 数据分流；2. 数据汇聚；3. 数据耦合方法（多机负载均衡/分流器）；4. 协议还原；5. 内容抽取。

a. 系统的并行架构设计

b. 报文抓取过程优化（零拷贝技术、DPDK等等）

6. 如何提高单机捕包能力？零拷贝捕包主要解决哪些核心问题?

提高单机捕包能力：减少内核态中无用内存操作，如数据拷贝/提高硬件性能/减少网卡中断次数/减少系统调用次数。

关键技术：1 DMA技术：把数据从网卡拷贝到主存的过程中，CPU完全不需要参与。不会有任何CPU资源消耗，带来的损耗就可以避免。
2 共享内存技术：内核中的某块内存区域被设定为网卡DMA映射的内存区，用户进程通过虚拟地址映射（server进程）关联到该区域。难点：修改内核和网卡驱动，并重新绑定。

零拷贝技术
一个网络数据包通过网卡进入内核，然后再进入用户空间，至少经过2次data copy。先从网卡到内核，再到用户空间，这个拷贝过程仅仅起到“传输”作用，而数据包内容没有任何变化。零拷贝技术的核心：取消这两次拷贝，网卡直接到用户缓冲区。

7. 网站信息爬取的主要思路是什么？用伪代码描述单机网页信息爬取算法。

网络信息爬取的主要思路
（1）选取一部分种子URL放入队列。
（2）取出URL，解析DNS，得到主机ip，下载存储入库。此外，将这些URL放进已抓取URL队列。
（3）分析已抓取URL队列中的URL得到新URL，进入
下一个循环。

```
DS:
Url_QUEUE  uqueue;
History_LIST  hlist;
PROCEDURE:
Crawler（seed_url）                    // seed_url是起始URL队列集合
{  in_queue (uqueue , seed_url);
   while   (u=out_queue(uqueue) )          //从uqueue队列中移除url地址
   {
   wpage=http_get(u);                  // 下载网页
     save wpage;                       // 保存网页
     for each url in wpage             // 解析网页中的URL，看是否被访问过
        { if  url not in hlist
          then in_queue (uqueue , url); }   // 未被访问加入到uqueue队列中
     }
}
```

8. 利用多机爬取多个网站的信息，核心解决哪些问题？主要策略有哪几种？用伪代码描述多机单网站网页信息爬取算法?

多机爬取利用多机爬取多个网站的信息
核心解决 Requests队列集中管理/去重集中管理问题

主要策略有深度优先遍历策略/广度优先遍历策略/最佳优先搜索策略

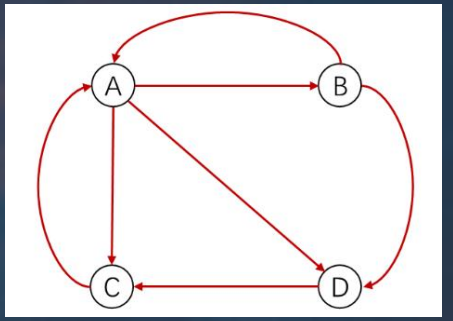
```
DS:
Url_QUEUE  uqueue;
History_LIST  hlist;
PROCEDURE:
Crawler  (seed_url)                    // seed_url是起始URL队列集合
{  in_queue (uqueue , seed_url);
   while   (u=out_queue(uqueue) )          //从uqueue队列中移除url地址
   {
     wpage=http_get(u);                  // 下载网页
     save wpage;                        // 保存网页
     for each url in wpage               // 解析网页中的URL，看是否被访问过
        { if  url not in hlist
          then{
               n=hash  (url )  mod  N;  // N为节点机总个数
               if  (n==my_node_ID)
               then in_queue (uqueue , url);
                  else  sendurl(url,n);  // 将url发到节点机n，并插入到n的uqueue
               }
        }  // 未被访问加入到uqueue队列中
   }
}
```

9. 简述基于主动获取技术的网站信息监测系统的构成。

网络信息搜索系统由搜索器、索引器、检索器和用户接口等四个部分组成。
搜索器的功能是在互联网中漫游，发现和搜集信息。它要尽可能多、尽可能快地搜集各种类型的新信息，同时因为互联网上的信息更新很快，所以还要定期更新已经搜集过的旧信息，以避免死连接和无效连接。
索引器的功能是理解搜索器所搜索的信息，从中抽取出索引项，用于表示文档以及生成文档库的索引表。
检索器的功能是根据用户的查询在索引库中快速检出文档，进行文档与查询的相关度评价，对将要输出的结果进行排序，并实现某种用户相关性反馈机制。检索器常用的信息检索模型有集合理论模型、代数模型、概率模型和混合模型四种。用户接口的作用是输入用户查询、显示查询结果、提供用户相关性反馈机制。主要的目的是方便用户使用搜索引擎，高效率、多方式地从搜索引擎中得到有效、及时的信息。
用户接口的设计和实现使用人机交互的理论和方法，以充分适应人类的思维习惯

10.简述 Pagerank 算法的主要思路。

在互联网上，如果一个网页被很多其它网页所链接，说明它受到普遍的承认和信赖，那么它的排名就高。连接流行度PageRank根据网站的外部链接和内部链接的数量和质量俩衡量网站的价值。
算法：如果网页T存在一个指向网页A的连接，则表明T的所有者认为A比较重要，从而把T的一部分重要性得分赋予A。1. 在初始阶段：网页通过链接关系构建起Web图，每个页面设置相同的PageRank值，通过若干轮的计算，会得到每个页面所获得的最终PageRank值。随着每一轮的计算进行，网页当前的PageRank值会不断得到更新。2. 在一轮中更新页面PageRank得分的计算方法：在一轮更新页面PageRank得分的计算中，每个页面将其当前的PageRank值平均分配到本页面包含的出链上，这样每个链接即获得了相应的权值。而每个页面将所有指向本页面的入链所传入的权值求和，即可得到新的PageRank得分。当每个页面都获得了更新后的PageRank值，就完成了一轮PageRank计算 如果网页T存在一个指向网页A的连接，则表明T的所有者认为A比较重要，从而把T的一部分重要性得分赋予A。这个重要性得分值为：PR(T)/C(T)。其中，PR(T)为T的PageRank值，C(T)为T的出链数；A的PageRank值为一系列类似于T的页面重要性得分值的累加。优点：是与查询无关的静态算法，所有网页的PageRank值通过离线计算获得，有效减少了在线查询时的计算量，极大降低了查询响应时间。不足：PageRank忽略了主题相关性，导致结果的相关性和主题性降低。另外，PageRank有很严重的对新网页的歧视。



两个重要假设

1.数量假设:在Web图模型中，如果一个页面节点接收到的其他网页指向的入链数量越多，那么这个页面越重要。2.质量假设:指向页面A的入链质量不同，质量高的页面会通过链接向其他页面传递更多的权重。所以越是质量高的页面指向页面A，则页面A越重要。

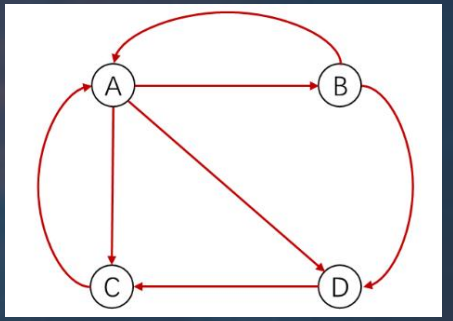
$$PR(A) = \left(\frac{PR(B)}{L(B)} + \frac{PR(C)}{L(C)} + \frac{PR(D)}{L(D)} + \cdots \right) q + 1 - q$$

此处q表示阻尼系数

PR(A)表示节点的PR值即A被访问的概率。B,C,D...表示指向A的节点。L(B),L(C),L(D)表示各自的出度。

PageRank算法的核心思想就是：假设有一个随机游走者，在图上进行随机游走。随机游走者经过多次游走后，对不同的节点有着不同的访问次数。显然，随机游走者访问次数多的节点有着更高的重要性。算法就是在模拟随机游走者在图上的随机游走过程，所以每次算法对各个节点PR值的迭代更新都对应着随机游走者在图上的一次随机游走。这里的q代表查看当前网页，1-q代表拒绝查看当前网页而是随机输入一个网页进行访问。

PS：随机游走者的"游走"体现为：如图 1 示若随机游走者当前所处位置为节点A，那么，随机游走者可以向BCD三个节点进行下一步的游走；随机游走者的"随机"体现在：随机游走者可以按照概率去随机选择下一个要游走到的节点



两个重要假设

每次迭代时，图结构不变，计算各节点的访问概率时，只有对应链接到该节点的对应节点PR值在变动。自然的想到用矩阵来存储图的链接结构，我们称该矩阵为转移矩阵M。如果网页 j 有 k 个出链，那么对于每一个出链所指向的网页 i，其 M[i][j] = 1/k，对于没有被出链指向的网页 t，其 M[t][j]=0。所以对于本题：

有矩阵：

$$M = \begin{bmatrix} 0 & 1/2 & 1 & 0 \\ 1/3 & 0 & 0 & 0 \\ 1/3 & 0 & 0 & 1 \\ 1/3 & 1/2 & 0 & 0 \end{bmatrix}$$

计算公式为PRi=M×PRi-1，初始化：

$$PR_0 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

第一次计算结果为：

$$PR_1 = M \otimes PR_1 = \begin{bmatrix} 0 & 1/2 & 1 & 0 \\ 1/3 & 0 & 0 & 0 \\ 1/3 & 0 & 0 & 1 \\ 1/3 & 1/2 & 0 & 0 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 3/2 \\ 1/3 \\ 4/3 \\ 5/6 \end{bmatrix}$$

1.最终结果收敛时得到A>C>D>B

2.需要图是强连通的，否则最后结果会趋向于0。

3.陷阱问题：有一个节点仅存在指向自己的边，导致结果除了自己外其他PR值均趋向于0.

本页是对没有指数平滑操作前的计算而言，平滑操作后，这些问题都不存在了

11.网页抓取的搜索策略有哪几种？各自适合哪类应用场景？

网页的抓取策略分为深度、广度和最佳优先三种。常见的是广度和最佳方法。

深度：从起始页开始，一个链接一个链接跟踪下去，在很多情况下会导致爬虫的陷入。获取界面所有链接

广度：覆盖尽可能多的网页。但是大量的无关网页将被下载并过滤 算法的效率将变低

最佳：预测候选URL与目标网页的相似度，或与主题的相关性，并选取评价最好的一个或几个URL进行抓取。减少无关网页。它只访问经过网页分析算法预测为“有用”的网页，但是很多相关网页可能被忽略，因为最佳优先策略是一种局部最优搜索算法。

12.试比较主动获取与被动获取技术的异同。

	被动获取	主动获取
是否需要协作	需要提前获得授权	无需协作，web 上只要获取 URL 即可获取
获取范围	根据捕获规则捕获经过网络接口的数据包（受限）	按照一定的规则，自动地抓取万维网（网页）公开发布的信息（不受限）
数据处理实时性	保留符合规则的数据包等待处理，要求对数据处理具有良好的实时性	按照要求采集处理数据，并在本地存储数据，可以后续进行处理，实时性要求不强
技术指标	1. 看重获取的吞吐量和丢包率，越大越好； 2. 应用场景越广泛越好； 3. 不必要的系统调用越少，被动获取的抓包效率越高，内核中，用户进程频繁的系统调用，消耗大量的资源	1. 着眼于获取的效率，单位时间获取的有效数据越多越好； 2. 网络越稳定越好，稳定性差断网、网络波动、对方服务器的问题等都会导致网站 URL 抓取失败； 3. 遵守相关法律法规的前提下进取信息获取，否则容易带来法律风险和隐私泄露； 4. 适应性越强越好，可以减少经常性的维护措施
技术难点	1. 协议类型多种多样，工作量大； 2. 定义分组过滤规则是一复杂的工作，在配置过滤规则时难免不发生错误和出现漏洞； 3. 分组过滤规则被配置后，难以测试和维护； 4. 频繁的内存读写操作影响速度；	1. 需要适应不同的发布方式，每增加一种发布方式，就需要更新获取模块，即需要对系统进行持续维护，一旦目标结构发生改变，就需要修改代码； 2. 部分系统会有反抓取机制； 3. 海量信息处理，音频信息增多使得爬取难度加大

小作文

School of CS
Academic achievement

这里给出几个热点的题目：

1.web1.0，2.0，3.0各自的安全风险与相互之间的辨析。

2.隐私保护相关的挑战，政策，法律法规。

3.舆情引导相关的技术，原则。

4.P2P网络技术与流量监控。

5.图聚类，Pagerank等相关热点技术

6.国家安全战略

7.社交网络检测审查技术方法

小作文

School of CS
Academic achievement

还需要注意咱们的作业：

针对《国家网络安全战略》中的9项任务，每组选择1项进行扩展阅读，以PPT的形式，总结国家在近5年的相关工作。

1. 社交网络异常传播行为有哪些？如何分类？有何危害？
2. 社交机器人是否合法性？主体责任如何归属？哪些行为不合法？
3. “机器人”与正常用户的本质区别是什么？
4. 有哪些新的异常传播的识别方法？

针对千万量级的模式集，如何加速。

比较AC算法和WM算法的优缺点并给出优化方法。

DBSCAN优化思路

围绕着“清明节加班员工怒怼领导”，利用舆情学知识，请给出相关主体的合理应对建议。

讲座结束
感谢各位观看

Skyrim