黑客攻防

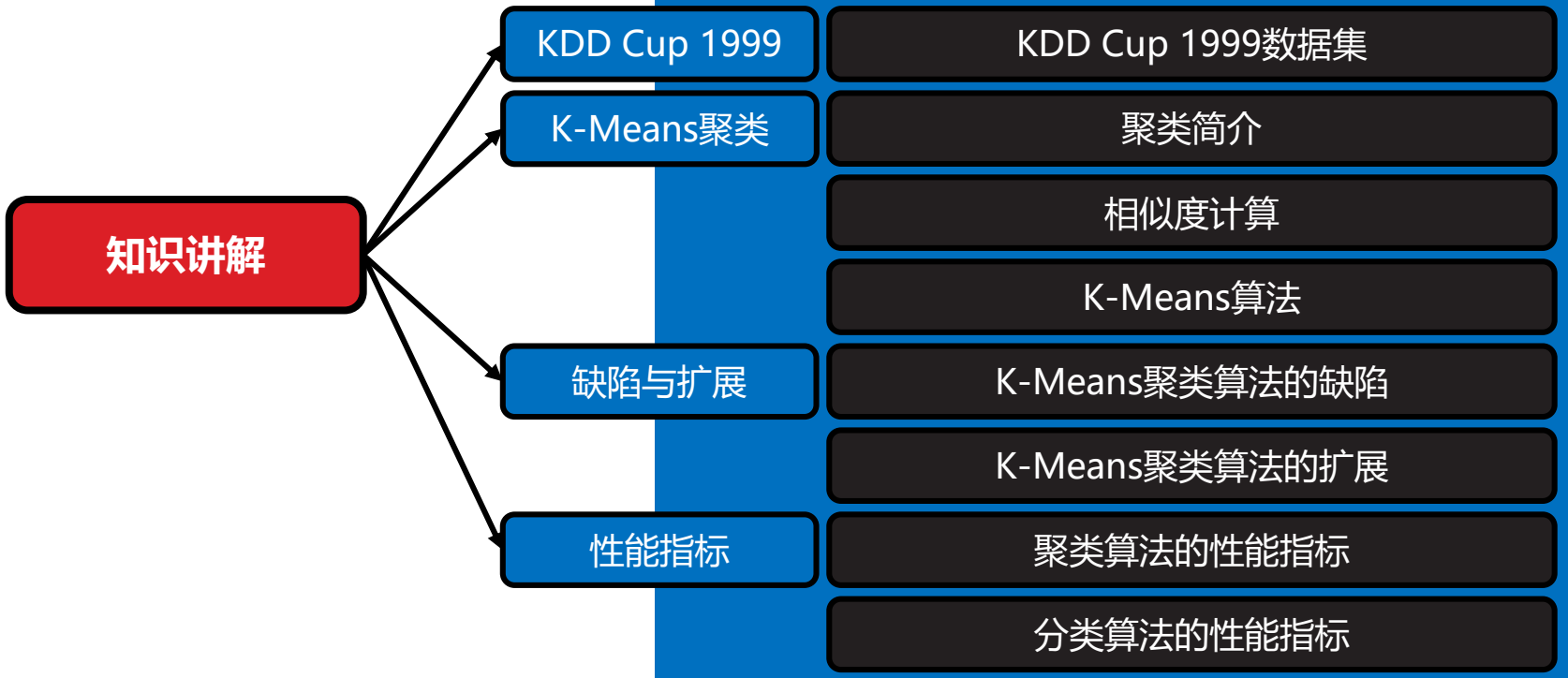
入侵检测

Unit10

内容

上午	09:00 ~ 09:30	作业讲解和回顾
	09:30 ~ 10:20	知识讲解
	10:30 ~ 11:20	
	11:30 ~ 12:00	
下午	14:00 ~ 14:50	实训案例
	15:00 ~ 15:50	
	16:00 ~ 16:50	扩展提高
	17:00 ~ 17:30	总结和答疑

知识讲解


知识讲解
KDD Cup 1999
K-Means聚类
缺陷与扩展
性能指标
KDD Cup 1999数据集
聚类简介
相似度计算
K-Means算法
K-Means聚类算法的缺陷
K-Means聚类算法的扩展
聚类算法的性能指标
分类算法的性能指标

KDD Cup 1999数据集

KDD Cup 1999数据集

知识讲解

- KDD Cup 1999数据集是第三届知识发现与数据挖掘竞赛所使用的数据集
- 源于模拟美国空军军事网络环境下的局域网连续九周原始TCP dump数据
- 共包含大约700万个样本，分为训练集和测试集两部分
- 每个样本包含41个特征，本单元使用其中的20个特征，前三个为标签型特征，其余为数值型特征，如下表所示：

序号	列号	特征	类型
1	2	协议类型	标签
2	3	服务类型	标签
3	4	状态标志	标签
4	5	源到目的的字节数	连续
5	6	目的到源的字节数	连续
6	11	登录失败次数	连续
7	14	是否获得root权限	二值
8	16	获得root权限的次数	连续
9	22	是否以guest身份登录	二值
10	23	两秒钟内连接同一台主机的次数	连续
11	24	两秒钟内连接同一个端口的次数	连续
12	27	REJ错误的比率	连续
13	29	目的端口相同的连接比率	连续
14	30	目的端口不同的连接比率	连续
15	33	目的地址相同，目的端口相同的连接次数	连续
16	34	目的地址相同，目的端口相同的连接比率	连续
17	35	目的地址相同，目的端口不同的连接比率	连续
18	36	目的地址相同，源端口相同的连接比率	连续
19	37	目的地址相同，目的端口相同，源地址不同的连接比率	连续
20	39	目的地址相同，目的端口相同，SYN错误的比率	连续

KDD Cup 1999数据集

- 每个样本对应一个类别，正常或者攻击，攻击分为四种类型：
  - 拒绝服务，如SYN洪泛
  - 监视与探测，如端口扫描
  - 非法获得超级用户权限，如缓冲区溢出
  - 非法入侵，如密码猜测

知识讲解

K-Means聚类算法

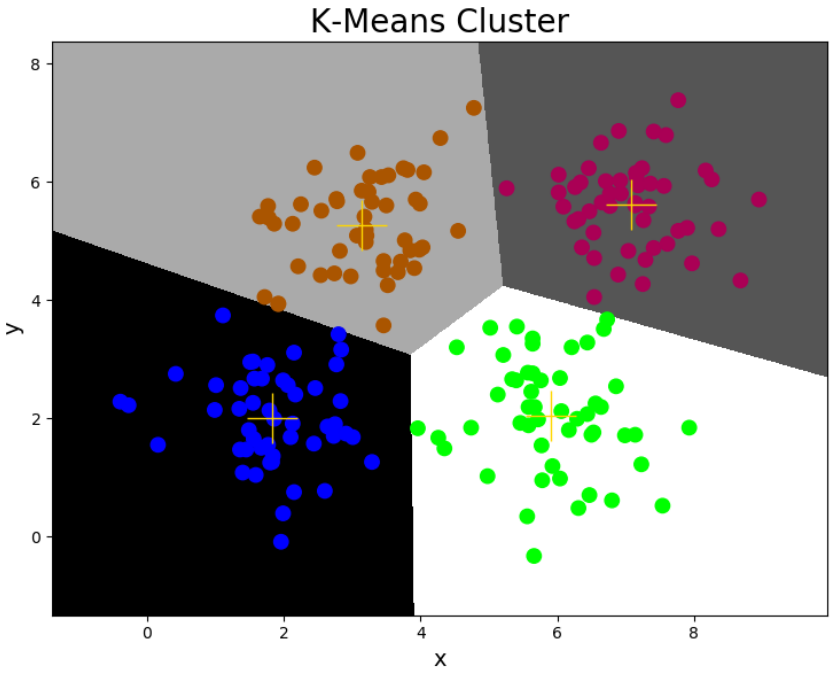
聚类简介

- 根据具体应用的不同，聚类算法可分为以下五类：
  - 基于划分的聚类算法，如K-Means算法
  - 基于层次的聚类算法，如凝聚层次算法
  - 基于密度的聚类算法，如DBSCAN算法
  - 基于网格的聚类算法，如CLIQUE算法
  - 基于模型的聚类算法，如均值漂移算法
- K-Means算法是一种基于划分的聚类方法，即在一个由n个样本组成的样本空间中构建k $(k < n)$个群落，其中每个群落即表示一个聚类。通过划分得到的聚类必须同时满足以下两个条件：
  - 每个聚类至少包含一个样本
  - 每个样本必须隶属于且仅隶属于一个聚类

知识讲解

聚类简介

- 如下图所示：



知识讲解

- 一般而言，判断划分效果优劣的基本原则是：
  - 同一聚类中的样本尽可能地密集，即内密
  - 不同聚类中的样本尽可能地疏远，即外疏
- 同时满足内密外疏的聚类划分就是好的聚类划分，而度量样本疏密的量化指标就是聚类划分的依据，即相似度，如曼哈顿距离、欧几里得距离、闵可夫斯基距离等

相似度计算

- 假设总样本空间由n个样本(记录、对象)组成，每个样本包含m个特征(字段、属性)，其中样本X和样本Y记作：

$$X = (x_1, x_2, \ldots, x_m)$$
$$Y = (y_1, y_2, \ldots, y_m)$$

- 则样本X和样本Y的相似度$D(X,Y)$可用如下距离表示：
  - 曼哈顿距离

$$D_{manhattan}(X,Y) = \sum_{i=1}^{m} |x_i - y_i|$$

  - 欧几里得距离

$$D_{euclidean}(X,Y) = \sqrt{\sum_{i=1}^{m} |x_i - y_i|^2}$$

知识讲解

相似度计算

- 则样本X和样本Y的相似度$D(X,Y)$可用如下距离表示：
  - 闵可夫斯基距离

$$D_{minkowski}(X,Y) = \sqrt[q]{\sum_{i=1}^{m}|x_i - y_i|^q} \quad (q = 1,2,\ldots)$$

  - 其中，q是一个大于或等于1的正整数：
    - $q = 1$的闵可夫斯基距离即曼哈顿距离
    - $q = 2$的闵可夫斯基距离即欧几里得距离

知识讲解

相似度计算

- 不同特征对相似度影响的差异，可通过特征权重加以体现，权重向量记作：

$$W = (w_1, w_2, \ldots, w_m)$$

- 表示样本相似度的三种距离公式为：

$$D_{manhattan}(X, Y, W) = \sum_{i=1}^{m} w_i |x_i - y_i|$$

$$D_{euclidean}(X, Y, W) = \sqrt{\sum_{i=1}^{m} w_i |x_i - y_i|^2}$$

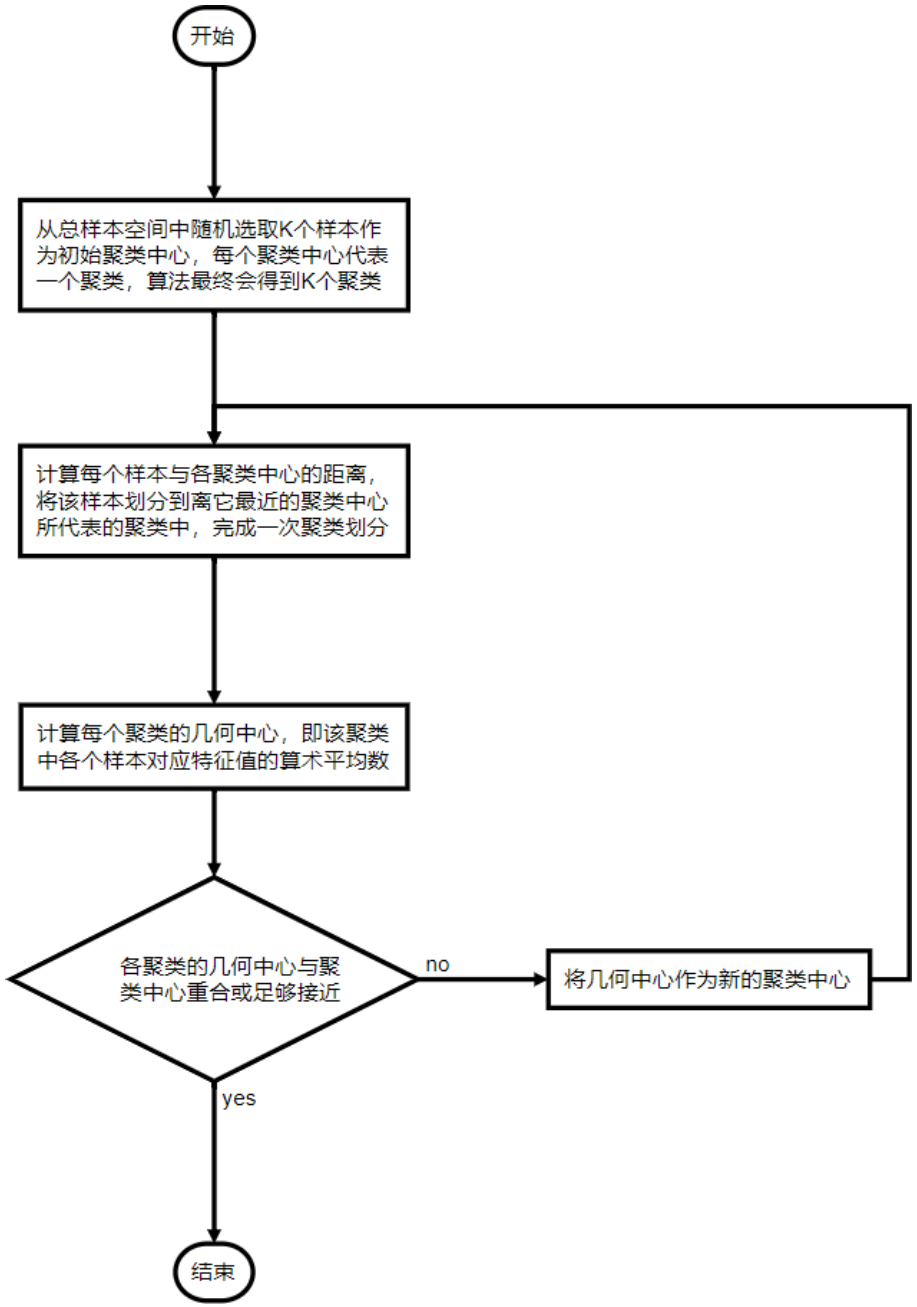
$$D_{minkowski}(X, Y, W) = \sqrt[q]{\sum_{i=1}^{m} w_i |x_i - y_i|^q} \quad (q = 1,2, \ldots)$$

知识讲解

K-Means算法

- K-Means聚类算法流程图如下所示：
  1. 从总样本空间中随机选取K个样本作为初始聚类中心，每个聚类中心代表一个聚类，算法最终会得到K个聚类
  2. 计算每个样本与各聚类中心的距离，将该样本划分到离它最近的聚类中心所代表的聚类中，完成一次聚类划分
  3. 计算每个聚类的几何中心，即该聚类中各个样本对应特征值的算术平均数
  4. 若各聚类几何中心与聚类中心重合或足够接近，则完成聚类，否则将几何中心作为新的聚类中心，返回第2步重新划分聚类

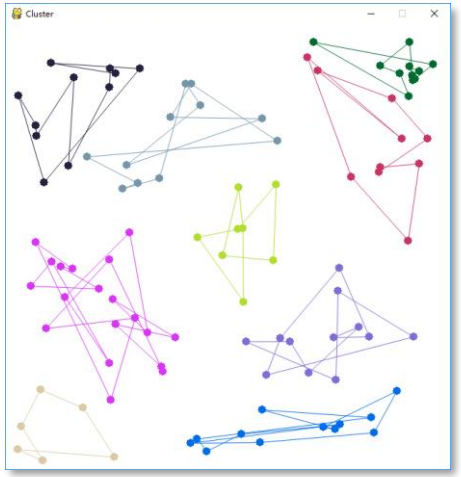
知识讲解



K-Means聚类算法的缺陷与扩展

K-Means聚类算法的缺陷

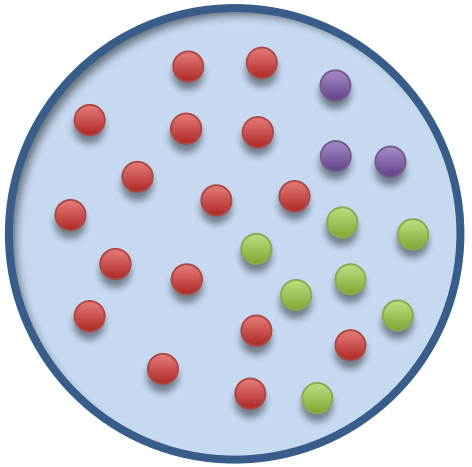
- KDD Cup 1999数据集包含正常样本和四类攻击样本，共计五个聚类，即K=5。但仅将初始聚类中心的个数设置为5并不一定能获得足够理想的聚类效果。究其原因：
  - 同一种类型的攻击，其相似度却可能很低
  - 某些攻击样本与正常样本的相似度甚至高于与同类攻击样本的相似度
- 因此，对KDD Cup 1999数据集应用K-Means算法做聚类分析，无论K值如何选取，仅对数据做一次聚类都是远远不够的



知识讲解

K-Means聚类算法的扩展

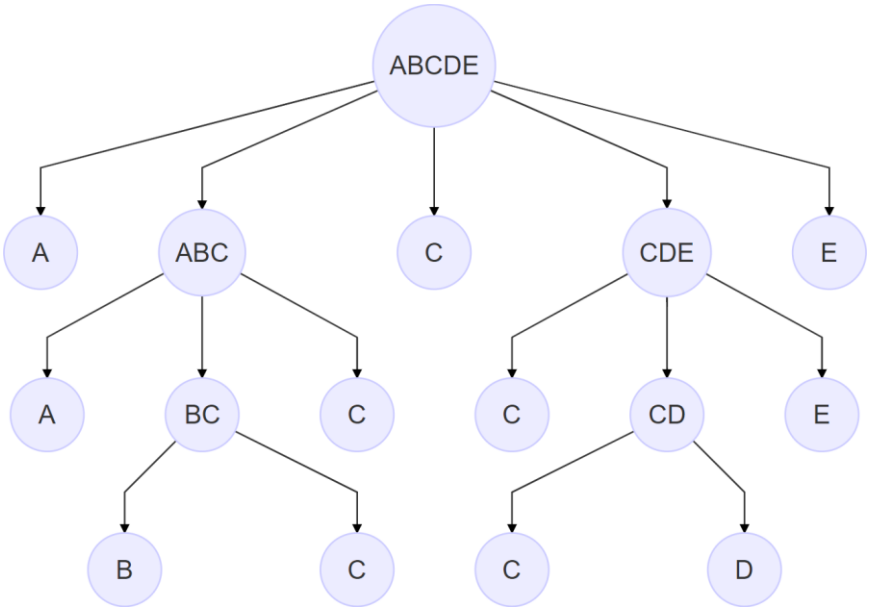
- 聚类不纯度(Cluster Impurity)
  - 理想情况下，一个聚类中应该只包含一种类别的样本，但事实上有可能包含多种类别的样本，即不够纯。聚类不纯度表征了一个聚类不纯的程度。计算过程如下：
    - 统计一个聚类中各个类别的样本数
      - 类别1：10个样本
      - 类别2：100个样本
      - 类别3：1000个样本
    - 在所有类别中找出样本数最多的类别作为该聚类的主类别
      - 类别3：1000个样本
    - 聚类不纯度为非主类别样本数与主类别样本数的比率
      - $CI = \frac{10+100}{1000} = 0.11$



知识讲解

K-Means聚类算法的扩展

- 聚类树(Cluster Tree)
  - 利用K-Means算法将总样本空间划分为K个聚类，计算其中每个聚类的聚类不纯度，如果发现某个聚类的聚类不纯度高于事先给定的上限，则按照其所包含的类别数继续进行聚类，不断重复这个过程，形成一棵聚类树，该树的叶级聚类的聚类不纯度均不高于事先给定的上限。如下图所示：



知识讲解

聚类和分类算法的性能指标

聚类算法的性能指标

- 纯粹的聚类算法属于无监督学习的一种，其用于模型测试的样本通常没有已知的类别标签，因此无法通过预测类别与实际类别之间的符合程度评估性能
- 轮廓系数(Silhouette Score)可以在无需类别标签的前提下，精确地刻画出聚类本身内密外疏的程度
  - 对参与模型测试的每一个样本，其轮廓系数可表示为：$s = \frac{b-a}{\max(a,b)}$
    - 其中，a代表该样本与其所在聚类中其它样本的平均距离；b则表示该样本与其最近的另一个聚类中所有样本的平均距离
    - 好的聚类，内部样本足够密集，聚类之间足够疏远，$a \ll b$，$s \to 1$
    - 坏的聚类，内部样本过于疏远，聚类之间过于密集，$a \gg b$，$s \to -1$
    - 聚类重叠，内部样本和聚类之间的疏密程度很接近，$a \approx b$，$s \to 0$
  - 整个聚类的轮廓系数可用所有测试样本轮廓系数的算术平均数表示：$S = \bar{s}$

知识讲解

分类算法的性能指标

- 混淆矩阵(Confusion Matrix)
  - 对于类别标签已知的测试样本，通常用混淆矩阵表示预测类别与已知类别之间的符合程度。混淆矩阵中的每一行表示一个实际类别的样本数，每一列表示一个预测类别的样本数。如下所示：

	0	1	2	3	4
0	1566	4	16	0	4
1	200	4272	1	0	0
2	6	4	42	0	0
3	0	0	0	2	0
4	100	0	0	0	3

  - 混淆矩阵的主对角线表示实际类别和预测类别一致的样本数。理想混淆矩阵中的所有非零元素，全部集中在主对角线上。换言之，混淆矩阵除主对角线以外的元素数值越小越好，最好都是0

知识讲解

分类算法的性能指标

- 查准率(Precision)
  - 某个特定类别的查准率表示模型预测该类别的正确性，即模型预测该类别样本中正确样本的占比：
    - $p_0 = \frac{1566}{1566+200+6+100} = 0.836538$
    - $p_1 = \frac{4272}{4+4272+4} = 0.998131$
    - $p_2 = \frac{42}{16+1+42} = 0.711864$
    - $p_3 = \frac{2}{2} = 1$
    - $p_4 = \frac{3}{4+3} = 0.428571$
  - 分类模型的查准率可用各个类别查准率的算术平均数表示：
    - $P = \frac{p_0+p_1+p_2+p_3+p_4}{5} = \frac{0.836538+0.998131+0.711864+1+0.428571}{5} = 0.795021$

知识讲解

分类算法的性能指标

- 召回率(Recall)
  - 某个特定类别的召回率表示模型预测该类别的完整性，即该类别样本中被模型正确预测的样本占比：
    - $r_0 = \frac{1566}{1566+4+16+4} = 0.984906$
    - $r_1 = \frac{4272}{200+4272+1} = 0.955064$
    - $r_2 = \frac{42}{6+4+42} = 0.807692$
    - $r_3 = \frac{2}{2} = 1$
    - $r_4 = \frac{3}{100+3} = 0.029126$
  - 分类模型的召回率可用各个类别召回率的算术平均数表示：
    - $R = \frac{r_0+r_1+r_2+r_3+r_4}{5} = \frac{0.984906+0.955064+0.807692+1+0.029126}{5} = 0.755358$

知识讲解

分类算法的性能指标

- F1得分(F1-Score)
  - 某个特定类别的F1得分用两倍该类别查准率、召回率之积与它们的和的比值表示：
    - $f_0 = 2 \times \frac{p_0 \times r_0}{p_0 + r_0} = 2 \times \frac{0.836538 \times 0.984906}{0.836538 + 0.984906} = 0.904679$
    - $f_1 = 2 \times \frac{p_1 \times r_1}{p_1 + r_1} = 2 \times \frac{0.998131 \times 0.955064}{0.998131 + 0.955064} = 0.976122$
    - $f_2 = 2 \times \frac{p_2 \times r_2}{p_2 + r_2} = 2 \times \frac{0.711864 \times 0.807692}{0.711864 + 0.807692} = 0.756757$
    - $f_3 = 2 \times \frac{p_3 \times r_3}{p_3 + r_3} = 2 \times \frac{1 \times 1}{1 + 1} = 1$
    - $f_4 = 2 \times \frac{p_4 \times r_4}{p_4 + r_4} = 2 \times \frac{0.428571 \times 0.029126}{0.428571 + 0.029126} = 0.054545$
  - 分类模型的F1得分可用各个类别F1得分的算术平均数表示：
    - $F = \frac{f_0 + f_1 + f_2 + f_3 + f_4}{5} = \frac{0.904679 + 0.976122 + 0.756757 + 1 + 0.054545}{5} = 0.738421$

知识讲解

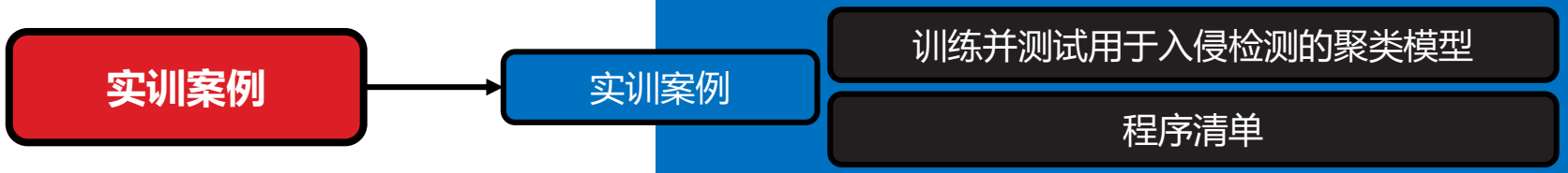
分类算法的性能指标

- 分类报告(Classification Report)
  - 将以上各指标以分类报告的形式加以汇总：

知识讲解

Label	Precision	Recall	F1-Score
0	0.836538	0.984906	0.904679
1	0.998131	0.955064	0.976122
2	0.711864	0.807692	0.756757
3	1	1	1
4	0.428571	0.029126	0.054545
Average	**0.795021**	**0.755358**	**0.738421**

实训案例



实训案例

训练并测试用于入侵检测的聚类模型

- 在Linux系统上编写预处理器和聚类器两个应用程序
  - 预处理器负责对KDD Cup 1999数据集做预处理
  - 聚类器使用K-Means算法对训练数据集进行多次聚类分析，形成聚类树模型
  - 聚类器利用训练所得聚类树模型，对测试数据集中的每条记录进行类别预测

代码实践

程序清单

- 声明Preprocessor类
  - preprocessor.h
- 实现Preprocessor类
  - preprocessor.cpp
- 测试Preprocessor类
  - preprocessor_test.cpp
- 测试Preprocessor类构建脚本
  - preprocessor_test.mak
- 声明KMeans类
  - kmeans.h
- 实现KMeans类
  - kmeans.cpp
- 测试KMeans类
  - kmeans_test.cpp
- 测试KMeans类构建脚本
  - kmeans_test.mak

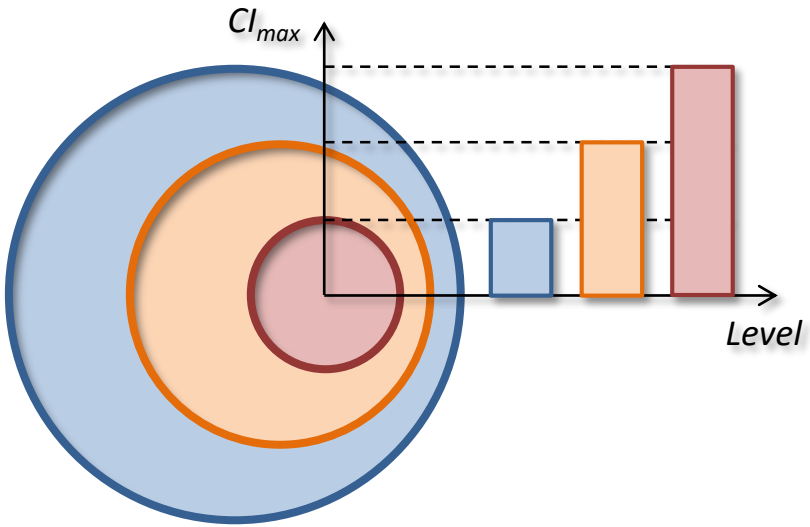
代码实践

扩展提高

聚类不纯度对模型性能的影响

聚类不纯度对入侵检测模型性能的影响

- 聚类不纯度是一个聚类中干扰数据占主类别数据的比例：
  - 聚类不纯度高，意味着不同类别样本混入一个聚类，降低入侵检测模型的准确性
  - 聚类不纯度低，延长聚类树生成的收敛过程，增加入侵检测模型的资源消耗
  - 选择合理的聚类不纯度标准有助于建立高效准确的入侵检测模型
- 聚类层次不同对聚类不纯度的上限也不同，层次越深上限越高



知识讲解

常用入侵检测工具

Snort

- Snort最早可以追溯到1998年由Martin Roesch编写的一款开源入侵检测工具。迄今为止，Snort已经发展为一个跨平台，能够提供实时流量分析和记录网络数据包的入侵检测与防御系统
- Snort具有三种工作模式：
  - 嗅探器：读取网络上的数据包，以连续数据流的形式显示在终端上
  - 记录器：捕获网络上的数据包，以纯文本文件的形式记录在硬盘上
  - 入侵检测：根据可定制的规则，检测网络上的数据包，对符合规则者执行操作
- 通过如下命令可以启动Snort的入侵检测工作模式：
  - snort -d -h 192.168.1.0/24 -l ./log -c snort.conf
    - 其中，snort.conf为描述规则的配置文件。Snort会将捕获到的数据包与配置文件中的每条规则逐一匹配，一旦发现其与某条规则匹配，即执行该规则所描述的操作，同时将符合规则的数据包以文本文件形式保存在-l选项指定的目录中

知识讲解

Snort

- Snort规则包括两个逻辑部分：
  - 规则头：匹配条件，如动作、协议、源和目的地址、源和目的端口等
  - 规则选项：执行操作，如检查内容、收集、报警、丢弃等
- 例如：

```
alert tcp any any -> 192.168.1.0/24 111 (content: "|00 01 86 a5|"; msg: "mountd access")
\_____________________________________/ \_____________________________________________/
              Rule Header                                Rule Options
```

  - 规则头中可以包含多个条件，它们是逻辑与的关系，而配置文件中的多条规则则是逻辑或的关系

知识讲解

PSAD

- 端口扫描攻击探测器(The Port Scan Attack Detector, PSAD)通过对防火墙日志的分析，检测包括端口扫描、拒绝服务攻击等在内的各种可疑流量，并通过自动制定防火墙规则，实现有针对性的防御
- PSAD可以和Snort密切配合，通过Snort检测各种后门程序、拒绝服务工具，以及高级端口扫描器的探测行为。PSAD能够检测出Snort规则集中描述的各种攻击行为
- PSAD能够通过分析TCP SYN数据包，提取发起端口扫描的远程主机指纹
- PSAD还提供数据输出接口。用户可将PSAD输出的数据导入诸如AfterGlow或GnuPlotd等软件中，绘制各种图表，进一步探查发起各种网络攻击的源头

知识讲解

总结和答疑