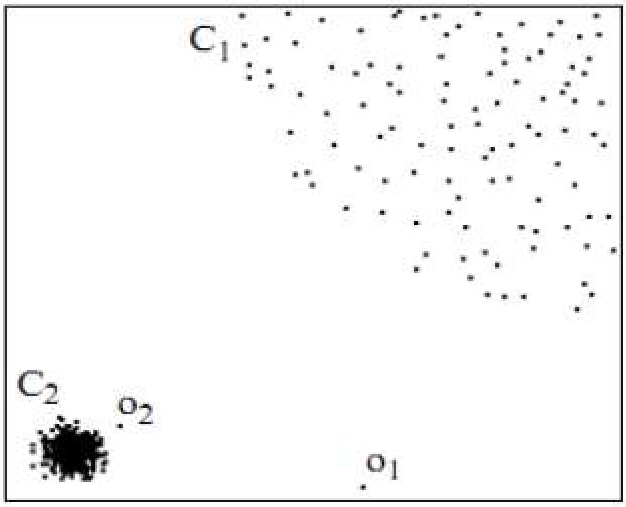
基于LOF的图像异常检测

何为异常

用视觉直观的感受一下如下图所示，对于C1集合的点，整体间距，密度，分散情况较为均匀一致，可以认为是同一簇；对于C2集合的点，同样可认为是一簇。o1、o2点相对孤立，可以认为是异常点或离散点。



异常检测应用场景

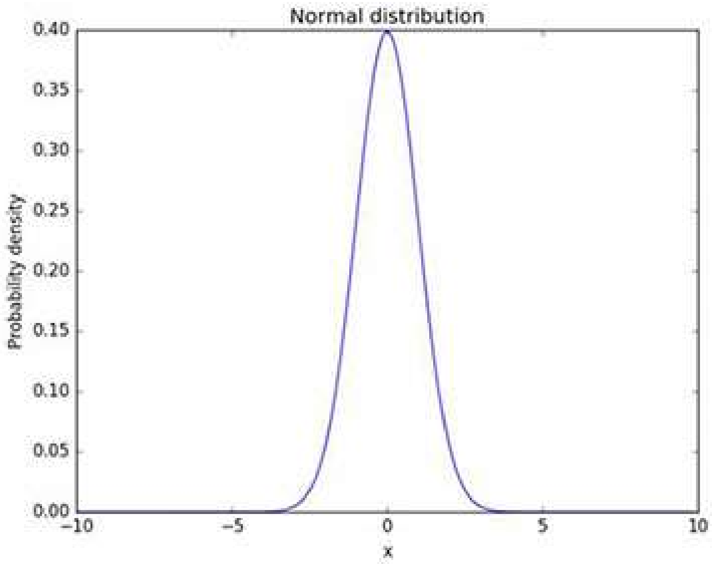
- 1.数据预处理
- 2.病毒木马检测
- 3.工业制造产品检测
  在上述场景中，异常的数据量都是很少的一部分，像SVM、逻辑回归等分类算法都不适用，因为：

> *“监督学习算法适用于有大量的正向样本，也有大量的负向样本，有足够的样本让算法去学习其特征，且未来新出现的样本与训练样本分布一致。”*

异常检测算法

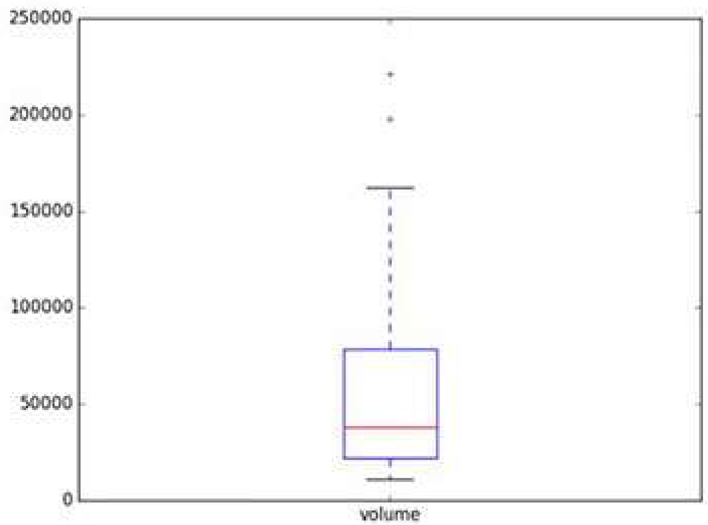
- **1.基于统计与数据分布**
  假设数据集应满足正态分布,即：

$$P(x;\mu,\sigma)=\frac{1}{\sqrt{2\pi\sigma^2}}exp(-\frac{(x-\mu)^2}{2\mu^2}),\quad x\in[-\infty;\infty]$$



给定一个新的数据x，如果x的值大于4或者小于-4，都可以认为是异常值。

- **2.箱线图分析**
  股票成交量的箱线图如下图所示。



  大体可知，该股票在成交量小于20000，或者成交量大于80000时，就应该提高警惕了。
- **3.基于距离/密度**
  典型的算法是：“局部异常因子算法-Local Outlier Factor”，该算法通过引入“k-distance，第k距离”、“k-distance neighborhood，第k距离邻域”、“reach-distance，可达距离”、以及“local reachability density，局部可达密度 ”和“local outlier factor，局部离群因子”，来发现异常点。
- **4.基于划分思想**
  典型的算法是“孤立森林， Isolation Forest”，其思想是：

> 假设我们用一个随机超平面来切割（split）数据空间（data space），切一次可以生成两个子空间（想象拿刀切蛋糕一分为二）。之后我们再继续用一个随机超平面来切割每个子空间，循环下去，直到每子空间里面只有一个数据点为止。直观上来讲，我们可以发现那些密度很高的簇是可以被切很多次才会停止切割，但是那些密度很低的点很容易很早的就停到一个子空间了。

我们考虑使用“基于距离/密度的检测算法”：LOF

LOF局部异常因子算法

LOF是一种基于密度的异常检测算法

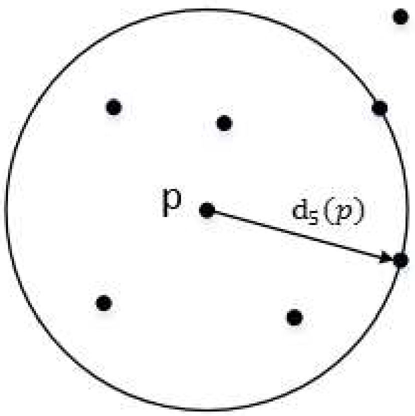
LOF算法相关定义

1. $d(p,o)$:两点p和o之间的距离

$$d(p,o) = \parallel p - o \parallel_2$$

2. k-distance：第K距离

对于点p的第k距离 $d_k(p) = d(p,o)$,并且满足:
a)在集合中至少有不包括p在内的k个点 $o' \in C\{x \neq p\}$,满足 $d(p,o') \le d(p,o)$;
b)在集合中最多有不包括p在内的k-1个点 $o' \in C\{x \neq p\}$,满足 $d(p,o') < d(p,o)$; p的第k距离，也就是距离p第k远的点的距离，不包括p。 如下图。

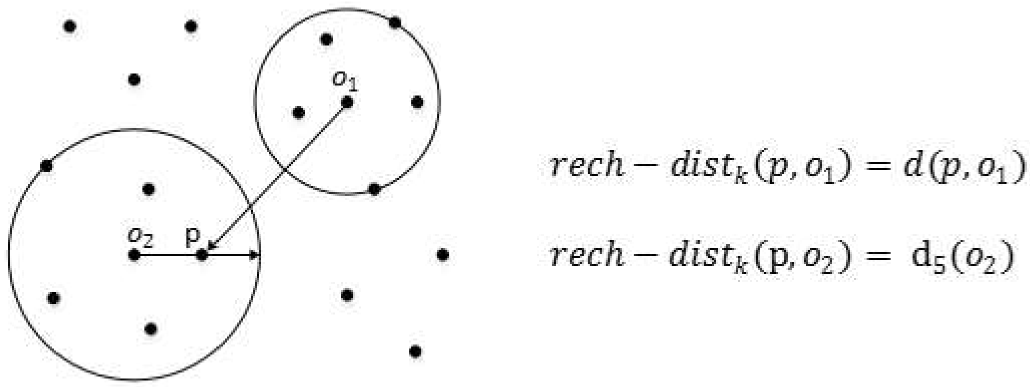


3.k-distance neighbourhood of p:第k距离邻域

点p的第k距离邻域 $N_k(p)$ ，就是p的第k距离即以内的所有点，包括第k距离。因此p的第k邻域点的个数 $|N_K(p)| \geq k$.

4.reach-distance:可达距离

点o到dianp的第k可达距离定义为： $reach-distance_k(p,o) = max\{k-distance(o), d(p,o)\}$ 也就是，点o到点p的第k可达距离，至少是o的第k距离，或者为o、p的真实距离。这也意味着，距离o最近的k个点，o到它们的可达距离被认为相等，且都等于 $d_k(o)$ 。如下图，o1到p的第5可达距离为 $d(p,o_1)$, o_2 到p的第5可达距离为 $d_5(o_2)$ 。



5.local reachability density:局部可达密度：

点p的局部可达密度表示为：

$$lrd_k(p) = 1/\frac{\sum_{o \in N_k(p)} reach-dist_k(p,o)}{|N_k(p)|}$$

表示点p的第k邻域内点到p的平均可达距离的倒数。 注意，是p的邻域点 $N_k(p)$ 到p的可达距离，不是p到 $N_k(p)$ 的可达距离，一定要弄清楚关系。并且，如果有重复点，那么分母的可达距离之和可能为0，则会导致lrd变为无限大，下面还会继续提到这一点。
这个值的含义可以这样理解，首先这代表一个密度，密度越高，我们认为越可能属于同一簇；密度越低，越可能是离群点。如果p和周围领域点是同一簇，那么可达密度越可能为较小的 $d_k(o)$ ，导致可达距离之和较小，密度值较高；如果p和周围邻居点较远，那么可达距离可能都会取较大值 $d(p,o)$ ，导致密度较小，越可能是离群点。

6.local outlier factor:局部离群因子

点p的局部离群因子表示为：

$$LOF_k(p) = \frac{\sum_{o \in N_k(p)} \frac{lrd_k(o)}{lrd_k(p)}}{|N_k(p)|} = \frac{\sum_{o \in N_k(p)} lrd_k(o)}{|N_k(p)|} / lrd_k(p)$$

表示点p的邻域点 $N_k(p)$ 的局部可达密度与点p的局部可达密度之比的平均数。
如果这个比值越接近1，说明p的其邻域点密度差不多，p可能和邻域同属一簇；如果这个比值越小于1，说明p的密度高于其邻域点密度，p为密集点；如果这个比值越大于1，说明p的密度小于其邻域点密度，p越可能是异常点。

监控图像异常检测

何为异常图像

下图中第一行为摄像头正常工作时拍摄的图片，第二行为摄像头转动，单色等异常情况下拍摄到的图片，即为异常图像。

图像特征

无论是对图像中物体的识别，还是图像异常识别都需要用一些特征对图像进行描述，进而根据特征之间的共性和差异来识别图像。
这里我们提取了图像的颜色矩特征。在提取特征之前，我们先将图像从RGB空间转换到HSV空间。

$$V \leftarrow \max(R, G, B)$$

$$S \leftarrow \begin{cases} \frac{V-\min(R,G,B)}{V} & \text{if } V \neq 0 \\ 0 & \text{otherwise} \end{cases}$$

$$H \leftarrow \begin{cases} 60(G-B)/(V-\min(R,G,B)) & \text{if } V = R \\ 120 + 60(B-R)/(V-\min(R,G,B)) & \text{if } V = G \\ 240 + 60(R-G)/(V-\min(R,G,B)) & \text{if } V = B \end{cases}$$

If $H < 0$ then $H \leftarrow H + 360$. On output $0 \leq V \leq 1, 0 \leq S \leq 1, 0 \leq H \leq 360$.

图像的颜色矩一共需要9个分量(3个颜色分量，每个分量上3个低阶矩)。

$$E_i = \frac{1}{N}\sum_{j=1}^{N} p_{ij}$$

$$\sigma_i = (\frac{1}{N}\sum_{j=1}^{N}(p_{ij}-E_i)^2)^{\frac{1}{2}}$$

$$s_i = (\frac{1}{N}\sum_{j=1}^{N}(p_{ij}-E_i)^3)^{\frac{1}{3}}$$

注：公式中，N 表示图片中的总的像素数，p_{ij} 表示第 j 个像素在第 i 个颜色通道上的像素值，E_i 表示第 i 个颜色通道上所有像素的均值，σ_i 表示第 i 个颜色通道上所有像素的标准差，s_i 表示第 i 个颜色通道上所有像素的斜度(skewness)的3次方根。

从图像中提取的特征值如下，每行代表一张图像。

```
[ 30.17  36.96 106.27  22.79  24.66  62.56  32.07  35.77  75.64]
[ 28.85  37.26 107.4   21.97  23.34  53.53  32.5   33.34  68.65]
[ 37.12  42.56 102.93  34.64  31.28  60.05  44.65  42.93  74.1 ]
[ 34.56  41.42 101.71  30.92  28.62  59.78  40.66  40.35  73.91]
[ 37.15  40.7  100.76  32.52  28.17  60.05  39.79  38.91  74.27]
[ 29.73  35.84 104.79  23.79  22.15  57.57  33.65  32.45  71.56]
[ 31.01  37.82 102.27  25.53  24.99  60.71  35.59  35.01  75.4 ]
[ 31.21  37.75 103.17  25.46  23.33  53.22  35.4   31.64  68.62]
[ 34.34  36.41 105.24  30.87  23.45  56.33  39.83  32.96  70.94]
[ 34.89  41.46 101.77  28.53  23.37  63.5   36.28  33.37  76.71]
```

LOF局部离群因子计算

instance 为一条待测试的样本数据，instances 为整个数据集，K为计算lof值时的邻域点数。instance的lof值为：

```
l = LOF(instances)
value = l.local_outlier_factor(k=10, instance)
```

根据上述所介绍的LOF理论可知：value越接近1，instance和其邻域同属一簇；value越小于1，instance为密集点；value越大于1，instance越可能是异常点。
这里，我们将t作为一个阈值，若value>t，则我们认为该数据为异常数据。
为了找出最佳的t，我们将t取为1.0~3.2,差值为 0.2的等差数列：

```
t = np.arange(1.0, 3.2 0.2)
```

并且得到不同t下的预测正确率：

```
k: 10 t: 1.000000 训练集正确率为: 0.384615  召回率: 35/99 = 0.353535  特效度: 5/5 = 1.000000
k: 10 t: 1.200000 训练集正确率为: 0.807692  召回率: 79/99 = 0.797980  特效度: 5/5 = 1.000000
k: 10 t: 1.400000 训练集正确率为: 0.923077  召回率: 91/99 = 0.919192  特效度: 5/5 = 1.000000
k: 10 t: 1.600000 训练集正确率为: 0.961538  召回率: 95/99 = 0.959596  特效度: 5/5 = 1.000000
k: 10 t: 1.800000 训练集正确率为: 0.990385  召回率: 98/99 = 0.989899  特效度: 5/5 = 1.000000
k: 10 t: 2.000000 训练集正确率为: 0.990385  召回率: 98/99 = 0.989899  特效度: 5/5 = 1.000000
k: 10 t: 2.200000 训练集正确率为: 0.990385  召回率: 99/99 = 1.000000  特效度: 4/5 = 0.800000
k: 10 t: 2.400000 训练集正确率为: 0.990385  召回率: 99/99 = 1.000000  特效度: 4/5 = 0.800000
k: 10 t: 2.600000 训练集正确率为: 0.990385  召回率: 99/99 = 1.000000  特效度: 4/5 = 0.800000
k: 10 t: 2.800000 训练集正确率为: 0.990385  召回率: 99/99 = 1.000000  特效度: 4/5 = 0.800000
k: 10 t: 3.000000 训练集正确率为: 0.990385  召回率: 99/99 = 1.000000  特效度: 4/5 = 0.800000
```

其中，数据集共有104张图片，正常图片为99张，异常图片为5张。正确率为算法对整个数据集的识别能力，召回率为算法对正类样本的识别能力，召回率为算法对负类样本的识别能力。

参考文献

异常点/离群点检测算法——LOF

异常检测概述

LOF算法实现