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，小象学院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，我们将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：大数据分析挖掘
  - 新浪微博：ChinaHadoop

分布式爬虫

大纲

- Daemon
- Text-Tag Ration
- K-Means
- 标签模板
- PyGoose

Daemon

Daemontool

Spider always get crash or halt so we need a daemon tool to monitor its status, start it automatically (can be controlled by master), especially restart it when it’s crashed

daemontools is a collection of tools for managing UNIX services.supervise monitors a service. It starts the service and restarts the service if it dies. Setting up a new service is easy: all supervise needs is a directory with a run script that runs the service.

Installation

http://cr.yp.to/daemontools/install.html

Installation

Create a /package directory:

mkdir -p /package

chmod 1755 /package

cd /package

Download daemontools-0.76.tar.gz into /package. Unpack the package:

wget http://cr.yp.to/daemontools/daemontools-0.76.tar.gz

gunzip daemontools-0.76.tar

tar -xpf daemontools-0.76.tar

rm -f daemontools-0.76.tar

Installation & Config

vim src/conf-cc

append -include /usr/include/errno.h to end of gcc command, like this gcc -O2 -Wimplicit -Wunused -Wcomment -Wchar-subscripts -Wuninitialized -Wshadow -Wcast-qual -Wcast-align -Wwrite-strings -include /usr/include/errno.hCompile

set up the daemontools programs:

package/install

Add csh -cf '/command/svscanboot &' to /etc/rc.local (a soft link to /etc/rc.d/rc.local)

Create service

```
# mkdir /root/spider
# vi /root/spider/run
```

Add below lines to run file

```
#!/bin/sh
exec service salt-minion start
```

```
# chmod 1755 /root/spider
# chmod 755 /root/spider/run
# ln -s /root/spider /service/spider
```

Once soft link is created (i.e. spider script is added to system service folder), the service is immediately started by daemontool

文本长度分析

HTML 中的换行

HTML 是标签解释的超文本，本身是没有换行的，也就是说，在HTML文本里的换行，显示的时候并不会被作为单独的一行被显示，所有的换行都是依靠行级元素、块级元素以及
来实现的

```
<!DOCTYPE html>
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
  <title>Title</title>
</head>
<body>
  <p>This is a test
  this is on a new line</p>
  <p>Here on the same line </p><p>Here on the same line</p>

  <p>参考消息网3月25日报道 外媒称，菲律宾总统杜特尔特23日说他愿与中国共享海牙国际仲裁法庭已"裁决判属"马尼拉专有的南海水域的资源。</p>
    <p>据法新社3月23日报道，杜特尔特说菲律宾人没有能力自行开采那里的自然资源。
    </p>
</body>
</html>
```

HTML 中的换行

- HTML 是标签解释的超文本，本身是没有换行的，也就是说，在HTML文本里的换行，显示的时候并不会被作为单独的一行被显示，所有的换行都是依靠行级元素、块级元素以及
来实现的
- 一般来说，一段正文，会是连续显示，直到
 或者 <p> 这样的标签
- 在HTM了里大量使用 <p> 来封装正文

HTML 中的换行

考虑下面的文本：This is a test this is on a new line 实际会显示在一行

而下面的两段 Here on the same line 会显示为两行

```
<!DOCTYPE html>
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
  <title>Title</title>
</head>
<body>
  <p>This is a test
  this is on a new line</p>
  <p>Here on the same line </p><p>Here on the same line</p>

  <p>参考消息网3月25日报道 外媒称，菲律宾总统杜特尔特23日说他愿与中国共享海牙国际仲裁法庭已"裁决判属"马尼拉专有的南海水域的资源。</p>
    <p>据法新社3月23日报道，杜特尔特说菲律宾人没有能力自行开采那里的自然资源。</p>
</body>
</html>
```

去除 Javascript 及 CSS

利用 lxml 的 clean 类，能删除HTML 里所包含 css 及 script

```
<script language="javascript" type="text/javascript">if (typeof M !==
"undefined" && typeof M.loadResource === "function")
{M.loadResource("http://js.mafengwo.net/js/cv/js+pageletcommon+pageHea
dUserInfoWWWDark:js++ACnzzGaLog:js+ARecruit:js+ALazyLoad^Z11V^148
9552560.js");}</script>
```

```
from lxml.html import clean

cleaner = clean.Cleaner(style=True, scripts=True, comments=True, javascript=True,
page_structure=False, safe_attrs_only=False)

content = cleaner.clean_html(content.decode('utf-8')).encode('utf-8')
```

出去所有HTML TAG

利用下面的正则表达式，把HTML的TAG和属性也都去除掉，最后只剩下正文部分

```
reg = re.compile("<[^>]*>")
content = reg.sub('', content)
```

```
<li><a href="http://m.sina.com.cn/m/weibo.shtml"
target="_blank"><i class="sina15-ico-client sina15-ico-weibo"></i>新浪微博</a></li>
```

新浪微博

每行的长度及文本密度

新浪首页

我要评论

分享文章

回到顶部

新浪专栏：读名家知天下

想成为专栏作家？戳这里

强势围观！政务微博大事件！

快来看！湖南微博有大事！

最火博文 大家都在看

一大波台湾美食正在向你逼近！

移动客户端

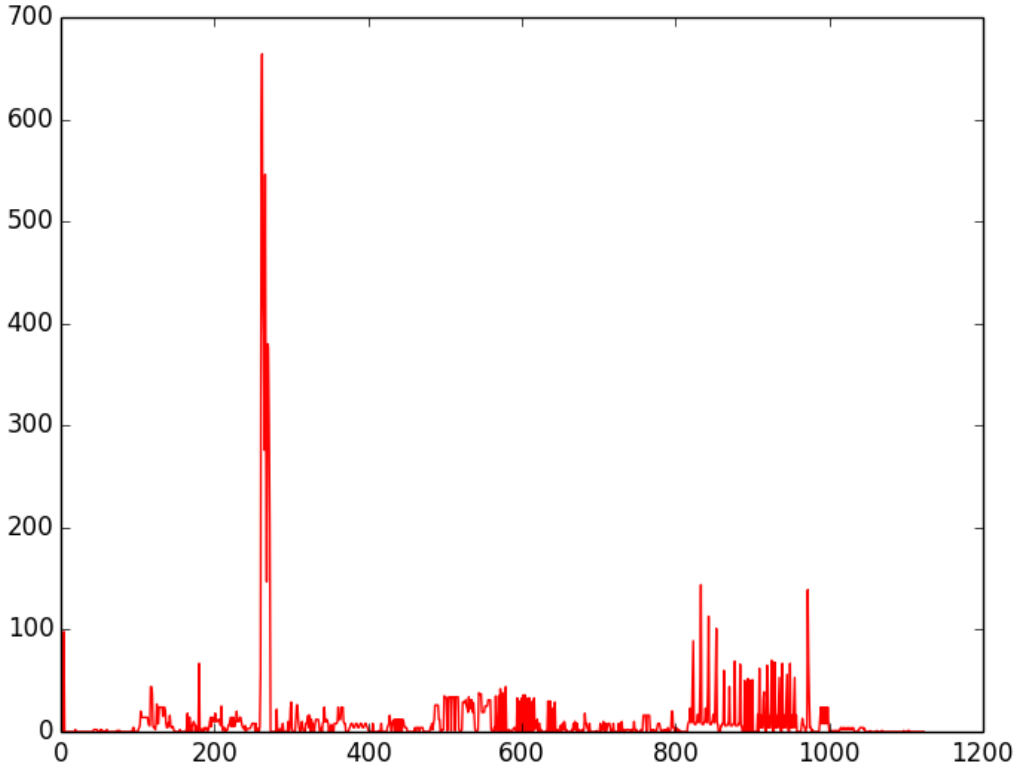
新浪微博
新浪新闻
新浪体育
新浪娱乐
新浪财经
新浪博客
新浪视频
新浪游戏
天气通

资料图片：菲律宾总统杜特尔特 新华社发

参考消息网3月25日报道 外媒称，菲律宾总统杜特尔特23日说他愿与中国共享海牙国际仲裁法庭已“裁决判属”马尼拉专有的南海水域的资源。
据法新社3月23日报道，杜特尔特说菲律宾人没有能力自行开采那里的自然资源。
杜特尔特在马尼拉说：“即使我想开采所有资源，我们也没钱开采。就连石油钻塔和所有设备我们都买不起。我愿意考虑共享资源。”
法新社称，杜特尔特的前任阿基诺三世曾对中国宣称享有南海大部分水域的控制权提出异议。2013年阿基诺政府向海牙常设仲裁法院提起诉讼。法院去年作出有利于马尼拉的裁决。
就在法院作出裁决数日前，杜特尔特上台，其后他逆转了阿基诺的政策，努力获得了北京的巨额投资和援助。
杜特尔特23日重申了早些时候宣布过的决定：他不会与中国因主权声索争议而交战。（编译/郑国仪）

基于文本长度的分析方法

对于一些新闻类的网页，正文分布往往比较集中，在正文集中的区域，每一行的文本数量都比较多



Text-Tag Ratio

文本与标签

```
<div class="ad_content ad_06 adNone" id="PublicRelation5">
    <div class="cmenu01">
      <span id="cbtn01" class="selected">应用中心</span>
      <span id="cbtn02">新浪公益</span>
      <span id="cbtn03">新浪游戏</span>
      <span id="cbtn04">互动活动</span>
      <span id="cbtn05">热点推荐</span>
    </div>
    <div class="ad_cont_03">
```

```
    <span class="img_descr">资料图片：菲律宾总统杜特尔特 新华社发</span>
  </div>

<p>    参考消息网3月25日报道 外媒称，菲律宾总统杜特尔特23日说他愿与中国共享海牙国际仲裁法庭已“裁决判属”马尼拉
<p>    据法新社3月23日报道，杜特尔特说菲律宾人没有能力自行开采那里的自然资源。</p>
<p>    杜特尔特在马尼拉说：“即使我想开采所有资源，我们也没钱开采。就连石油钻塔和所有设备我们都买不起。我愿意考虑
<p>    法新社称，杜特尔特的前任阿基诺三世曾对中国宣称享有南海大部分水域的控制权提出异议。2013年阿基诺政府向海牙
<p>    就在法院作出裁决数日前，杜特尔特上台，其后他逆转了阿基诺的政策，努力获得了北京的巨额投资和援助。</p>
<p>    杜特尔特23日重申了早些时候宣布过的决定：他不会与中国因主权声索争议而交战。（编译/郑国仪）</p>
```

文本与标签

```
<div id="sinaads_box" class="adNone">
    <div id="sinaads_fixed">
        <ins class="sinaads" data-ad-pdps="PDPS000000055163"></ins>
<div id="sinaads-float-close"></div>


    </div>
</div>
```

```
<div class="feed-wrap" id="relatedNewsWrap">
  <div class="feed-title">
    <h3>相关阅读</h3>
  </div>
  <div class="feed-c">
    <div id="feedCard"></div>
  </div>
</div>
```

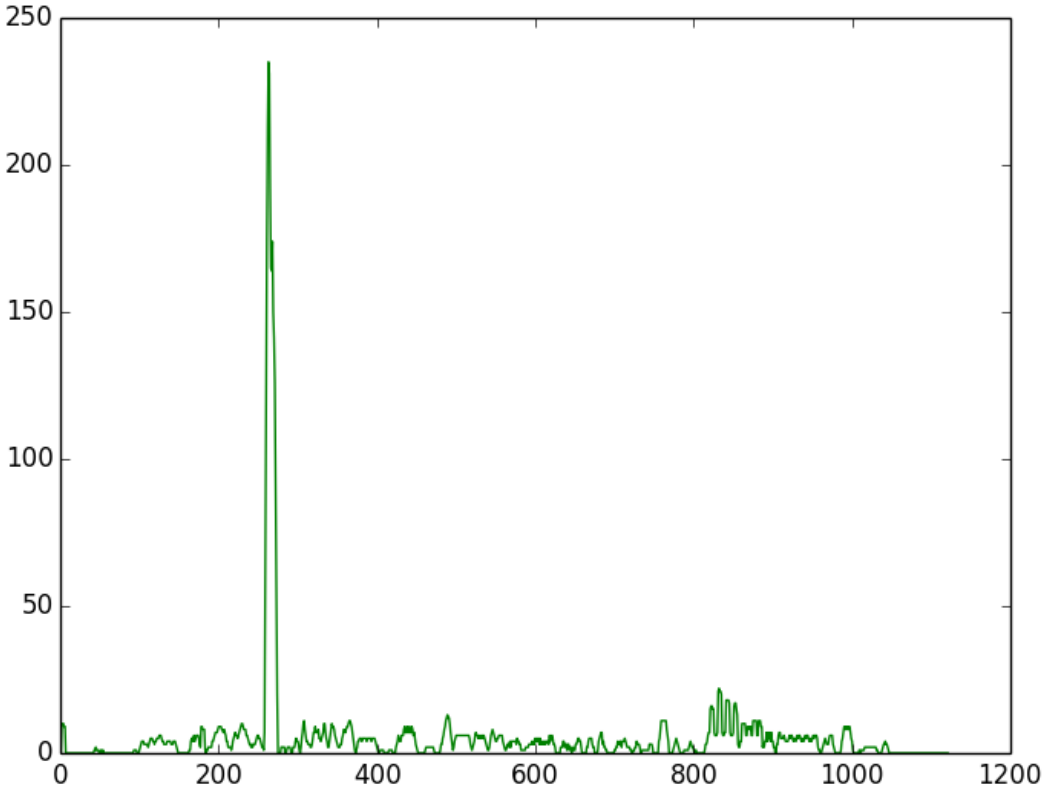
文本与标签密度

对比前面的几页，我们发现正文的标签比较少，文本数量与标签数量的比值很大，而噪声部分，标签很多，正文很少，所以我们考虑利用文本与标签的比值来进一步区分，采用下面算法计算 TEXT/TAG 的比值

```
input
   h ← HTML source code
begin
Remove all script and remark tags and empty lines for each line k to numLines( h )
do
    x ← number of non-tag ASCII characters in h[k]
    y ← number of tags in h[k]
if y = 0 then
   TTRArray[i] ← x
else
   TTRArray[i] ← x / y
end if
end for
return TTRArray
end
```

文本与标签密度

使用 text/tag 的比值后，有效去除了一些噪声

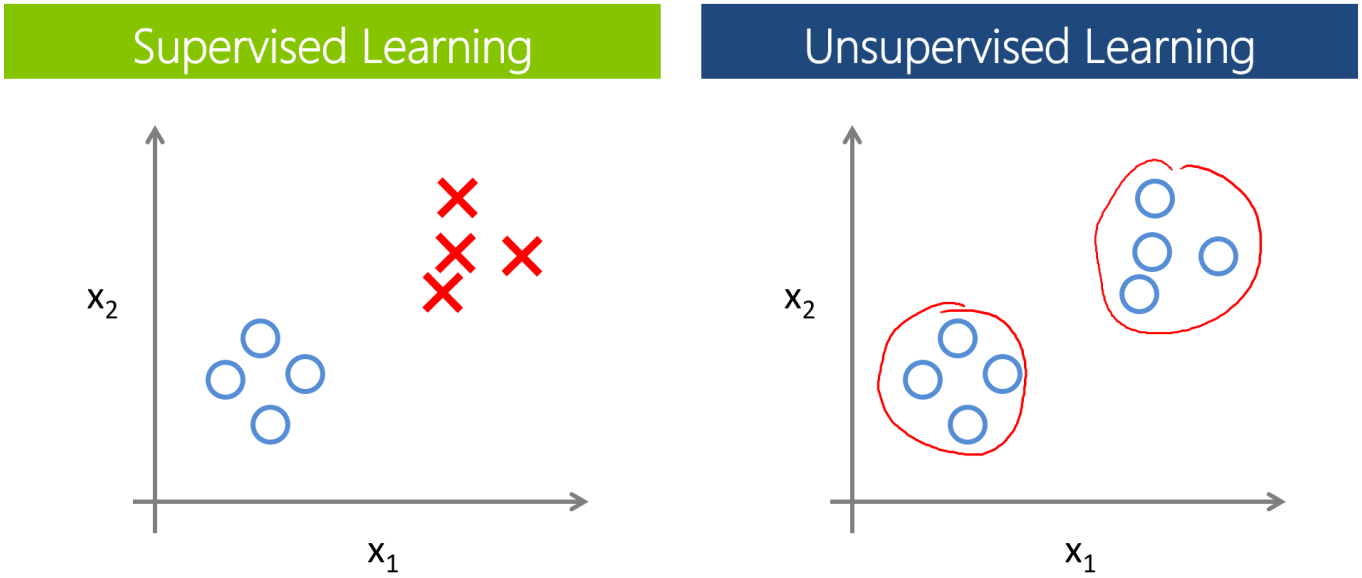


K-Means

监督学习与无监督学习

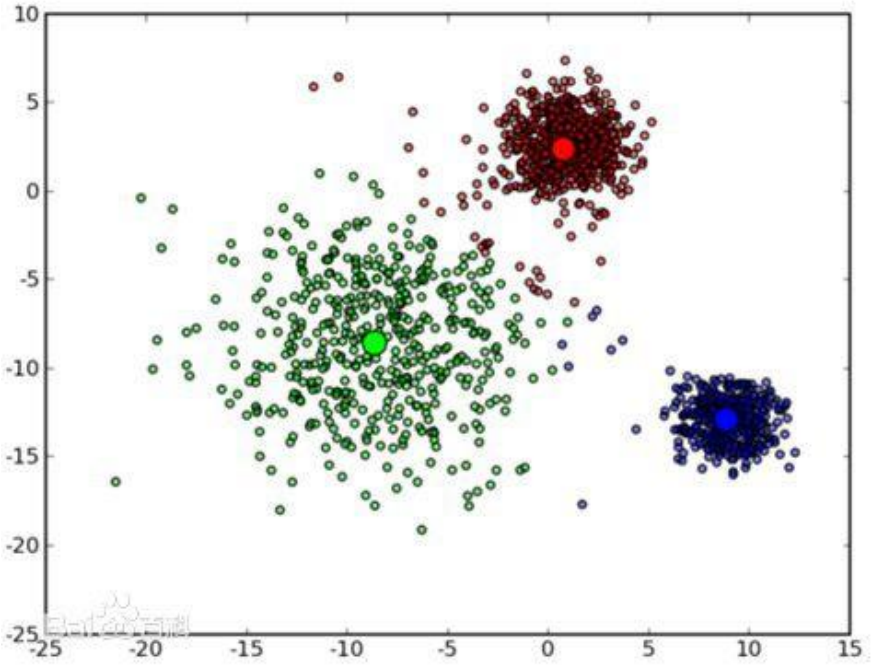
监督学习：有标签，根据标签来预测结果并与实际结果对比，修正数据最终得到一个模型

无监督学习：没有标签，根据一些特性来自动分类



K-Means

1. 随机选择k个中心点
2. 计算每个点与k个中心点的距离，把这个点划归到距离最小的点那个类别里
3. 重新计算每个类别的中心点
4. 重复步骤 2 ~ 3，直到目标函数达到最优或者迭代次数达到上限



K-Means 计算

- 计算每个点与k个中心点的距离，欧拉距离 $d = \sqrt[2]{(c-k)^2}$
- 重新计算每个类别的中心点：$k_x = \frac{\sum_{i=0}^{n-1} x_i}{n}$
- 结束条件，综合距离方差最小 $min(\sum_{i=1}^{k} \sum_{x \in c_i} dist(c_i - x)^2)$

  可以转化为求两次方差之间的差值小于设定的阈值

$$\Delta \sum_{i=1}^{k} \sum_{x \in c_i} dist(c_i - x)^2 < \varepsilon$$

算法的优点

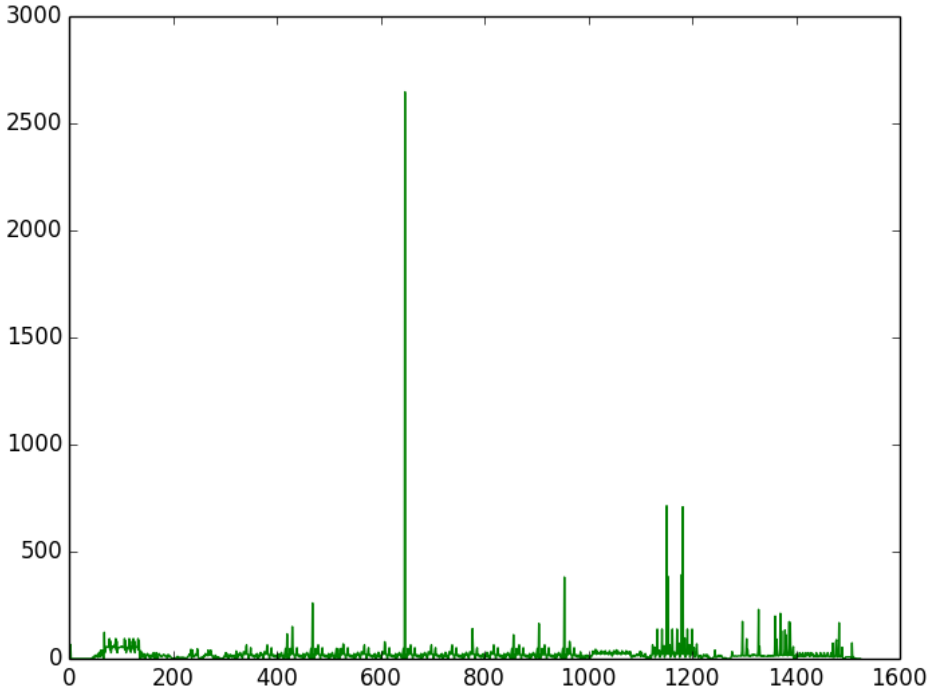
- 算法快速、简单
- 大数据集有较高的效率并且是可伸缩性的
- 时间复杂度近于线性，而且适合挖掘大规模数据集。K-Means聚类算法的时间复杂度是O(nkt) ,其中n代表数据集中对象的数量，t代表着算法迭代的次数，k代表着簇的数目
- 每个簇接近高斯分布的时候，效果比较好

算法的缺点

- 簇的平均值可被定义的情况下，才能使用
- 必须给定k的数目，而且对k很敏感
- 对噪声和孤立点很敏感

考虑这样的情况

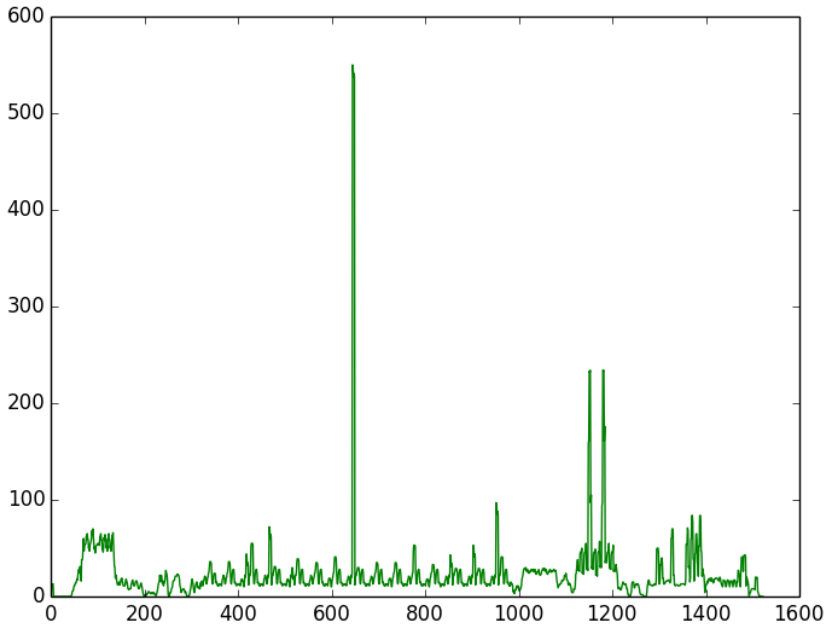
一段游记，有大量几个字一句话的描述，有一部分长段的介绍，因此会出现下面的情况，如果我们按照平均值来计算的话，大多数正文都会被认为是噪声



对数据进行平滑

正文整体是成块出现的，因此可以对奇点、噪点进行平滑，平滑的办法就是对前后 r 行进行平均，这样能提高 K-Means 算法的精度

$$e_k = \frac{\sum_{i=k-r}^{k+r} TTRArray_i}{2r+1}$$



利用 K-Means 聚类

```
# 转化为 二维的数组
# reshape 讲一个数组改变为任意 m * n 维度 的数组
# m 为 -1 的时候，数组的数量由总数和每个数组的元素来确定
feature = np.array(ratio_smoth).reshape(-1,1)

# 聚合，预测每个元素所处的聚类，设置k = 3
kmeans = KMeans(3).fit(feature)
labels = kmeans.predict(feature)

# 找出聚合中心点
centers = kmeans.cluster_centers_

# 得到均方差
std = np.std(ratio_smoth)
```

中心点均值

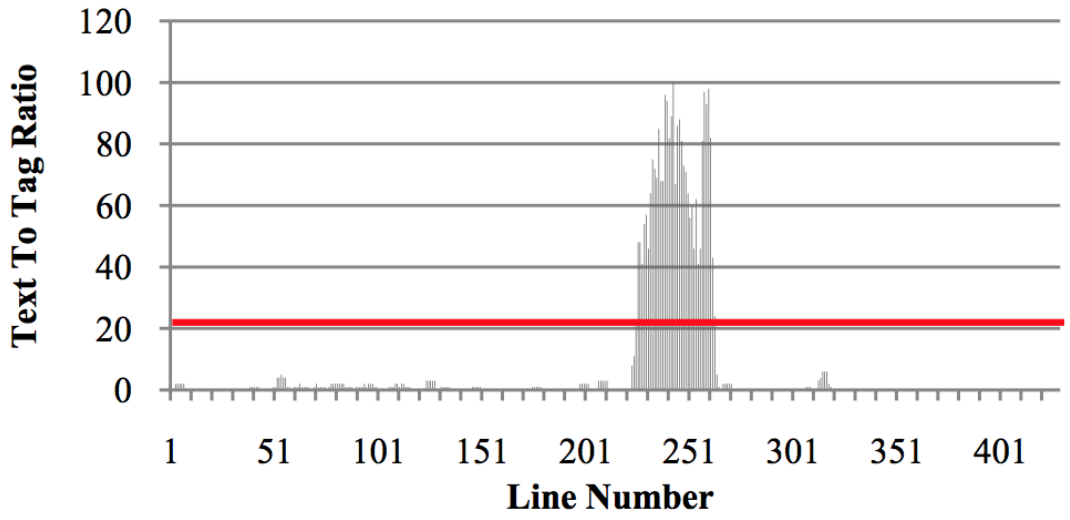
cluster	k0	k1	k2
center	23.87065003		
center	374	21	
center	16.40404798	542.8	63.90217391

均值： 23.8706500328
标准差： 36.8512610629

k 分别取1、2、3，合并3次聚类中认为的正文部分

中心点均值

分布更平均的时候，标准差减小；分布不均匀的时候，标准差增加。正文簇的均值一定是高于标准差的



计算过程代码

```
for k in range(1,4):

    # 聚合，预测每个元素所处的聚类，设置k = 2
    kmeans = KMeans(k).fit(feature)
    labels = kmeans.predict(feature)

    # 找出聚合中心点
    centers = kmeans.cluster_centers_

    print centers

    # 根据聚类的结果来分类
    clusters = {}
    n = 0
    for item in labels:
        if item in clusters:
            clusters[item].append(n)
        else:
            clusters[item] = [n]
        n +=1

    index = 0
    for i in centers:
        if i[0] > std:
            result_list += clusters[index]
        index += 1
```

标签模板

通用模板与配置

```
# 定义正文的tag

tags = {
    'title': '//h3[@class="b_tle"]',
    'content': '//td[@class="editor bbsDetailContainer"]//*[self::p or self::span or
self::h1]'
}
```

针对特定类型的网站，可以快速找出它们所使用的标签类型，我们把这些选择器以模板的方式来配置

xpath 支持 or 的选项，因此我们可以用

self::p or self::span

来合并多种标签标记的正文

使用场景

Text/Tag：

- 大规模抓取，没有模板的网站
- 优势：不需要规则，适用广泛
- 劣势：精准度差

模板：

- 有针对性的抓取，对于核心网站可以考虑使用模板
- 优势：精度高，质量好，速度快
- 劣势：只能对特定网站使用

PyGoose

Introduction

Goose will try to extract the following information:

- Main text of an article
- Main image of article
- Any YouTube/Vimeo movies embedded in article
- Meta Description
- Meta tags

Setup

git clone https://github.com/grangier/python-goose.git

cd python-goose pip install -r requirements.txt

python setup.py install

使用

配置：

```
g = Goose({'use_meta_language': False, 'target_language':'es'})
g = Goose({'stopwords_class': StopWordsChinese})
```

提取，可以传入 url 或者 html 文本：

```
article = g.extract(url=url)
article = g.extract(raw_html=html)

article.cleaned_text
```

疑问

- 问题答疑： http://www.xxwenda.com/
  - 可邀请老师或者其他人回答问题

联系我们

小象学院：互联网新技术在线教育领航者

- 微信公众号：大数据分析挖掘
- 新浪微博：ChinaHadoop