管理学院信息管理专业
《商务智能与数据挖掘方法》课程作业

基于GMM模型的电商平台用户评论聚类研究

组员1 :管信1801班　徐安琪　201805020

组员2 :管信1801班　邱健铭　201805022

组员3 :管信1801班　肖文建　201802017

组员4 :管物1701班　米唯嘉　201600028

指 导 教 师 :　裘江南

大连理工大学

Dalian University of Technology

目录

1. 引言

1.1 选题背景

在当下的大数据时代，文本等非结构化数据呈现爆发式的增长趋势。如何从这些海量的、杂乱无章的数据中挖掘出有价值的信息，是一个许多学者在关注的问题。因此，寻找一种高效的挖掘这些信息的方法体系显得尤为重要。

近年来，随着我国电商行业的日渐崛起，网络购物用户数量和网购交易规模的不断扩大，线上购物成了许多消费者的主要购买方式。大量的线上交易同时产生了大量的用户评论数据，用户也倾向在购买前浏览商品的评论信息。通过对用户评论文本进行挖掘，可以识别评论的热点所在，分析出不同产品的优势和不足，来实现对消费者

用户的有效划分，为后续的分析提供良好的分析基础。

随着信息技术的迅猛发展，笔记本电脑已经成为当下人们学习和工作不可或缺的工具，进而也成为各大电商平台热销的电子产品之一。由于笔记本电脑的耐用性和较高的价格，顾客在购买前会更加关注已有的评论信息，来帮助自己做出明智的判断。同时，在收到线上购买的笔记本之后，用户也会提供自己关于客服服务、物流快慢、商品质量等方面的评价和感受。因此，电商平台笔记本电脑的用户评论信息，是研究消费者购买行为非常有价值的数据源之一。

1.2 现实意义

对在线用户评论进行挖掘，无论是对电商平台而言，还是商品生产商，以及潜在消费者来说，都有着重要的参考价值。对于电商平台来说，通过对用户评论信息的挖掘，可以得知消费者对卖家提供的相关服务是否满意，比如在物流快慢、店铺客服、快递包装等方面；对于生产商而言，通过分析这些用户留下的评论，能够从消费者的角度发现该品牌下的产品优点与不足之处，对于改进产品与提升产品质量有一定的指导意义；对于那些潜在的消费者来说，通过对某品牌下的商品用户评论的挖掘分析，可以从过来人的眼里发现此商品的优点与缺点，对其选购有一定指导意义。

具体来说，本文通过爬取与分析京东商城网站中关于苹果和华为两个品牌的笔记本电脑的用户评论，寻找消费者选购不同品牌电脑时最直观的关注热点及看法，以便从海量的用户评论中挖掘出有价值的信息。对于笔记本电脑制造商而言，可以从顾客的角度发现自身产品的优点与缺点，这对于改进产品方向上具有一定的参考意义与价值。对于京东商城而言，可以根据分析结果总结出顾客在进行线上选购笔记本电脑时，对其所提供的物流、包装及客服方面服务的满意程度与看法，在服务质量提升方面有一定的参考价值。对于那些潜在的消费者来说，通过分析不同笔记本电脑品牌的消费者的评论，可以发现不同品牌的笔记本电脑的优势和不足，进而得以综合多方面因素，做出明智的购买决定。

2. 主要算法介绍

1.1 数据的获取与预处理技术

2.1.1 网络爬虫技术

由于研究的原始数据来自京东商城网站，所以需要Python的爬虫技术来获得最新的用户评论。网络爬虫技术，又叫做网络蜘蛛人，是一种按照封装好的爬取规则，专注于进行自动地爬取网站页面相关信息的程序。利用Python的网络爬虫技术，工作原理向用户正常访问网页一样，先对服务器发起 Request 请求，接收到Response 回馈响应后再进行网页的访问和信息爬取。由于网络爬虫不是我们的技术重点，所以基础概念和相关技术（包括正则表达式和超文本协议等）我们不再赘述。

目前来看，按照爬虫的体系结构不同，网络爬虫主要被分为三大类，分别为通用网络爬虫和主题爬虫以及增量式网络爬虫，对于本文研究的内容，所用到的爬虫类型是主题网络爬虫。主题网络爬虫是一种有选择的爬虫技术，在爬行之前，事先确定好对某一主题的爬取，在爬行时会有针对地爬取与之前定义好的主题相关的网页信息。这在很大程度上节省了硬件方面以及网络方面的资源，同时相比于通用网络爬虫，具有速度快的优势。

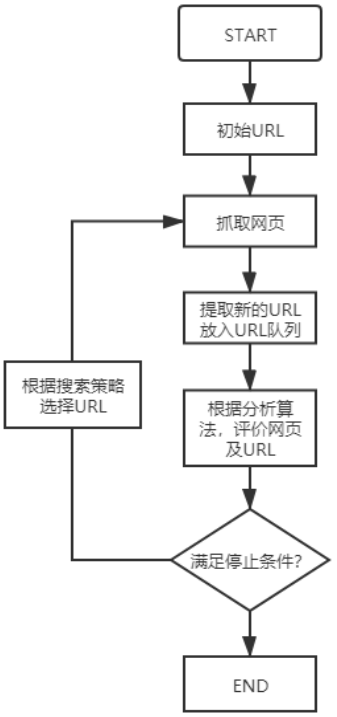


图2-1 主题网络爬虫流程

2.1.2 文本预处理技术

在数据驱动的研究中，数据质量的高低直接影响模型分析的结果，因此对数据进行预处理非常重要。对于非结构化的文本数据来说，通常的数据预处理则称为文本预处理。本研究对原始文本数据进行的预处理包括数据清洗、文本分词、去除停用词、特征提取及降维。

1) 数据清洗：主要包括对大量评论数据的去短、去重、去除垃圾停用词等步骤，从而提高数据的整体质量。
2) 文本分词：文本分词技术，是指把一句句比较长的句子，依照一定的规则与逻辑拆分成一个个单独存在的词过程。本研究主要是通过 Python 中的 Jieba 分词工具来实现对评论文本数据的分词处理。对于分词算法的选择，本文研究选用的是基于精确模式的分词算法，这种分词算法使分词更加精确，提升了分词的效果，从而为

后续的分析做了充足准备。

3) 去除停用词：停用词去除主要是为了过滤掉一些对后续研究毫无意义的字或词，一般情况下停用词主要包括两类词，一类是指功能词，而另一类是指应用十分广泛的虚词与助词等。比如，类似“的”、“想”、“好想”、“了”这样的词语，把这些无意义的词去掉，有利于提高分词质量以及挖掘文本信息的价值。在实际的分析中，需要通过构建好的停用词表来进行去停用词的处理工作。本研究用到的停用词表是哈工大平台构建的停用词表。

4) 特征提取及降维：在特征提取之前，需要对文本信息进行文本向量化处理，把对文本内容的处理简化为向量空间中的向量运算，并且它以空间上的相似度表达语义的相似度。本研究采用TF-IDF算法来描述用户评论。设一篇文档可以表示为一组n个关键词对应的矢量$d_j(\omega_{1j}, \omega_{2j}, ..., \omega_{nj})$。其中对应的是关键词$k_i$对于文档$d_j$的权重，可以通过词频和文档频率来计算。词频通常用$tf$表示，代表的是一个词在文档中出现的频率。假设一个词k_i在文档d_j中出现的次数为f_{ij}，文档d_j中出现次数最多的关键词为k_l。词频tf_{ij}的计算公式如下：

$$tf_{ij} = \frac{f_{ij}}{f_{lj}}$$

假设文档的个数为N，词k_i在n_i文档中出现，则其文档频率df_i可以定义为$df_i = n_i/N$。在计算词k_i的权重时使用逆文档频率idf_i，定义如下：

$$idf_i = log\frac{N}{n_i}$$

综合以上两个指标来计算关键词权重，一般使用相乘法，即$w_{ij} = tf_i \times idf_i$。最终，可以得到文档—词语向量矩阵用来进一步分析文档之间的相似度。

1.2 高斯混合聚类模型概述

高斯密度函数估计是一种参数化模型。高斯混合模型（Gaussian Mixture Model, GMM）是单一高斯概率密度函数的延伸，GMM能够平滑地近似任意形状的密度分布。高斯混合模型种类有单高斯模型（Single Gaussian Model, SGM）和高斯混合模型（Gaussian Mixture Model, GMM）两类。类似于聚类，根据高斯概率密度函数（Probability Density Function, PDF）参数不同，每一个高斯模型可以看作一种类别，输入一个样本x，即可通过PDF计算其值，然后通过一个阈值来判断该样本是否属于高斯模型。很明显，SGM适合于仅有两类别问题的划分，而GMM由于具有多个模型，划分更为精细，适用于多类别的划分，可以应用于复杂对象建模。

2.2.1 单高斯模型

多维高斯（正态）分布概率密度函数PDF定义如下：

$$N(x;\mu,\Sigma)=\frac{1}{\sqrt{(2\pi)|\Sigma|}}\exp\left[-\frac{1}{2}(x-\mu)^T\sum^{-1}(x-\mu)\right] \quad (1)$$

注意与一维高斯分布不同，其中x是维数为d的样本向量（列向量），是模型期望，Σ是模型方差。

对于单高斯模型，由于可以明确训练样本是Σ否属于该高斯模型，故通常由训练样本均值代替，由样本方差代替。为了将高斯分布用于模式分类，假设训练样本属于类别C，那么，式(1)可以改为如下形式：

$$N(x/C)=\frac{1}{\sqrt{(2\pi)|\Sigma|}}\exp\left[-\frac{1}{2}(x-\mu)^T\sum^{-1}(x-\mu)\right] \quad (2)$$

式子(2)表明样本属于类别C的概率大小。从而将任意测试样本 x_i 输入式(2),均可以得到一个标量 $N(x;\mu,\Sigma)$，然后根据阈值t来确定该样本是否属于该类别。

*阈值t的确定：可以为经验值，也可以通过实验确定。另外也有一些策略可以参考，如：首先令t=0.7,以0.05为步长一直减到0.1左右，选择使样本变化最小的那个阈值做为最终t值，也就是意味着所选的t值所构造的分类模型最稳定。

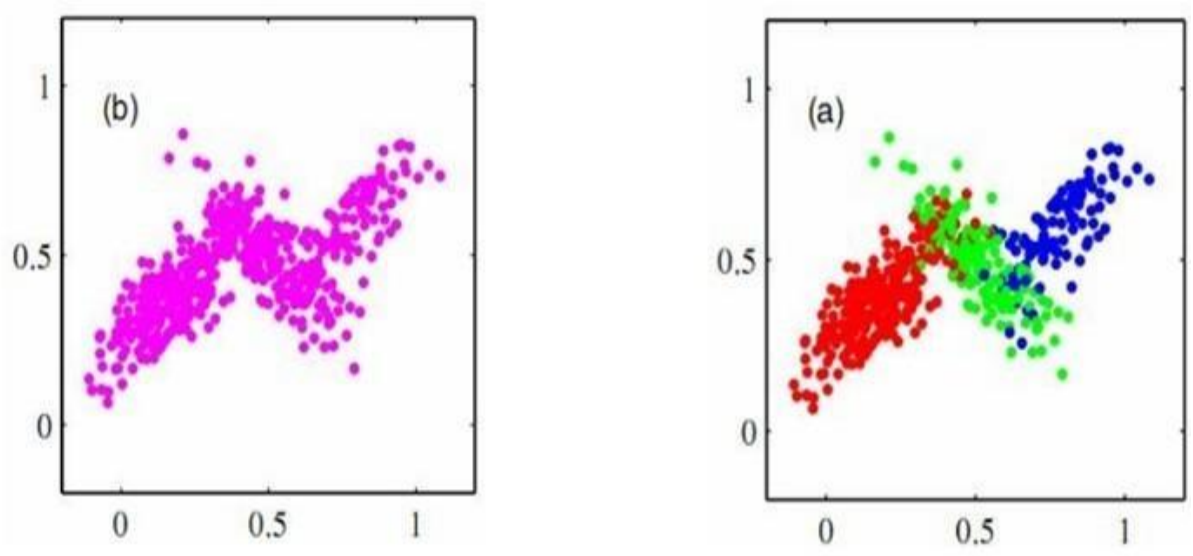
*几何意义理解：单高斯分布模型在二维空间应该近似于椭圆，在三维空间上近似于椭球。遗憾的是在很多分类问题中，属于同一类别的样本点并不满足“椭圆”分布的特性。这就引入了高斯混合模型。

2.2.2 高斯混合模型

高斯混合模型是单一高斯机率密度函数的延伸，由于 GMM 能够平滑地近似任意形状的密度分布，因此近年来常被用在语音、图像识别等方面，得到不错的效果。

例如：例：有一批观察数据 $X=\{x_1,x_2,\cdots,x_n\}$ 数据个数为n，在d维空间中的分布不是椭球状（如图1(a)），那么就不适合以一个单一的高斯密度函数来描述这些数据点的机率密度函数。此时我们采用一个变通方案，假设每个点均由一个单高斯分布生成（如图1(b)，具体参数 μ_j,Σ_j 未知），而这一批数据共由M（明确）个单高斯模型生成，具体某个数据属于哪个单高斯模型未知，且每个单高斯模型在混合模型中占的比例未知，将所有来自不同分布的数据点混在一起，该分布称为高斯混合分布。

聚类前后示意图



从数学上讲，我们认为这些数据的概率分布密度函数可以通过加权函数表示：

$$p(x_i)=\sum_{j=1}^{M} a_j N_j(x_i;\mu_j,\Sigma_j) \quad (3)$$

上式即称为GMM，$\sum_{j=1}^{M} a_j = 1$，其中：

$$N_j(x;\mu_j,\Sigma_j)=\frac{1}{\sqrt{(2\pi)^m|\Sigma_j|}}\exp\left[-\frac{1}{2}(x-\mu_j)^T\sum_j{}^{-1}(x-\mu_j)\right] \quad (4)$$

表示第j个SGM的PDF。j需要事先确定好，就像K-means中的K一样。a_j 是权值因子。其中的任意一个高斯分布 $N_j(x;\mu_j,\Sigma_j)$ 叫做这个模型的一个component。GMM是一种聚类算法，每个component就是一个聚类中心。即在只有样本点，不知道样本分类（含有隐含变量）的情况下，计算出模型参数 (a,μ,Σ) 。这显然可以用最大期望算法（EM算法）来求解。再用训练好的模型去差别样本所属的分类，方法是：

Step1:随机选择K个component中的一个（被选中的概率是 a_j ）

Step2:把样本代入刚选好的component，判断是否属于这个类别，如果不属于则回到step1。

2.3.高斯混合模型参数估计

2.3.1 样本分类已知情况下的GMM

当每个样本所属分类已知时，GMM的参数非常好确定，直接利用极大似然估计方法（Maximum Likelihood Estimate，MLE）。设样本容量为N，属于K个分类的样本数量分别是，N_1, N_2，...，N_K，属于第k个分类的样本集合是L(k)。

$$a_k = \frac{N_k}{N} (5)$$

$$\mu_k = \frac{1}{N_k} \sum_{x \in L(k)} x (6)$$

$$\sum_k = \frac{1}{N_k} \sum_{x \in L(k)} (x - \mu_k)(x - \mu_k)^T (7)$$

2.3.2 样本分类未知情况下的GMM

有N个数据点，服从某种分布 $Pr(x;\theta)$，我们想要找到一组参数 θ，使得生成这些数据点的概率最大，这个概率就是：$\prod_{i=1}^{N} \Pr(xi;\theta) (8)$

称为似然函数(Likelihood Function)。通常单个点的概率很小，连乘之后数据会更小，容易造成浮点数下溢，所以一般取对数，变成：$\sum_{i=1}^{N} \log \Pr(xi;\theta) (9)$，称为最大化对数似然函数（log-likelihood function）。

GMM的log-likelihood function就是：

$$\sum_{i=1}^{N} \log \{ \sum_{k=1}^{K} a_k N(x_i; \mu_k, \sum k) \} (10)$$

这里每个样本 $\mathbf{X}_i$ 所属的类别 Z_k 是不知道的。Z是隐含变量。我们就是要找到最佳的模型参数，使得(10)式所示的期望最大，“期望最大化算法”名字由此而来。

运用EM法求解：

EM要求解的问题一般形式是:

$$\theta^* = \arg\max_{\theta} \prod_{j=1}^{|x|} \sum_{y \in Y} \Pr(X=\mathbf{x}_j, Y=y; \theta) (11)$$

Y是隐含变量，我们已经知道如果数据点的分类标签Y是已知的，那么求解模型参数直接利用Maximum Likelihood就可以了。EM算法的基本思路是：随机初始化一组参数 $\theta^{(0)}$ 根据后验概率 $Pr(Y \mid X;\theta)$ 来更新Y的期望E(Y)，然后用E(Y)代替Y求出新的

模型参数 $\theta^{(t+1)}$ 。如此迭代直到 θ 趋于稳定。

E-Step。E就是Expectation的意思，就是假设模型参数已知的情况下求隐含变量Z分别取 Z_1、Z_2…… 的期望，亦即Z分别取 Z_1、Z_2…… 的概率。在GMM中就是求数据点由各个 component生成的概率。

$$\gamma(i,k) = b_k \Pr(Z_k \mid x_i; a, \mu, \sum) \quad (12)$$

注意到我们在Z的后验概率前面乘以了一个权值因子 b_k，它表示在训练集中数据点属于类别 z_k 的频率，在GMM中它就是 a_k 。

$$\gamma(i,k) = \frac{a_k N(x_i \mid \mu_k, \sum k)}{\sum_{j=1}^{K} a_j N(x_i \mid \mu_k, \sum j)} \quad (13)$$

M-Step。M就是Maximization的意思，就是用最大似然的方法求出模型参数。现在我们认为上一步求出的 $\gamma(i,k)$ 就是“数据点 x_i 由component k生成的概率”。根据公式(5),(6),(7)可以推出：

$$N_k = \sum_{j=1}^{N} \gamma(i,k) \quad (14)$$

$$\mu_k = \frac{1}{N_k} \sum_{i=1}^{N} \gamma(i,k)\, x_i \quad (15)$$

$$\sum k = \frac{1}{N_k} \sum_{i=1}^{N} \gamma(i,k)(x_i - \mu_k)(x_i - \mu_k)^T \quad (16)$$

$$a_k = \frac{N_k}{N} \quad (17)$$

根据上述过程推导出的基本数学关系进行模型参数求解。

3. 程序实现

让我们首先导入Python库：

```
import imageio
import matplotlib.animation as ani
import matplotlib.cm as cmx
import matplotlib.colors as colors
import matplotlib.pyplot as plt
import numpy as np
from matplotlib.patches import Ellipse
from PIL import Image
from sklearn import datasets
from sklearn.cluster import KMeans
```

我们将使用Iris机器学习数据集，使用sklearn提供的load_iris函数获取：

```
iris = datasets.load_iris()
X = iris.data
```

现在我们来实现高斯密度函数：

```
def gaussian(X, mu, cov):
 n = X.shape[1]
 diff = (X - mu).T
 return np.diagonal(1 / ((2 * np.pi) ** (n / 2) * np.linalg.det(cov) ** 0.5) * np.exp(-0.5 * np.dot(np.dot(diff.T, np.linalg.inv(cov)), diff))).reshape(-1, 1)
x0 = np.array([[0.05, 1.413, 0.212], [0.85, -0.3, 1.11], [11.1, 0.4, 1.5], [0.27, 0.12, 1.44], [88, 12.33, 1.44]])
mu = np.mean(x0, axis=0)
cov = np.dot((x0 - mu).T, x0 - mu) / (x0.shape[0] - 1)
y = gaussian(x0, mu=mu, cov=cov)
y
```

- Step 1

这是GMM的初始化步骤。此时，我们必须初始化参数 π 、 μ 和 Σ 。在本例中，我们将使用KMeans的结果作为 μ 的初始值，将 π 设置为聚类数量1，并将 Σ 设置为单位矩阵。我们也可以使用随机数，但是使用一个合理的初始化过程将帮助算法获得更好的结果。

```python
def initialize_clusters(X, n_clusters):
    clusters = []
    idx = np.arange(X.shape[0])

    # We use the KMeans centroids to initialise the GMM

    kmeans = KMeans().fit(X)
    mu_k = kmeans.cluster_centers_

    for i in range(n_clusters):
        clusters.append({
            'pi_k': 1.0 / n_clusters,
            'mu_k': mu_k[i],
            'cov_k': np.identity(X.shape[1], dtype=np.float64)
        })

    return clusters
```

- Step 2 (Expectation step)

我们可以通过以下表达式来实现:

$$\gamma\left(z_{nk}\right)=\frac{\pi_k N\left(x_n|\mu_k,\Sigma_k\right)}{\Sigma_{j=1}^{K}\pi_j N\left(x_n|\mu_j,\Sigma_j\right)}$$

为了方便起见，我们只是将分母计算为分子中所有项的和，然后将其分配给一个名为total的变量。

```python
def expectation_step(X, clusters):
    totals = np.zeros((X.shape[0], 1), dtype=np.float64)

    for cluster in clusters:
        pi_k = cluster['pi_k']
    mu_k = cluster['mu_k']
    cov_k = cluster['cov_k']

    gamma_nk = (pi_k * gaussian(X, mu_k, cov_k)).astype(np.float64)

    for i in range(X.shape[0]):
        totals[i] += gamma_nk[i]

    cluster['gamma_nk'] = gamma_nk
    cluster['totals'] = totals

    for cluster in clusters:
        cluster['gamma_nk'] /= cluster['totals']
```

- Step 3 (Maximization step):

现在我们来实现最大化步骤。我们可以简单地定义:

$$N_k = \sum_{n=1}^{N} \gamma(z_{nk})$$

然后我们可以通过以下方法计算修正后的参数:

$$\pi_k^* = \frac{\sum_{n=1}^{N} \gamma(z_{nk})}{N}$$

$$\mu_k^* = \frac{\sum_{n=1}^{N} \gamma(z_{nk}) x_n}{\sum_{n=1}^{N} \gamma(z_{nk})}$$

$$\Sigma_k^* = \frac{\sum_{n=1}^{N} \gamma(z_{nk})(x_n - \mu_k)(x_n - \mu_k)^T}{\sum_{n=1}^{N} \gamma(z_{nk})}$$

注意:为了计算协方差，我们定义了一个包含(xn- μ k)T的辅助变量diff。

```
def maximization_step(X, clusters):
    N = float(X.shape[0])

    for cluster in clusters:
        gamma_nk = cluster['gamma_nk']
    cov_k = np.zeros((X.shape[1], X.shape[1]))

    N_k = np.sum(gamma_nk, axis=0)

    pi_k = N_k / N
    mu_k = np.sum(gamma_nk * X, axis=0) / N_k

    for j in range(X.shape[0]):
        diff = (X[j] - mu_k).reshape(-1, 1)
    cov_k += gamma_nk[j] * np.dot(diff, diff.T)

    cov_k /= N_k

    cluster['pi_k'] = pi_k
    cluster['mu_k'] = mu_k
    cluster['cov_k'] = cov_k
```

我们现在确定模型的对数似然。由下式给出

$$\ln ln\, p(X) = \sum_{n=1}^{N} \ln ln \sum_{k=1}^{K} \pi_k N(x_n | \mu_k, \Sigma_k)$$

然而，第二个求和已经在expectation_step函数中计算过，并且可以在total变量中使

用。

```
def get_likelihood(X, clusters):
  likelihood = []
  sample_likelihoods = np.log(np.array([cluster['totals'] for cluster in clusters]))
  return np.sum(sample_likelihoods), sample_likelihoods
```

最后，让我们把所有的东西放在一起。首先，我们将使用initialise_clusters函数初始化参数，然后执行几个expectation-maximization steps。在本例中，我们将训练过程的迭代次数设置为固定的n_epochs次数。我这样做是为了以后生成对数似然图。

```
def train_gmm(X, n_clusters, n_epochs):
    clusters = initialize_clusters(X, n_clusters)
    likelihoods = np.zeros((n_epochs,))
    scores = np.zeros((X.shape[0], n_clusters))
    history = []
    for i in range(n_epochs):
        clusters_snapshot = []

    # This is just for our later use in the graphs
    for cluster in clusters:
        clusters_snapshot.append({
            'mu_k': cluster['mu_k'].copy(),
            'cov_k': cluster['cov_k'].copy()
        })

    history.append(clusters_snapshot)

    expectation_step(X, clusters)
    maximization_step(X, clusters)
    likelihood, sample_likelihoods = get_likelihood(X, clusters)
    likelihoods[i] = likelihood
    print('Epoch: ', i + 1, 'Likelihood: ', likelihood)

    for i, cluster in enumerate(clusters):
        scores[:, i] = np.log(cluster['gamma_nk']).reshape(-1)

    return clusters, likelihoods, scores, sample_likelihoods, history
```

训练模型：

```
n_clusters = 3
n_epochs = 50
clusters, likelihoods, scores, sample_likelihoods, history = train_gmm(X, n_clusters, n_epochs)
```

查看对数似然：

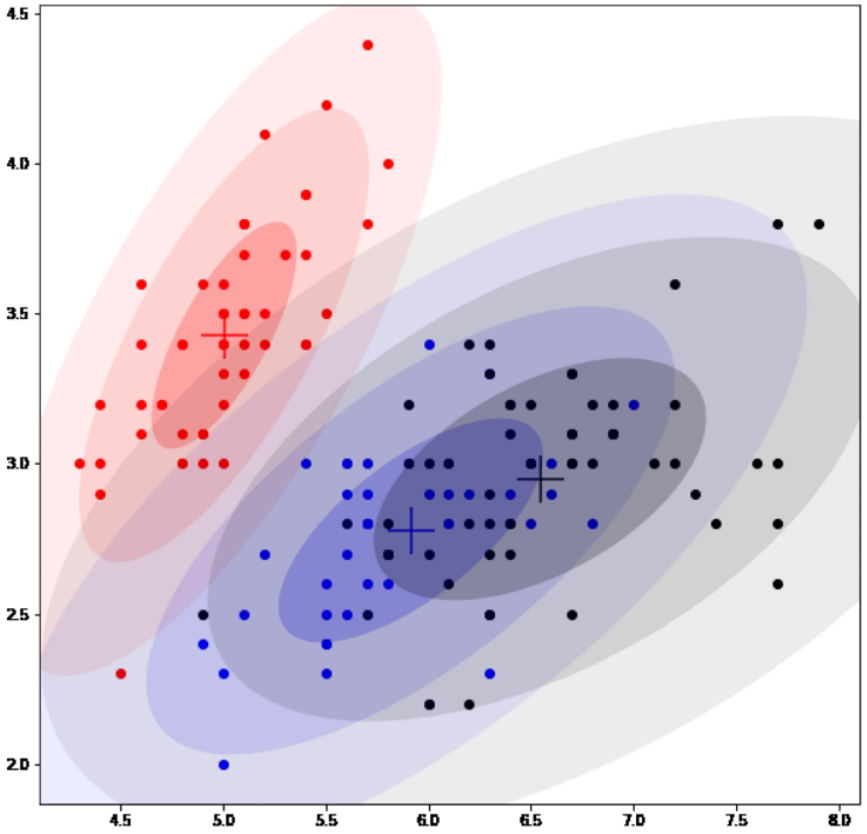
```
plt.figure(figsize=(10, 10))
plt.title('Log-Likelihood')
plt.plot(np.arange(1, n_epochs + 1), likelihoods)
plt.show()
```

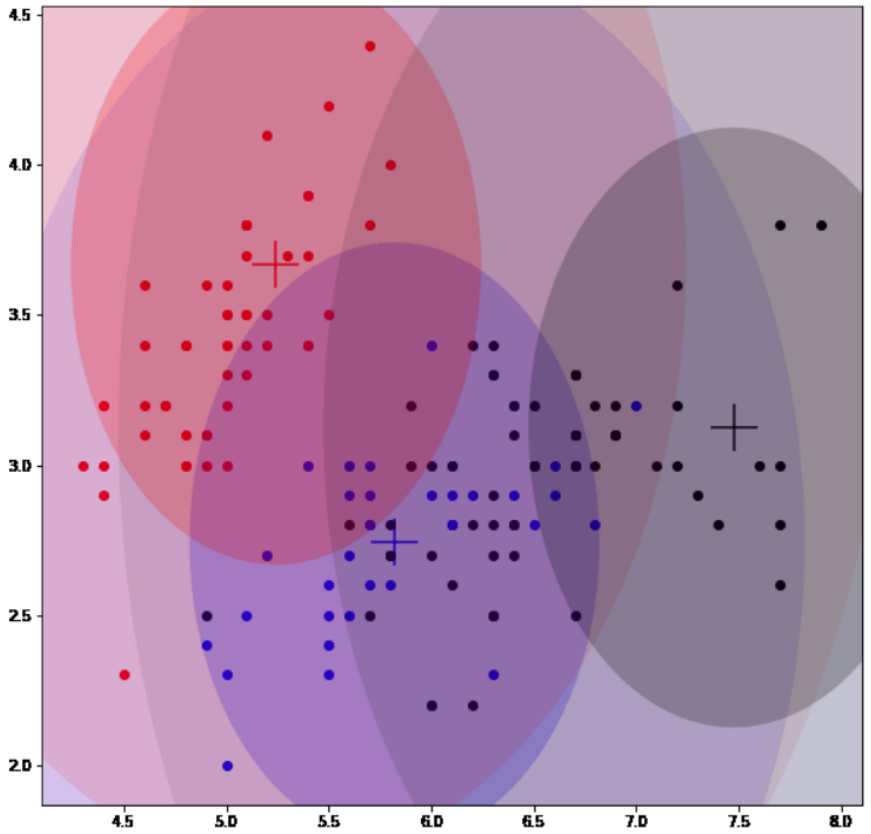
我们知道创建一个图形来可视化我们的聚类以及高斯混合的参数。实际上，我们所做

的是创建不同尺度的椭圆以便映射到每个高斯函数的坐标。

```
def create_cluster_animation(X, history, scores):
    fig, ax = plt.subplots(1, 1, figsize=(10, 10))
    colorset = ['blue', 'red', 'black']
    images = []

    for j, clusters in enumerate(history):

    idx = 0

    if j % 3 != 0:
        continue

    plt.cla()

    for cluster in clusters:
        mu = cluster['mu_k']
    cov = cluster['cov_k']
    eigenvalues, eigenvectors = np.linalg.eigh(cov)
    order = eigenvalues.argsort()[::-1]
    eigenvalues, eigenvectors = eigenvalues[order], eigenvectors[:, order]
    vx, vy = eigenvectors[:, 0][0], eigenvectors[:, 0][1]
    theta = np.arctan2(vy, vx)
    color = colors.to_rgba(colorset[idx])
    for cov_factor in range(1, 4):
        ell = Ellipse(xy=mu, width=np.sqrt(eigenvalues[0]) * cov_factor * 2,
                      height=np.sqrt(eigenvalues[1]) * cov_factor * 2, angle=np.degrees(theta), linewidth=2)
    ell.set_facecolor((color[0], color[1], color[2], 1.0 / (cov_factor * 4.5)))
    ax.add_artist(ell)
    ax.scatter(cluster['mu_k'][0], cluster['mu_k'][1], c=colorset[idx], s=1000, marker='+')
```

```
    idx += 1
    for i in range(X.shape[0]):
        ax.scatter(X[i, 0], X[i, 1], c=colorset[np.argmax(scores[i])], marker='o')

    fig.canvas.draw()

    image = np.frombuffer(fig.canvas.tostring_rgb(), dtype='uint8')
    image = image.reshape(fig.canvas.get_width_height()[::-1] + (3,))
    images.append(image)

    kwargs_write = {'fps': 1.0, 'quantizer': 'nq'}
    imageio.mimsave('./gmm.gif', images, fps=1)
    plt.show(Image.open('gmm.gif').convert('RGB'))


create_cluster_animation(X, history, scores)
```


4.5
4.0
3.5
3.0
2.5
2.0
4.5
5.0
5.5
6.0
6.5
7.0
7.5
8.0


4.5
4.0
3.5
3.0
2.5
2.0
4.5
5.0
5.5
6.0
6.5
7.0
7.5
8.0

在本例中，我们使用sklearn的GMM实现来检查参数和概率。

```
from sklearn.mixture import GaussianMixture
gmm = GaussianMixture(n_components=n_clusters, max_iter=50).fit(X)
gmm_scores = gmm.score_samples(X)
print('Means by sklearn', gmm.means_)
print('Means by our implementation:', np.array([cluster['mu_k'].tolist() for cluster in clusters]))
print('Scores by sklearn:', gmm_scores[0:20])
print('Scores by our implementation:', sample_likelihoods.reshape(-1)[0:20])
```

4.算法实际应用（以京东商城笔记本电脑评论为例）

4.1基于LDA模型的主题提取

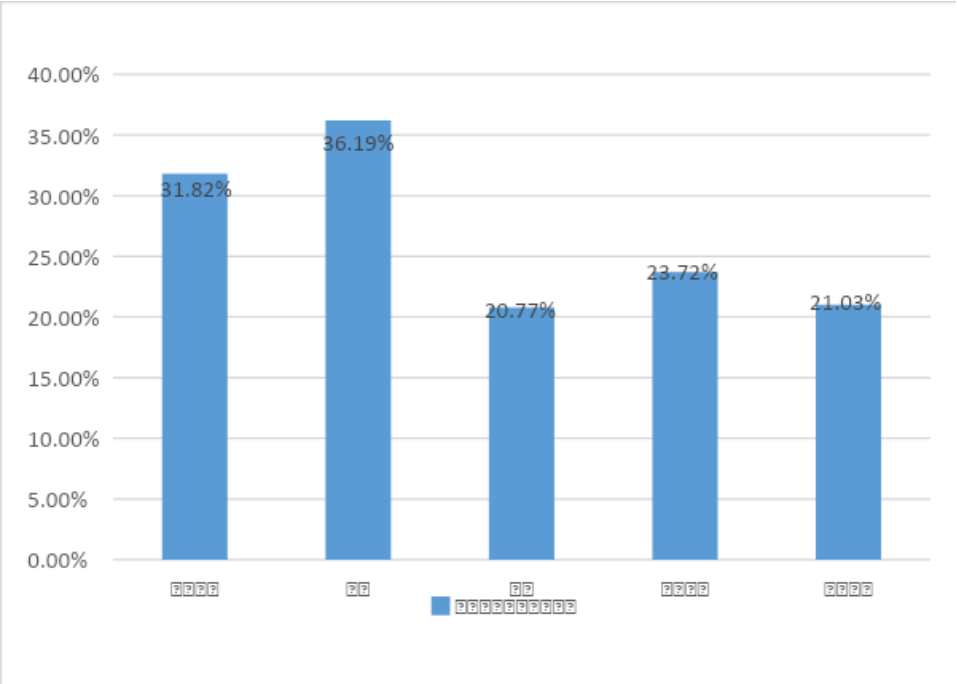
我们基于已经获得的用户评论文本数据，首先通过构建LDA模型完成对用户评论的主题提取工作,以挖掘出用户真正的关注内容。比如，可以区分出哪些用户评论是关于京东提供的服务方面主题的，哪些用户评论是关于笔记本电脑自身特征主题的等。本文对国内华为品牌(9033条）和国外苹果品牌（7321条）两大品牌作为代表分别进行实证分析。

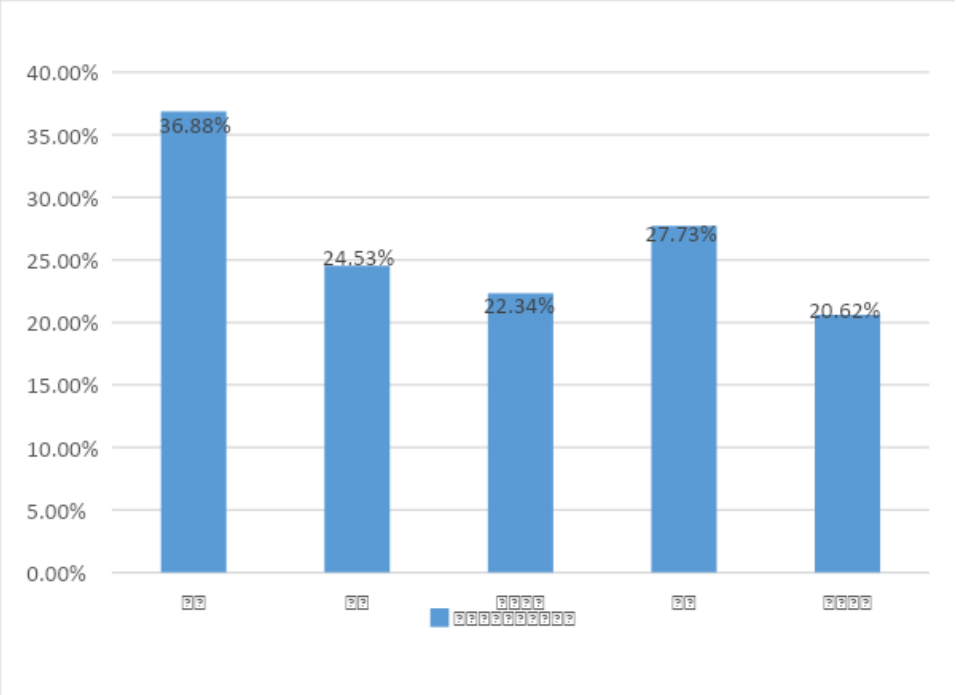
LDA主题提取模型，包含词语、主题以及文档等三层的结构，模型运算结果得到了“文档-主题”和“主题-单词”两个概率分布。本文研究所用的参数估计方法是Gibbs抽样技术，来完成对LDA模型参数进行估计。

对于该模型中关于参数设置问题,通过查看大量文献,设置α=0.1,β=0.02，运算迭代次数G=500，主题数T=5。结合现实中的情况，第一，取得主题数过多不方便进行归类解释，第二，主题数过少则模型的准确度不高，第三，结合用户评论挖掘方面的有关研究，最后决定将主题数确定为5，本文通过Python中的gensim库完成LDA主题模型估计，得到的概率分布，并将结果进行归纳展示。

在对华为与苹果两大电脑品牌分别完成用户评论主题提取后，接着通过Excel软件完成了各主题下的文档数目的统计工作，一方面为了更清晰地比较出消费者在购买电脑时对这两大电脑品牌的关注点的共同点与不同点，另一方面也为后续进一步挖掘主题下的内容做准备。经过筛选与统计后，得出华为与苹果用户评论主题的统计表与统计图。

	主题1	主题2	主题3	主题4	主题5
华为	硬件配置（31.82%）	物流（36.19%）	客服（20.77%）	外观颜值（23.72%）	整体性能（21.03%）
苹果	系统（36.88%）	客服（24.53%）	硬件配置（22.34%）	物流（27.73%）	外观颜值（20.62%）





由图可知，对于华为笔记本用户而言，在选购华为笔记本电脑时主要关注点在主题1与主题2上,即大多数评论都涉及到华为电脑的硬件配置以及电子商城物流方面的内容。其中，涉及快递运输、物流快慢内容的用户评论最多，与此相关评论占比达到了华为用户总评论的36.19%。其次，用尸评论最多的定对笔记本电脑自身硬件配置方面的评价，评论占比达到31.82%，即对于屏幕、键盘、鼠标等硬件的相关评价。在商家客服、外观颜值以及整体性能方面，相关评论占比均衡，大约占华为用户总评的1/5左右。

对于苹果笔记本用户来说，在选购笔记本电脑时苹果系统成为了其主要的关注点，用户关于苹果系统的相关评论占总评论的36.88%，结合实际情况而言，对于苹果用户来说，苹果系统相比于安卓系统有一定的优势。除了关注苹果系统以外,京东物流快递方面的也成为了苹果电脑用户的热评点。其次，苹果用户的关注点是对于电商提供的相关服务评价，评论占比分别为24.53%与22.34%，主要包括服务态度、服务质量等内容。需要说明的是，通过表5-11的统计结果会发现，无论是对于华为用户还是对于苹果用户来说，各主题下的文档占比相加之和大于100%，这是由于部分文档的评论语料较长，可能涉及多个主题内容，因此文档主题之间存在交集。

4.2基于GMM模型的聚类分析

4.2.1 华为用户评论分析

对于华为笔记本电脑，针对上述主题提取的结果，以关注度比较高的硬件配置为例，继续探索消费者对于华为电脑硬件配置方面的关注热点，对该主题下的2876条评

论进行分析，通过中文分词技术，将涉及到硬件配置这一维度的评论进行分词处理，并对一个字的词以及停用词进行去除，做出统计出高频词统计表与词云图，运行结果如下所示:

高频词统计

词语	词频	词语	词频
屏幕	1285	性能	397
使用	865	性价比	3811
很快	722	问题	367
满意	552	体验	345
轻薄	508	手感	344
运行	439	舒服	286
做工	422	配置	283

硬件配置主题下的词云图（华为）

从词云图中可以直观的看出，华为笔记本用户关于电脑硬件配置方面的热评词主要有:“屏幕”、“轻薄”、“满意”、“键盘”、“使用”、“硬盘”等。

接下来对华为笔记本用户有关电脑硬件配置方面的评论（2876条）进行聚类分析，进一步挖掘用户对笔记本电脑哪方面的硬件配置比较关注。由于文本聚类属于高维聚类，为了提升聚类分析效果增加聚类的准确性，本文在聚类之前通过提取tf-idf特征值进行了特征降维处理。在参数设置上，设置min_df=0.2，max_df=0.9，即规定所提取的特征词，至少在整个语料库20%的文档出现过，最多在90%的文档中出现过。由于本文比较关注在高维数据上的聚类效果，故做了简单的假设，即设置covariance_type='spherical'，即不同类的方差协方差矩阵都是相同的，且都为对角矩阵。

通过构建的高斯混合聚类模型，对上述华为用户评论进行聚类分析。本文从现实情况出发，通过查阅大量有关笔记本电脑方面的文献资料，并结合笔记本电脑的基础知识，最终将聚类模型的类别个数确定为6，最终得到的聚类结果及各个类别下的评论数见下表。

硬件配置主题下的评论聚类结果

类别	评论数	类别	评论数
1、显示屏	1254	4、电池	281
2、机身外壳	510	5、硬盘内存	240
3、鼠标键盘	376	6、CPU	215

由表可以看出，华为用户关于电脑硬件配置方面的评论涉及的热点主要包括显示屏、机身外壳、鼠标键盘、电池、硬盘以及CPU等。其中，用户对于华为笔记本的显示屏（屏幕）的提及量最多，在2876条的有效评论中，有1254条左右的评论主要是用户评价笔记本电脑屏幕方面的，评论占比达到43.46%。除了显示屏之外，其他的类别占比分别为17.73%、13.07%、9.75%和8.61%。

类别	评论数	评论内容
显示屏	1254	感觉屏幕大了一点，办公13寸正好，14寸看视频更爽。开机速度也很快，而且屏幕细腻,很清晰,款式好看! 我真的觉得屏幕很好啊2333看起来和我的IPAD屏幕没什么区别,可能是我真的没用过太好的屏幕吧,但我觉得对于普通人来说这款的屏幕绝对够用!!屏幕画质很清晰，外观做的也不错，跟我14寸电脑比较，很小巧，手感也相当好。 总体感觉很值得，打开包装看到的第一眼就让人爱不释手，轻薄机身、触感细腻、88%屏占比+100%的 zB广色域,不会像我的老电脑那样眼睛没看多久就疲劳了，适合长时间阅读办公，就这点，就必须要赞一下这块屏幕!
机身外壳	510	颜值满分，真的很轻薄，刚拿到的时候真的惊艳到了，对比上一台笔记本简直天壤之别，全金属外壳很满意。特地对比了舍友的宏碁的swift，他的塑料背面手感完全比不上金属的，而且因为他有机械硬盘的缘故，所以也厚多了，这台真的是很薄很漂亮- 感觉挺薄的，全金属，第一次用这样的本本，还成。不会很热，适合我办公 简单需要，电池也还可以，做工够用级别吧，要求不高，总体来说一

		台性价 比很高的便携笔记本。 轻薄，负重减少，出差使用也不错- 京东的速度一如既往的快，本来没想这么快收到的。在说说这一周的使用情况,笔记本很轻薄,屏幕尺寸也很大，外观很漂亮
电池	281	电池，充电速度快，充电头挺大的。还在摸索中,电池显示还是比较抗用的,后续在补充。 说说电脑,很轻薄，出门带着确定不用带充电器，一天够用，星空灰在强光下有点泛紫，很骚气,键盘打字没有声音，也很舒服- 电量给力!内置57.4机大电池，日雷续航提升15%，半小时可充入40%电量，其电源适配器也特别小巧，跟手机充电头差不多，而且可为手机充电, 也就是说出门只需带一个充电器就足够，便携性很高- 买了一台送人，看了感觉很不错。又买了一台自己用。开机五秒钟，屏幕看着很舒服，电池续航也很不错。关键是手机是华为的，传输文件，出差充电都很方便-

结合表可以看出用户对华为笔记本显示屏的主要评价内容。通过聚类结果可以发现，对于大多数消费者来说，对华为笔记本屏幕的设计还是比较认可的，屏幕画质感比较清晰、细腻，款式好看。其次在电脑硬件配置的方面，华为用户关注最多的就是机身外壳材质以及重量等方面了,与之相关的评论占比达到了17.73%。由聚类结果可知，在机身外壳重量与材质方面，华为笔记本的轻薄款成为了大多数消费者的热爱，重量轻，携带方面，其次大多数用户比较在意笔记本外壳是否是金属材质。华为用户对于笔记本电脑电池方面的评论，主要关注电池是否续航长久、充电是否快速。通过观察发现，大多数评论对于华为笔记本电脑内置电池还是比较满意的，续航时间长，充电速度快等，对于追求性价比的消费者或者经常出差旅行的用户来说，华为品牌电脑是一种不错选择。

4.2.2 苹果用户评论分析

对于苹果笔记本电脑，针对上述主题提取的结果，同样以关注度比较高的硬件配置为例，继续探索用户对于苹果电脑硬件配置方面的主要关注点，对该主题下的1656条评论进行分析，首先得到其词频统计表与词云图，分析结果如下:

词语	词频	词语	词频
屏幕	792	使用	104
好用	452	服务	102

轻薄	369	满意	80
很棒	322	好用	80
键盘	311	超薄	79
鼠标	272	完美	78
问题	247	性能	78
电池	208	质量	77
运行	129	体验	77

硬件配置主题下的词云图（苹果）

从图可以直观的看出苹果用户对于苹果电脑硬件配置方面的热评词语有:“屏幕”、“好用”、“轻薄”、“手感”、“键盘”、“鼠标”、“音质”以及“电池”等。

接下来对苹果笔用户关于电脑硬件配置方面的评论进行聚类分析,进一步挖掘苹果用户对哪些方面的硬件配置比较关注。本文同样对苹果用户评论语料采用基于高斯混合模型（GMM)的聚类分析，并将聚类模型的类别个数确定为6类，最终得到的聚类结果及各个类别下的评论数见表5-17。

类别	评论数	类别	评论数
1、显示屏	654	4、电池	182

2、鼠标键盘	450	5、声卡音质	60
3、机身外壳	239	6、硬盘内存	71

由表可以看出，苹果用户对于苹果电脑硬件配置方面的评论涉及的热评词主要是显示屏、鼠标键盘、电池及声卡等方面。其中，用户对于笔记本的显示屏（屏幕）的提及量最多，该类别下的文档数目有654条，占了该主题下品论总数的39.49%。其次关注比较多的是键盘、鼠标、电池方面，各类别下的评论数占比分别为27.17%、14.43%、10.99%等。具体聚类内容如下表，由于篇幅的限制，下表给出的为部分聚类详情。

类别	评论数	评论内容
鼠标键盘	450	用了快半个月，真心很不错。而且第二天降价了七百，找客服立刻就给我补了差价，简直不要太爽啊!键盘码字感觉特别爽，手感很不错，值了，性价比很高。 键盘之前说键程短，不过用了以后觉得手感挺好的，很有感觉!还没有用到大软件，不过在系统忙的时候，发热倒是有一点，不过没有到风扇嗡嗡响的程度。 电脑外观非常不错，电脑按键按着舒服，系统自然是没话说，非常流畅，可以满足日常的各种需要，最让我满意的是触控板，太丝滑了，总体而言这款电脑还是非常不错的 一如精品，到手就觉得值。做工上，每个细节都凸显精致，相比较其他笔记本，真是精细的太多。键盘很好用，至于键程问题，适应一下即可。
电池	239	续航满意，充电很快，电源循环两次，总之很满意，希望用mac创造价值!电池使用方面，充满电可以使用六个小时（轻度使用，上网看视频等，剪片子或者编程还没试），总的来说，还是比较满意的。 全新air 风格跟MacBook相似，相当于MacBook plus，屏幕更大一点，续航还强，很满意 非常棒，速度很快整合一下办公室都是用苹果电脑!反应也比较灵敏，电池很耐用，基本充满电可以用两天左右。轻盈，还推荐了几个同事购买!
显示屏	654	笔记本很好，运行速度很流畅，画质清晰，京东快递很快……很满意 非常完美的设计。速度非常快。同时跑两个系统装了虚拟机。感觉非常快非常快。128G办公足够了，屏幕质量非常高，颜色非常亮丽。

		买给媳妇的，上上网看视频可以，不错,屏幕虽然没pro 牛，但是比我的x270要好很多很多…… 挺好的关注很久了刚好做活动就赶紧买了拿到手就迫不及待打开用了画面清晰质量很好

通过聚类结果可以看出，对于大多数苹果用户来说，在购买电脑时，与华为一样，对苹果笔记本屏幕的评价是最多的，从他们的评论中看以看出，苹果电脑屏幕画质感比较清晰，质量很好。除了屏幕以外，苹果用户关注最多的就是电脑鼠标键盘了，设计键盘的相关评论占比达到27.17%，大多数消费者都认为苹果笔记本的键盘做工精致，按起来比较舒服，有手感。相比之下，苹果用户对电脑声卡及硬盘方面的评论比较少，这可能由于大多数消费者缺乏专业的电脑知识，对这些方面关注度不是很高。

4.2.3 聚类模型的评价

为了评价聚类系统的整体性能，判断聚类的精准度，本文共抽出60%的华为用户在硬件配置主题下评论(1725条）进行判别。具体的做法是将硬件配置主题下的各个类别中所包含的用户评论作为总的抽样框，样通过Excel软件编写公式，来实现对样本数据的系统抽样，再根据本文原理部分模型评价的相关指标进行运算，通过统计计算，得出以下的各指标值。

类别	实际值	正确值	偏离值	未入类值	准确率 T	召回率 R	F 值
1	740	518	125	97	80.56%	70.00%	0.7491
2	350	231	67	52	77.52%	66.00%	0.7130
3	235	175	46	14	79.19%	74.47%	0.7675
4	172	118	39	15	75.16%	68.60%	0.7173
5	138	97	31	10	75.78%	70.29%	0.7293
6	90	58	27	5	68.24%	64.44%	0.6629
合计	1725	1197	335	193	---	---	---

由表计算得出T=76.07%，R= 68.97%，F=0.7232。从各个类别的评价指标值可以看出，平均准确率为76.07%，平均召回率为68.97%，平均F值为0.7232。总体来说，基于混合高斯模型模型聚类效果比较好，但对于部分类别的聚类效果不是很好，比如类别6，可能是因为评论中存在大量语意模糊的词语，导致聚类模糊，对聚类结果造成一定的不良影响。

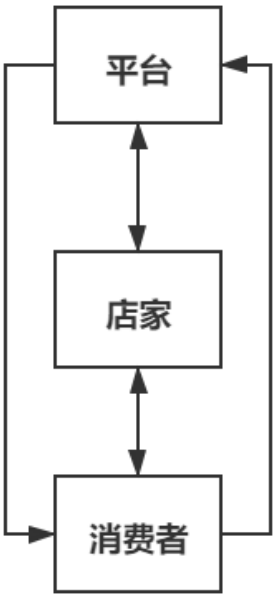
4.3算法实际应用总结

在实际应用上，该算法主要可用于两个方面：网购优化以及网络舆情分析。

4.3.1 网购

当前不少电商平台都存在这些问题：卖家缺乏对消费者关注点的认识以及对当前平台的状态认知不够，消费者不清楚商品的实际使用体验以及之后网购过程的顺畅与否，平台缺乏对店家以及消费者的认识等等。而这些问题的实质都是由于它们对彼此之间的了解不够所导致的，而聚类算法就能够很有效的解决这些问题。

平台店家消费者三方关系图



对于店家来说，在当下电商迅猛发展的时代，能否快速发觉用户的需求对于电商卖家来说显得尤为重要。而借助该算法我们可以对用户的评价进行有效的分析来让店家发现用户真正的需求。正如算法实现这部分所提到的，我们利用该算法对京东商城网站中苹果和华为这两个品牌的笔记本电脑的用户评论进行了分析，从而得出了用户的关注热点。而获得的这些信息对于这两个品牌在新产品的开发，销售等活动中又有

着不小的指导作用，它们可以借此完善自家的产品，让他们更符合消费者的需求。同时，根据用户的反馈，还能让店家发现消费者在购物过程中遇到的其余问题，例如：消费者在快递，售后，支付等方面遇到的问题都可以及时反馈给店家，再由店家同平台等合力解决。

对于消费者来说采用该算法也可以让店家及时了解到其自身的关注点，从而更好的满足消费者的需求。通过对用户评论进行聚类分析，以关键词的形式发布出来，还能让更多的消费者在购物之前对各个商家与商品有个大致的了解，让消费者可以更加快捷的购买到自己想要的产品。

同样的，该算法也可以帮助电商平台发掘出当前网购过程中的不合理地方，例如推送不准确，页面卡顿，支付环节不便捷等等问题，平台可以依据这些信息不断改善用户的购物体验。平台也可以依据分析的结果对消费者进行细分，可以依据这些类别对顾客进行针对性的推送，吸引顾客多次消费。平台还可以利用该算法将现有评论中的关注热点展现给消费者，让消费者可以了解到某一商品大体上的使用体验，吸引消费者消费的同时还可以督促店家提升商品质量，提升平台整体的口碑。

总体来看，对于店家而言，适当的采用该算法可以有效的加强自身的竞争力，进一步发展市场。而对于消费者来说，采用该算法也可以让店家更好的满足消费者自身的需求。同样的，该算法也可以帮助电商平台不断优化系统，平台可以依据这些信息不断改善用户的购物体验。在网购这一方面，采用聚类算法实质是缩短了平台，消费者以及店家这三者之间的距离，让这三者能够更加了解彼此，若是能有效的利用该算法，就可以实现这三者利益上的共赢。

4.3.2 网络舆情分析

当前网络提供的方便促使越来越多的网民选择利用互联网来表达自己的看法，在网络上发表民意，逐渐形成了网络舆情。而这样的环境也带来了不少问题，首先由于发布信息的成本和门槛极低，这就导致网上目前存在不少虚假，不良信息；其次网络的大面积普及导致网上的信息量变得极大，真正需要被注意的热点新闻反倒被淹没在这信息的海洋中了。能否有效的甄别出虚假的信息与真正需要被关注的热点在当前就变得愈加重要了。

网络舆情的分析对于维护社会稳定、促进国家发展具有重要的意义。借助算法对网民的意见与观点进行分析，能够帮助政府部门快速发现，处理问题。尤其在疫情期间，针对随时可能出现的社会现象，政府可以借助该算法快速解决日常生活中出现的问题，维持社会稳定。以前段时间发生的抗美援朝老兵遗失纪念章事件为例，老兵家属在事件发生后在微博上发出求助，短短两天内就获得了大量的网友转发，在事件发生后的第二天晚上，北京退役军人事务局就主动赠送老人一枚新的奖章。这次事件从

网上发布到有关部门作出反应只用了不到36个小时，这也从中反映了算法在控制引导舆论方面的重要性。在往后出现更加紧迫的状况时，若是能有效的采用算法来控制引导舆情的发展，就能让有关部门反应的更加及时。

现如今网上虚假和不良信息经常引发错误舆情导向，形成了大量虚假的负面信息，一旦被网民采信就会造成较大危害。若是能在早期对舆情进行分析，对不良与虚假信息进行跟踪与控制，也就能有效减少谣言所带来的危害。在这个控制与监测的过程中，采用该算法可以快速帮助有关机构获取热点信息，从而能够帮助其快速甄别出虚假与不良信息，控制谣言的传播。特别是在疫情期间，往往有不少针对物价的谣言，为了维持的社会的稳定，就要求有关部门能够及时的察觉到谣言的出现，从而进一步跟踪与控制。

在网络舆情分析方面，采用该算法可以让有关部门及时了解到当前网民的想法与关注热点。在网络飞速发展的今天，越来越多的信息开始在网上传播，要从这茫茫无际的信息海洋中敏锐的发现其中蕴含的重要信息变得愈发的艰难。聚类算法的应用在当前开始变得越来越重要，通过对大量文本进行分析，来让使用者发现信息的重点所在，这不仅仅只适用于上述这两方面。在信息量持续爆炸的当前，聚类算法势必会在其余许多领域内发挥不小的作用。

4.3.3 针对电商平台的商业模式

围绕着算法，我们设计了针对电商平台的商业模式。

我们计划使用文本聚类分析的方法来优化整个平台。具体是通过对某一具体的商品类型的所有评论进行聚类分析，采取出关注热点并对评论进行深入的分析。然后将这些信息打包并发送给制造商，销售商与平台等利益相关者。我们首先通过LDA主题提取模型发现了消费者的关注热点，其次运用GMM模型进行了细致的分析，发现了用户对商品具体某一方面的使用感受，再将信息汇总。以京东商城两个品牌的笔记本电脑为例，我们首先提取出了用户的关注的主题所在，这两个品牌的消费者关注的热点有较大的不同，其中苹果用户很大程度上更关注系统，而华为用户更关注配置和性能。接下来我们对这两个品牌在不同的主题下进行了更细致的分析，得到了用户在具体方面的使用感受。最后我们再对这些信息进行打包处理，发送给利益相关者。

其中对于消费者，我们倾向于通过建立网站来吸引这一群体。将一定时期内某件商品的评论信息提取分析，为有购买意愿的消费者提供信息。当前的用户面对越来越多的商品，往往很难抉择，而许多决定消费者是否购买的重要因素往往都是非常主观的使用体验。通过该算法消费者可以更加直观的了解到产品是否能够让自己满意，确定是否购买。而且通过用户的口碑宣传也可以快速提高网站的访问人数，网站也就能够借助广告位等实现盈利。

对于制造商和销售商，我们会对某一类商品的评价进行分析，提取出主题，再将

其有偿提供给制造商和销售商。同消费者获取的不同，为制造商和销售商提供的是对某一类商品的分析。制造商和销售商可以了解到消费者在这一类商品上更想看到什么，是更强的性能还是更好的外观。制造商可以根据这些信息不断改进产品，满足消费者的需求，而销售商也可以根据信息不断调整在架的商品，让他们更加符合消费者的需求，提升营业额。

对于平台，我们主要是获取某些店家的所有评论，针对其中重复的评论进行分析，来分析是否存在刷单，恶意差评等情况，再将各个店铺的数据汇总形成报告有偿发给平台方。平台方借助这些信息可以更快的对各个店铺进行分类，对于数据优异的店家，平台方可以为其引流进一步提升店铺人气，拉动消费；对于存在刷单的店家，由平台方给与警告或惩罚等等。同时借助该算法还可以快速识别出被恶意差评的商家，平台可以快速行动，维护平台的经营环境。良好的经营环境可以进一步吸引其余店家的入驻，形成的良性循环能够帮助平台不断壮大，提升自身的影响力。

5. 分析与展望

本文主要对高斯混合聚类模型进行算法研究，将高斯混合聚类模型进行实际应用。

首先通过Python爬虫技术，对京东商城中关于笔记本电脑的用户评论进行了爬取，得到原始评论数据185289条。在完成相关预处理工作后，得到有效评论数据147586条，其中包括华为笔记本电脑用户评论数9033条，苹果笔记本电脑用户评论7321条。在完成数据清洗的基础上，本文采用了基于隐马尔科夫模型的Jieba精确模式分词的中文分词方法，完成了有效评论的分词处理。进行去停用词、词频统计、词云图特征等分析工作之后，通过LDA模型分别对华为与苹果用户评论进行主题提取，分别对华为与苹果用户在硬件配置主题下的评论进行聚类分析,利用构建的GMM模型完成了该主题下的评论聚类，旨在进一步挖掘了该主题的核心内容，这在进一步分析消费者行为，以及在指导产品改进方面发挥了一定的作用。

对未来的展望主要有以下三个方面:第一，在文本聚类算法模型的研究方面，可以在传统的模型基础上，加入其它的聚类算法进行结果对比分析，选择出最优的聚类模型来进行分析，以尽可能地提升聚类效果。第二，在用户评论文本分析方面，可以加入用户的情感分析研究，通过分析用户评论的极性，可以更加精确地感知用户的态度。第三，在算法的推广应用方面，本文研究的聚类算法可以迁移到其他电商平台产品，通过对用户评论的有效分析和挖掘，能为消费者、生产者以及电商平台提供精准、有价值的建议。