# 第一章 引言

## 1.11 项目背景

情绪是人在探索和发现周围环境时，对外界环境刺激做出的心理或生理反应，是一种综合了人的思想，感觉，和行为的状态。常见的情绪有高兴，平静，悲伤，生气等。积极的情绪可以让人精神焕发，干劲十足，消极的情绪可以让人萎靡不振，日渐消沉，人的情绪极大影响了人们工作生活的效率。因此在这种情况下，对情绪的了解显得极为重要。

人的情绪变化时通常伴随着人体的一系列生理表现，或者行为表现，比如心跳加快，血压升高，脑电波活跃，激素分泌，又比如欢呼雀跃，手舞足蹈，唉声叹气，愁眉苦脸等等。正是因为情绪变化时，人体会产生这些变化，那么，如果我们可以通过愈来愈发达的技术手段准确的获取人体的这些信息，并且准确有效，再加上科学正确的分析，我们就可以判断出用户此刻的心情。情绪分析和识别是一项多领域交叉的科研问题，目前，多个领域都在对于情绪进行相关的研究，涉及的领域有医学，心理学，神经科学，人工智能等。

目前解决情绪分析问题主要有两种思路，一类是基于生理信号进行情绪识别，一类是基于非生理信号进行识别。非生理信号的情绪识别研究主要分析面部表情，语音，语调等，也包括人的行为动作。不同的表情常常会反映不同的情绪，所以我们可以通过分析人的面部表情以及伴随的面部肌肉动作来判断人的情绪。比如嘴角上扬时，并且眼角出现皱纹，我们通常可以判断此时的情绪状态为高兴；眼睛瞪大，眉头紧皱，我们通常可以判断此时情绪状态为愤怒。不同的语言表达方式通常也意味着不同的情绪状态，比如，人的心情愉悦时，语音会轻快，语调会上扬，心情失落时，语调会变得低沉，语调会放缓。基于非生理信号的情绪识别研究简单易行，操作难度小，但是存在识别不准确和不可靠的问题，毕竟人是种复杂的动物，常常会为了隐藏自己的情绪而伪装自己的声音表情。另一方面，对于一些病患人群可能无法提供非生理信号情绪识别所需的特征，所以对于这一人群也不适用。另一大类是基于生理信号的情绪识别。这类研究关注的重点通常是人体的心率，血压，皮肤阻抗，呼吸，脑电波等生理信号，这些信息相比非生理信号而言，不易被伪装，并且通过这种方式，在获得准确信息的前提下，分析结果准确率高。但是该方法也存在着存在收集困难，如收集难度大，信号准确率无法保障，易被干扰的缺点。

情绪识别这项技术的应用前景十分广泛，比如听音乐时，如果能够识别出用户此刻的心情，进而推送给用户符合当前心情状态的歌曲，可以极大提高用户的使用体验。再比如，在医疗护理过程中，如果可以准确分辨出患者此刻的心情，尤其是具有表达沟通障碍的患者，进而对其提供针对性的护理，无疑对于治疗过程是有极大帮助的。再比如，如果可以准确获知用户使用一款产品时的情绪状态，那么开发人员就可以针对用户使用时的情绪进行更有针对性的优化处理，从而持续提供更好的服务。

## 1.2研究现状

1.2.1 文本情绪分析

1.2.2

1.2.3

## 1.3本文的研究内容

下文主要研究基于机器学习的安卓移动用户情绪分析系统的设计与实现，即将从安卓设备采集来的数据，通过经典的机器学习算法进行分析，从而预测出使用者现在的心情状态。机器学习经过多年的发展，相关理论已经非常成熟。随着科技水平的飞速发展，并且伴随着数据量的暴增，人们进入“大数据”时代，在这样的时代背景下，机器学习具有传统统计难以匹及的优势，因此本文选用了机器学习的方式来进行情绪的识别。之所以选用安卓手机来收集这些信息，是因为，首先，越来越多的人拥有智能手机,其中绝大部分是安卓手机（下文“手机”，“智能手机”都指代“安卓手机”），安卓手机的大规模使用保障了数据的充足，在之前是不存在这样优越的研究条件的。其次，随着手机越来越智能，功能越来越多，浏览网页，聊天阅读，拍照娱乐，移动支付等等，使用手机的场景越来越多，人们越来越离不开手机，这使得很多人有了随身携带手机的习惯。并且，手机的携带对被测试者的正常生活影响较小，这使得持续获取数据成为了可能。最重要的一点是，手机智能化的过程中承载的传感器越来越多，cpu处理能力越来越强，这些都为我们进行实验提供了便利。

# 第二章 相关技术研究

## 2.1 数据预处理

通常我们收集来的大量数据，是不能直接被我们使用的。收集来的原始数据有可能会存在包含大量噪声，数据不完整（比如有的属性可能缺失，或者不确定），数据不一致（比如在不同表中的同一属性名称不一致或者数据矛盾），度量单位不一致等问题。可能产生的原因有，比如收集过程中设备使用方法不正确，用户操作不当，或者设备出现故障和异常，信息收集过程受到干扰和中断，再比如数据收集后数据存储不当，工作人员误操作等等。这就需要我们通过科学的数据预处理技术对数据进行清洗，剔除坏值，非法值，异常值，填补空值等操作，来消除数据中的噪声，保证数据数据一致性及数据完整性，保证收集来的数据有效和可用。除此之外，我们还需要数据进一步进行集成，规约，变换等操作，使得数据满足我们研究分析的要求。

2.1.1 数据清洗

1. 孤立值处理：筛选出数据中与周围数据的范围明显不符的异常值，即孤立值，分析其产生原因，对于错误数据，通常采取空值替换，直接删除，平均值修正等处理。

2. 缺失值处理：我们收集来的大量数据难免有属性缺失的情况存在，我们一般的处理方法有中位数替代法，平均值替代法，频率最高值替代法，默认值替代法，邻近值替代法或者根据需要直接删除属性缺失的行或者列。

3. 噪声平滑：对于收集来的数据中的噪声，通常先采取分箱方法，将数据按照一定规律排好序，然后依据等深分箱原则或者等宽分箱原则将数据分成几组，然后根据需要，选择边界平滑，均值平滑，中值平滑中的一种或者几种对数据进行处理，即使用每组中的边界值、平均值或者中值代替同一组的数据。

2.1.2 数据集成

数据集成是把多组源数据融合成一组数据，这多组源数据可能来自多个不同的数据库或者不同的文件，所以集成的过程中要消除数据不一致和数据冗余。数据不一致主要表现为属性名称不一致或者数据矛盾。数据冗余一般是同一属性名称多次出现或者属性间存在线性关系。一般通过相关性分析来消除属性间线性相关。

属性A和属性B的相关性计算：

如果，则有A与B正相关。

如果，则有A与B负相关。

如果，则有A与B相互独立。

因此，如果很大，那么就说明A，B相关性很强，可以删除其中一个。

2.1.3 数据变换

数据不变换主要是把数据标准化，消除量纲的影响，比如把属性按比例缩小，把属性放到一个。

1. 最小最大规范化：

已知属性区间，将属性的取值范围由[old\_min,old\_max]映射到[new\_min,new\_max]

该种方法保留了原来数据中存在的关系，但如果将来遇到超过目前属性[old\_min,old\_max]取值范围的数值，将会引发错误。

1. Z-score标准化：

根据属性的均值和方差来对属性进行规范化，一般在最大最小化规范化出现异常数据时使用。

其中的和分别为属性A的均值和方差。

## 2.2 特征提取

## 2.3 主成分分析（PCA）

2.3.1 主成分分析概述

在一般的研究分析过程中，常常需要对反映研究对象特征的多个属性进行大规模的数据收集来进行规律的发现和寻找。毫无疑问，样本容量大，属性多可以为研究和分析过程提供给更多的有用信息，为分析数据中的规律提供便利。但与此同时，也带来了数据收集整理工作量增加等问题，而且多维度数据的特征提取极容易造成维度危机，给分析工作造成极大挑战。而如果只分析单个指标，又会导致分析是孤立的，缺少全局性和综合性。单纯的为了降低工作量去减少分析指标，很容易得出错误的分析结果。因此，需要找到一个有效方法，在尽可能少丢失原属性带有的信息的同时，降低需要分析的属性个数，以达到既能全面分析所收集的数据，又能在一定程度上降低工作量的目的。由于各属性之间不是完全孤立存在的，这就为我们用较少的属性表现全体属性所携带的规律提供了可能。主成分分析就是这样一种降维方法。主成分分析在日常研究中应用十分广泛，在统计学，地理学，经济学，社会学等多个领域都有应用，是一种常用的多元数学分析方法。

2.3.2 PCA降维原理

一些高维度数据，数据中的不同属性之间，常常有一定的关联。为了解释降维原理，以淘宝的交易数据为例，假设订单中有商品编号，购买数量，商品单价，商品总价四个属性，那么明显可以看出商品总价可以由单价和数量的关系反映出来，所以这项属性就是冗余的。删除掉这项属性，也就是降低了维度，并不会损失数据的信息，但却可以有效降低数据信息的分析、整理、计算过程中的工作量，有效降低后面构建模型时模型所需的空间时间开销。我们降维的目的就是将近乎无损的数据信息用尽可能少的属性或维度表现出来，最后的效果就是各属性间不存在线性关系。

PCA降维是通过把高维数据投射到低维子空间进行降维，比如把二维平面的点投射成线，数据集中的值由两个值变成一个值表示，这就实现了二维数据的降维。同理，把三维数据投射到一个平面，就实现了三维数据到二维数据的降维操作。n维数据集以此类推，通过映射，将n维数据降维到k（k<=n）维子空间。

降维步骤：

假设原数据是m行n维，首先，由原数据生成n行m列的矩阵。然后对训练集中的数据特征进行归一化处理，然后计算协方差矩阵，接下来要求出协方差矩阵的特征向量和特征值，这里要用到奇异值分解算法。最后一步，进行降维计算，通过获得的降维矩阵，就可以将高维数据映射到低维子空间了。

经过降维之后的数据，在低维下更容易处理，更容易使用。相关特征，特别是重要特征更能在数据中明确的显示出来；如果只有两维或者三维的话，更便于可视化展示。此外，应用PCA进行特征降维还可以去除数据噪声，降低算法的开销。

## 2.4 机器学习分类器

2.4.1 k-NearestNeighbor算法

K最近邻 (k-Nearest Neighbors，KNN) 算法工作原理：一组每个数据都带有标签的数据集，被称为样本集。样本集中的数据和其对应分类是已知的。输入样本集后，通过对样本集数据特征进行分析，训练出模型。等再输入不带标签的新数据时，提取出新数据的数据特征，与训练好的数据模型进行比对，从训练集中提取出k个与新数据最相似的样本的标签，选出这k个数据里面比例最高的标签作为新数据的标签，从而实现分类的目的。通常情况下，k的取值不大于20。 K最近邻的主要过程：

1、计算测试对象到训练集中每个对象的距离

2、按照距离远近排序，这个距离可以是欧式距离，马氏距离，曼哈顿距离。

3、选取与当前测试对象最近的k的训练对象，作为该测试对象的邻居

4、统计这k个邻居的类别频率

5、k个邻居里频率最高的类别，即为测试对象的类别

k值选取较小时，也就意味着使用待分类点周围较少的邻居点进行预测，比如极端值k=1,那么也就是由待分类点最近的一个点来对该点进行分类，如果一旦该邻居点是噪声点，那么就会对预测结果造成较大误差。K值选取较大时，可以有效降低噪声的影响，但是容易导致分类界限不明显。k的具体取值一般与数据情况有关，不合理的k值会在一定程度上产生过拟合或者欠拟合的问题。具体的k值选择可以通过经验判断或者交叉验证来确定。

2.4.2 SVM 算法

2.4.3 朴素贝叶斯算法

朴素贝叶斯算法一种简单而且效果比较不错的弱分类器，其理论基础是概率论中的贝叶斯理论。朴素贝叶斯算法虽然构建简单，分类效果却很优秀，甚至比许多复杂算法还要高效，尤其是在大型数据集，表现更佳。之所以说其朴素，是因为朴素贝叶斯算法是基于各个样本特征相互独立的假设的。举个例子，比如一个男生具有长得高，皮肤白，性格好的特点，我们得出结论该男生受女生喜欢，虽然可能这些特征之间具有一定的关联，或者相互依赖，但在朴素贝叶斯算法看来，这些特征在判断男生是不是受女孩喜欢的问题上，特征之间是相互独立的，并且对事件的影响是相同的，即权重相同。

先验概率：在事件发生前，基于历史事件统计，或者背景常识，或者人的经验判断得出的事件可能发生的概率。比如，天空中有乌云，历史上有乌云会下雨的可能性是70%，那么70%就是先验概率。

后验概率：在事件发生后，依据事件发生的结果反推该事件是由某因素引起的概率，即执果寻因。举个例子，中午吃了苹果，下午肚子疼，我们想算一下肚子疼是由吃苹果导致的概率，这就是后验概率。

贝叶斯理论：

事件X在事件Y发生的条件下的概率，与事件Y在事件X发生的条件下的概率是不一样的，贝叶斯公式就是用来描述这种关系的。

其中P(X|Y)是在Y发生的情况下X发生的概率。X，Y都是事件，并且P(Y)不为0。P(X)是事件X发生的先验概率，P(X|Y)是B的后验概率。

当上式被如下表示，，y表示类变量，X是特征向量：

由朴素贝叶斯的朴素假设，也就是每个特征变量之间相互独立，即有：

所以，前面式子又可以表示为：

由于输入数据与分母是常量相关，于是进一步推导：

所以，我们的目的是选择出类变量y的所有可能值的中使概率最大的那个可能值，可以用公式表示为：

最后，通过与的计算得出结果。

朴素贝叶斯分类器具有算法逻辑简单，实现难度小的的优点，而且因为朴素贝叶斯分类器建立在特征相互独立的假设下，只涉及二维存储，所以分类过程中时间复杂度低。朴素贝叶斯分类器在样本特征比较少，并且特征之间相关性小时，具有较为良好的分类效果，而且理论上，相比其他分类方法，朴素贝叶斯算法误差率最低。但实际应用中，情况往往比较复杂，样本特征比较多的时候往往难以保证特征间相互独立的假设，分类效果不好。

2.4.4 决策树

决策树原理简单，易于理解。

决策树对于数据要求简单，不必进行数据清洗，数据预处理等操作，允许数据保留冗余或者空白的属性。

不必像其他分类方法一样统一数据属性，决策树算法可以同时处理常规型和数据型属性。

决策树是一种“白盒模型”，

决策树的优点：

一、 决策树易于理解和解释.人们在通过解释后都有能力去理解决策树所表达的意义。

二、 对于决策树，数据的准备往往是简单或者是不必要的.其他的技术往往要求先把数据一般化，比如去掉多余的或者空白的属性。

三、 能够同时处理数据型和常规型属性。其他的技术往往要求数据属性的单一。

四、 决策树是一个白盒模型。如果给定一个观察的模型，那么根据所产生的决策树很容易推出相应的逻辑表达式。

五、 易于通过静态测试来对模型进行评测。表示有可能测量该模型的可信度。

六、 在相对短的时间内能够对大型数据源做出可行且效果良好的结果。

七、 可以对有许多属性的数据集构造决策树。

八、 决策树可很好地扩展到大型数据库中，同时它的大小独立于数据库的大小。

决策树的缺点：

一、 对于那些各类别样本数量不一致的数据，在决策树当中,信息增益的结果偏向于那些具有更多数值的特征。

二、 决策树处理缺失数据时的困难。

三、 过度拟合问题的出现。

四、 忽略数据集中属性之间的相关性。

2.4.5 随机森林

2.4.6 Adaboost算法

Boosting是一种集成学习技术，也被称为增强学习法，可以通过训练出多个预测精度较低的弱分类器，然后把这些组合起来成为一个分类精度高的强分类器。在难以直接构造出高预测精度分类器的情况下，这种方法无疑为研究人员提供了一种行之有效的算法设计思路。其中Adaboost是boosting算法中一种十分成功的应用。

Adaboost算法，即自适应增强算法，是一种迭代算法。通过对同一数据集迭代训练不同的分类器，每次找到一个最优的分类器，然后下一次迭代时增大前一个分类器错误分类样本的权值，减小正确分类样本的权值。最后将得到的多个最优的分类器组合起来就得到一个强分类器。

Adaboost算法实现步骤：

1、初始化各样本数据的权值。假设有N个训练数据，第一次开始迭代时，各个样本被赋予相同的权值

2、对数据集迭代训练弱分类器。如果训练过程中，某个训练数据被准确分类，那么在下一次迭代过程中，降低该训练数据的权值，同时提高被错误分类的训练数据的权值。一次迭代过程完成后，使用权重值更新后的训练数据集进行下一次迭代，构造新的弱分类器。如此迭代下去，完成整个训练过程。

3、集成各个弱分类器构建一个新的强分类器。为了让分类准确率高的弱分类器发挥更大的作用，按照分类过程中各个弱分类器的误差大小情况，为各个弱分类器分配权重。误差率越小的分类器，在构建强分类器的过程中，所占权重越高，否则，所占权重越小。这样，一个强分类器就构建完成了

Adaboost是一种简单有效的分类算法，很好地利用了不同弱分类器进行级联，并且在构建过程中充分考虑了不同分类器的权重问题，分类结果精度高。主要缺点有，分类精度可能会受数据不平的影响而下降，时间复杂度高，弱分类器的数目也就是迭代次数不易确定。

## 2.5 验证方法

交叉验证：