中山大学移动信息工程学院本科生实验报告

（2017年秋季学期）

课程名称:*Artificial Intelligence*

年级	1501	专业（方向）	移动互联网
学号	15352015	姓名	曹广杰
电话	13727022190	Email	1553118845@qq.com

一、实验题目

Bayes for Regression

二、实现内容

使用Bayes处理回归问题，使用相关系数衡量回归结果。

算法原理

Bayes方法是对于同样的事件，通过先验概率推测后验概率的方法。分为伯努利模型与多项式模型，本次实验中实现多项式模型。

在多项式模型中，遵循一个假设——即所有词的出现都是各自独立的，不会因为出现了“Same”前面就一定要加一个定冠词。这种情况下，对于每一个词出现的概率就可以估算了，本次实验中使用频率代替概率。

所以当一个句子中出现了很多词，每一个词在出现的概率都各自独立，既然这些词恰好组成了当前的句子，就应该把这些词出现的概率相乘。而对于不同的情感来说，每一个词在不同的情感中出现的概率都不太一样，这样计算出的不同情感的权值也就不太一样。如此这般，也就知道所给出的数据在特定的感情下所出现的概率。

然而，这个概率的前提是感情是明确的，事实上，这些感情信息一直都作为系数存在，感情的系数也是不明确的。为了获得当前测试句子的最大概率，我们应该在已有的词汇概率的基础上，乘以当前的感情可能出现的概率——这才是在该感情下，当前句子可能出现的概率。

通过这些概率，我们可以在后续处理无论是分类问题还是回归问题中都有所依仗。

伪代码

通过两个文件对算法的符合度进行进行比较。实验中给定了训练集数据与validation数据，笔者的实现步骤基于二者的数据，并对数据进行整合和运算。

整体思路

- 整合训练数据
- 对validation数据进行回归运算
- 输出运算结果

```
int main(int argc, char** argv) {
    train("训练集文本名称");// 自定义函数train
    for(int i=0; i<训练集的行数; i++){
        // 更新每一行的计算结果
    }

    valid("测试集数据文件名");// 自定义函数validation
    for(int i=0; i<测试集数据行数; i++){
      // 将测试集数据的一行，转化为一个Tuple
      // 根据测试集的数据计算情感系数
    }
    // 将得到的信息输出到CSV文本中
    return 0;
}
```

关键代码截图

在笔者阅读了实验要求之后，进行实验之前，笔者意识到应该建立几个全局变量，这些全局变量将会在多处计算中派上用场：

- 所有出现过的词的有序集合（以出现顺序为序）
- 所有的元组的集合（实际上就是train本身）
- 所有的测试数据的文本的集合

```
extern vector<string> vocbase;
extern vector<Tuple> Tuplebase;
extern vector<string> Validbase;
```

结构体框架

笔者在原有的元组基础上，为了建立新的情感数据管理机制，对该类进行了发展：

```
class Tuple{
public:
//  For updating; 部分接口
    void Collectvoc(string s, bool fix_base);// 拆词，更新voca数据，bool 表是否入库
    void Updatebaserow(); // 更新baserow数据
    double Distanceof(Tuple tuple);
    void taste(); // 由VOCA推算情感系数

    vector<string> voca; // 所有句子中的词
    vector<double> baserow; // 全局TF矩阵的一行
    vector<double> vocarow; // 与voca中的词对应的TF矩阵中的系数
private:
    vector<double> emotionbar; // 储存六欲的容器
};
```

目前已经将输入的数据转化为了储存的元组信息，这些信息将可以用于判断当前处理的数据的情感系数。

根据实验要求和Bayes的理论，当一个句子中出现了很多词，每一个词在出现的概率都各自独立，既然这些词恰好组成了当前的句子，就应该把这些词出现的概率相乘。而对于不同的情感来说，每一个词在不同的情感中出现的概率都不太一样，这样计算出的不同情感的权值也就不太一样。如此这般，也就知道所给出的数据在特定的感情下所出现的概率。

测试数据某词的概率估算

由于在实现的过程中，测试数据总可能出现训练数据中未曾出现的数据，这就直接导致输出信息为0，为此实验要求使用Laplace平滑对数据进行加工，即：

> 词A出现的概率 = (A在句子S中出现的次数+1)/(句子S的总词数 + K);

代码如下：

```
// 遍历句子中出现的所有的词
for(int vwpin=0; vwpin<vwgts.size(); vwpin++){
  // vwgts-value's weight 表示每一个词出现的次数/频率
  double iniwgt = vwgts[vwpin];
  // vwgtsum 是总词数    vwgts.size表示当前句子有多少个词
  vwgts[vwpin] = (iniwgt+1)/(vwgtsum + vwgts.size());
}
```

经过Laplace平滑的计算，我们可以得到这个词在这一行中出现的频率。接下来就是对其他的词也使用同样的算法，获得频率表示概率再相乘。

与当前情感概率相乘

然而，这个概率的前提是感情是明确的，事实上，这些感情信息一直都作为系数存在，感情的系数也是不明确的。为了获得当前测试句子的最大概率，我们应该在已有的词汇概率的基础上，乘以当前的感情可能出现的概率：

> 句子的概率 =
>
> 词A在感情e下出现的概率 x 词B在感情e下出现的概率 x.....x 最后一个词在感情e下出现的概率 x 感情e出现的概率

这才是在感情e的条件下，当前句子可能出现的概率：

```
if(crt_tp.emotionbar[empin] == 0){ // 若该行当前情感系数为0
  crt_em = 0;// 之前的一切计算作废
}else{
  // vwmultp 表示之前的（词的概率）的（乘积）
  crt_em = vwmultp * crt_tp.emotionbar[empin];
}
```

这样，我们就可以完成对一行的信息的比较了，测试集中还有很多数据，这些数据都需要进行比较，并将每一行都得到的概率做和。

```
for(int tpin=0; tpin<Tuplebase.size(); tpin++){
    ...
  emoj[empin] += (10000*crt_em);
}
```

至此，不同的感情系数对预测的影响才算是计算结束。目前的emoj数据已经是对训练集数据的所有因素都考虑过计算过的值的归纳了。之后的工作就是对这个数组中的数据进行归一化、更新当前数据的情感系数。

创新点与优化

笔者进行优化是基于Laplace平滑的基础上进行的优化操作。分析如下：

在Laplace平滑之前：

```
    double iniwgt = vwgts[vwpin];
    vwgts[vwpin] = iniwgt;
```

而在Laplace平滑之后：

> vwgts[vwpin] = (iniwgt+1)/(vwgtsum+vwgts.size());

所得出的数据是在原有的权值上除以一个大数进行消减，再为之添加一个常量。由分析可知，这种方式对于当前权值的波动影响有巨大的削弱。这种削弱将会掩盖真实的波动情况，为此，笔者修改当前的数学模型，以放大当前权值的影响。

而在对于数学模型的选择上，使用正比例函数的扩大是不合理的，因为之后的归一化函数将会对所有已经放大过同等倍数的数据除以这些数据的和——因为6种情感的emotion值都是放大了相同的倍数，所以归一化的时候我们所做的乘积操作就会毁于一旦。

除了正比例函数，还是有很多的函数可以用于放大差距，以区分当前数据的波动的，但是我们还要在这些函数中进行筛选，这种情况下我们就需要考虑我们正在处理的数据的特点：

- 处理的数据是词在当前的句子中出现的频率
- 频率处于0-1之间

为了在0-1之间实现大的区分度，可以使用一个有渐近线的函数，且渐近线为直线x = 1，但是这个函数有可能会造成数据的过度区分，这与我们预期的平滑目的相违背。因此，我们需要选择的模型：

- 在0-1区间内，没有过分的攀升
- 也不会造成原有的0的状况
- 同时还要避免常数项的覆盖影响。

所以，笔者使用指数函数在0-1部分的走向进行拟合：

> vwgts[vwpin] = pow(1.001, iniwgt) - 1 + Tiny;

得出结果：

	anger	disgust	fear	joy	sad	surprise
r	0.320878709	0.25812482	0.34382751	0.39414226	0.385189188	0.350457243
average	0.342103288					
evaluation	低度相关 666					

三、实验结果与分析

1. 实验结果示例展示

	anger	disgust	fear	joy	sad	surprise
r	0.213867689	0.099397069	0.08786095	0.171637159	0.14087297	0.158063714
average	0.145283259					
evaluation	极弱相关 加油哦小辣鸡					

优化后：

	anger	disgust	fear	joy	sad	surprise
r	0.320878709	0.25812482	0.34382751	0.39414226	0.385189188	0.350457243
average	0.342103288					
evaluation	低度相关 666					

2. 评测指标展示及分析

嗯，又是小辣鸡……为什么会是小辣鸡呢？首先我们分析一下什么是小辣鸡——

考虑我们的输出数据全部是0的情况，假如不考虑被除数为0，那么可以说我们的输出数据与处理的数据毫无关系。为了简化模型，笔者将这种情况考虑为0——其实并不是。

那么小辣鸡就是在原有的毫无关联的基础上有一点正向的波动，但是波动的程度不够，换句话说，就是每一行的数据都差不多，或者——都差不多地和预期的数据没什么关系。

笔者认为有如下几个原因：

- 训练集中未出现的测试词集过多

  在尚未实现Laplace的情况下，如果当前的词在训练集中没有出现过，那么该词所占的比重就一定是0了，这种情况使得输出数据有一半都是0——就没有什么参考价值。而在实现了Laplace平滑的操作之后，所有本应该是0的数据都转变为一个平滑了的数据，虽然不是常数，但是也和常数差不多，因为训练集没有这个词的出现，所以就没有参考价值。这种情况使得本来半篇是0的情况转变为了半篇都是常熟的情况，本质上并没有发生过多的变化。

- 数学模型中用于平滑处理的常数项所占的比例过大

  在尚未使用Laplace平滑之前，如果对输出的结果进行归一化，由于有很多的信息都是0，导致归一化之后的数据情感分明特别明显，或者都是1，或者是0.80以上，当然这未必符合我们的预期值。但是Laplace平滑之后，由于分子分母添加的常数值，如果遇到分母极小的情况，很可能会使得有微小波动的分子数据的波动被隐藏在小数点后4位到5位，这同样会使得数据的波动程度不够。

四、思考

- 为什么Knn相似度加权的时候，要将距离的倒数作为权重

  距离表示当前检查的元组与比较的元组之间的距离，距离越长则这两个元组的相似度越低，对照的元组参考性就越低，这种情况下，参考性与距离之间就呈负相关关系。因而，使用反比例函数是符合要求的。

- 如何归一化

  如果当前的情感系数不都为0，那么可以对所有的系数做和，再将每一个系数除以系数之和，以达到归一化的目的。

  如果当前的情感系数都未被探测到，可以都设置为1/6，但是这种方法的对该问题的符合度不高。

- 矩阵稀疏度不同的时候，曼哈顿距离与欧式距离有什么不同

  曼哈顿距离与欧式距离都是距离范式中的一种形式，曼哈顿距离取参数值为1，欧式距离取参数为2。

  当矩阵非常稀疏的时候，进行计算的数组中都具有很多的0，而非零的数据密度极低。在计算距离的时候，如果非零的值大于1，则曼哈顿距离计算得到的距离更大，而值小于1的时候，计算的距离更小——所谓波动都比欧式距离更小，因此，欧式距离的计算结构更加稳定。

  结合矩阵的稀疏度，对于不同的矩阵：

  - 使用曼哈顿距离计算得到的数据波动幅度更大，可以作为区分矩阵的特征量
  - 使用欧式距离计算得到的数据幅度波动较小，可以限定对数据的过度拟合分析

- 伯努利模型与多项式模型各有什么优缺点

伯努利模型的重点集中于文本的数量，伯努利模型是以文本为计算单位的。而多项式模型是以词为单位的。

伯努利模型在应对区分度大的文本集的时候，优势大于多项式模型，伯努利模型的处理方式决定了其能够很清楚地分清，当前这个词，在文本中到底有多大的价值，如果在此类文本中，出现这个词的概率过于小，那就说明该词并不能被人们接受地表达该类别文本的意思。但是伯努利模型在处理少量数据的时候就显现出巨大的劣势——如果我们的文本数量少得可怜，但是文本的内部有具有大量的信息可以挖掘，伯努利模型在这方面的表现很显然是不足的。

多项式模型对文本之间的区分并不是非常敏感，多项式模型在分析一个词的时候，参考的主体是整个词库，这个词库表达的内容繁多而且值得挖掘，如此，对于这个词的理解就很深刻。但是同时多项式模型淡化了文本的概念，过度放大了出现次数所占的比重，这种特点的结果就是对介词副词等无实际意义的连接词没有任何抵抗力，非常容易混淆视听。

- 如果测试集中出现了一个全词典都没有出现的词应该如何处理

  常规来讲我们是没有任何办法的，在Bayes的regression实验中，有大量的测试数据都没有在训练集中出现过，这种情况下，就不得不放弃对于整个句子的句意的猜测。

  如果该句子中还有其他的词汇，我们可以降低这个未知的词在分析过程中所占的比重，将重点转移到其他已知内涵的词语中。

  但是正如Bayes的regression实验中，有的句子中一个词汇都没有出现过，这种情况下就真的是一筹莫展，就算是人，也鲜有办法。