

基于朴素贝叶斯的 情感分析

赵海臣

背景

🌀 为了能够及时分辨出用户评论的情感，用于评论归类，以及将用户的负面情感交给客服进行善后，需要开发一套能够及时识别用户情感的算法。

🌀 已经开发了一套基于关键词识别的情感分析算法，但是基于手工筛选的关键词覆盖率有限，并且识别规则过于死板。于是想到开发一套基于朴素贝叶斯的全局性词频统计算法。

基于单词分析的情感分类

基于单词分析的情感分类

- ★ 假设：不同情感的文本之间的单词使用是有明显区别的
- ★ 途径：文本分类通过分析文本的单词组成来判断
- ★ 数据来源：将所有评论分解成BOW向量，基于朴素贝叶斯公式计算出对应的类先验概率、类条件概率。
- ★ 算法类型：朴素贝叶斯
- ★ 预测方法：在预测时先对评论分解成BOW向量，再调用计算好的类先验概率、类条件概率来预测评论所属类别。

训练数据来源

🌀 训练数据两种来源：

★ 人工标注：

- 通过人工的方式审视每一条评论，对评论的情感进行打分，从而获得训练数据。缺点是，人工标注很容易引入标注者的主观色彩，与标注者的性格、心情息息相关。

★ 基于用户反馈的标注：

- 用户在写完评论后，有一个主观星级打分，要求用户输入购物时的心情。评论+用户的主观心情经过处理可以作为我们的用户反馈标注：
 - 5星：标注成好评评论(1)
 - 1星：标注成差评评论(-1)
 - 2-4星：弃用，确保训练数据有明确的极端情感倾向
- 原理：在分析用户评论的时候，我们可以看到5星好评里偶尔会有极端差评，1星差评里偶尔会有极端好评，但是基于贝叶斯，这种小概率噪声事件并不会对全局统计造成很大的影响，因此，基于用户星级反馈的训练数据是合理的。

贝叶斯定理

✎ 计算某个评论情感倾向的贝叶斯定理：

$$P(c | \vec{x}) = \frac{P(c)P(\vec{x} | c)}{P(\vec{x})}$$

✎ 其中，c是对应的评价情感分类， \vec{x} 是评论对应的BOW(Bag Of Words)向量。

★ $P(c)$ 是类先验概率

- 根据大数定律，当训练集包含充足的独立同分布(IID)样本时， $P(c)$ 可以通过各类样本出现的频率来进行估计。

★ $P(x | c)$ 是类条件概率(似然)

- 由于涉及关于 \vec{x} 所有属性联合概率，不能直接根据样本来估计。

朴素贝叶斯

✎ 朴素贝叶斯采用“属性条件独立性假设”：对已知类别，假设所有属性相互独立，换言之，假设每个属性独立地对分类结果发生影响。

$$P(\vec{x} | c) = \prod_{i=1}^d P(x_i | c)$$

✎ 按照属性条件独立性假设，贝叶斯定理可以写成(朴素贝叶斯)：

$$P(c | \vec{x}) = \frac{P(c)P(\vec{x} | c)}{P(\vec{x})} = \frac{P(c)}{P(\vec{x})} \prod_{i=1}^d P(x_i | c)$$

✎ 使用朴素贝叶斯公式计算出 \vec{x} 下各个类别的概率，选出概率最大的一个类别作为 \vec{x} 的类别：

$$c_{\vec{x}} = \arg \max_c P(c) \prod_{i=1}^d P(x_i | c)$$

概率值的平滑

✧ 若有些属性 x_i 在训练数据的某个类 c 中未曾出现，则

$$P(\vec{x} | c) = \prod_{i=1}^d P(x_i | c) = 0$$

从而将对应的类条件概率“抹去”，为了避免此类现象，常用拉普拉斯修正(Laplacian correction)来平滑：

★ 类先验概率统计修正为：

● 其中， N 为训练集 D 中的总类别数

$$\hat{P}(c) = \frac{|D_c| + 1}{|D| + N}$$

★ 类条件概率统计修正为：

● 其中， N_i 为第 i 个属性可能的取值数

$$\hat{P}(x_i | c) = \frac{|D_{c,x_i}| + 1}{|D_c| + N_i}$$

✧ 拉普拉斯修正避免了因训练集样本不充分而导致概率估值为零的问题，在训练集变大时，修正过程导致的偏差也会逐渐变得可忽略不计。

算法过程

- 将所有评论进行分词，构建BOW向量映射模型
- 统计各个类(“好评”，“差评”)的频次，并存于数据库之中： D_c
- 统计各个类下各个词的频次，并存于数据库之中： D_{c, x_i}
- 对需要预测的评论进行分词，分解成BOW向量
- 基于评论的词调用数据库中现有的频次数据，使用朴素贝叶斯公式预测类

$$\begin{aligned} c_{\vec{x}} &= \arg \max_c \hat{P}(c) \prod_{i=1}^d \hat{P}(x_i | c) \\ &= \arg \max_c \frac{|D_c| + 1}{|D| + N} \prod_{i=1}^d \frac{|D_{c, x_i}| + 1}{|D_c| + N_i} \end{aligned}$$

★ 若对应的 D_c 不存在，或者 D_{c, x_i} 不存在，则 $D_c=0$ 或 $D_{c, x_i}=0$

情感判断与打分

- ✎ 基于朴素贝叶斯进行分类预测，最后的结果就是“褒义”或者“贬义”，到此为止，朴素贝叶斯算法基本上能够胜任“褒义”或者“贬义”的判断
- ✎ 但是我们需要进一步分析句子的情感等级，并对“褒义”或者“贬义”程度用一个合理数值进行打分。

评论情感打分的问题与定义

🌀 对情感进行分类容易，但是要对情感强烈度进行客观数值的映射是一个很棘手的工作，原因是情感强烈程度是一个主观感受，不同人的感受并不一样

- ★ 在客服部门做过一次调研，给定200个抽样评论，要求同一个人独立完成所有评论的情感判断，返回的问卷中虽然只有2份有效答卷，但是结果差异很大，我自己的主观判断也与问卷的结果大相径庭。显示人与人之间对情感的判断差异性很大。

🌀 根据我们的实际需要，我们需要定义一个情感烈度的合理指标

- ★ 业务目标：体现出用户的情感强烈度，并及时筛选出用户负面情绪很重的评论，交给客服进行善后处理，以后后续对评论的自动情感分类
- ★ 情感强烈度：可以根据用户的句子长度，平均每个词的情感倾向综合考虑

评论情感分定义

考虑以下几种评论：

- ★ a. 句子很长，且句子的每个词都带有明显的负面倾向
- ★ b. 句子较短，且句子的每个词都带有明显的负面倾向
- ★ c. 句子很长，且句子的用词情感倾向有褒义词也有贬义词
- ★ d. 句子较短，且句子的每个词都带有明显的褒义倾向
- ★ e. 句子很长，且句子的每个词都带有明显的褒义倾向

我们需要找的强负面评论往往是a类型，原因是用户花了大代价(时间/精力)写了很多负面倾向的词语，说明用户的怨气很大，需要进行及时安抚。

用户的情感强烈度可以从措辞与长度综合考虑：

- ★ 评论越长，用户的代价越大(需要更多时间与精力)，反映用户对自己情感的发泄的迫切感，是情感强烈度的一个直观指标
- ★ 评论的措辞，如果是极端好评，组成句子的词语大部分是褒义，如果是极端差评，组成句子的词语大部分是贬义。

基于朴素贝叶斯的情感打分

✎ 分析朴素贝叶斯的类概率公式：

$$P(c_x) = \hat{P}(c) \prod_{i=1}^d \hat{P}(x_i | c)$$

- ★ 一个句子越长，连乘的词概率越多，最后的类概率越小，甚至能到 10^{-100} ，所以句子与句子之间类概率的横向对比没有任何意义，所以单概率无法进行情感打分度量。
- ★ 为了能让句子的情感概率能反应出句子与句子间情感烈度的横向对比指标，可以使用概率除法：

$$sentimentRate = \frac{P(c1_x)}{P(c2_x)} = \frac{\hat{P}(c1) \prod_{i=1}^d \hat{P}(x_i | c1)}{\hat{P}(c2) \prod_{i=1}^d \hat{P}(x_i | c2)}$$

概率除法的意义

✎ 概率除法直接意义在于：衡量了一个句子积极情感相对消极情感的对比强度：

$$sentimentRate = \frac{P(c(P)_{\vec{x}})}{P(c(N)_{\vec{x}})} = \frac{\hat{P}(c(P)) \prod_{i=1}^d \hat{P}(x_i | c(P))}{\hat{P}(c(N)) \prod_{i=1}^d \hat{P}(x_i | c(N))}$$

- ★ 在句子与句子横向对比中，若a句子的积极情感相对消极情感的对比比b句子大，可以认为a句子的积极情感程度比b句子越强烈。

情感概率除法的意义

🌀 分解情感概率除法：

$$\begin{aligned} \text{sentimentRate} &= \frac{P(c(P)_{\bar{x}})}{P(c(N)_{\bar{x}})} = \frac{\hat{P}(c(P)) \prod_{i=1}^d \hat{P}(x_i | c(P))}{\hat{P}(c(N)) \prod_{i=1}^d \hat{P}(x_i | c(N))} \\ &= \frac{\hat{P}(c(P))}{\hat{P}(c(N))} * \frac{\hat{P}(x_0 | c(P))}{\hat{P}(x_0 | c(N))} * \frac{\hat{P}(x_1 | c(P))}{\hat{P}(x_1 | c(N))} * \dots * \frac{\hat{P}(x_n | c(P))}{\hat{P}(x_n | c(N))} \\ &= \text{classSenRate} * \text{word0SenRate} * \text{word1SenRate} * \dots * \text{wordNSenRate} \end{aligned}$$

🌀 所以，情感概率除法的实际意义是：

- ★ 组成句子的每个词好评与差评概率的比值的积
- ★ 若将每个词的好评与差评概率的比值作为词的情感倾向值，则情感概率除法是结合每个词的情感倾向值的乘积。

情感概率除法与实际需求的切合

符合我们情感打分假设：

- ★ 句子越长，积极情感倾向词越多，最后的分数越高；
- ★ 句子越长，消极情感倾向词越多，最后的分数越小；
- ★ 积极消极倾向词都有，相互会产生抵消效应，导致结果倾向于中性。
- ★ 按照朴素贝叶斯分类标准：
 - sentimentRate > 1 即 $P(\text{积极}) > P(\text{消极})$ 为积极评论
 - sentimentRate < 1 为 $P(\text{积极}) < P(\text{消极})$ 为消极评论

情感概率除法分数的log处理

🌀 sentimentRate的最终分数，若句子较长，往往数值在10的指数函数描述上，例如， 3.2×10^{37} ，或者 $1.67 \times 10^{(-47)}$ ，这个数值并不符合人的直观感受，需要进行进一步的处理。

- ★ log函数，是一个单调递增函数，适合对10的指数函数进行处理，并且不会造成排序的差异，因此，对sentimentRate结果取对数
- ★ 并且，log函数能将“乘法”变成“加法”，“除法”变成“减法”

$$\log(a * b) = \log(a) + \log(b)$$

$$\log(a / b) = \log(a) - \log(b)$$

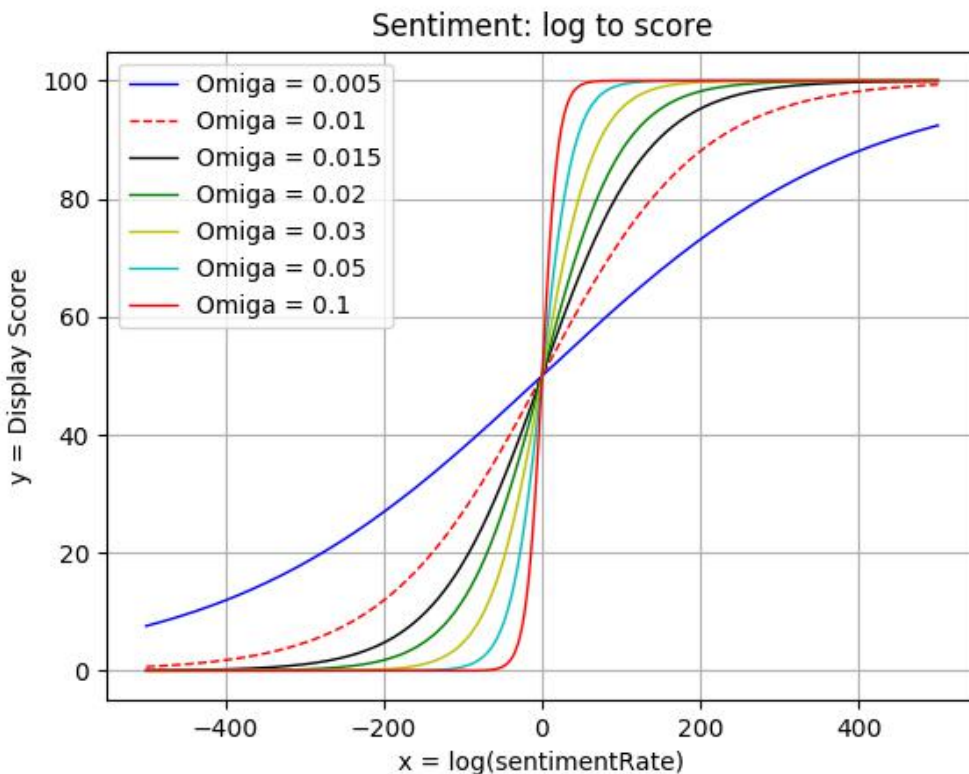
- ★ 从而，原式可以变为

$$\begin{aligned} \log(\text{sentimentRate}) &= \log\left(\frac{\hat{P}(c(P))}{\hat{P}(c(N))} * \frac{\hat{P}(x_0 | c(P))}{\hat{P}(x_0 | c(N))} * \frac{\hat{P}(x_1 | c(P))}{\hat{P}(x_1 | c(N))} * \dots * \frac{\hat{P}(x_n | c(P))}{\hat{P}(x_n | c(N))}\right) \\ &= (\log(\hat{P}(c(P))) - \log(\hat{P}(c(N)))) \\ &\quad + (\log(\hat{P}(x_0 | c(P))) - \log(\hat{P}(x_0 | c(N)))) + \dots \\ &\quad + (\log(\hat{P}(x_n | c(P))) - \log(\hat{P}(x_n | c(N)))) \end{aligned}$$

使用sigmoid函数映射到0-100百分制阌

✎ 由于sigmoid函数可以将负无穷到正无穷映射到0~1之间，所以，我们可以通过对sigmoid函数进行变换，来让我们的数值达到一个合理的百分区间。

$$SentimentScore = 100 * sigmoid(x) = 100 * \frac{1}{1 + e^{-\omega * x}}$$



参数确定

根据对数据的结果进行调试，最后确定的参数是：

- ★ log : 取以5为底的log
- ★ sigmoid: omiga = 0.1

$$\begin{aligned} \textit{SentimentScore} &= 100 * \textit{sigmoid}(\log_5(\textit{sentimentRate})) \\ &= 100 * \frac{1}{1 + e^{-0.1 * \log_5(\textit{sentimentRate})}} \end{aligned}$$

THE END

THANK YOU!