

多分类器融合的情感分析方法研究

王超名

北京交通大学 软件学院

摘要: 近年来,随着 O2O 的迅速发展,外卖领域产生了大量的文本评论。在大数据时代,这些包含用户主观情感倾向的文本蕴含着大量有用的信息,人工分析长篇文本的情感倾向效率极低,所以,利用计算机对文本进行情感分析具有巨大的应用价值。本文针对外卖领域文本评论,分析了基于词典、k-近邻、朴素贝叶斯、最大熵模型、支持向量机的情感极性分析算法。在基于词典的情感分析算法上,本文构建了一个外卖领域的情感词典,创造性地提出短语词典,自动识别更多句式和短语。在基于 k-近邻算法上,本文提出了 Multiple-K 的新型决策机制,有效消除 K 值大小对分类结果的影响。在情感分析算法上,本文采用多分类器融合的投票机制,来决定单个句子的情感极性。实验结果表明,这种方法是有有效,准确率能提升 1 到 2 个百分点。

关键词: 多分类器融合;情感分析;K-近邻;贝叶斯;最大熵;支持向量机;情感词典

0 引言

文本信息可大致分为两类:事实和观点。事实是关于实体、事件及实体与事件的属性的客观表达描述。情感是对事物的积极或消极的情绪表达。随着互联网的飞速发展,特别是微博、微信、论坛等的兴起,人们越来越多地通过网络发表自己的评论看法。在大数据时代,这些包含用户主观情感倾向的文本蕴含着大量有用的信息,人工分析长篇文本的情感倾向效率极低,所以,利用计算机对文本进行情感分析具有巨大的应用价值。¹

情感分析的研究已比较成熟,在粒度上,情感分析可以有三种划分:词语粒度、句子粒度和篇章粒度。²而每个粒度下,又有其相应的子任务。³

篇章粒度。篇章粒度上的情感分析主要考虑整个篇章的情感极性。Lee et al. 将传统机器学习方法应用于电影评论的情感分析,最后得到从篇章中提取出 unigrams 后 SVM 的分类效果最好。⁴这三种机器学习方法都是有监督的方法。而 Peter D. Turney 提出了一种无监督的方法进行篇章情感极性的自动分类。⁵篇章粒度有几个子任务,都有相应的研究尝试解决。比如,主客观分类⁶、review helpfulness⁷、垃圾观点侦测⁸。

句子粒度。句子粒度的情感分析首先是判断句子是否主观或客观,然后将主观语句进行极性分类。Weibe et al. 提出主观性分析在句子中的有效性及其在自然语言处理的应用。⁹熊德兰等人提出了基于 HowNet 的语义距离和语法距离相结合的句子褒贬倾向性计算方法利用夹角余弦法对语义倾向进行了改进。¹⁰句子粒度的子任务有:观点检索¹¹、观点问答¹²、观点摘要¹³。

词语粒度。词语粒度的情感分析主要是提取出评价对象,评价观点,实体,属性,

特征及其情感倾向。¹⁴¹⁵¹⁶

本文尝试对外卖领域的文本，进行句子粒度的情感分析。构建了适应外卖领域的情感词典，并提出 K-NN、Naïve Bayes、MaxEnt、SVM 的多分类器融合和规则辅助的情感分析方法。

1 预处理

1.1 构建爬虫

传统爬虫向服务器发送请求后，服务器返回相应的静态网页信息。而如今大多数网站，很多信息都利用 js 进行动态加载，传统爬虫根本无法获取到。所以，本文利用 Python 上 selenium 和 BeautifulSoup，构建了一个终极爬虫，能自动打开浏览器，模拟实际的网页点击事件和浏览，与 js 网页动态交互，能成功解析 js 动态网页，爬取外卖网站上的文本评论。

1.2 特征选择

文本处理中，特征选择是一个非常重要的步骤，其目的在于从原始特征信息中，挑选出最具有代表性的、分类性能优异的特征进行分类，故合适的特征选择方法将很大程度上决定最终分类效果的好坏。通常情况下，文本分类领域都是选取词语作为特征，通过计算词语与类别之间的关系，度量各个词语对类别的贡献度大小，从而归属于某个类别。由于在文本数据处理中，往往词语的数量非常多，若把所有的词语都选为特征项，则特征空间的维度过大，不仅会增加运算量，还会影响分类的精度。因此须对文本内容作降维处理，合适的特征选择方法就尤为重要。情感分类中的特征选择，一方面需要去除与情感无关、类别关联度较小的特征，排除不必要干扰。另一方面，特征选择方法要能获取与情感分类有关联的特征信息，才能提高情感倾向性判别的准确性。¹⁷因此，必须针对外卖评论选择合适的特征抽取方法，才能提高情感识别的分类效果。本文采用 X^2 统计方法：

$$X^2(t, C) = \frac{N(AD - BC)^2}{(A + C)(B + D)(A + B)(C + D)}$$

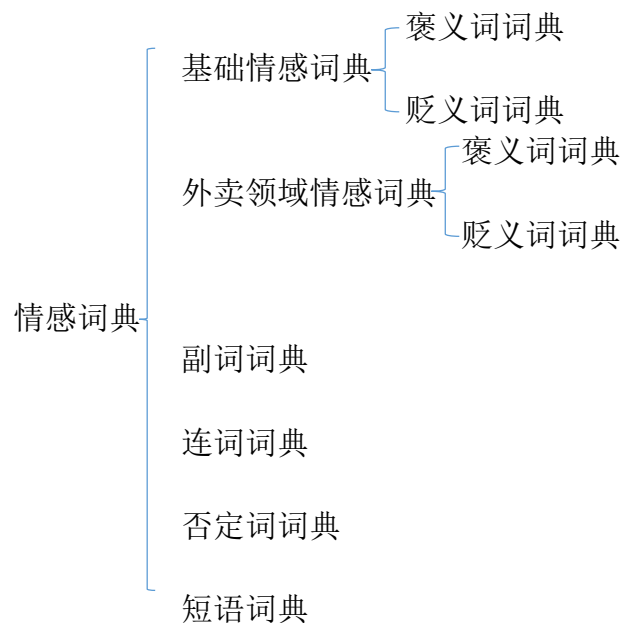
X^2 统计方法度量特征词语 t 和句子类别 c 之间的相关程度，并假设 t 和 c 之间符合具有一阶自由度的 X^2 分布。如果 A 表示包含词条 t 且属于类别 c 的句子频数， B 为包含 t 但是不属于类别 c 的句子频数， C 表示属于类别 c 但是不包含 t 的句子频数， N 表示语料中句子总数， D 表示既不属于 c 也不包含 t 的句子频数。¹⁸

实际上，如果类别是一个二分类别，则 $X^2(t, c1) = X^2(t, c2)$ 。

2 基于词典的情感分析

2.1 情感词典的构建

文本的情感倾向大多通过情感词语来体现，情感词典能否覆盖全面在一定程度上影响着情感分类效果，故情感词典的构建是情感分类研究的基础。文本情感分析研究领域还没有一部完整且通用的情感词典。若构建一个面向外卖领域的情感词典，一方面须对当前的已有相关资源进行总结与整理，另一方面需要构建一个基于外卖的领域情感词典。本文选择中国科学院董振东教授构建的知网(HowNet)为基础情感词典，在此基础上，加上自己的副词词典、连词词典、否定词词典，完成情感词典的构建。另外，本文创造性地提出短语词典，自动识别更多句式 and 短语。



```
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
只送……菜 -1
只有……菜 -1

(把)?……撒……了 -2
(把)?……洒……了 -2

味……不对 -2

没有……特点 -1
看不见……肉 -1
没……味道 -1 start:1 end:6
没……味 -1 start:1 end:6
没……肉……多 -1
没……肉 -1 between_tag:m

比……少 -1
没……多 -1
没……好吃 -1
没……好 -1

吃出…… -2 head:[^没]?
扔了…… -1
等……很久 -2

(和|跟)……不(同|一样) -1
```

2.2 情感分析算法

2.2.1 程序的调用层次

1. sentiment_analysis_from_corpus_file(path, encoding)
2. sentiment_analyse_a_sentence(sentence, runout_filepath=None)
3. divide_sentence_into_clauses(the_sentence)
4. sentiment_analyse_a_clause(the_clause, seg_result=None)
5. clause_is_pattern_2(the_clause)
6. clause_is_pattern_3(the_clause, seg_result)
7. emotional_word_analyse(core_word, value, segments, index)
8. sentiment_analyse_a_word(the_word, seg_result=None, index=-1)
9. word_is_conjunction(the_word)
10. word_is_punctuation(the_word)
11. word_is_positive(the_word, seg_result, index)
12. emotional_word_analyse(core_word, value, segments, index)
13. word_is_negative(the_word, seg_result, index)
14. emotional_word_analyse(core_word, value, segments, index)
15. clause_is_pattern_1(the_clause)



2.2.2 程序的算法分析

1、首先将语句分成几个子句，分句为后续的情感分析做铺垫，若分句错误，后续的情感分析将无从谈起。另外，分句要能识别出各种句式，诸如“如果……就……”。接下来对每一个分句进行分析，得到每个分句的情感分值，整句的情感分值为每个分句情感分值之和。

2、判断分句是否是句式二，即“如果……就”，“要是……就好了”。若是，则不管句式内有多少正向词语，直接判定为负向情感。比如，“要是肥牛再多点就好了”。

3、判断分句是否是句式三，句式三是各种短语，写进文件。每次读取文件，解析成正则表达式，进行匹配。比如，文件内为：“提高了……质量 2”会被解析成“提高了[\u4e00-\u9fa5]* 质量”，为正向短语，分值为 2；而“希望……提高……质量 -1”会被解析成“希望[\u4e00-\u9fa5]*提高[\u4e00-\u9fa5]*质量”，为负向短语，分值为-1。另外，还有一些特殊短语的解析，如：“没……味道 -1 start:1 end:6”会解析成“没[\u4e00-\u9fa5]{1, 6}味道”，而“没……肉 -1 between_tag:m”会解析成“没[\u4e00-\u9fa5]*肉”，并且中间必须有一个数词（词性 tag 为 m），匹配“没有几块肉”之类的句子。

4、将分句分词，分词利用 jieba 分词。然后逐个分析分词。首先判断是否是连词，是，则记录；判断是否是标点符号，是，则记录。判断是否是正向情感词，是则深度分析前三个视窗内的词。若有奇数个否定词，则正向情感分值变负；偶数个情感词，情感分值不变。若有副词，判断副词强度，“很”之类的词情感分值加倍，“不太”这类的副词，情感分值减半。

5、最后，加上之前记录的连词得倍数，标点符号的倍数，综合得到整个分句的情感分值。

2.3 实验结果

本文对正负各 1000 条外卖评论进行测试，实验结果如下：

pos-precision	pos-recall	pos-f1	neg-precision	neg-recall	neg-f1	total-precision
79.82	80.7	80.26	80.49	79.6	80.04	80.15

3 基于 K-近邻的情感分析

3.1 k-近邻算法

k-近邻算法简单、直观。给定一个训练数据集，对新的输入实例，在训练数据集中找到与该实例最近邻的 k 个案例，这 k 个案例的多数属于某个类，就把该输入实例分为这个类。¹⁹

其算法步骤为：

- step.1---初始化距离为最大值
- step.2---计算未知样本和每个训练样本的距离 dist
- step.3---得到目前 K 个最临近样本中的最大距离 maxdist
- step.4---如果 dist 小于 maxdist，则将该训练样本作为 K-最近邻样本
- step.5---重复步骤 2、3、4，直到未知样本和所有训练样本的距离都算完
- step.6---统计 K-最近邻样本中每个类标号出现的次数
- step.7---选择出现频率最大的类标号作为未知样本的类标号

3.2 Multiple K

k 值的选择、距离的度量、分类决策规则是 k-近邻算法的三个基本要素。¹⁹

k 值的选择较小的优点是：“学习”的近似误差（approximation error）会减小，只有与输入值较近的（相似的）训练实例才会对预测结果起作用。缺点是：“学习”的估计误差（estimation error）会增大，预测结果会对近邻的实例点非常敏感。如果邻近的实例点恰巧是噪声，预测会出错。

k 值的选择较大的优点：“学习”的估计误差（estimation error）会减小，“学习”的近似误差（approximation error）会增大。这时与输入实例较远的（不相似的）训练实例也会对预测起作用。使预测发生错误。

既然 k 大 k 小都有问题，于是本文尝试在每一个训练数据内，取多个 k 值，计算出分类结果，再在分类结果里面看，哪种分类数目更多，然后取其值。k 值的数量应该为奇数。这样可以消除 k 很大的影响和 k 很小的影响。本文称这种算法为“Multiple K”。

3.3 实验结果

我们使用正负 3500 条训练评论，正负 1000 条作为测试数据。实验结果如下：

k	Featur	pos-	pos-	pos-f1	neg-	neg-	neg-	total-
---	--------	------	------	--------	------	------	------	--------

	e	precision	recall		precision	recall	f1	precision
1,3,5,7,9,11,13	4000	83.91	77.42	89.54	79.04	85.15	81.99	81.29
1	4000	75.29	75.02	75.16	75.11	75.38	75.24	75.2
3	4000	79.39	76.35	77.84	77.23	80.18	78.67	78.27
5	4000	80.79	75.91	78.27	77.28	81.96	79.55	78.93
7	4000	82.56	77.87	80.915	79.06	83.56	81.24	80.71
9	4000	83.04	76.62	79.7	78.30	84.36	81.22	80.49
11	4000	83.65	76.44	79.89	78.31	85.07	81.55	80.76
13	4000	83.43	76.53	79.83	78.33	84.8	81.43	80.67
21,25,29,33,37,41,45	4000	77.49	85.6	80.79	83.65	73.69	78.35	80.64
21	4000	77.17	84.44	80.65	82.83	75.02	78.73	79.73
25	4000	77.24	85.07	80.96	83.38	74.93	78.93	80
29	4000	77.39	84.89	80.97	83.27	75.2	79.03	80.04
33	4000	75.77	87.02	81.01	84.76	72.18	77.96	79.6
37	4000	74.9	87.29	80.62	84.77	70.76	77.13	79.02
41	4000	75.51	85.78	80.32	83.54	72.18	77.44	78.98
45	4000	75.53	85.07	80.02	82.91	72.44	77.32	78.76

4 基于 Naïve Bayes 的情感分析

4.1 Naïve Bayes

假设句子 X 包含词汇 (a_1, a_2, \dots, a_m) 。 Y 为两类情感极性的集合 (y_1, y_2) 。则 X 的情感极性为 $\arg\max(P(y_i|X)) = \arg\max(P(y_i|x_1, x_2, \dots, x_m))$ 。

由贝叶斯公式

$$P(y_i|x) = \frac{P(x|y_i)P(y_i)}{P(x)}$$

$P(x)$ 是相同的，可直接忽略。

Naïve Bayes 被称为 naïve 是因为假设各个词之间是相互独立的，所以我们可以得到：

$$P(x|y_i)P(y_i) = P(a_1|y_i)P(a_2|y_i)\dots P(a_m|y_i)P(y_i) = P(y_i) \prod_{j=1}^m P(a_j|y_i)$$

Naïve Bayes 的条件独立假设看上去很傻很天真，但结果去很强大。因为有些独立假设在各个分类之间的分布都是均匀的所以对于似然的相对大小不产生影响；即便不是如此，也有很大的可能性各个独立假设所产生的消极影响或积极影响互相抵消，最终导致结果受到的影响不大。²⁰

4.2 实验结果

我们使用正负 3500 条训练评论，正负 1000 条作为测试数据。实验结果如下：

feature	pos-precision	pos-recall	pos-f1	neg-precision	neg-recall	neg-f1	total-precision
all	86.77	66.18	75.09	72.67	89.91	80.37	78.05
4000	91.10	69.6	78.91	75.40	93.2	83.36	81.4

5 基于最大熵的情感分析

5.1 最大熵

1957 年 Jaynest²¹基于信息熵理论建立了最大熵模型。最大熵模型是根据样本信息对某个未知分布做出推断的一种方法。最大熵在自然语言处理等各领域应用非常广泛。²²最大熵原理指出，当我们需要对一个随机事件的概率分布进行预测时，我们的预测应当满足全部已知的条件，而对未知的情况不要做任何主观假设。在这种情况下，概率分布最均匀，预测的风险最小。因为这时概率分布的信息熵最大，所以人们称这种模型叫"最大熵模型"。

5.1.1 定义

我们假设 \mathcal{P} 为最大熵模型的学习后的所有条件概率， $P(y|x)$ 为 \mathcal{P} 中的一个元素。

训练集内包含大量的样本， $(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)$ 。我们定义 $\tilde{p}(x, y)$

为样本 (x, y) 出现的概率。从训练集中我们可以得到很多有用的特征，我们定义一个特征函数：

$$f(x, y) = \begin{cases} 1 & \text{满足特征条件} \\ 0 & \text{不满足特征条件} \end{cases}$$

特征函数在满足成为一个特征时取值 1，否则取值 0。则特征函数的经验概率为：

$$\tilde{p}(f) \equiv \sum_{x, y} \tilde{p}(x, y) f(x, y)$$

5.1.2 满足所有已知条件

而在最大熵模型中，特征函数的期望概率为：

$$p(f) \equiv \sum_{x, y} \tilde{p}(x) p(y|x) f(x, y)$$

最大熵模型的训练，就是满足这样一个约束条件——模型的期望概率与训练集中得到的经验概率相等：

$$p(f) = \tilde{p}(f)$$

即，

$$\sum_{x, y} \tilde{p}(x) p(y|x) f(x, y) = \sum_{x, y} \tilde{p}(x, y) f(x, y)$$

所有这样的约束组成一个集合 \mathcal{C} 。

5.1.3 对未知条件尽可能均匀

现在，最大熵模型已经满足了所有约束条件了。那对于未知条件，我们要尽可能均匀（uniform）。那怎样才能达到“尽可能均匀”呢？一个衡量均匀分布的数学度量为熵：

$$H(p) \equiv - \sum_{x, y} \tilde{p}(x) p(y|x) \log p(y|x)$$

当所有情况概率都相等时，熵最大。

5.1.4 训练

最大熵模型的训练，就是要找到一个模型 $p^* \in \mathcal{C}$ ，并且具有最大熵 $H(p)$ 。即，

$$\begin{aligned} p^* &= \operatorname{argmax}_{p \in \mathcal{C}} H(p) \\ &= \operatorname{argmax}_{p \in \mathcal{C}} \left(- \sum_{x,y} \tilde{p}(x) p(y|x) \log p(y|x) \right) \end{aligned}$$

利用拉格朗日乘子法，我们得到：

$$\begin{aligned} \xi(p, \Lambda, \gamma) \equiv & - \sum_{x,y} \tilde{p}(x) p(y|x) \log p(y|x) \\ & + \sum_i \lambda_i \left(\sum_{x,y} \tilde{p}(x, y) f_i(x, y) - \tilde{p}(x) p(y|x) f_i(x, y) \right) \\ & + \gamma \sum_x p(y|x) - 1 \end{aligned}$$

通用的训练算法有 GIS²³和 IIS²⁴算法。

5.2 实验结果

我们使用正负 3500 条训练评论，正负 1000 条作为测试数据。实验结果如下：

feature	pos-precision	pos-recall	pos-f1	neg-precision	neg-recall	neg-f1	total-precision
all	88.68	93.3	90.93	80.24	69.55	78.51	86.63
4000	89.7	90.55	91.59	81.47	72.55	76.75	87.64

6 基于 SVM 的情感分析

6.1 支持向量机

支持向量机的基本思想是：求解能够正确划分训练数据集并且几何间隔最大的分离超平面。对于线性可分的训练数据集，线性可分离超平面有无穷多个（等价于感知机），但是几何间隔最大的分离超平面只有一个。间隔最大的直观解释是：对训练集找到几何间隔最大的超平面意味着以充分大的确信度对训练数据集进行分类。¹⁹

6.1.1 划分数据集

一个线性分类器的学习目标就是要在 n 维的数据空间中找到一个分类超平面，其方程可以表示为：

$$w^T x + b = 0$$

于是，我们得到一个分类函数：

$$f(x) = w^T x + b$$

为了便于计算距离，我们将分类标签设为 1 和 -1。

6.1.2 最大化间隔

这个分类超平面要满足这样的条件：找到离分割超平面最近的点，确保它们离分割超平面的距离尽可能远。²⁵

令 γ 为离分割超平面最近的点 x 到分类超平面的距离：

$$\gamma = \frac{w^T x + b}{\|w\|} = \frac{f(x)}{\|w\|}$$

不过这里的 γ 是带符号的，我们需要的只是它的绝对值，因此类似地，也乘上对应的类别 y 即可：

$$\tilde{\gamma} = y\gamma = \frac{\hat{\gamma}}{\|w\|}$$

其中， $\hat{\gamma} = y * (w^T x + b) = y * f(x)$ 。我们的目标便是： $\max \tilde{\gamma}$ 。方便推导和

优化的目的，我们可以令 $\hat{\gamma} = 1$ 。上述的目标函数 $\tilde{\gamma}$ 转化为：

$$\max \frac{1}{\|w\|}, \quad s.t., y_i(w^T x_i + b) \geq 1, i = 1, \dots, n$$

这个问题等价于：

$$\min \frac{1}{2} \|w\|^2 \quad s.t., y_i(w^T x_i + b) \geq 1, i = 1, \dots, n$$

6.1.3 求解

支持向量机的求解过程比较复杂，应该说书迄今为止机器学习领域中最复杂的推导过程之一。²⁶

我们引入拉格朗日乘子法，得到：

$$\mathcal{L}(w, b, \alpha) = \frac{1}{2} \|w\|^2 - \sum_{i=1}^n \alpha_i (y_i (w^T x_i + b) - 1)$$

然后令

$$\theta(w) = \max_{\alpha_i \geq 0} \mathcal{L}(w, b, \alpha)$$

根据 KKT 条件，上述最优化问题必须满足：

$$1. \quad h_j(\mathbf{x}_*) = 0, j = 1, \dots, p, \quad g_k(\mathbf{x}_*) \leq 0, k = 1, \dots, q,$$

$$2. \quad \nabla f(\mathbf{x}_*) + \sum_{j=1}^p \lambda_j \nabla h_j(\mathbf{x}_*) + \sum_{k=1}^q \mu_k \nabla g_k(\mathbf{x}_*) = \mathbf{0},$$

$$\lambda_j \neq 0, \quad \mu_k \geq 0, \quad \mu_k g_k(\mathbf{x}_*) = 0.$$

要求解这个最优化问题，首先固定 α ，要让 \mathcal{L} 关于 w, b 最小化，我们分别对 w, b 求导：

$$\frac{\partial \mathcal{L}}{\partial w} = 0 \Rightarrow w = \sum_{i=1}^n \alpha_i y_i x_i$$

$$\frac{\partial \mathcal{L}}{\partial b} = 0 \Rightarrow \sum_{i=1}^n \alpha_i y_i = 0$$

将上述结果带回 \mathcal{L} ：

$$\mathcal{L}(w, b, \alpha) = \frac{1}{2} \|w\|^2 - \sum_{i=1}^n \alpha_i (y_i (w^T x_i + b) - 1)$$

得到：

$$\begin{aligned}\mathcal{L}(w, b, \alpha) &= \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j x_i^T x_j - \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j x_i^T x_j - b \sum_{i=1}^n \alpha_i y_i + \sum_{i=1}^n \alpha_i \\ &= \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j x_i^T x_j\end{aligned}$$

此时的 $\mathcal{L}(w, b, \alpha)$ 函数只剩下一个变量 α 。只要求出 α , w, b 便能求解出来。

分类函数 $f(x) = w^T x + b$ 也就能轻而易举求出来了。

6.2 实验结果

我们采用 Python 机器学习包 `scikit-learn` 来进行 SVM 的情感分析。使用正负 3500 条训练评论，正负 1000 条作为测试数据。实验结果如下：

Feature	pos-precision	pos-recall	pos-f1	neg-precision	neg-recall	neg-f1	total-precision
TFIDF	83.93	84.60	84.26	84.48	83.80	84.14	84.20

7 多分类器融合的情感分析

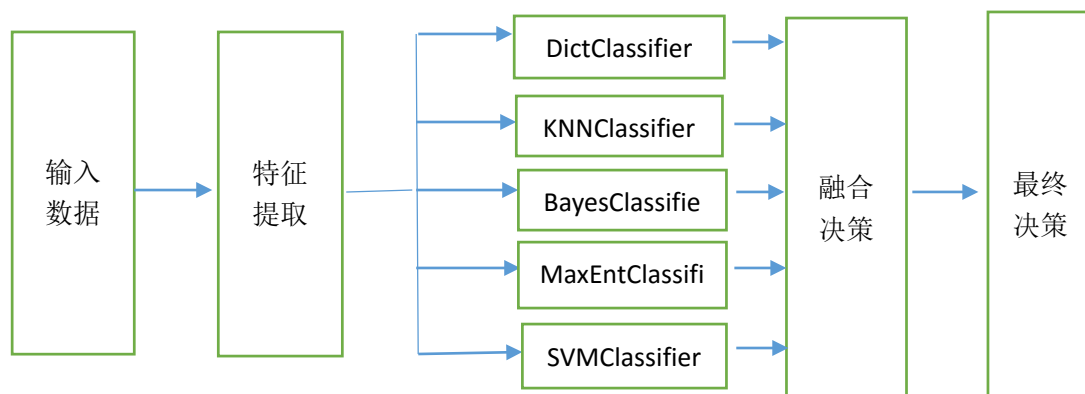
7.1 多分类器比较

从上述测试数据可以看出，部分分类器正向分类精度大于负向分类精度，而部分分类器负向分类精度大于正向分类精度。

分类器	pos-precision	neg-precision
Dict-Based	79.82	80.49
KNN	76.49	83.65
Bayes	91.10	75.4
最大熵	89.7	81.47
SVM	83.93	84.48

7.2 多分类器融合

于是，本文尝试利用多分类器的并联融合，采用投票机制，决定最终分类结果。



对于测试数据，本文定义 i 为其分类标签， C_i 为分类为 i 的分类器个数则，多分类器融合下的情感分析分类为

$$i = \operatorname{argmax}(C_i).$$

7.3 实验结果

我们使用正负 3500 条训练评论，正负 1000 条作为测试数据。实验结果如下：

feature	pos-precision	pos-recall	pos-f1	neg-precision	neg-recall	neg-f1	total-precision
4000	94.65116	81.4	87.52688	83.68421	95.4	89.15888	88.4

8 结论

由于不同分类器对正负语料分类精度的不同，我们利用多分类器的并联融合，采用投票机制，可以使感分析的精度提升 1 到 2 个百分点。

参考文献：

- ¹ 赵妍妍，秦兵，刘挺. 文本情感分析 [J]. 软件学报，2010，21(8)：1834-1848.
- ² Pang, Bo, and Lillian Lee. 2008. "Opinion Mining and Sentiment Analysis." Foundations and Trends® in Information Retrieval 2: 1-135.
- ³ K. Vivekanandan, J. Soonu Aravindan. 2014. Aspect-based Opinion Mining: A Survey.

- ⁴ Pang, Bo, Lillian Lee, and Shivakumar Vaithyanathan. 2002. "Thumbs up? Sentiment Classification Using Machine Learning Techniques." In Conference on Empirical Methods in Natural Language Processing (EMNLP 2002), 79–86.
- ⁵ Turney, Peter D. 2001. "Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews." In Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, ACL '02 417- 424.
- ⁶ Yu, Lei, Jia Ma, Seiji Tsuchiya, and Fuji Ren. 2008. "Opinion Mining: A Study on Semantic Orientation Analysis for Online Document." In Proceedings of the World Congress on Intelligent Control and Automation (WCICA), 4548–52.
- ⁷ Liu, Jingjing et al. 2007. "Low-Quality Product Review Detection in Opinion Summarization." In Computational Linguistics, 334–42.
- ⁸ Lim, Ee-Peng et al. 2010. "Detecting Product Review Spammers Using Rating Behaviors." Proceedings of the 19th ACM International Conference on Information and Knowledge Management: 939–48.
- ⁹ Wiebe, Janyce et al. 2004. "Learning Subjective Language." Computational Linguistics 30: 277–308.
- ¹⁰ 熊德兰, 程菊明, 田胜利. 基于 HowNet 的句子褒贬倾向性研究. 计算机工程与应用, 2008, 44(22): 143—144.
- ¹¹ Huang, Shen et al. 2009. "Improving Product Review Search Experiences on General Search Engines." In Proceedings of the 11th International Conference on Electronic Commerce - ICEC '09, 107.
- ¹² Moghaddam, Samaneh, and Martin Ester. 2011. "AQA: Aspect-Based Opinion Question Answering." In Proceedings - IEEE International Conference on Data Mining, ICDM, 89–96.
- ¹³ Hu, Mingqing, and Bing Liu. 2004. "Mining and Summarizing Customer Reviews." Proceedings of the 2004 ACM SIGKDD international conference on Knowledge discovery and data mining KDD 04 04: 168.
- ¹⁴ Lu, Yue, ChengXiang Zhai, and Neel Sundaresan. 2009. "Rated Aspect Summarization of Short Comments." Proceedings of the 18th international conference on World wide web - WWW '09: 131.
- ¹⁵ Wong, Tak-Lam, Wai Lam, and Tik-Shun Wong. 2008. "An Unsupervised Framework for Extracting and Normalizing Product Attributes from Multiple Web Sites." Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval SIGIR 08: 35.
- ¹⁶ Liu, Bing, and South Morgan Street. 2005. "Opinion Observer : Analyzing and Comparing Opinions on the Web." In Proceedings of the 14th International Conference on World Wide Web, 342–51.
- ¹⁷ 硕士论文《基于情感词典的中文微博情感倾向分析研究》陈晓东. 2012 年 1 月 12 日
- ¹⁸ MADEIRA H. COSTA J, VIEIRA M. The OLAP and data warehousing approaches for analysis and sharing of results from dependability evaluation experiments[C] / / International Conference on Dependable Systems and Networks. [S. 1.]: IEEE Press, 2003: 86—91.
- ¹⁹ 《统计学习方法》,李航 著, 清华大学出版社,2012 年 3 月出版
- ²⁰ Harry Zhang. The Optimality of Naive Bayes. 2004, American Association for Artificial Intelligence
- ²¹ T. Jaynes. Information Theory and Statistical Mechanics. Physics Reviews. 1 957(1 06): 620. 630
- ²² Adwait Ratnaparkhi, A Simple Introduction to Maximum Entropy Models for Natural Language Processing. IRCS Report 97—08
- ²³ Darroch, J. N. and Ratcliff, D. (1972).Generalized Iterative Scaling for Log-linear Models. Annals of Mathematical Statistics, no. 43, 1470-1480
- ²⁴ A Berger. The improved iterative scaling algorithm: A gentle introduction. 《Unpublished Manuscript》, 1998

²⁵ Peter Harrington. Machine Learning In Action.2013. Manning Publications

²⁶ 《机器学习——算法原理与编程实践》，郑捷 著，中国工信出版社，电子工业出版社，2015.11