基于 word2vec 方法的情感分析研究及应用

2017 年 12 月 24 日

目录

1 研究背景

随着信息时代的蓬勃发展，互联网逐渐倡导“以用户为中心，用户参与”的开放式构架理念。互联网用户由单纯的“读”网页，开始向“写”网页、“共同建设”互联网发展，并由被动地接收互联网信息向主动创造互联网信息迈进。因此，互联网（如博客和论坛）上产生了大量的用户参与的、对于诸如人物、事件、产品等有价值的评论信息。这些评论信息表达了人们的各种情感色彩和情感倾向性，如喜、怒、哀、乐和批评、赞扬等。基于此，潜在的用户就可以通过浏览这些主观色彩的评论来了解大众舆论对于某一事件或产品的看法。由于越来越多的用户乐于在互联网上分享自己的观点或体验，这类评论信息迅速膨胀，互联网上的数据急剧膨胀。根据国际数据公司（IDC）的统计和预测，2011 年全球网络数据量已经达到 1.8ZB，到 2020 年，全球数据总量预计还将增长 50 倍。仅靠人工的方法已难以应对网上海量信息的收集和处理，因此迫切需要计算机帮助用户快速获取和整理这些相关评价信息。情感分析（sentiment analysis）技术应运而生（本文中提及的情感分析，都是指文本情感分析）。

文本情感分析又称意见挖掘。简单而言，是对带有情感色彩的主观性文本进行分析、处理、归纳和推理的过程。最初的情感分析源自前人对带有情感色彩的词语的分析，如“美好”是带有褒义色彩的词语，而“丑陋”是带有贬义色彩的词语。随着互联网上大量的带有情感色彩的主观性文本的出现，研究者们逐渐从简单的情感词语的分析研究过渡到更为复杂的情感句研究以及情感篇章的研究。

近年来，深度学习在图像和语音信号中取得了巨大的成功。语言是人类认知过程中产生的较高级抽象信息，将深度学习应用到自然语言处理中的研究进展较为缓慢。目前取得最基础也是最有趣的一个成果就是词向量，它是利用基于神经网络模型学习得到词的分布式语义表达向量。这种低维的、连续的数值型向量可以很方便的应用到其他的 NLP 任务中，例如词性解析、命名实体识别、语言模型等，并取得了一定的进展。

随着技术的发展，情感分析的应用越来越广泛。目前，国内外有很多研究机构根据现实生活中的具体需求研发出各个领域的情感分析系统，帮助用户对海量信息进行分析和决策。例如，Liu 等人研发的 OpinionObserver 系统可以处理网上在线顾客产品评价，采用可视化方式对若干种产品评价对象的综合质量进行比较。

Wilson 等人研发的 OpinionFinder 系统可以自动识别主观性句子以及抽取句子中情感信息。上海交通大学则开发了一个用于汉语汽车论坛的情感分析系统，挖掘并概括人们对各种汽车品牌的评论和意见。除了用户评论分析与决策、舆情监控以及信息预测领域以外，情感分析在其他一些自然语言处理领域也扮演着重要的角色。例如，在信息抽取领域，抽取对象一般是反映客观事实的文本，情感分析技术可用于将文本中的主观句和客观句进行分离，提高信息抽取的准确率。情感分析技术还可以用于问答系统中，当用户所问问题是情感相关的问题时，该技术可以帮助问答系统提供更真实的答案。此外，情感分析技术还可以用于情感文摘的生成，进而达到汇总归纳的目的。情感分析技术的快速发展在很大程度上源于人们改进人机交互现状的愿望。该技术在以上众多研究领域的应用使其成为一个非常重要的研究方向。

2 文献综述

自 2003 年 Nasukawa 提出情感分析概念来，大量研究者对情感分析展开了深入而广泛的研究。Liu 在 2012 年系统介绍了情感分析的各个方面，按照不同的归类方式将情感分析任务划分成不同的层次。按照待处理文本的类型，可将情感分析任务分为词或短语级别（word- or phrase-level）、句子级别（sentence-level）和文挡级别（document-level）的情感分析。按照情感分析任务的输出结果，可将其划分为情感极性分类（sentiment polarity classification）、情感强度预测（sentiment strength prediction）等。按照研究方法，可将其分为有监督的学习（supervised learning）和无监督的学习（unsupervised learning）等。目前大部分研究都是采用有监督学习算法。近年来，深度学习在自然语言处理领域的研究也逐渐成为研究的热点。

在基于机器学习的情感分析思路中主要分为中文分词、特征提取、特征选择、分类模型以及评价指标等方面。其中对于文本的特征选择和分类模型为其核心部分。

2.1 特征选取

基于神经网络的分布表示一般称为词向量、词嵌入（word embedding）或分布式表示（distributed representation）。神经网络词向量表示技术通过神经网络技术对上下文，以及上下文与目标词之间的关系进行建模。由于神经网络较为灵活，这类方法的最大优势在于可以表示复杂的上下文。在前面基于矩阵的分布表示方法中，最常用的上下文是词。如果使用包含词序信息的 n-gram 作为上下文，当 n 增加时，n-gram 的总数会呈指数级增长，此时会遇到维数灾难问题。而神经网络在表示 n-gram 时，可以通过一些组合方式对 n 个词进行组合，参数个数仅以线性速度增长。有了这一优势，神经网络模型可以对更复杂的上下文进行建模，在词向量中包含更丰富的语义信息。

Xu 等人在 2000 年首次尝试使用神经网络求解二元语言模型。2001 年，Bengio 等人正式提出神经网络语言模型（Neural Network Language Model，NNLM）[1]。该模型在学习语言模型的同时，也得到了词向量。具体而言，对语料中一段长度为 n 的序列$w_i, w_{i-1,} \dots, w_{i-(n-1)}$，$n$元语言模型需要最大化

$$P(w_i|w_{i-1},\dots,w_{i-(n-1)}) \tag{2.1}$$

其中，w_i为通过语言模型预测的词（目标词）。神经网络语言模型采用普通的三层前馈神经网络结构，其中第一层为输入层：

$$\boldsymbol{x}=[\boldsymbol{e}(w_{i-1});\boldsymbol{e}(w_{i-2});\dots;\boldsymbol{e}(w_{i-(n-1)})] \tag{2.2}$$

当输入层完成对上文的表示$\boldsymbol{x}$之后，模型将其送入剩下两层神经网络，依次得到隐藏层 h 和输出层 y：

$$\boldsymbol{h}=\tanh(\boldsymbol{b}^{(1)}+\boldsymbol{H}\boldsymbol{x}) \tag{2.3}$$

$$\mathbf{y}=\boldsymbol{b}^{(2)}+\boldsymbol{W}\boldsymbol{x}+\boldsymbol{U}\boldsymbol{h} \tag{2.4}$$

其中$\boldsymbol{H}$为输入层到隐藏层的权重矩阵，$\boldsymbol{U}$ 为隐藏层到输出层的权重矩阵，$|\boldsymbol{V}|$表示词表的大小，$|\boldsymbol{e}|$表示词向量的维度，$|\boldsymbol{h}|$为隐藏层的维度。$\boldsymbol{b}^{(1)}$、$\boldsymbol{b}^{(2)}$均为模型中的偏置项。矩阵$|\boldsymbol{W}|$表示从输入层到输出层的直连边权重矩阵。2007 年，Mnih 和 Hinton 在神经网络语言模型（NNLM）的基础上提出了 log 双线性语言模型（Log-Bilinear Language Model，LBL）[2]。之后的几年中，Mnih 等人在 LBL 模型的基础上做了一系列改进工作。其中最重要的模型有两个：层级 log 双线性语言模型（Hierarchical LBL，HLBL）和基于向量的逆语言模型（inverse vector LBL，ivLBL）。Mikolov 等人提出的循环神经网络语言模型（Recurrent Neural Network based Language Model，RNNLM）则直接对文本进行建模[3]。因此，RNNLM 可以利用所有的上文信息，预测下一个词。与前面的三个基于语言模型的词向量生成方法不同，Collobert 和 Weston 在 2008 年提出的 C&W 模型[4]，是第一个直接以生成词向量为目标的模型。之后 Mikolov 等人在 2013 年的文献[5]中，同时提出了 CBOW（Continuous Bag of Words）和 Skip-gram 模型。他们设计这两个模型的主要目的是希望用更高效的方法获取词向量。因此，他们根据前人在 NNLM、RNNLM 和 C&W 模型上的经验，简化现有模型，保留核心部分，得到了这两个模型。CBOW 模型和 Skip-gram 模型为了有更高的性能，在神经网络语言模型或者 log 双线性语言模型的基础上，同时去掉了隐藏层和词序信息。为了更好地分析词序信息对词向量性能的影响，来斯惟提出一个新模型[6]，名为“Order”，意为保留了词序信息。该模型在保留词序信息的同时去除了隐藏层。

2.2 文献评述

2.2.1 词向量方法总结

上文介绍的各种神经网络词向量模型中，除了 Skip-gram 模型使用词作为上下文表示之外，其它模型均使用 n-gram 作为上下文表示。如 CBOW 模型使用 n-gram 中各词词向量的平均值作为上下文表示；Order 模型使用 n-gram 中各词词向量的拼接作为上下文表示，这种方法可以看做词向量的线性组合；LBL 模型则是直接对 n-gram 中各词的词向量做了线性变换；NNLM 和 C&W 模型更是做了非线性变换。

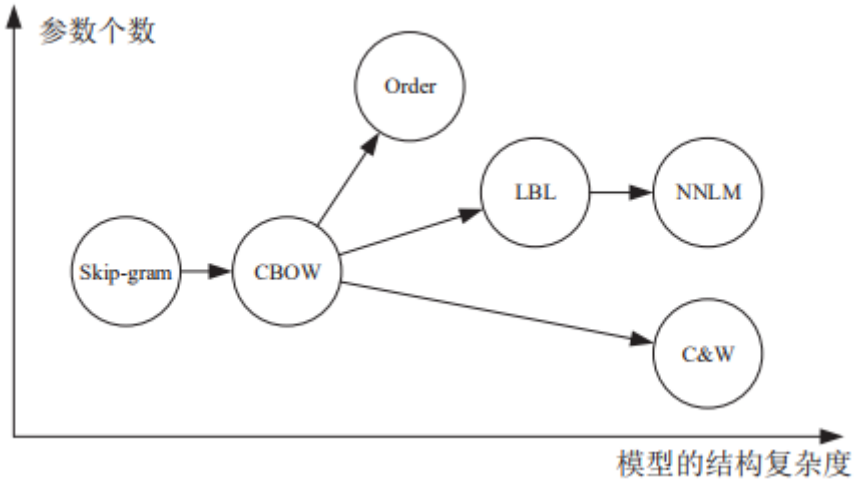

图 1：神经网络词向量模型复杂程度对比图

这些不同的策略可以从复杂度的角度进行分析。图 1 中展示了各词向量模型复杂度的关系图。图中箭头方向表示各模型复杂程度的拓扑顺序，从简单的模型指向复杂的模型。水平方向的相对位置表示模型结构的复杂程度，从简单到复杂。垂直方向的相对位置表示模型的参数个数，越靠上的模型参数越多。

2.2.2 情感分析研究存在的问题

情感分析经过十多年的发展，在某些领域上（例如产品评论、影评、宾馆、餐馆等）已经取得了相对成熟的发展和应用，并且在某些领域上达到了可完全实用的水准。但从一般意义上来说，情感分析还需要进行长期研究和探索，其最本质的难题还是语言文字的理解问题，依然存在非常多的挑战和待解的问题：

（1）针对社交媒体文本的情感分析任务仍极具挑战性。由于微博、微信等社交媒体上用户生成文本的开放性、自由性和不规范性，现有方法针对社交媒体

开放域文本的情感分析效果并不理想。针对社交媒体的开放话题进行情感分析的任务具有以下几个特点：评论对象或属性更加难以抽取，表达更加隐晦，甚至不存在明显属性描述词；许多话题不存在明显的观点评价词；理解情感表达需要更多的上下文。这些问题都还没有得到很好的解决，值得深入探索。

（2）在基于深度学习的端到端情感分析方面，对于语言文字这种高度概括和抽象数据的处理，简单的端到端并不能完全解决问题，还需要考虑更多语言学知识。如包括：情感词典如何有效地利用在端到端学习方法中；句法、语法、语义信息如何有效地结合在端到端的深度学习方法中。因此，将语言学知识和深度学习方法进行深度结合，有可能形成情感分析方法新的性能突破。

3　研究方法及结果

3.1 数据预处理

采用中科院计算所谭松波博士提供的较大规模的中文酒店评论语料：共有 10,000 篇。其中积极评论 7,000 篇，消极评论 3,000 篇。训练集：8,000 篇（含消极评论 2,400 篇，积极评论 5,600 篇）。测试集：2,000 篇（含消极评论 600 篇，积极评论 1,400 篇）

对数据进行清洗：去除数字、字母、标点符号、空格，分词以及去停用词。共 292,871 个词，不重复出现为 23,641 个。

3.2 描述性分析

本文将处理后所得的语料绘制词云图如下：

图 1：语料库词云图

同时得到词频最高的 10 个词：

表 1：语料库词频

	Words	Freq
1	酒店	10,171
2	房间	7,051
3	不错	4,026
4	服务	3,536
5	没有	3,146
6	入住	2,716

7	比较	2,355
8	感觉	2,079
9	早餐	2,066
10	非常	1,883

由于词汇较多，本文对所有词汇进行卡方检验，挑选出卡方值大于 3.84 的词汇。共有 306 个作为特征。前十个词汇及卡方值如表 2。

表 2：卡方检验后语料库词频

	Wods	Value
1	不错	1,245
2	不会	422
3	没有	363
4	根本	316
5	不要	307
6	携程	272
7	最差	259
8	很差	252
9	尽然	228
10	太差	223

3.3 TFIDF 词向量

本文共采用了 3 中词向量表示方法：TFIDF、Word2Vec 以及 TFIDF 乘以 Word2Vec 得到的词向量，同时对每种词向量表示方法运用 3 种分类方法来进行情感分析。

3.3.1 TFIDF + Logistic

基于 R 软件我们得到了如下结果：

表 3：TFIDF + Logistic 实验结果

	neg	pos
neg	495	216
pos	105	1178

此时模型测试集的召回率为 84%，精度为 91%，f1 为 88%，准确率为 83%。这里结合 ROC 曲线进一步考虑整体效果。如图所示，AUC 为 90%。

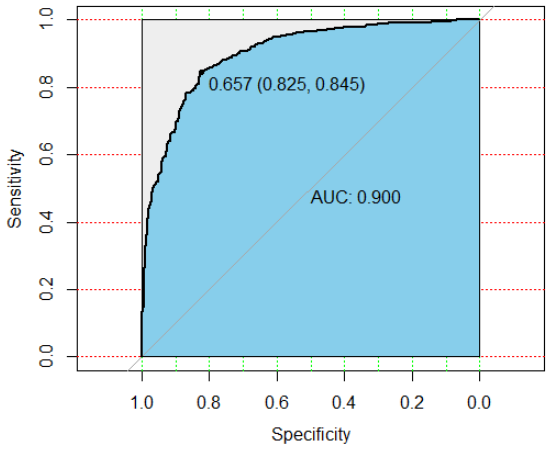

图 2 ROC 曲线图

可以看出实验效果已经非常好了能够掌握语料库中大部分特征。

3.3.2 TFIDF + LASSO

在用 LASSO 进行回归时选取λ为 0.036，选取了 219 个特征作为变量。相比较逻辑回归少了好多。同时得到了系数路径图：

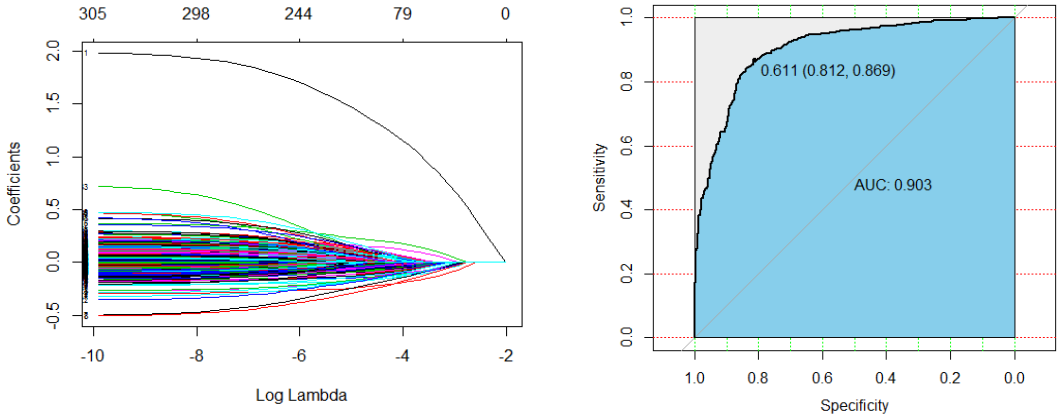

图 3 系数路径图、ROC 曲线图

实验结果为：

表 4：TFIDF + LASSO 实验结果

	neg	pos
neg	487	183
pos	113	1211

此时模型测试集的召回率为 87%，精度为 91%，f1 为 89%，准确率为 85%。这里结合 ROC 曲线进一步考虑整体效果。如图所示，AUC 为 90.3%。

3.3.3 TFIDF + Naïve Bayes

基于 R 软件我们得到了如下结果：

表 5：TFIDF + Naïve Bayes 实验结果

	neg	pos
neg	495	420
pos	105	974

此时模型测试集的召回率为 70%，精度为 90%，f1 为 79%，准确率为 74%。这里结合 ROC 曲线进一步考虑整体效果。如图所示，AUC 为 81.3%。

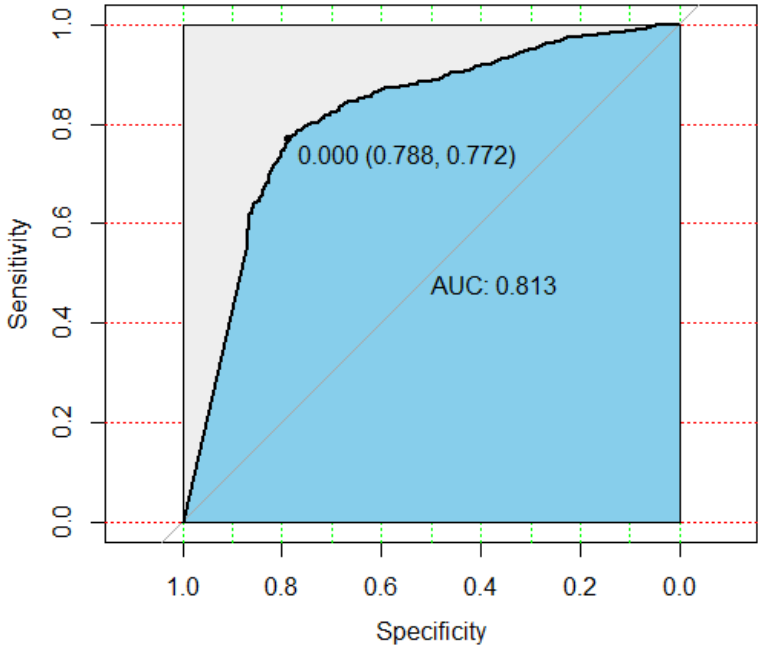

图 4 ROC 曲线图

可以看出运用 TFIDF + Naïve Bayes 方法实验效果已经非常好了能够掌握语料库中大部分特征。但效果不如前两种方法。

3.4 Word2Vec 词向量

接下来本文将采用 Word2Vec 词向量表示方法来进行情感分析。

3.4.1 Word2Vec + Logistic

基于 R 软件我们得到了如下结果：

表 6：Word2Vec + Logistic 实验结果

	neg	pos
neg	500	190
pos	100	1210

此时模型测试集的召回率为 90%，精度为 93%，f1 为 87%，准确率为 86%。这里结合 ROC 曲线进一步考虑整体效果。如图所示，AUC 为 91.6%。

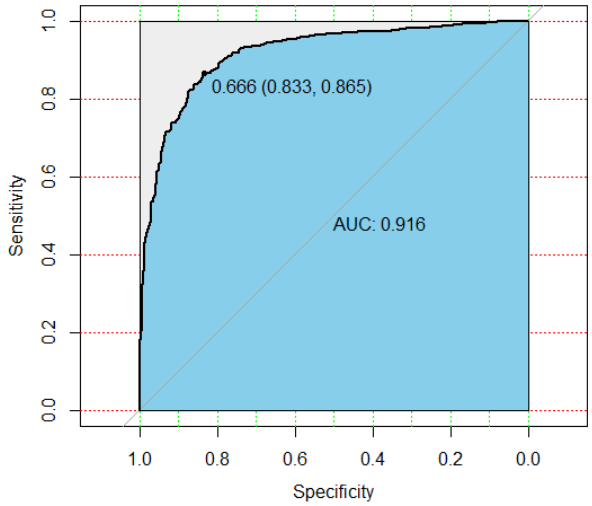

图 5 ROC 曲线图

可以看出实验效果已经非常好了，使用 Word2Vec 词向量比 TFIDF 效果要好。

3.4.2 Word2Vec+ LASSO

在用 LASSO 进行回归时选取λ为 0.049，选取了 61 个特征作为变量。相比较逻辑回归少了好多。同时得到了系数路径图：

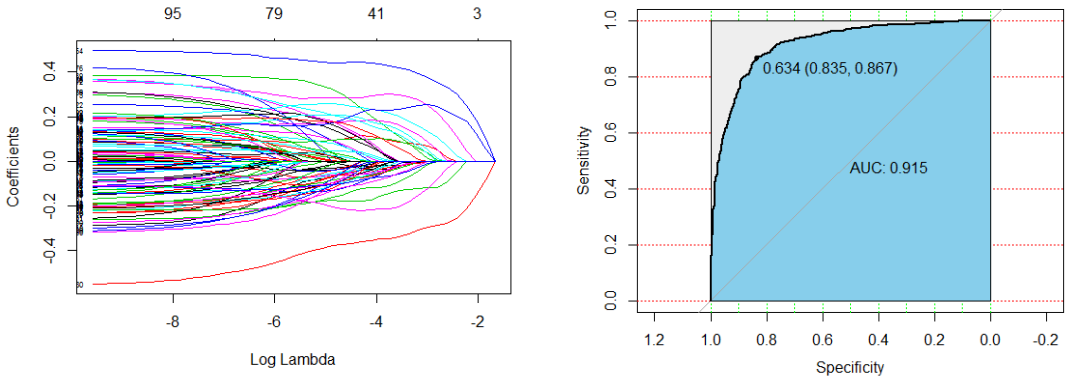

图 6 系数路径图、ROC 曲线图

实验结果为：

表 7：Word2Vec + LASSO 实验结果

	neg	pos
neg	512	236
pos	88	1164

此时模型测试集的召回率为 83%，精度为 93%，f1 为 88%，准确率为 84%。这里结合 ROC 曲线进一步考虑整体效果。如图所示，AUC 为 91.5%。

3.4.3 Word2Vec+ Naïve Bayes

基于 R 软件我们得到了如下结果：

表 8：Word2Vec+ Naïve Bayes 实验结果

	neg	pos
neg	503	245
pos	97	1155

此时模型测试集的召回率为 87%，精度为 86%，f1 为 88%，准确率为 82%。这里结合 ROC 曲线进一步考虑整体效果。如图所示，AUC 为 82.8%。

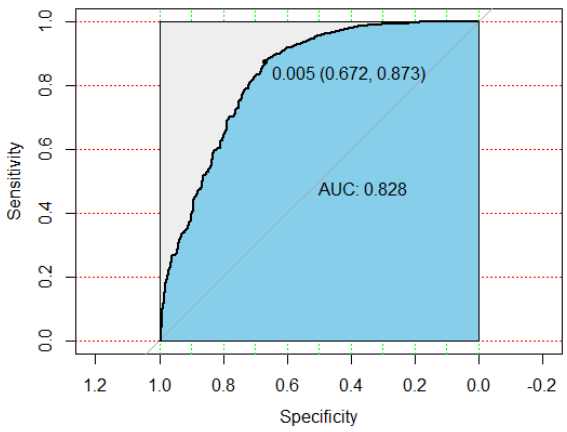

图 7 ROC 曲线图

3.5 Word2Vec × TFIDF 词向量

接下来本文将采用 Word2Vec × TFIDF 词向量表示方法来进行情感分析。

3.5.1 Word2Vec × TFIDF + Logistic

基于 R 软件我们得到了如下结果：

表 9：Word2Vec × TFIDF + Logistic 实验结果

	neg	pos
neg	509	220
pos	91	1180

此时模型测试集的召回率为 86 %，精度为 92%，f1 为 89%，准确率为 85%。这里结合 ROC 曲线进一步考虑整体效果。如图所示，AUC 为 91.2%。

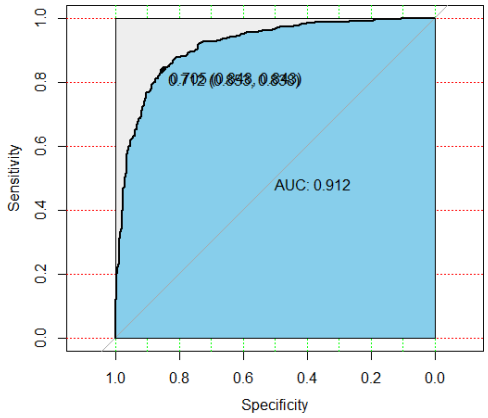

图 8 ROC 曲线图

可以看出实验效果已经非常好了，使用 Word2Vec × TFIDF 词向量与 Word2Vec 效果基本相同。

3.5.2 Word2Vec × TFIDF + LASSO

在用 LASSO 进行回归时选取λ为 0.039，选取了 66 个特征作为变量。同时得到了系数路径图：

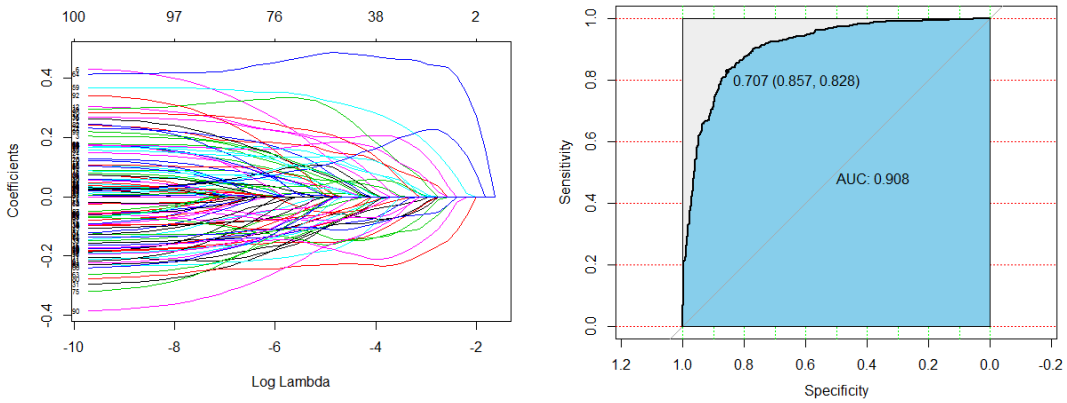

图 9 系数路径图、ROC 曲线图

实验结果为：

表 10Word2Vec × TFIDF + LASSO 实验结果

	neg	pos
neg	514	243
pos	86	1157

此时模型测试集的召回率为 84%，精度为 93%，f1 为 88%，准确率为 84%。这里结合 ROC 曲线进一步考虑整体效果。如图所示，AUC 为 90.8%。

3.5.3 Word2Vec × TFIDF + Naïve Bayes

基于 R 软件我们得到了如下结果：

表 11：Word2Vec × TFIDF + Naïve Bayes 实验结果

	neg	pos
neg	492	309
pos	108	1091

此时模型测试集的召回率为 78%，精度为 91%，f1 为 83%，准确率为 79%。这里结合 ROC 曲线进一步考虑整体效果。如图所示，AUC 为 86.4%。

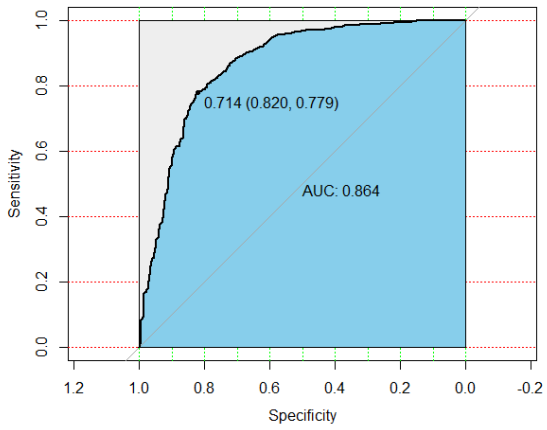

图 10 ROC 曲线图

4 研究结论

通过以上 3 种词向量表示以及 3 中分类方法可以看出 Word2Vec 方法明显要好于 TFIDF 词向量。有研究显示运用 TFIDF 与 Word2Vec 相乘所得的词向量会更加好，但本文实验结果为两者几乎没有差别，可能与本文实验数据集词汇量较少有关。以下给出了本文中 3 种方法的准确度汇总。

表 12：各词向量以及方法汇总

Accuracy	TFIDF	Word2Vec	Word2Vec × TFIDF
Logistic	0.83	0.86	0.85
LASSO	0.85	0.84	0.84
Naïve Bayes	0.74	0.82	0.79