第5章

舆情分析和预测分析

本章重点

- 舆情分析
- 预测分析

本章难点

- 基于机器学习的情感分析
- 基于机器学习的预测分析

5.1 舆情分析概述

5.1.1 舆情分析研究对象

舆情分析很多情况下涉及用户的情感分析，或者也称为观点挖掘，是指用自然语言处理、文本挖掘技术以及计算机语言学等方法来正确识别和提取文本素材中的主观信息，通过对情感因素的主观性文本进行分析，确定文本的情感倾向。

在商务和经济领域，情感分析的应用非常广泛，比如商品销售评论挖掘、产品推荐、金融股市预测、社会事件的评论等。文本情感分析的基本步骤是对文本信息的情感属性进行分类，诸如褒义贬义，正向负向，或者更复杂的情绪状态等。

在部分统计分类中，不带情感的中性类经常被忽略，在特定场合下，区分出中性类可以帮助提高分类算法的整体准确率。判定文本情绪的常用方法是利用比例换算系统，当一个词语通常被认为与消极、中性或积极的情感有关联时，可以给对象词赋予指定范围内的数值，数值的大小用于区分级类别，这使得非结构化文本数据的情感分析变得可能。第二种方法是根据研究目的，计算文本情感力度分数。第三种方法是主观和客观识别，这个

问题有时比其他级分类问题更难解决。因为主观属性判定可能是依赖于前后文语意，而客观文本也有可能包含主观信息。此外，结论在很大程度上可能依赖于主观概念的具体定义。第四种方法则基于功能和属性，针对实体在某方面或者特定功能下表现出来的意见或情感，功能可以理解为实体的属性或者组成部分，然后判断其情感属性。

现有的文本情感分析的途径大致可以分成关键词识别、词汇关联、统计方法和概念算法四大类别。关键词识别是利用文本中出现的定义实现分类。词汇关联则调查词汇和特定情感的内在关联。统计方法通过删除机器学习元素，比如潜在语意分析（Latent Semantic Analysis，LSA）、支持向量机（Support Vector Machines，SVM）等分析情感因素。概念算法则权衡了知识表达（Knowledge Representation）的元素，比如知识本体以及语义网络（Semantic Networks），因此也可以检测到文本信息的细微情感表达。例如，分析尚无明确相关信息的文本，可以通过分析与明确概念的潜在联系来获取信息。

5.1.2 情感分析方法

目前主流的情感分析方法主要有两种：基于情感词典的分析法和基于机器学习的分析法。

1. 基于情感词典的情感分析

基于情感词典的情感分析是指根据已构建的专家情感词典，针对对象分析文本进行文本处理抽取关键情感词，计算对象文本的情感倾向。分类质量很大程度上取决于专家词典的完善度和准确度。目前比较具有代表性的中文情感词典，包括知网情感分析用词语集、台湾大学情感词典以及清华大学褒贬词词典等。

2. 基于机器学习的情感分析

情感分析有时具有二分类特征，可以采用机器学习的方法识别，选取文本中的情感词作为特征词，将文本矩阵化，利用逻辑回归模型（Logistic Regression）、朴素贝叶斯模型（Naive Bayes）、支持向量机模型、K 均值模型（K-means）以及 K 均值改进模型等进行分类，训练文本的选择以及情感标注的准确性都可能影响分类效果。

下面重点介绍 K 均值算法，它是常用的聚类算法之一。算法的输入是一个样本集，通过该算法可以将用户或者用户情感样本进行聚类，具有相似特征的样本聚为同一类。针对选定的点，首先计算特定点与所有中心点的距离并取得最近中心点；然后将此点归类为此中心点所代表的簇，其他点执行类似运算；一次迭代结束之后，针对每个簇类，重新计算中心点，然后针对各点，重新分析各自最近的中心点；如此循环，直到前后两次迭代的簇类没有变化。根据 Hartigan-Wong 算法，簇类内部的差异性可以定义为式(5-1)：

$$\omega(C_k)=\sum_{x_i\in C_k}(x_i-\mu_k)^2 \tag{5-1}$$

其中，x_i 属于簇类 C_k 的数据点；μ_k 是簇类的平均值，即中心点。我们的目的就是求解所有簇类的点到其中心点的距离的平方和的最小值，参见式(5-2)：

$$\sum_{k=1}^{k}\omega(C_k)=\sum_{k=1}^{k}\sum_{x_i\in C_k}(x_i-\mu_k)^2 \tag{5-2}$$

K 均值算法的基本步骤如下。

(1) 指定聚类数量 K(可以通过优化算法获得)。

(2) 从数据集中任意随机选取 K 个对象作为初始聚类中心点或者平均值。

(3) 将每个数据点分给距离其最近的中心点,计算欧几里得距离。

(4) 基于 K 聚类,计算聚类数据点的新平均值,更新其聚类中心点,第 K 个聚类的中心点是一个向量,该向量包含此聚类所有观察点变量的平均值的长度。

(5) 迭代计算并最小化总的聚类平方和。重复执行步骤(3)和步骤(4),直到聚类分类结果不再变化或者达到最大迭代次数为止。

图 5-1 是基于 K 均值算法的用户聚类效果图,图中用户分为三类,相同颜色的样本表示同一种类别。

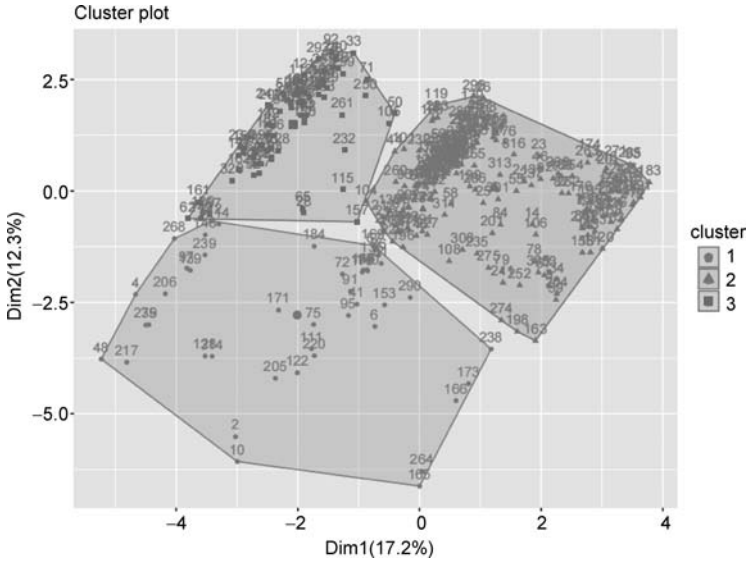

图 5-1　K 均值算法聚类结果示例

5.2　预测分析概述

在经济和社会活动中,很多场合需要进行预测分析,比如金融股票市场走势分析、用户罹患特定疾病的分析、天气预报等,影响预测算法的主要因素包括问题的复杂度、模型复杂度以及数据的质量等。

(1) 问题复杂度:如果问题异常复杂,描述问题存在困难,转换成计算机可以理解的问题描述过程中需要简化问题,而简化操作可能导致信息失真。

(2) 模型复杂度:复杂的问题,需要对匹配的复杂模型加以描述,对于复杂问题,如果模型过于简单,可能导致边界界定模糊,运算困难。

(3) 数据质量:用于预测的数据质量高低,将影响预测结果。

在预测评估中,常用的方法包括线性回归、逻辑回归和非线性回归。线性回归相对简

单，因变量和自变量之间属于线性关系，非线性回归则比较复杂，需要根据实际情况具体分析。逻辑回归的基本原理与线性回归基本相同，因变量为二分类变量或事件发生概率，主要用于分类问题，比如罹患疾病风险的评估、经济预测分析和舆情危机预警等场景。

假定因变量 y 代表特定事件，事件发生表述为1，没有发生表述为0，则表示该事件发生的概率为 $p(y=1)$，事件未发生的概率为 $1-p(y=1)$，假定二者之间的对数比是自变量 $x=(x_0,x_1,\cdots,x_n)^{\mathrm{T}}$ 的线性函数，如式(5-3)所示：

$$\ln\left(\frac{p}{1-p}\right)=\beta x=\beta_0+\beta_1 x_1+\beta_2 x_2+\cdots+\beta_n x_n \tag{5-3}$$

则 $\frac{p}{1-p}=\mathrm{e}^{\beta x}\Rightarrow p=\frac{\mathrm{e}^{\beta x}}{1+\mathrm{e}^{\beta x}}$，即事件发生的概率，取值范围为0～1，也是逻辑回归的数学原理。

衡量预测模型的性能，需要根据不同问题采取不同的方法。

(1) 回归问题，可以使用均方误差(Mean Squared Error，MSE)，平均绝对误差(Mean Absolute Error，MAE)和根均方误差即均方误差的算术平方根(Root Mean Square Error，RMSE)等作为性能评价指标。

(2) 分类问题，可以使用误分类率、受试者工作特征曲线(Receiver Operating Characteristic Curve，ROC)和曲线下面积(Area Under The Curve，AUC)作为性能评价指标。

5.3 电影评论情感分析实例

下面基于用户的电影评论，使用IMDB数据集，进行情感分析，原始数据总共包括正面和负面评价各25000条。

1. 导入库文件

先导入需要使用到的库文件，实例使用到的主要库信息包括Sklearn的CountVectorizer和TfidfVectorizer，Nltk，sklearn.linear_model中的LogisticRegression和SGDClassifier，以及CountVectorizer中的fit_transform()函数和transform()函数等。

2. 数据读取以及分类

读入电影评论原始数据，分别统计积极评价和消极评价的数量。

```
data = pds.read_csv('data/IMDB Dataset.csv')
data['sentiment'].value_counts()
```

分别输出获得正面、负面情感的数值统计结果，各自包含25000条记录。

```
positive    25000
negative    25000
```

划分训练集和测试集数据，关注评论详细信息以及正面评价或者负面评价的标识信

息，分别选定数据的最初 45000 条记录为训练集，而剩余的 5000 条为测试集，因此测试数据所占比例为 10%，此处可以适当调整，评估的结果也会相应发生变化。

```
train_evaluate = data.evaluate[:45000]
test_evaluate = data.evaluate[45000:]
train_flag = data.flag[:45000]
test_flag = data.flag[45000:]
```

3. 词袋模型文本向量化

利用词袋模型将文本信息向量化，调用 fit_transform()和 transform()函数分别将训练文本和验证文本转换为向量表达。

```
text_to_vector = CountVectorizer(encoding = 'utf - 8', decode_error = 'strict', strip_accents
 = None, lowercase = True, min_df = 0,max_df = 1,binary = False,ngram_range = (1,3))
train_evaluate_normalize_vector = text_to_vector.fit_transform(train_evaluate_normalize)
test_evaluate_normalize_vector = text_to_vector.transform(test_evaluate_normalize)

print('词袋模型训练数据维度信息:',train_evaluate_normalize_vector.shape)
print('词袋模型验证数据维度信息:',test_evaluate_normalize_vector.shape)
```

4. 词频-逆文档频率模型文本向量化

利用词频-逆文档频率模型将文本信息向量化，编码方式选择 UTF-8，英文字符转换为小写，调用 fit_transform()和 transform()函数分别将训练文本和验证文本转换为向量表达结果。

```
tfidf_vector = TfidfVectorizer(encoding = 'utf - 8', decode_error = 'strict', strip_accents =
None, lowercase = True, min_df = 0,max_df = 1,binary = False,ngram_range = (1,3))
train_evaluate_normalize_tfidf = tfidf_vector.fit_transform(train_evaluate_normalize)
test_evaluate_normalize_tfidf = tfidf_vector.transform(test_evaluate_normalize)
```

5. 逻辑回归模型

使用逻辑回归模型执行回归处理，设置最大迭代次数为 1500 次，设定随机状态以便对数据进行随机化处理，决策函数设定附带截距，CPU Cores 值为 1，设定 multi_class 为 ovr，代表模型根据二分类问题特征进行拟合。

```
logistic = LogisticRegression(C = 1, class_weight = None, dual = False, fit_intercept = True,
intercept_scaling = 1, max_iter = 1500, multi_class = 'ovr', n_jobs = 1,penalty = 'l2', random_
state = 42, solver = 'liblinear', tol = 0.0001,verbose = 0, warm_start = False)
```

6. 拟合分析

基于词袋模型对逻辑回归结果进行拟合分析。

```
logistic_fit = logistic.fit(train_evaluate_normalize_vector,train_transform)
print(logistic_fit)
```

基于词频-逆文档频率模型对逻辑回归结果进行拟合分析。

```
logistic_tfidf = logistic.fit(train_evaluate_normalize_tfidf,train_transform)
print(logistic_tfidf)
```

7. 输出结果

获得评论的统计结果，其分布特征参见图 5-2。

```
data['words'] = data['evaluate'].apply(lambda y: len(y.split()))
sns.displot(data = data, x = "words")
```

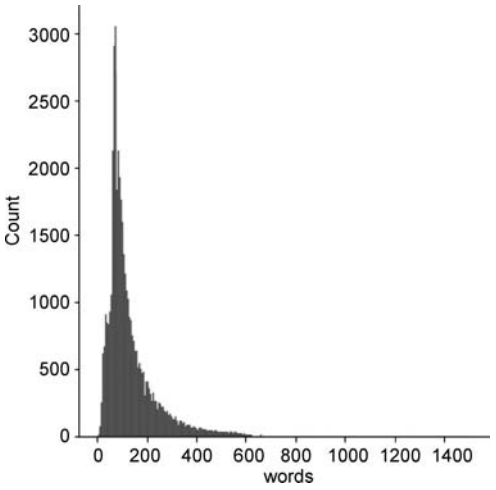

图 5-2 用户评论统计结果

对比两种模型的准确性评估异同。

```
bow_accuracy = accuracy_score(test_transform,bow_predict)
tfidf_accuracy = accuracy_score(test_transform,tfidf_predict)
print("词袋模型准确性:",bow_accuracy)
print("词频 - 逆文档频率模型准确性:",tfidf_accuracy)
```

输出两种模型的准确性结果：词袋模型准确性为 0.7614，词频-逆文档频率模型准确性为 0.761。执行基于词袋模型和词频-逆文档频率模型的分类评估指标对比。

```
bow_report = classification_report(test_sm, bow_pred, digits = 3, labels = None, sample_
weight = None,target_names = ['Positive','Negative'])
tfidf_report = classification_report(test_sm,tfidf_pred,digits = 3, labels = None, sample_
weight = None,target_names = ['Positive','Negative'])
```

```
print('词袋模型:\n',bow_report)
print('词频 - 逆文档频率模型:\n',tfidf_report)
```

从结果分析，两种模型的准确性评估结果为 0.75～0.77，维持在大致相当的水平，用户的正面评价和负面评价的指标分析结果没有发生很大差异。

```
词袋模型:
              precision  recall  f1 - score  support
   Positive     0.766     0.761     0.763      2530
   Negative     0.757     0.762     0.759      2470

词频 - 逆文档频率模型:
              precision  recall  f1 - score  support
   Positive     0.764     0.763     0.764      2530
   Negative     0.758     0.758     0.758      2470
```

5.4　用户罹患癌症预测实例

1. 实例背景

本节以加州大学维护的 UCI 机器学习存储库罹患乳腺癌病人数据集为基础，进行罹患癌症的预测分析，使用多种预测法，原始数据是 569 名用户的异常组织检测数据，其中包括良性样本 357 例，恶性样本 212 例。诊断指标包含半径、纹理、周长、面积、平滑度、紧凑性、凹度、凹点、对称性和分形维数。每个指标又包括平均值、标准误差和极值三类评价方法，本实例仅从半径、周长和面积平均值维度进行评估。

2. 导入库文件

首先，导入分析使用的库文件，主要使用的库信息包括 sklearn 中的 DecisionTreeClassifier 类文件库包、RandomForestClassifier 类文件库、GradientBoostingClassifier、AdaBoostClassifier、accuracy_score、confusion_matrix、LogisticRegression 以及 KNeighborsClassifier。实例的结果部分涉及中文显示，需要设置参数支持中文文字输出。

```
fontP = font_manager.FontProperties()
fontP.set_family('SimHei')
fontP.set_size(14)
```

3. 数据清洗和样本显示

根据实例研究的对象指标，输出部分样本的指标分布特征，如图 5-3 所示。

剔除其他指标，仅保留研究对象半径、周长和面积平均值指标，各观察指标的数据类型特征，参见图 5-4。

diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean
M	17.99	10.38	122.80	1001.0
M	20.57	17.77	132.90	1326.0
M	19.69	21.25	130.00	1203.0
M	11.42	20.38	77.58	386.1
M	20.29	14.34	135.10	1297.0

图 5-3　部分样本指标分布

```
#   Column          Non-Null Count  Dtype
---  ------          --------------  -----
0   diagnosis       569 non-null    object
1   radius_mean     569 non-null    float64
2   perimeter_mean  569 non-null    float64
3   area_mean       569 non-null    float64
```

图 5-4　观察指标数据类型

```
dr.drop(['id','Unnamed: 32','texture_mean','smoothness_mean','compactness_mean','concavity_
mean','concave points_mean','symmetry_mean','fractal_dimension_mean','fractal_dimension_
worst','symmetry_worst','concave points_worst','concavity_worst','compactness_worst',
'smoothness_worst','area_worst','perimeter_worst','texture_worst','radius_worst','fractal_
dimension_se','symmetry_se','concave points_se','concavity_se','compactness_se','smoothness_
se','area_se','perimeter_se','texture_se','radius_se','radius_se'], axis = 1 , inplace =
True)
```

4. 相关分析

对病人诊断结果进行分类，恶性肿瘤分类为 1，非恶性或者良性肿瘤分类为 0，具体函数为：

```
diagnosis.replace({"B":0,"M":1},inplace = True)
```

绘制各半径、周长和面积指标之间的相关关系，参见图 5-5。

```
correlation = dr.corr()
plt.figure(figsize = (7,7))
sn.set(font_scale = 1.2)
sn.set(font = 'SimHei')
sn.heatmap(correlation, cmap = 'Purples', xticklabels = ['诊断结果', '半径均值', '周长均值',
'面积均值'],yticklabels = ['诊断结果','半径均值','周长均值', '面积均值'],annot = True,
annot_kws = {"size": 12}, linewidths = 0, linecolor = 'white', cbar = True, cbar_kws = None,
cbar_ax = None, square = False)
```

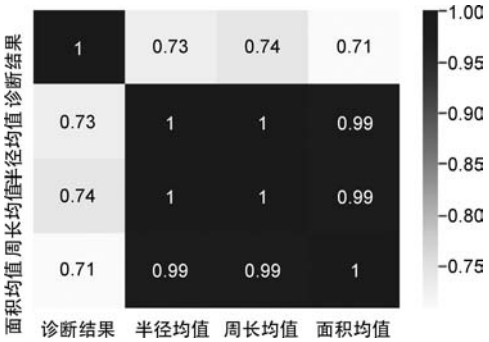

图 5-5　指标相关分析

选择相关性不小于68%作为阈值，分析特征值与罹患恶性肿瘤的关系，结果如图5-6所示。

```
correlation = dr.corr()
cancer_threshold = 0.68
filter = np.abs(correlation["diagnosis"]) >= cancer_threshold
correlation.filter = correlation.columns[filter].tolist()
sn.set(font_scale=2.5)
sn.set(font='SimHei')
sn.clustermap(dr[cf].corr(), figsize=(6, 6),annot = True, annot_kws={"size": 16},cmap=
"Purples").fig.suptitle("罹患肿瘤相关性分析", fontproperties=fontP,x=0.6, y=1.0)
```

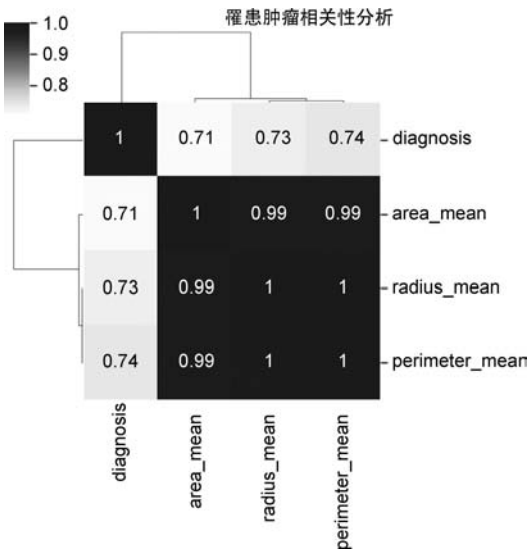

图5-6　指标与罹患肿瘤相关分析

描画三个指标之间的对关系图，参见图5-7。

```
sn.set(font_scale=1.5)
sn.set_style(style='white')
sn.pairplot(dr[correlation.filter],dropna=True,grid_kws=None, diag_kind = "kde",
markers = ".", hue = "diagnosis", palette='Purples')
```

5. 划分自变量和因变量

将数据划分为自变量和因变量，为后续模型分析准备数据。

```
y=dr["diagnosis"]
x=dr.drop(["diagnosis"],axis=1)
threshold= -2.5
data_filter= pds.dataframe["score"]< threshold
pds_filter_tolist= pds.dataframe[data_filter].index.tolist()
```

```
x = x.drop(pds_filter_tolist)
y = y.drop(pds_filter_tolist).values
```

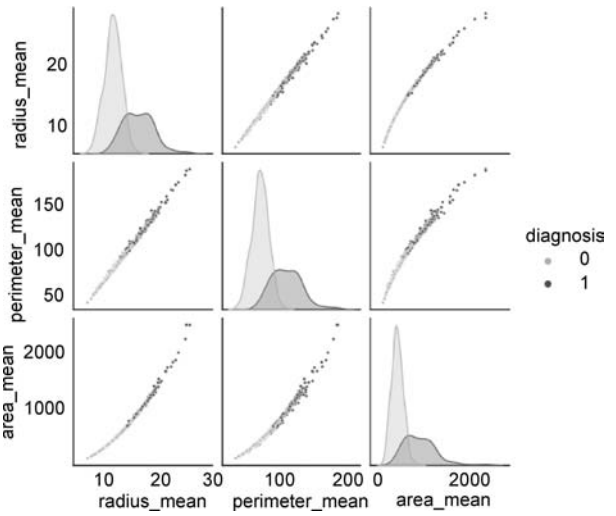

图 5-7　指标对关系图

6. 模型分析

使用不同分类方法，获得分析结果。

```
index = ['梯度法','逻辑回归法','近邻法','支持向量机法','决策树法','随机森林法']
model_list = [GradientBoostingClassifier(),LogisticRegression(), KNeighborsClassifier(n_
neighbors = 4), SVC(kernel = "rbf"), DecisionTreeClassifier(), RandomForestClassifier(n_
estimators = 600)]
model_dict = dict(zip(index,model_list))
```

根据不同模型预测，获得预测结果。

```
pediction_outcome_list = []
for i,j in model_dict.items():
    j.fit(x_train,y_train)
    x_prediction = j.predict(x_test)
    accuracy = accuracy_score(y_test, x_prediction)
    pediction_outcome_list.append(accuracy)
    print(i,"{:.4f}".format(accuracy))
```

输出不同模型的预测结果，随机森林法精度最高，决策树法精度最低。

```
梯度法 0.9027
逻辑回归法 0.9292
近邻法 0.9115
```

```
支持向量机法 0.9204
决策树法 0.8761
随机森林法 0.9292
```

不同模型的输出结果图形表示，参见图 5-8。

```
plt.figure(figsize = (10,4))
sn.barplot(x = index , y = pediction_outcome_list, palette = "vlag", ci = 95, edgecolor =
'purple',n_boot = 1000, dodge = False)
plt.rcParams['font.sans-serif'] = ['SimHei']
plt.ylabel("准确度",fontsize = 14)
```

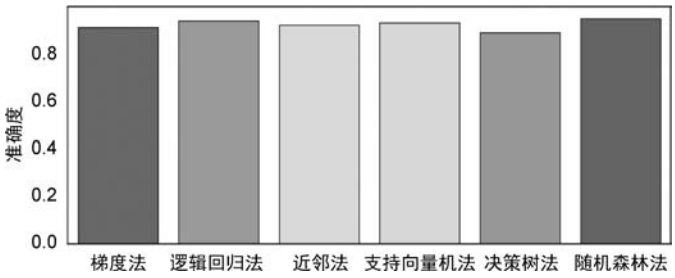

图 5-8　模型准确性比较

7. 梯度模型准确性变化趋势分析

选择梯度模型，分析其在迭代过程中的准确性变化规律。

```
gradient_boost = GradientBoostingClassifier(loss = 'deviance', learning_rate = 0.1)
result_list = []
accuracy = []
for j in range(1,28,1):
    x_train,x_test,y_train,y_test = train_test_split(x, y, test_size = 0.25, random_
state = j)
    gradient_boost_fit = gradient_boost.fit(x_train, y_train)
    fit_predict_x = gradient_boost_fit.predict(x_test)
    accuracy.append(accuracy_score(y_test, fit_predict_x))
    result_list.append(j)

plt.figure(figsize = (10,8))
plt.xlabel("迭代次数",fontsize = 14)
plt.ylabel("准确度",fontsize = 14)

std_accuracy = 0.5 * np.std(accuracy) / np.mean(accuracy)
plt.fill_between(result_list, (accuracy - std_accuracy), (accuracy + std_accuracy), color =
'purple', alpha = 0.1)
plt.plot(result_list, accuracy, color = 'purple')
```

```
for k in range(len(result_list)):
    accuracy[k] = "{:.5f}".format(accuracy[k])
    print(result_list[k],accuracy[k])
plt.show()
```

输出梯度法迭代的准确度信息变化趋势。

```
1 0.84507,2 0.87324,3 0.88028,4 0.85915
5 0.91549,6 0.88028,7 0.86620,8 0.86620
9 0.83803,10 0.90845,11 0.86620,12 0.88732
13 0.88732,14 0.91549,15 0.88028,16 0.85211
17 0.89437,18 0.85915,19 0.85211,20 0.85915
21 0.93662,22 0.91549,23 0.88732,24 0.88028
25 0.88028,26 0.88732,27 0.85211
```

梯度法分析获得的变化数据转换成图形显示，第 21 次迭代时达到极值，图形阴影部分表示置信区间，参见图 5-9。

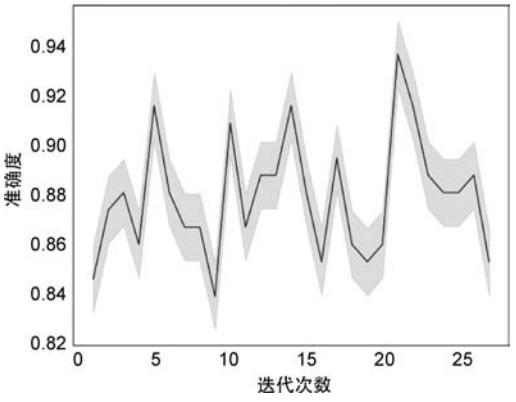

图 5-9　梯度模型准确度变化趋势

小结

本章分析了舆情分析和预测分析的基本概念和主要方法，并列举具体实例说明其在医学诊断和电影评论中的应用。

关键术语

基于机器学习的情感分析、受试者工作特征曲线、曲线下面积、梯度法、逻辑回归法、近邻法、支持向量机法、决策树法、随机森林法

习题

1. 简述舆情分析。
2. 情感分析的应用领域包括哪些？
3. 简述文本情感分析的基本步骤。
4. 简述中性类的作用。
5. 简述判定文本情绪的常用方法。
6. 简述文本情绪分析的四种方法。
7. 基于主观和客观的文本分析方法为什么有时比其他级分类问题更难解决？
8. 描述基于功能和属性的文本情感分析。
9. 现有的文本情感分析大致可以分为哪四类？
10. 简述关键词识别。
11. 简述词汇关联。
12. 简述统计方法。
13. 简述概念算法。
14. 目前主流的情感分析方法有哪两种？
15. 简述基于情感词典的情感分析。
16. 简述基于机器学习的情感分析。
17. 影响预测算法的主要因素包括哪些？
18. 描述如何根据不同问题采取不同的模型评估方法。
19. 根据教材提供的电影评论预测代码，选用其他用户评论素材，获得相应分析结果。
20. 根据教材提供的疾病预测代码，选用其他用户数据和评价指标，获得相应分析结果。