机器学习概论-集成学习实验报告

计 63 李志鹏 2016011283

2019 年 5 月 16 日

1 实验基础

1.1 任务描述

实现集成学习，对亚马逊网站上的五万个商品评价进行学习，给出其是高质量评价的概率，使用 AUC 方法进行评测。要求手写集成学习核心代码。并对不同的集成学习方法和机器学习算法的组合进行评价。至少要求比较 Bagging + SVM, Bagging + Decision Tree, AdaBoost + SVM 和 AdaBoost + Decision Tree 四种方法。

1.2 Bagging 方法

从训练集从进行随机采样组成每个基模型所需要的子训练集，对所有基模型预测的结果进行综合产生最终的预测结果。在随机采样时，每采集一个样本后，都将样本放回。对于我们的 Bagging 算法，随机采集和训练集样本数 m 一样个数的样本。这样得到的采样集和训练集样本的个数相同，但是样本内容不同。如果我们对有 m 个样本训练集做 T 次的随机采样，则由于随机性，T 个采样集各不相同。这种方法可以降低方差，但不可以降低 bias。

1.3 AdaBoost 方法

Boosting 的主要原则是训练一系列的弱学习器，对于训练好的弱分类器，如果是分类任务按照权重进行投票，而对于回归任务进行加权，然后再进行预测。

Algorithm 4 AdaBoost.M1

Input: Training set $S = \{\mathbf{x}_i, y_i\}$, $i = 1, \ldots, N$; and $y_i \in \mathbb{C}, \mathbb{C} = \{c_1, \ldots, c_m\}$; T: Number of iterations; I: Weak learner

Output: Boosted classifier:

$$H(x) = \arg\max_{y \in \mathbb{C}} \sum_{t=1}^{T} \ln\left(\frac{1}{\beta_t}\right) [h_t(x) = y] \text{ where } h_t, \beta_t$$

are the induced classifiers (with $h_t(x) \in \mathbb{C}$) and their assigned weights, respectively

1: $D_1(i) \leftarrow 1/N$ for $i = 1, \ldots, N$
2: **for** $t = 1$ to T **do**
3: $h_t \leftarrow I(S, D_t)$
4: $\varepsilon_t \leftarrow \sum_{i=1}^{N} D_t(i)[h_t(\mathbf{x}_i) \neq y_i]$
5: **if** $\varepsilon_t > 0.5$ **then**
6: $T \leftarrow t - 1$
7: **return**
8: **end if**
9: $\beta_t = \frac{\varepsilon_t}{1-\varepsilon_t}$
10: $D_{t+1}(i) = D_t(i) \cdot \beta_t^{1-[h_t(\mathbf{x}_i) \neq y_i]}$ for $i = 1, \ldots, N$
11: Normalize D_{t+1} to be a proper distribution
12: **end for**

图 1: AdaBoost 算法

AdaBoost 算法每一轮都要判断当前基学习器是否满足条件，一旦条件不满足，则当前学习器被抛弃，且学习过程停止。AdaBoost 算法中的个体学习器存在着强依赖关系，应用的是串行生成的序列化方法。每一个基生成器的目标，都是为了最小化损失函数。所以，可以说 AdaBoost 算法注重减小偏差。

2 实验设计

2.1 特征选择和提取

训练集中给出的数据种类有 reviewerID, asin, reviewText, overall, votes_up 和 votes_all。其中 votes_up 和 votes_all 只在训练集中出现，作为 label 的依据。由于测试集中并不会出现这两种数据，因此我认为这两类数据可以忽略，避免造成特征维数的增加。

对于其他数据种类，除了 reviewText 外，都为整数或浮点数，较容易整合为特征。而 reviewText 中的信息为自然语言, 较难提取出特征向量, 与其他种类的数据不能用同一种方法提取特征。因此单独将 reviewText 提取出考虑。我选择使用 TF-IDF 方法对 reviewText 进行特征提取。

使用 sklearn.feature_extraction 中的 CountVectorizer 和 TfidfTransformer 实现 TF-IDF 方法提取特征。对于其他数据，使用 DictVectorizer 提取特征。

2.1.1 TF-IDF 方法

TF-IDF(term frequency–inverse document frequency) 是一种统计方法，用以评估一字词对于一个文件集或一个语料库中的其中一份文件的重要程度。字词的重要性随着它在文件中出现的次数成正比增加，但同时会随着它在语料库中出现的频率成反比下降。

词频 (term frequency, TF) 指的是某一个给定的词语在该文件中出现的次数.

逆向文件频率 (inverse document frequency, IDF) 是一个词语普遍重要性的度量。某一特定词语的 IDF，可以由总文件数目除以包含该词语之文件的数目，再将得到的商取对数得到。

2.2 基础分类器的选择

为了探索不同的基础分类器对集成学习效果的影响，除了实验要求使用的决策树和 SVM 外，我还尝试了 KNN 和朴素贝叶斯分类器。

其中朴素贝叶斯分类方法我还具体尝试了伯努利朴素贝叶斯和多项分布朴素贝叶斯方法。伯努利朴素贝叶斯用于多重伯努利分布数据的贝叶算法。即有多个特征，但每个特征都假设是一个二元 (Bernoulli, boolean) 变量。因此，这类算法要求样本以二元值特征向量表示。多项分布朴素贝叶斯实现服从多项分布数据的贝叶斯算法，使用 TF-IDF 向量在实际项目中表现得很好。

使用 sklearn 库中的五种分类器实现，与两种集成学习的方法组合一共十种组合，列表如下。

Bagging+DTree	AdaBoost+DTree
Bagging+SVM	AdaBoost+SVM
Bagging+KNN	AdaBoost+KNN
Bagging+BernuliBayes	AdaBoost+BernuliBayes
Bagging+MultinomialBayes	AdaBoost+MultinomialBayes

表 1: 基础分类器与集成学习方法的组合

2.3 集成学习的参数

2.3.1 Bagging 的参数

Bagging 方法中的一个重要参数是参与投票的基础分类器的个数。为了研究参与投票的基模型的数量对集成学习效果的影响，设定多个不同的值进行测试观察。

2.3.2 AdaBoost 的参数

AdaBoost 中的重要参数为总的迭代次数。当 σ 小于 0.5 时，持续进行迭代，不断更新权值，直到到达迭代次数。因此当 AdaBoost 方法不中途退出时，何时终止迭代也是一个值得探索研究的问题。

3 结果分析和对比

3.1 特征提取分析

如实验设计中所提到的，对于文本特征可以进行特殊的 TF-IDF 处理，也可以将文本特征与其他特征合并起来共同提取特征。同时我也测试了不使用文本数据，只使用其他特征进行分类的结果。为了保证其他变量一致，均使用 Bagging+SVM 进行训练和预测，Bagging 参数为 100。

特征选择	AUC 表现
只使用 ReviewText	0.77807
使用除 ReviewText 之外的特征	0.76785
使用全部特征	0.79568

表 2: Bagging(100 次)+SVM 对不同特征的表现

可以发现，单独使用文本信息或单独使用 overall 等评分信息，都可以较好的实现分类和预测，但是当两者组合起来，共同作为特征时，效果更好。我认为共同考虑文本和其他数据特征时，并不会造成特征维数过多的情况，也不会使得训练时间较长。因此在后续实验中，均提取了所有的特征。

3.1.1 关于文本提取的发现

当进行文本特征提取时，我发现如果只使用训练数据中的文本，则可能在自己划分出的验证集上表现较好，但是在测试集上表现明显变差。经过思考后我认为，对于文本来讲，

会有很多不同的单词或其他特征，那么很有可能出现一些测试集中的单词从没有在训练集中出现，导致缺失相关的信息，从而导致表现较差。为了验证这一想法，我在使用 TF-IDF 方法时，将训练集和测试集的所有文本信息共同考虑，再分别给训练集和测试集提取特征。实践后发现分类器的表现的确变好，验证了之前的猜想。

3.2 基础分类器的分析

我一共使用了四种基础分类器，均使用 sklearn 库中的接口进行调用。

3.2.1 决策树

决策树作为 baseline 分类器，对于已经提取好 feature 的数据，得到的结果如下。

集成学习方法	AUC 表现
Bagging(40 times)	0.78895
AdaBoost(40 times)	0.78677

表 3: 决策树的得分

但是决策树对于维度较大的特征，训练较慢，且对于训练集很容易出现过拟合问题，因此我对决策树的深度进行了限制，并探索了不同深度限制下决策树的表现。

集成学习方法	AUC 表现
Bagging(40 times)	0.78176
AdaBoost(40 times)	0.79409

表 4: 最大深度为 1000 的决策树的得分

结果表明在限制深度的情况下，AdaBoost 的表现较好，Bagging 的效果反而变差，这一点在 Section3.3 中进行详细讨论。

由于 AdaBoost 的表现符合预期，选择 AdaBoost 方法继续探索不同深度限制下的表现。

深度限制	AUC 表现
500	0.78467
1000	0.79409
1500	0.79053

表 5: 不同深度限制的决策树 +AdaBoost(40 times) 的得分

从结果中可以看出，决策树的深度并不是越大越好。深度限制为 1500 的决策树表现不如深度限制为 1000 的决策树，说明决策树的确存在过拟合问题，而深度限制为 500 的决策树中，由于分类器表现太差，在 AdaBoost 循环中提前退出，也说明深度限制过于严格的决策树不适合在 AdaBoost 中使用。

3.2.2 SVM

不管使用 TF-IDF 方法处理的特征，还是直接使用 DictVectorizer 提取的特征，维度都较大，比较稀疏。对于普通的 SVM，训练过程较长，且稀疏的特征一般而言可以很好地被线性分类，因此我选择使用 LinearSVC 分类器。同时为了避免文本特征过多，使分类器忽略掉其他重要特征，我在 LinearSVC 中使 class_weight=balanced，对不同特征进行加权。

集成学习方法	AUC 表现
Bagging(20000 times)	0.82085
AdaBoost(2000 times)	0.79811

表 6: SVM 的得分

SVM 的表现要比决策树显著的好，而且 LinearSVC 分类器的训练速度快，可以使用较大的集成学习参数进一步提升性能，具体讨论在 Section3.3 中讨论。

3.2.3 KNN

除了决策树和 SVM 外，我还尝试使用了 KNN 进行训练，效果如下。

集成学习方法	AUC 表现
Bagging(20 times)	0.77724
AdaBoost(10 times)	0.76313

表 7: KNN 的得分

由于 KNN 对于维数较大的数据训练十分缓慢，所以我选取的集成学习参数都较小。可以看到 KNN 的效果明显不如决策树和 SVM，我认为这是因为 KNN 相对于其他分类器，不太适合用于进行包含大量文本的分类问题，因为文本内的信息关系过于复杂，文本和最后的 label 对应关系不是十分明确，所以 KNN 在这样一个分类问题上表现不是很好。

3.2.4 朴素贝叶斯

朴素贝叶斯分类中我又分别尝试使用伯努利朴素贝叶斯和多项分布朴素贝叶斯方法。效果如下。

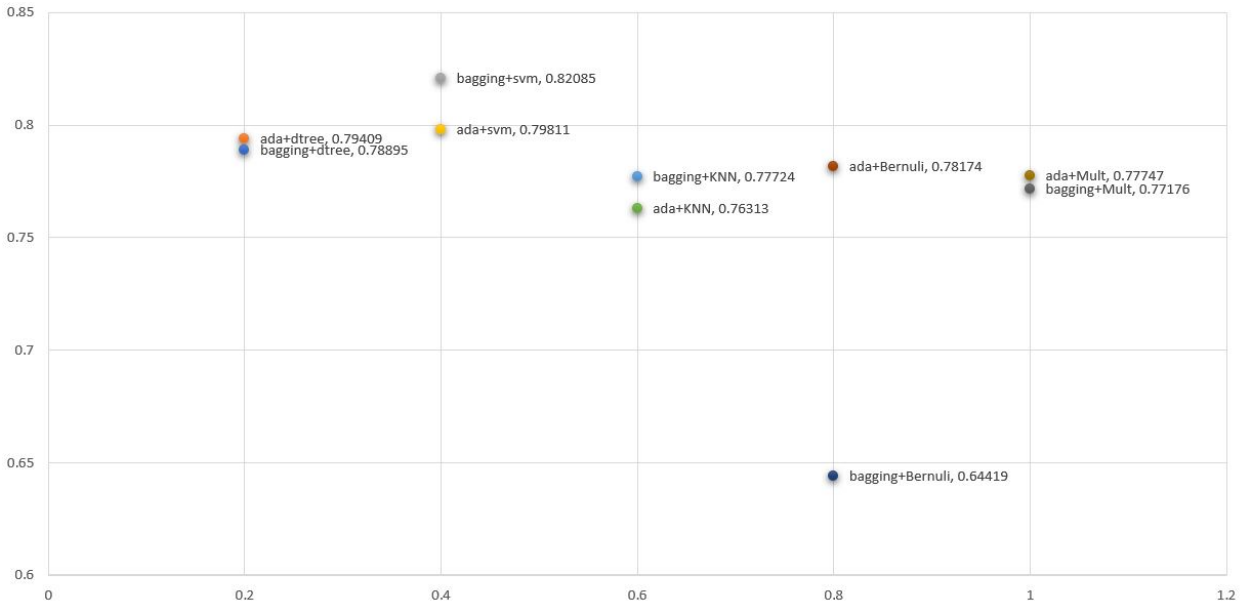
集成学习方法和基础分类器	AUC 表现
Bagging(100 times) + Bernuli	0.64419
AdaBoost(100 times) + Bernuli	0.78174
Bagging(100 times) + Multinomial	0.77176
AdaBoost(100 times) + Multinomial	0.77747

表 8: 朴素贝叶斯的得分

可以看到，除了 Bagging + Bernuli Bayes 的效果显著的不好之外，其他的方法组合效果基本处于 KNN 和决策树之间。对于 Bagging + Bernuli 和 AdaBoost + Bernuli 的区别也 Section3.3 中进行讨论。

3.2.5 不同分类器的比较

通过以上对不同分类器的实验，我发现 SVM 的表现最好，决策树次之，而 KNN 的表现最差。总体情况如下图。



我认为主要原因有两个。第一个原因是实验的数据集中包含大量的文本，导致提取的特征维数较大，使得 KNN 表现较差，决策树容易出现过拟合，同时也使得 SVM 脱颖而

出。第二个原因就是 LinearSVC 训练速度相对于其他分类器较快，使得可以进行更多次数的迭代，使得最后给出的概率值更加准确，在 AUC 评分体系中更有优势。

3.3 集成学习的参数分析

在上面的实验过程中，我对 Bagging 和 AdaBoost 方法中的迭代次数做了调整和测试，结果如下。

Bagging 参数	AUC 表现
100	0.79568
1000	0.81609
10000	0.82027
20000	0.82085

表 9: Bagging+SVM 中迭代次数和表现

AdaBoost 参数	AUC 表现
200	0.79652
2000	0.79811
5000	0.79072

表 10: AdaBoost+SVM 中迭代次数和表现

分析如下。

3.3.1 Bagging 的参数

在实验中，我发现对于 Bagging 的迭代次数，一般来讲是迭代次数越多，得到的效果越好。通过我自己的分析和学习，我认为原因如下。

对于 bagging 来说，每个基分类器的权重等于 1/m，且期望近似相等。

$$
\begin{aligned}
&\mathrm{E}(F) = \gamma * \sum_{i}^{m} E(f_i) \\
&= \frac{1}{m} * m * \mu \\
&= \mu \\
&\mathrm{Var}(F) = m^2 * \gamma^2 * \sigma^2 * \rho + m * \gamma^2 * \sigma^2 * (1-\rho) \\
&= m^2 * \frac{1}{m^2} * \sigma^2 * \rho + m * \frac{1}{m^2} * \sigma^2 * (1-\rho) \\
&= \sigma^2 * \rho + \frac{\sigma^2 * (1-\rho)}{m}
\end{aligned}
$$

上面的计算表明，整体模型的期望近似于基模型的期望，这也就意味着整体模型的偏差和基模型的偏差近似。随着 m 的增大，整体的方差减小，防止过拟合的能力增强，模型的准确度得到提高。当 m 大到一定数量级后，第二项对整体的影响就很小了，从而达到一个上界。这说明 m 的值对于整体模型的增强效果是有上界的，这也很好的解释了实验现象。此外，这个式子也说明 Bagging 方法使用基模型要为强分类器，否则就会使整体模型的准确度降低。

3.3.2 AdaBoost 的参数

对于 boosting 来说，基模型的训练集抽样是强相关的，那么模型的相关系数近似等于 1，所以公式简化如下。

$$
\begin{aligned}
&\mathrm{E}(F) = \gamma * \sum_{i}^{m} E(f_i) \\
&\mathrm{Var}(F) = m^2 * \gamma^2 * \sigma^2 * \rho + m * \gamma^2 * \sigma^2 * (1-\rho) \\
&= m^2 * \gamma^2 * \sigma^2 * 1 + m * \gamma^2 * \sigma^2 * (1-1) \\
&= m^2 * \gamma^2 * \sigma^2
\end{aligned}
$$

观察方差的表达式，发现如果使用的基模型不是弱分类器的话，方差会较大，无法防止过拟合。而使用弱分类器，使得每个基模型的准确度都不高，此时增加基模型的数量，可以使整体模型的期望增加，提升整体准确度。但是如果模型数量进一步增大，又会使方差变大，导致防止过拟合的能力降低，准确度又会降低。这也很好的解释了 AdaBoost+SVM 的实验结果。

3.3.3 Bagging 和 AdaBoost 的比较

在实验中，我发现在限制决策树深度的情况下，AdaBoost 的表现较好，Bagging 的效果反而变差。通过上面的分析，这一现象很好解释。因为当限制了深度时，决策树分类器

的强弱程度变弱，而 Bagging 方法要求使用强分类器，AdaBoost 方法偏向于弱分类器。所以当分类器变弱时，使用 Bagging 方法的效果就会变差，而 AdaBoost 效果会变好。

对于 Bernuli Bayes 的情况也是类似的。当使用 Bernuli Bayes 作为基模型时，由于 Bernuli Bayes 是一个十分弱的分类器，所以 Bagging + Bernuli 的效果就会十分的差，而 AdaBoost + Bernuli 就不会变差，反而还会相对变好。

通过以上的实例和分析，Bagging 和 AdaBoost 在实际使用中的相同点在于，适当的增加迭代次数都会使效果变好。区别主要在于迭代次数的增加会使 Bagging 逐渐达到上界，而会使 AdaBoost 达到上界后反而下降。同时其对基模型的要求不同，对于弱分类器，AdaBoost 方法适合一些，而对于强分类器，Bagging 方法的表现更好。

4 实验总结

在本次实验中，我最终达到了 0.82085 的得分，在所有参加比赛的队伍中排名第六。

除了上面讨论分析的问题外，我在实验中还发现了一些可以改进之处。比如对文本数据进行进一步处理，去除掉 a，the 等词，使得文本可以更有效的提供信息。同时，在基模型的调参上还有一些调整的空间。

在本次实验中，我不仅通过实践进一步学习掌握了集成学习的方法和技巧，同时还通过自己的思考和探索验证了一些集成学习相关的特点，并且辅以数学推导进行证明。此外，我还掌握了对于特征提取的一些方法和技巧，也了解到更多的提取特征的注意点和易错处。对以后的课程学习和机器学习相关的知识有了更浓厚的兴趣。在此我也特别感谢老师和助教的教导和付出。