数据挖掘练习实验报告

丁云翔 [1+]

[1](南京大学 软件学院,江苏省 南京市 210000)

The experiment report of data mining practice

DING Yun-Xiang[1+]

[1](Department of Software, Nanjing University, Nanjing 210000, China)

+ Corresponding author: Phn +86-18851159919, E-mail: 191250026@smail.nju.edu.cn

Abstract: For a given data set, find the relationship between the data, summarize and screen out some data features, and then preprocess the data. Finally, some data mining and machine learning algorithms are used to predict whether a given paper is written by a given author.

Key words: data mining, machine learning, data preprocessing, feature selection

摘 要: 对于给定的数据集，发现数据之间的关系，归纳并筛选出一些数据特征，然后对数据进行预处理，最后运用了一些数据挖掘和机器学习算法，以预测给定论文是否为给定作者所著。

关键词: 数据挖掘、机器学习、数据预处理、特征选择

本次任务需要预测给定论文是否为给定作者所著。我了解所有数据信息后，根据知识对数据进行了预处理，选择出对预测有帮助的那部分特征，并利用部分选出的特征和基于其他选出的特征构建出的新特征，作为实际的训练数据特征，尝试不同的机器学习算法，比较预测的性能。

1 对本次任务的理解

本次任务要求预测给定论文是否为给定作者所著，是一个真实场景中的数据挖掘任务。所掌握的可用来训练的数据有：作者信息、论文信息、论文与其作者信息（作者 id 不准确）、会议信息、期刊信息和训练数据集。训练数据集包括了已确认是和不是某个作者发表的论文 id。而测试数据集也是只有作者 id 和论文 id，要预测该论文是否为该作者所著。

如果只靠作者 id 和论文 id 作为特征，则学习的模型对于不在训练集中出现的 id 组合无法有效预测，泛化能力几乎为 0，显然不是我们所需要的。所以我们需要根据作者 id 和论文 id，在作者数据集、论文数据集和论文与其作者数据集中根据作者 id 或论文 id 找到有助于预测的特征，并作为训练模型的特征。

有了这些特征以后，模型的学习过程就可以看作是一个二分类分类器的学习过程了，一类为给定论文是给定作者所著，另一类是给定论文不是给定作者所著，只需要选择最优的模型和参数，即可对测试数据进行有效预测。

2 选择树/森林型集成学习算法的理由

这次的数据特征几乎都为离散值，决策过程类似于通过一条条规则来确定类别，所以使用树/森林类型的分类算法比较适宜，可处理混合类型特征，且可以自行进行特征选择和组合。

又对于大规模的数据集，使用集成模型效果会比单个模型要更好，故选择了一些集成的树/森林类型的算法进行测试。

在这些算法中，XGBoost 自带正则化，可缓解过拟合，且支持并行处理，模型训练耗时更少，可以在更短的时间内学得更多的基分类器，因为训练和测试得时间有限，训练和测试时间对我而言也很重要，最后 XGBoost 的结果提交的得分也最高（任务一）。但在任务二中效果并不理想，所以我在任务二中使用了 GradientBoosting。

3 方法细节介绍

3.1 数据预处理和特征提取

查看所有数据表以后，我认为获取特征需要从作者 id 和论文 id 出发，在别的数据集中获取对应的有用信息作为特征。由于论文与其作者数据中作者 id 并不准确，所以作者信息只能从作者数据表中获取，包括作者 id 对应的作者名和所属的机构。由于名字可能有多种表示格式，故同一个作者 id 对应一或多个作者名和所属机构信息对。而论文 id 方面，论文数据表内并无与作者有关的信息，我感觉那些特征对性能的提升相对较小，所以先考虑论文与其作者数据集中的信息，虽然其中的作者 id 并不准确，但是论文 id 与论文作者名和作者所属机构还是准确匹配的。故可以根据用作者 id 和论文 id 分别获取的作者名和作者所属机构是否对应来作为特征。且该特征不依赖训练集数据，即使是对于训练集中没出现的作者和论文也依然有效，泛化能力较好。

特征具体表示上，除了原有的作者 id 和论文 id 外，还另外添加了两维特征。第一维是：若用作者 id 获取的作者名集合中有一个名字表示与用论文 id 获取的作者名相同，则该维的值为 1；否则为 0。第二维是：若用作者 id 获取的作者所属机构集合中有一个机构与用论文 id 获取的作者集合中的某个作者所属的机构相同，则该维的值为 1；否则为 0。因为同一篇论文的作者多属于同机构，故可以使用该方式还计算特征的值。

不过由于时间和 Kaggle 提交测试的次数限制，最佳的特征表示形式还未完全对比测试。比如当作者的所属机构信息缺失时的处理方式，按上述的表示会将对应的特征的值置 0，但该情况与机构不匹配并不相同，都置为 0 并不合适。至于这种情况是应该标记为缺失值还是用 0 和-1 将这两种情况区分开，因为提交机会已用完，无法再测试得出结论。

由于后续使用的部分分类器不支持直接使用 string 类型作为特征，故引入 OneHotEncoder 对 string 类型的 id 特征进行编码，用得到的稀疏表示作为输入数据（常规表示特征表示矩阵太大，超出内存限制）。同样，对于支持 string 类型特征的分类器，有些测试是不使用编码和稀疏表示进行的。

由于任务二的表现较差，我还考虑了删除作者 id 和论文 id 的特征，只保留新增的两维特征进行训练和测试。但是由于这两维特征所能表示的信息实在太少，性能还不如保留 id 特征。

3.2 机器学习算法

基于 2 中的理由，我选取了 XGBClassifier、GradientBoostingClassifier、RandomForestClassifier 和 CascadeForestClassifier 共四种树/森林类型的分类器进行对比测试，由于剩余时间和提交次数的限制，并没有办法完全控制变量进行对比并测试各种参数组合下的性能。分类器的具体实现是使用现成的 sklearn 和 deepforest 库。

4 与基线方法的对比

任务一使用 XGBoost 作为对比方法，其他方法作为基线方法；任务二由于 XGBoost 的输出全为 1（即 AUC 为 0.5），故使用 GradientBoostingClassifier 作为对比方法，其他方法作为基线方法。由于提交次数有限，只能基

于已提交的部分参数组合来进行对比。

4.1 任务一

XGBClassifier，n_estimators=5000，输入数据包含新增的两维特征，且经过了编码和稀疏化表示，任务一的 AUC 为 0.66007。

RandomForestClassifier，n_estimators=100，输入数据不包含新增的两维特征，不经过编码和稀疏化表示，任务一的 AUC 为 0.60429。

XGBClassifier，n_estimators=5000，输入数据的特征只有作者 id 和论文 id，经过了编码和稀疏化表示，任务一的 AUC 为 0.65472。

XGBClassifier，n_estimators=5000，输入数据的新增特征只有对作者机构是否匹配的判断，且经过了编码和稀疏化表示，任务一的 AUC 为 0.65991。

GradientBoostingClassifier，n_estimators=5000，输入数据包含新增的两维特征，不经过编码和稀疏化表示，任务一的 AUC 为 0.65160。

4.2 任务二

GradientBoostingClassifier，n_estimators=5000，输入数据的新增特征只有对作者机构是否匹配的判断，不经过编码和稀疏化表示，任务二的 AUC 为 0.51835。

RandomForestClassifier，n_estimators=100，输入数据不包含新增的两维特征，不经过编码和稀疏化表示，任务二的 AUC 为 0.50244。

RandomForestClassifier，n_estimators=100，输入数据包含新增的两维特征，不经过编码和稀疏化表示，任务二的 AUC 为 0.49803。

GradientBoostingClassifier，n_estimators=5000，输入数据包含新增的两维特征，不经过编码和稀疏化表示，任务二的 AUC 为 0.51506。

XGBClassifier，n_estimators=5000，输入数据包含新增的两维特征，且经过了编码和稀疏化表示，任务二的 AUC 为 0.50000。（测试集输出全为 1）

4.3 对比

虽然性能仍不算好，但 XGBoost 和 GradientBoosting 相较于 RandomForest 仍然要更好一些。XGBoost 在任务一的表现最好，但在任务二中只能输出全 1，并不可用；而 GradientBoosting 在任务二表现最好，在任务一与 XGBoost 差距也很小。二者在任务一都要优于 RandomForest。

另外由于测试 CascadeForestClassifier 时提交次数已用尽，故无法得到确切的结果。且由于 CascadeForestClassifier 不支持 string 类型的特征，也不支持稀疏表示的特征，故只能在去掉作者 id 和论文 id，仅将新增的两维特征作为输入数据时使用。但由于这两维包含的信息实在太少，这四种分类算法的训练集准确率均只能达到 53%，测试集输出均为全 1，并不可用。

5 结论与总结

5.1 结论

由实验结果可得，新提取出的两个特征对预测性能还是有提升的。而在模型选择上，XGBoost 和 GradientBoosting 相较于 RandomForest，泛化性能要更好一些。因为在训练集上的准确率，RandomForest 要高于 XGBoost 和 GradientBoosting，而在测试集上则相反。但是，以目前提取出的两个特征，仍无法取得较好的性能。

5.2 总结

由于是第一次直接在原始数据上进行数据挖掘，不同于以前的数据可以直接用来训练，这次任务需要自己对

原始数据进行预处理，并根据原始数据选择出合适的特征和表示方法，有时还需要新建特征。我对数据预处理和特征选择等步骤还是不太熟悉，而且 ddl 又正好是期末周，大三复习压力较大，抽不出充足的时间来仔细对比各种特征表示和算法模型的效果，且提交次数也只有几次机会了。虽然没能取得较好的性能，但还是学到了很多东西。

References:

[1] Sklearn 说明文档

[2] Deepforest 说明文档