

Bagging 集成方法在保险欺诈识别中的应用研究

李秀芳 黄志国 陈孝伟

[摘要] 保险欺诈不仅危及保险公司的正常经营,增加投保人的负担,甚至有可能影响到国家的金融稳定。随着大数据时代的到来,保险反欺诈亟需引入革命性技术。Bagging 集成方法以其可调节模型结构、易于部署、参数空间可控、支持并行运算等特点成为保险公司进行保险反欺诈一个好的选择。Bagging 方法主要包括 Bagging 算法、Random Subspace 算法、Random Patches 算法,它们又能与不同基学习器结合构成新的分支算法及算法特例。本文基于这些算法对保险欺诈问题进行了实证检验,分析了各算法及与基学习器的适用性问题,以及基学习器个数对算法表现的影响。分析发现:针对保险欺诈识别问题,在 Bagging、Random Subspace、Random Patches 三者之中,Random Patches 算法的表现最好,Bagging 的运行时间最短;不同算法适用的基学习器不同,但总体来说最适合 Bagging 集成方法的是决策树;基于决策树的方法都一致选择是否委托律师代理作为最重要的特征;基学习器个数对不同 Bagging 算法表现的影响并不一致。

[关键词] Bagging; 保险欺诈; 极端随机树; 随机森林

[中图分类号] F84; G623 **[文献标识码]** A **[文章编号]** 1004-3306(2019)04-0066-19

DOI: 10.13497/j.cnki.is.2019.04.006

一、引言

自改革开放以来,我国保险业取得了长足的发展,据 2018 年 1 月保监会发布的保险业发展报告,我国 2017 年全年实现原保费收入 3.66 万亿,同比增长 18.16%,规模已达世界第二。而据 2018 年 7 月发布的《财富》世界 500 强企业排行榜,我国有 8 家保险企业上榜,实现了历史性的突破。但与此同时,保险欺诈的形势也日益严峻,保险欺诈案件的频率和损失也逐年上升,据《新金融》报道,截至 2018 年 11 月,人保财险 2018 年稽查到的、因保险欺诈而减损的总金额超过 80 亿,据人保财险的市场份额推算,行业总减损金额不低于 200 亿,而这只占未识别的保险欺诈案件的一小部分,据估计我国商业保险欺诈损失率在 10% 以上,车险的损失率更高达 20%(叶明华,2011),部分省份可能高达 30%(刘坤坤,2012),而国际保险监督官协会(IAIS)等国际经验数据来看,保险行业欺诈金额约占总赔付金额的 10%~20%。本文所使用的美国 Massachusetts 州 AIB 索赔数据集(Francis,2016)显示,所有索赔中,欺诈索赔的比例高达 31%。保险欺诈的情况不容乐观,为此,保监会于 2018 年 2 月专门印发了《反保险欺诈指引》以指导保险公司和保险行业进行反欺诈制度建设。保险欺诈已经对保险业的健康发展产生了巨大的影响,甚至有可能影响到我国的金融稳定。

[基金项目] 本文受到国家自然科学基金面上项目“保险公司经济资本预测与最优配置问题研究”(NO. 71573143)、“不确定全面风险分析框架下供应链风险建模与优化研究”(NO. 61673225)和中央高校基本科研业务费专项资金“随机最优控制与金融风险管理交叉研究”(NO. 63185019)的资助。

[作者简介] 李秀芳,南开大学金融学院教授、博士生导师,研究方向:精算学、保险学;黄志国,南开大学金融学院博士研究生,研究方向:风险管理、动态经济学、机器学习;陈孝伟,南开大学金融学院副教授、硕士生导师,研究方向:不确定理论、资本市场定价、动态经济学。

鉴于保险欺诈的严重影响,保险学界和实务界进行了保险识别的多种尝试,如 Probit 模型、PRIDIT 模型、专家系统等,这些方式不但效率低而且识别准确率也不高。虽然近年来有部分学者尝试将 Logistic 回归、SVM、ELM 等技术引入保险欺诈识别领域,但这些方法仅能处理低维数据,在处理大规模数据时表示能力十分有限。随着大数据时代的到来,保险数据将呈现出高维、高频、多尺度等特点,而针对这类数据,不管是人工识别还是 Logistic 回归、SVM、ELM 等技术显然都无能为力,保险业迫切需要引入新技术,而 Bagging 集成方法就是一个非常好的选择。

相比 Logistic 回归、SVM 等技术,Bagging 集成学习具有如下特点:第一,能够充分利用简单学习器的优点,易于部署;第二,可以根据数据的规模进行模型结构复杂度的调整;第三,参数空间可控,从而可以通过控制调参的粒度控制模型的计算复杂度;第四,基于决策树的实现具有一定的可解释性;第五,也是最重要的一点,它是一种并行集成方法,天然适合并行运算,从而可以充分利用大数据的分布式存储特点进行分布式训练。因此,理论上,Bagging 集成方法非常适合进行保险欺诈识别。但 Bagging 集成方法应用在保险欺诈识别问题上还有很多待解决的问题:首要的就是它在实际数据中的预测表现,机器学习方法的效果因数据不同而不同,并不存在一种通用的方法对所有数据都表现良好,Bagging 集成方法是否在实践中适合处理保险欺诈数据仍需通过实证来检验;其次,不同的 Bagging 算法适合的基学习器不同,针对保险欺诈数据,各个 Bagging 算法适合的基学习器类型目前尚无定论;再次,以决策树为基础的算法能够对特征重要度进行排序,我们可以预期不同的算法将会产生不同的排序,但不同的排序选择的最重要特征是否一致,这对于特征排序结果的可信度至关重要;最后,对于集成学习,基学习器个数是最重要的参数,针对保险欺诈数据,基学习器个数对 Bagging 集成表现的影响有怎样的影响,这决定了调参的方向和区间。本文将对这些问题展开研究,从而为 Bagging 集成方法应用于保险欺诈问题提供理论和实证支撑。

二、文献综述

在以往的研究中,学者进行保险欺诈识别多是以 Logistic 回归(或称 Logit 模型)以及一些传统机器学习方法。如:叶明华(2011)针对机动车保险欺诈问题基于 BP 神经进行了实证检验,并与基于 Logit 模型的特征工程后的实证检验进行了对比,结果表明经过 Logit 特征工程后,模型的预测表现有所上升。李聪(2015)则基于与叶明华(2011)同样的思路对健康保险欺诈识别问题进行了实证分析。刘崇(2018)采用与之类似的思路对医疗保险欺诈问题进行了研究。Stefano 和 Gisella (2001)采用模糊专家系统进行保险欺诈识别。与主流的监督方法不同,Peng et al(2006)采用了无监督的聚类方法进行保险欺诈识别。Bhowmik(2011)则采用了朴素贝叶斯和决策树方法进行保险欺诈识别。Šubelj et al(2011)利用基于社会网络的专家系统进行汽车保险欺诈识别。Kirlioglu 和 Asuk(2012)利用基于 SVM 的异常检测进行健康险的保险欺诈识别研究。

另一些文献则注重进行国际经验分析、经济学分析或者其他非机器学习方法进行研究,如:叶明华(2007)对保险欺诈的经济学理论进行了详细分析,并对决定保险欺诈识别的因子进行了分析。王碧波(2012)对保险欺诈的定义与经济学分析、历史与现状、国外反欺诈经验进行了系统性总结。李连友、林源(2011)及林源、李连友(2014)基于聚合风险模型、PSD-LDA 模型测度了新农合欺诈风险,为保险欺诈问题提供了新的思路。喻祎(2017)基于计算广义团伙的碰撞网络矩阵相似度等方法进行保险团伙欺诈识别研究。

少部分文献关注了基于集成学习的保险欺诈识别,但都是基于 Bagging 方法的直接应用,如:闫春等(2017)结合随机森林和蚁群优化算法对汽车保险欺诈识别问题进行了研究,研究发现结合蚁群优化算法的随机森林的模型表现要优于传统的随机森林算法。Li et al(2018)则采用基于主成分分析进行节点分支、基于潜在近邻分类进行投票的随机森林进行保险欺诈识别研究。近来也出现了一些采用非结构数据的深度学习方法进行保险欺诈识别的尝试,如 Wang 和 Xu(2018)。

由此可见，无论国内或国外，针对保险欺诈识别的研究还停留在传统机器学习方法的应用或延伸，或者集成学习的直接应用，或者深度学习的初步尝试，目前未有文献对 Bagging 集成方法进行全面的讨论，尤其是基学习的适用性问题和基学习器个数的影响分析。作为并行集成方法的代表，Bagging 方法具有很多天然的优势，但 Bagging 集成学习方法的保险欺诈识别效果，基学习的适用性问题，各 Bagging 集成学习算法的表现差异，基学习器个数对各个 Bagging 算法保险欺诈识别表现的影响，所有这些问题的研究目前仍未见诸文献。保险大数据时代已经到来，人工智能应用其中是大势所趋，对这些问题的讨论将为人工智能技术应用于保险大数据提供理论基础和实证支持。

三、理论基础

(一) 算法描述

在已有的文献中,学者多是直接应用 Bagging 方法的某些算法特例(如随机森林、极端随机树),但从理论模型角度来说已有文献存在如下不足:一些算法是 Bagging 某种算法在决策树实现的算法特例,已有文献并没有明确这种从属关系;Bagging 方法可与其他基学习器结合构成新的分支算法,而这需要对理论基础进行深入梳理。本文将从这些角度对 Bagging 方法进行梳理,明确算法的适用性以及从属关系,为分支算法研究提供理论基础。

Bagging 方法是并行集成的代表 ,又根据抽样策略不同分为 Bagging 算法(仅对样本集抽样) 、 Random Subspace 算法(仅对特征集抽样) 和 Random Patches 算法(既对样本集抽样也对特征集抽样) ,这三者又可以与不同学习器结合构成新的算法分支 ,而随机森林、极端随机树等则分别是它们在决策树上的算法特例。

Bagging 算法使用全部特征通过一次并行采样大量数据子集训练基学习器 ,然后通过结合策略组合各基学习器预测结果进行输出。一般而言 ,如果希望最终的集成学习器有较强的泛化能力 ,必须满足:(1) 基学习器具有尽量大的差异 (2) 基学习器应当充分训练。然而 ,在样本有限的情况下 ,这两个条件是相互矛盾的。如果要满足基学习器具有尽量大的差异 ,则样本子集必须不能重合 ,如此则可用于训练单个学习器的子集样本量必然有限 ,基学习器因而不能得到充分训练。因此必须对采样方法做出一定的妥协——使用自助采样法 ,自助采样方法能够在保证样本子集规模的同时可以保留一部分数据用于包外估计或防止过拟合 非常适合进行并行训练。Bagging 算法的一大优势在于它的计算复杂度 ,Bagging 算法的复杂度几乎与其基学习器的复杂度同阶 ,Bagging 的这种特性对于大规模集成场景来说是非常具有诱惑力的优点。

算法流程 1 Bagging 算法

```

输入: 训练集 D ,个体学习器 L ,训练轮数 T;
for t = 1 2 3 ;···;T:
    1. 对样本集进行自助采样获得样本子集 D_t;
    2. 根据样本子集对个体学习器进行训练获得学习器 h_t;
end
输出: H( x ) = arg max  $\sum_{i=1}^T \mathbf{1}( h_i( x ) = y )$ 

```

有时我们会面临数据集有限但特征集却非常丰富的情形。此时为使基学习器获得尽量大的差异，可以选择对特征集进行采样而使用全样本集进行训练。如果特征集足够大，每个基学习器所学习的特征子集的差异也会非常大，从而针对特征子集进行全样本训练也会使基学习器产生足够大的差异，进而取得令人满意的效果。

算法流程 2 Random Subspaces 算法

```
输入: 训练集 D ,个体学习器 L ,训练轮数 T 特征集 F;  
for t = 1 2 3 ,··· ,T:  
    1. 对特征集进行随机采样获得特征子集 Ft;  
    2. 根据特征子集 采用全样本对个体学习器进行训练获得学习器 ht;  
end  
输出: H( x ) = arg max  $\sum_{t=1}^T 1( h_t( x ) = y )$ 
```

极端随机树(Extremely Randomized Trees) 是 Random Subspaces 在决策树上的算法特例。它使用所有训练样本 ,但是特征尤其是分支阈值选择是随机的 ,由于分支阈值选择是决策树生长过程中最耗时的部分 ,因此引入随机阈值能够大大提高模型的训练效率。具体来说就是: 当属特征为类别时 极端随机树会随机选择一些类别的样本作为右分支 其余样本作为左分支; 当特征为数值时 随机选择介于最大值、最小值之间的某个数作为分支阈值。对于所有分支结果计算信息增益 根据信息增益选择最优划分特征 ,这相当于在一个特征子集中选择最优划分特征 ,但划分样本集仍然是全部训练样本。

某些时候 我们会遇到样本集较小而特征集也不大的情况 ,这种情况下单纯进行样本集或特征集采样都不足以获得大量有足够差异的、充分训练的基学习器 ,此时需要同时对特征集和数据集进行采样 即 Random Patches 算法。这种方式能够使基学习器获得比 Bagging 或者 Random Subspaces 更大的差异 ,从而取得理论上更好的集成效果。然而这也同时带来了一个问题: 基学习器既不能获得样本集的整体信息 ,也不能获得特征集的整体信息 ,从而不同基学习器对数据集的理解完全不同 导致基学习之间差异过于巨大而不能获得合意的效果 ,正因为如此 ,能够取得成功的 Random Patches 分支算法并不多 随机森林是其中之一。

算法流程 3 Random Patches 算法

```
输入: 训练集 D ,个体学习器 L ,训练轮数 T 特征集 F;  
for t = 1 2 3 ,··· ,T:  
    1. 对特征集进行随机采样获得特征子集 Ft;  
    2. 对样本集进行自助采样获得样本子集 Dt;  
    3. 根据特征子集和样本子集对个体学习器进行训练获得学习器 ht;  
end  
输出: H( x ) = arg max  $\sum_{t=1}^T 1( h_t( x ) = y )$ 
```

随机森林(Random Forest) 是 Random Patches 算法在决策树上一个算法特例 ,具体来说就是: 经典的决策树分支是在前节点的特征集中进行最优划分特征选择 ,而在随机森林中 ,算法首先在当前节点所有特征的集合中随机选择一定规模的一个特征子集 ,再从这个特征子集中选择最优划分特征 特征子集的规模代表随机的程度 规模越小 随机性越强 ,但随机性越强并不代表随机森林的泛化能力越强 特征子集规模存在一个最优的区间。

(二) 理论逻辑

保险欺诈识别是一种典型的分类问题 ,分类问题是 Bagging 方法最擅长的应用场景。最典型的保险欺诈识别问题是一种二分类问题 ,即将所有索赔区分为正常索赔和欺诈索赔。在粒度更细的场景中 ,可能要区分: 隐瞒真实情况欺诈; 倒签单欺诈; 无标的欺诈; 故意制造出险事故欺诈; 虚构出险事故欺诈; 夸大损失欺诈等等。但无论粒度多么细 ,保险欺诈识别问题都是一种典型的分类问题。虽然 Bagging 方法可以不经修改

地应用于分类和回归问题,但它最擅长的还是分类问题。首先,它最重要的基学习器是决策树,除 CART 可以既用于分类也用于回归外,包括 C4.5、ID3、OC1 等都适用于分类问题,而 SVM、kNN、贝叶斯分类器等基学习器都原生适用于分类问题,适用于回归问题还需要进行额外的修改。所以,Bagging 方法适合进行保险欺诈识别。

保险欺诈数据是一类典型的结构化数据,Bagging 方法适合处理结构化数据。所谓结构化数据是一种由二维表结构来逻辑表达和实现的数据,严格地遵循数据格式与长度规范,主要通过关系型数据库进行存储和管理。在处理大规模数据的问题上,学界一直有两种路线:深度学习和集成学习。深度学习技术的主要优势在于自动进行特征工程,深入挖掘数据间的隐含关系。但保险欺诈数据通常是结构化数据,具有良好的结构,不需要进行深入的特征工程,从而将主要的时间复杂度集中于模型的训练和预测中。而且,深度学习的计算十分庞大,需要强大的算力与很长的计算时间,而一些反欺诈场景对时效要求较高而且算力有限。因此,从这个角度,Bagging 方法适合进行保险欺诈识别。

基于决策树的 Bagging 方法可以为保险欺诈识别问题提供一定的解释性。多数机器学习方法虽然有较好的效果,但通常不能为待预测的问题提供合理的解释,被视为一种“黑箱”。但基于决策树的 Bagging 方法可以像通常的统计学模型一样,根据预测效果对解释变量进行重要性排序,这对于保险公司设计产品、制定营销策略、欺诈索赔预判、经营状况分析都具有重要的指导意义,因此,Bagging 方法适合进行保险欺诈识别。

四、实证分析

(一) 数据与评估标准

本文使用的数据基于美国 Massachusetts 州 AIB 索赔数据集,原数据集包含 100 多个变量,但经过 Francis(2016) 的特征工程发现,多数变量是冗余的,Francis(2016) 基于该数据集和特征工程的结果生成了一个公开的数据集用于研究目的,该数据集包括 27 个变量共 1500 个样本,该数据集与原始数据集拥有相似的变量名和相关程度,各变量的描述性统计如表 1。

变量描述性统计

表 1

变量名	mean	std	min	25%	50%	75%	max
Suspicion	0.31	0.462647	0	0	0	1	1
legalrep	0.364667	0.481497	0	0	0	1	1
sprain	0.498667	0.500165	0	0	0	1	1
chiropt	0.64	0.48016	0	0	1	1	1
emtreat	0.839333	0.367346	0	1	1	1	1
police	0.508	0.500103	0	0	1	1	1
prior	0.156	0.362976	0	0	0	0	1
NumProv	3.29	1.940896	0	2	3	4	10
NumTreat	4.62	2.970547	0	3	4	6	19
RptLag	6.881333	11.89438	0	2	3	8	182
TrtLag	5.064	11.59685	-1	-1	1	6	143
PolLag	199.5067	124.8646	3	107	173	262.25	749
Thresh	0.62	0.485548	0	0	1	1	1
Fault	34.80333	26.59148	0	11	33	54	100

(续表)

变量名	mean	std	min	25%	50%	75%	max
ambulcost	187.6392	188.9836	0	0	171.4215	285.1294	1643.686
Inj01	0.468667	0.499184	0	0	0	1	1
Inj02	0.444667	0.497095	0	0	0	1	1
Inj06	0.848667	0.358493	0	1	1	1	1
Ins06	0.934	0.248365	0	1	1	1	1
Acc04	0.821333	0.383201	0	1	1	1	1
Acc09	0.013333	0.114736	0	0	0	0	1
Acc10	0.890667	0.738744	0	0	1	1	2
Acc15	1.636	0.693656	0	2	2	2	2
Acc19	0.979333	0.142313	0	1	1	1	1
Clt07	1.165333	0.612847	0	1	1	2	2
HospitalBill	207.5104	435.5749	0	0	0	133.0578	3389.101

由 Suspicious 项可以看出欺诈嫌疑索赔已经达到了整体索赔的 31% ,这个数字对保险公司而言是十分巨大的损失 ,所以保险公司进行欺诈识别的研究是极其有必要的 ,欺诈识别是风控的重要一环 ,应当引起保险公司足够的重视。

为充分评估模型的表现 本文选择 Accuracy、Precision、Recall、F1、AUC 作为模型评估标准 ,其中 Accuracy、Precision、Recall、F1 为模型预测表现评估标准 ,AUC 为训练表现评估标准 ,并绘制 Precision – Recall 曲线、ROC 曲线作为训练的图像评估标准。同时记录运行时间以作为时间复杂度的测度。

(二) 基学习器实证分析

无论是 Bagging、Random Subspace 还是 Random Patches 都能够与不同的基学习器构成新的算法分支 ,本文致力于 Bagging 方法的详尽讨论 ,为此选择了一些有代表性的基学习器进行研究 ,其中 Logistic 回归是广义线性模型的代表 ,SVM 是非线性方法的代表 ,决策树是非基于距离方法的代表 ,kNN 是基于样本学习的代表(其特点是没有训练过程) 朴素贝叶斯(包括伯努利贝叶斯分类器 BNB、多项式贝叶斯分类器 MNB、高斯贝叶斯分类器 GNB) 是生成式方法的代表。为研究 Bagging 集成学习的表现 本文首先对基学习的表现进行了实证分析 ,一则作为前期的预备知识 ,二则可以为后续的集成表现提供一个标准 ,三则基于对基学习器超参数的寻优 ,避免后续在集成学习超参数寻优的同时对个体学习进行寻优 ,进而大大降低训练时间。

各基学习器得分及运行时间

表 2

	Accuracy	Precision	Recall	F1	AUC	Time(s)
Logistic	0.97	0.96667	0.93548	0.95082	0.98577	2.9302
SVM	0.97333	0.96667	0.94565	0.95604	0.99034	3.69887
决策树	0.93	0.88889	0.87912	0.88398	0.92009	0.03801
kNN	0.97	0.94444	0.95506	0.94972	0.98307	8.89664
BNB	0.90333	0.93333	0.78505	0.85279	0.94399	0.17362
MNB	0.90333	0.93333	0.78505	0.85279	0.94628	0.14595
GNB	0.89	0.93333	0.75676	0.83582	0.93527	0.00039

注: 黑体为最优得分 由于本文记录调节参数、训练、预测的整体时间 而 GNB 由于没有可调节参数故而速度很快 ,若其他学习器省略调优 时间与 GNB 相差无几。

表2中依次列示了针对保险欺诈数据,各基学习器的得分和运行时间。整体来看,表现最好的算法是SVM,运行时间最短的是决策树。我们可以预期Bagging集成方法对Logistic回归、SVM、kNN算法表现的提升可能有限,而对决策树、BNB、MNB、GNB算法的表现会有较好的提升。因此总体上,对于小规模保险欺诈数据,从识别效果的角度来看,SVM是最优选择,从时间复杂度角度来看,决策树是最优选择。SVM、决策树的Precision – Recall曲线和ROC如图1、图2所示。

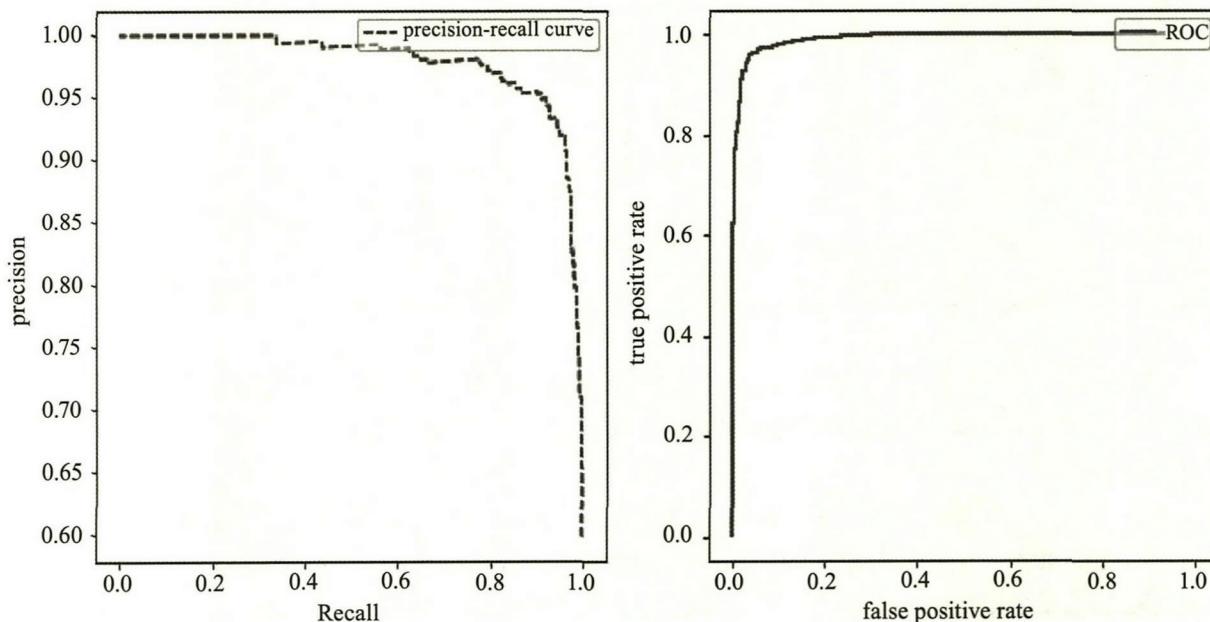


图1 SVM 的 Precision – Recall 曲线和 ROC

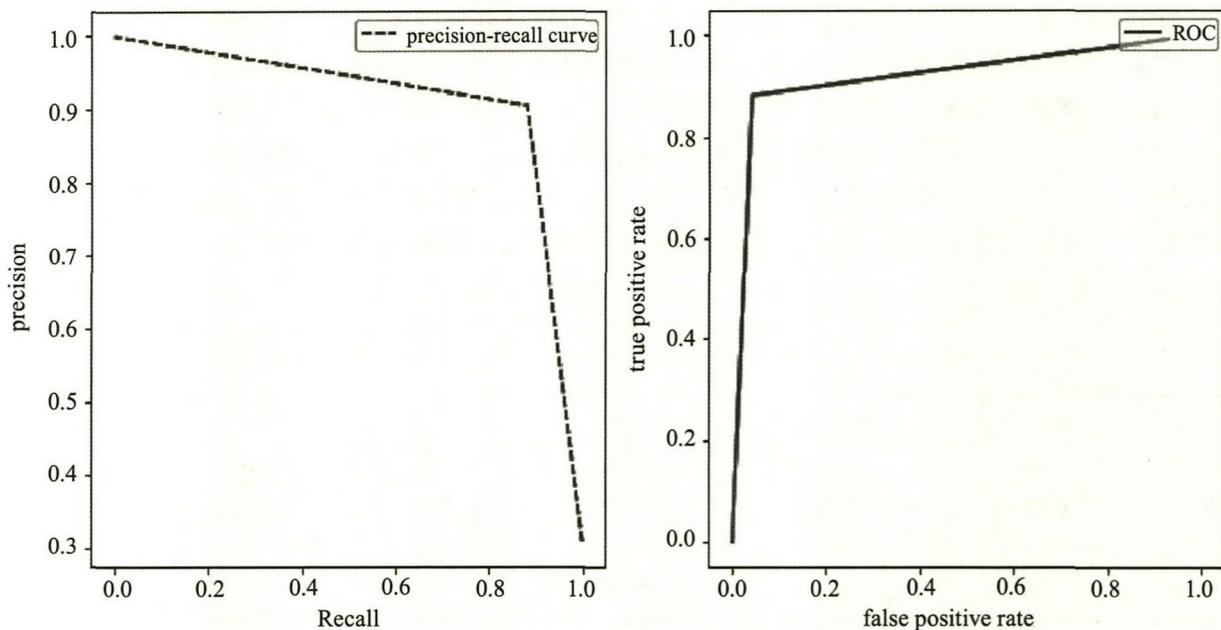


图2 决策树的 Precision – Recall 曲线和 ROC

决策树方法可以基于预测表现计算数据特征的重要程度,据此本文绘制特征相对重要性图如图3。由图3可以看出,对保险欺诈识别最重要的指标是legalrep,即是否聘请律师是判断保险索赔是否为欺诈索赔的重要指标。

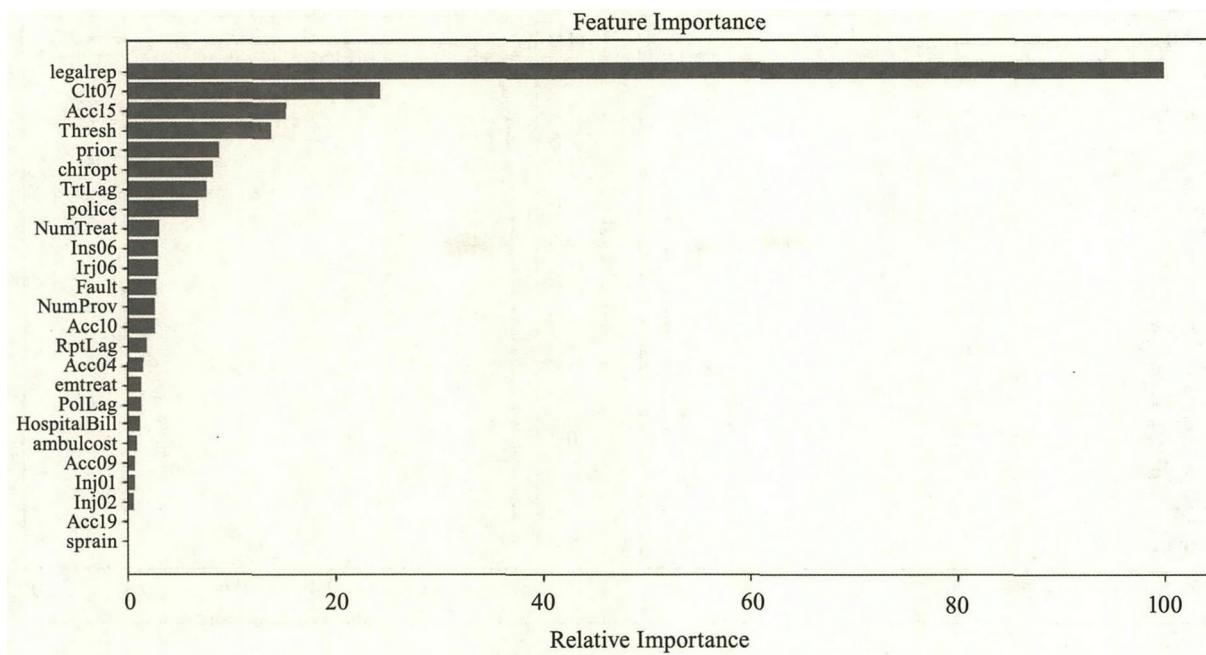


图 3 决策树分类器的特征相对重要性

(三) Bagging 集成各算法的模型表现

基学习器和 Bagging 集成各算法的结合,可构成多个分支算法,包括 Bagging – 基学习器算法分支、Random Subspace – 基学习器算法分支、Random Patches – 基学习器算法分支,以及极端随机树、随机森林等算法。其中 Bagging 算法各分支算法的保险欺诈识别效果如表 3。

Bagging 算法各分支算法的得分与运行时间

表 3

	Accuracy	Precision	Recall	F1	AUC	Time(s)
Logistic	0.97	0.96667	0.93548	0.95082	0.98571	239.034
SVM	0.97	0.96667	0.93548	0.95082	0.99023	297.453
决策树	0.97333	0.96667	0.94565	0.95604	0.98682	33.7423
kNN	0.96333	0.92222	0.95402	0.93785	0.99114	730.006
BNB	0.90333	0.93333	0.78505	0.85279	0.94519	60.6950
MNB	0.90333	0.93333	0.78505	0.85279	0.94628	33.5604
GNB	0.89	0.93333	0.75676	0.83582	0.93527	48.5629

总体来看,针对保险欺诈数据,除决策树外,Bagging 对基学习器表现的改善十分有限。Bagging 各分支算法中,表现最好、提升最大的是 Bagging – 决策树分支算法,运行时间最短的是 Bagging – MNB 分支算法,但也仅比 Bagging – 决策树稍短,因此表现最好的还是 Bagging – 决策树,之后的超参数分析中将以 Bagging – 决策树作为 Bagging 算法的代表。

Random Subspace 算法是针对属性采样而是用全样本进行训练的并行集成方法,它同样可以与不同基学习器结合构成新的分支算法,各分支算法针对保险欺诈数据的表现如表 4。

由表 4 可以看出,总体来看,针对保险欺诈数据,表现最好的是 Random Subspace – SVM 分支算法,提升最大的是 Random Subspace – BNB、Random Subspace – MNB,对其他基学习器,Random Subspace 未能明显提升其预测表现。由于极端随机树是 Random Subspace 在决策树上的一个算法特例,从多样性和 Accuracy 提升程度的角度,本文之后选择 Random Subspace – BNB 作为 Random Subspace 分支算法的代表。

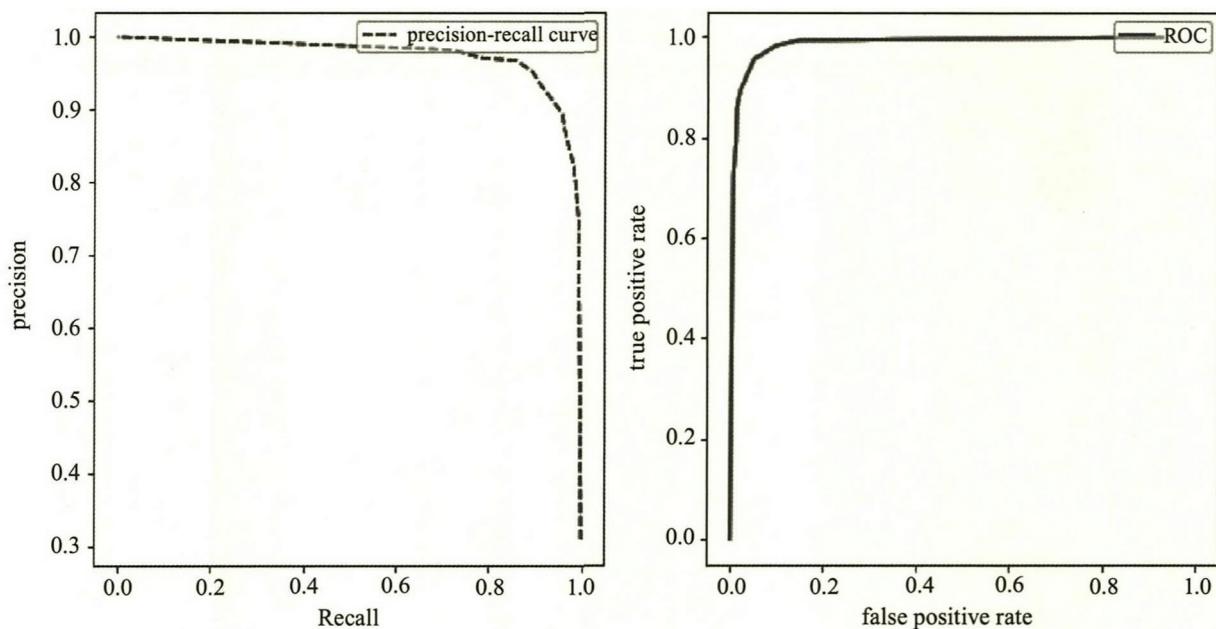


图4 Bagging – 决策树的 Precision – Recall 曲线和 ROC

Random Subspace 各分支算法的得分和运行时间

表4

	Accuracy	Precision	Recall	F1	AUC	Time(s)
Logistic	0.97	0.96667	0.93548	0.95082	0.98571	239. 034
SVM	0.97333	0.96667	0.94565	0.95604	0.98991	562. 083
决策树	0.97	0.94444	0.95506	0.94972	0.98423	40. 4133
kNN	0.97	0.94444	0.95506	0.94972	0.99104	623. 676
BNB	0.92667	0.91111	0.85417	0.88172	0.94437	47. 3980
MNB	0.92	0.82222	0.90244	0.86047	0.94708	30. 50501
GNB	0.89333	0.86667	0.79592	0.82978	0.94106	38. 3646

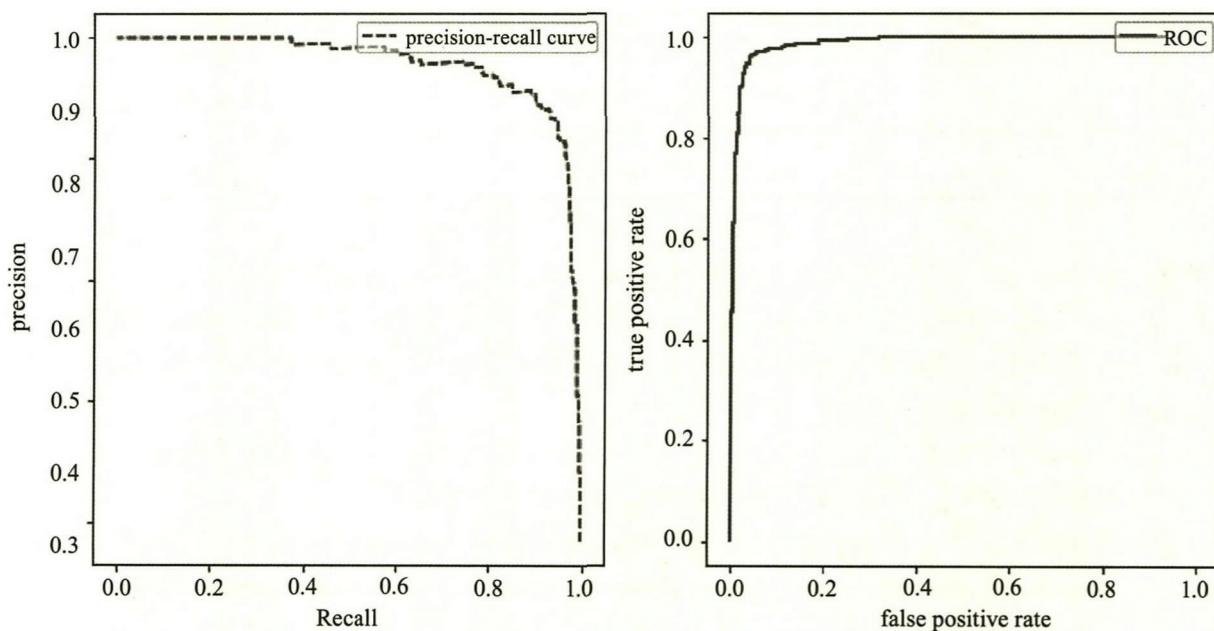


图5 Random Subspace – SVM 的 Precision – Recall 曲线和 ROC

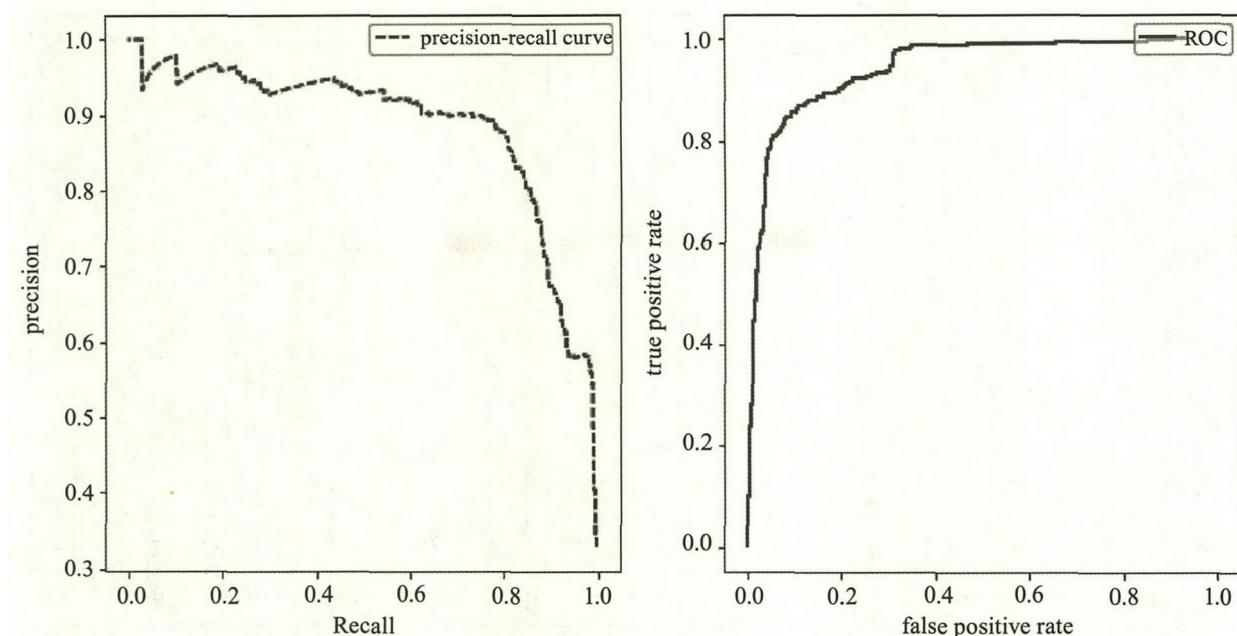


图 6 Random Subspace – BNB 的 Precision – Recall 曲线和 ROC

RandomPatches 是既对属性采样又对样本集采样的并行集成方法 ,它同样可以与不同基学习器结合构成新的分支算法 ,各分支算法针对保险欺诈数据的表现如表 5。

Random Patches 各分支算法的得分和运行时间

表 5

	Accuracy	Precision	Recall	F1	AUC	Time(s)
Logistic	0.97	0.96667	0.93548	0.95082	0.98572	1750. 25
SVM	0.97333	0.96667	0.94565	0.95604	0.99031	1917. 03
决策树	0.98	0.94444	0.98837	0.96591	0.98178	275. 118
kNN	0.97333	0.95556	0.95556	0.95556	0.99017	3253. 67
BNB	0.85667	0.9	0.70435	0.79024	0.94688	415. 384
MNB	0.93	0.92222	0.85567	0.88770	0.94634	262. 394
GNB	0.88333	0.76667	0.83133	0.79769	0.93803	330. 126

由表 5 可以看出 ,针对保险欺诈数据 ,表现最好的 RandomPatches 分支算法是 RandomPatches – 决策树 ,提升最大的是 RandomPatches – MNB ,对其他基学习器的表现则提升十分有限。 RandomPatches – 决策树与随机森林算法十分相似 ,因此从多样性角度 ,在之后本文将以 RandomPatches – MNB 作为 RandomPatches 分支算法的代表。

极端随机数是 RandomSubspace 算法针对决策树的一个算法特例 ,因此也属于 Bagging 方法 ,其针对保险欺诈数据的表现如表 6、图 9 所示。

极端随机树的得分和运行时间

表 6

Accuracy	Precision	Recall	F1	AUC	Time(s)
0.97	0.95556	0.94505	0.95028	0.98754	592. 277

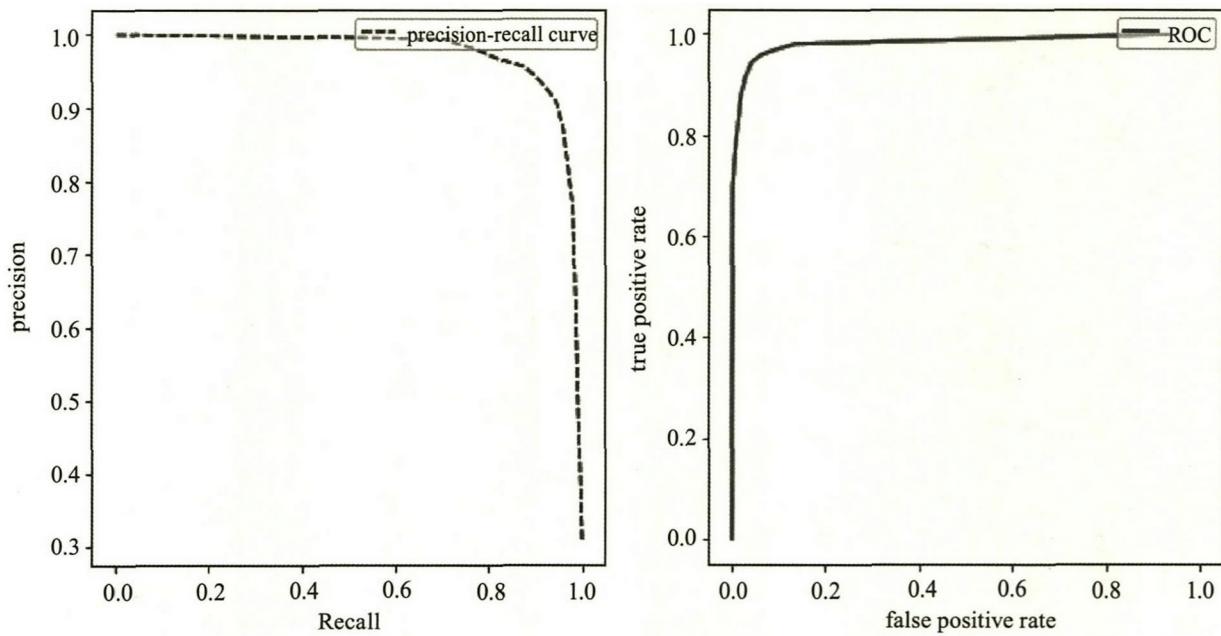


图 7 Random Patches – 决策树的 Precision – Recall 曲线和 ROC

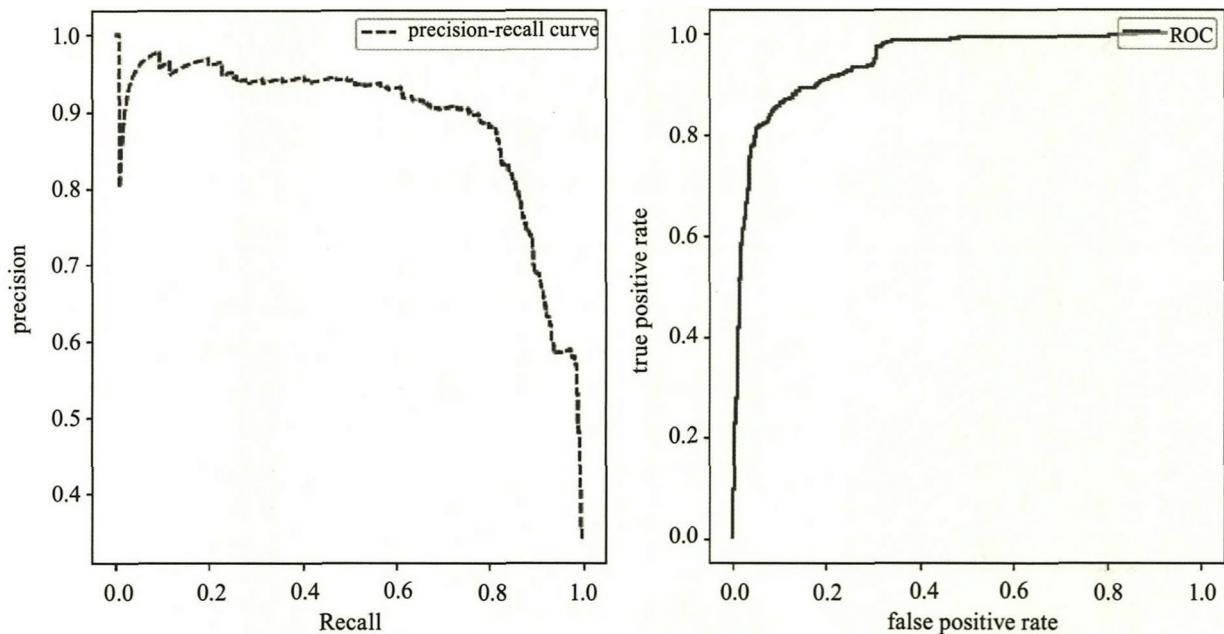


图 8 Random Patches – BNB 的 Precision – Recall 曲线和 ROC

由极端随机树的表现可以看出，无论是训练表现还是预测表现，极端随机树的各项指标都达到了较高的水平。而由图 10 的特征相对重要性排序可以看出，极端随机树所选定的特征重要度与决策树有一定程度的相似，它们选定的最重要特征都是 legalrep。

随机森林是 RandomPatches 算法针对决策树的一个算法特例，因此也属于 Bagging 方法，其针对保险欺诈数据的表现如表 7、图 11 所示。

随机森林的得分和运行时间

表 7

Accuracy	Precision	Recall	F1	AUC	Time(s)
0.97333	0.94444	0.96591	0.95506	0.98860	594.043

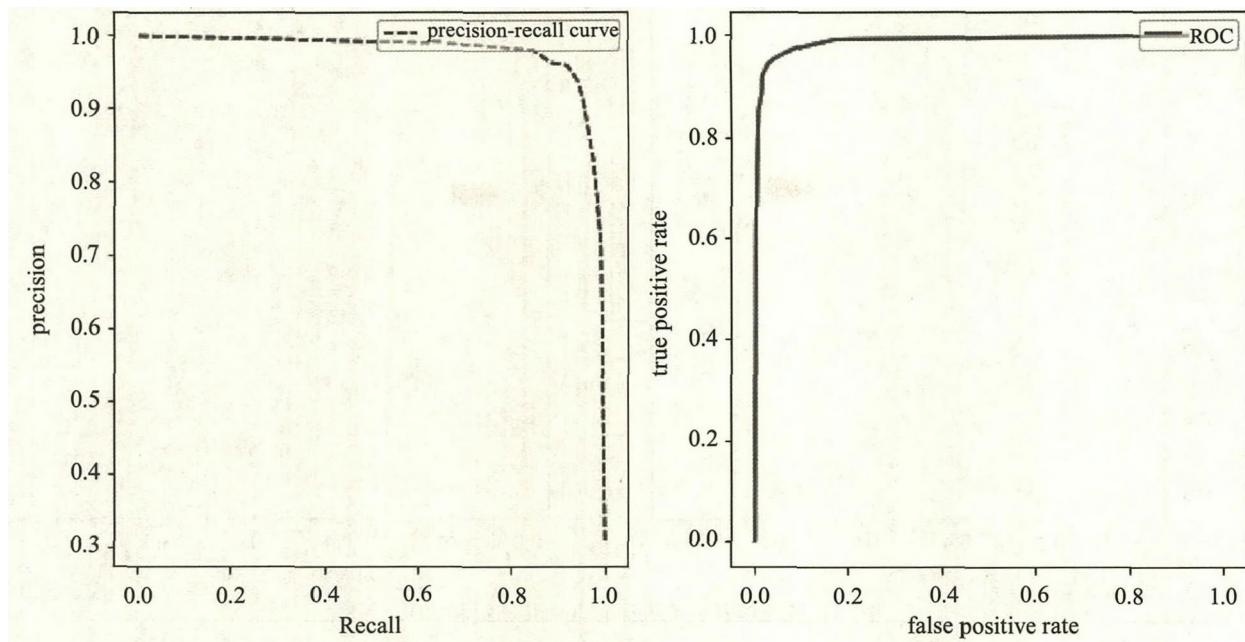


图9 极端随机树的 Precision – Recall 曲线和 ROC

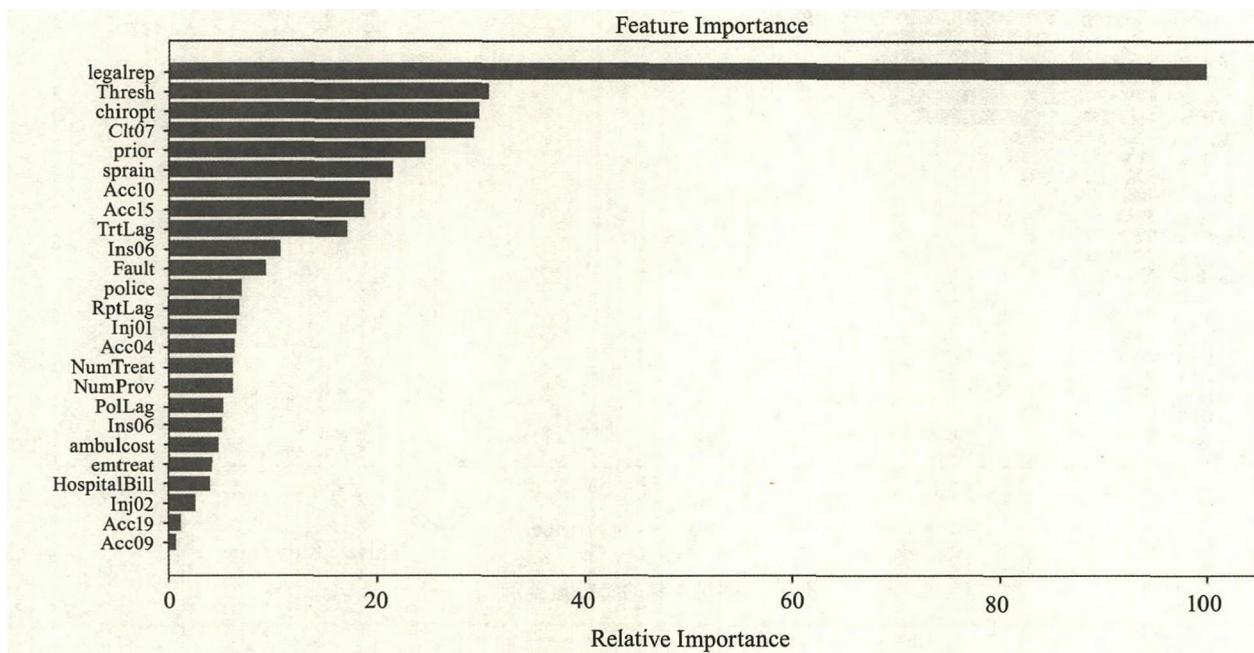


图10 极端随机树的特征重要度

由随机森林的表现看来,无论从训练表现还是预测表现,随机森林与极端随机树的表现十分接近。从图12的特征重要度来看,随机森林与决策树、极端随机树选定的最重要特征是一致的,即也选择 legalrep 作为最重要的特征。

根据对 Bagging 分支算法的分析可以看出,Bagging 各算法表现最好的分支算法分别是 Bagging – 决策树、Random Subspace – SVM、Random Patches – 决策树,而提升最大的分别是 Subspace – BNB、Random Patches – MNB。因此以下将表现最好、提升最大的算法分支与极端随机树、随机森林一起列示如下,但分析将分开进行,Bagging 各算法保险欺诈识别效果如表 8 所示。

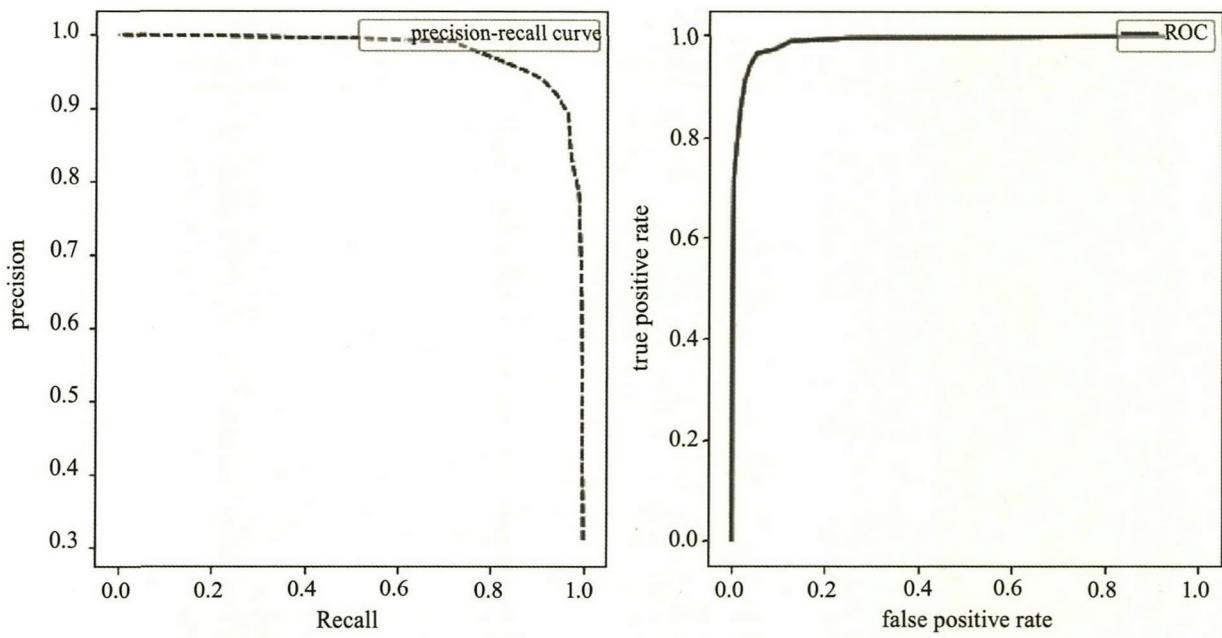


图 11 随机森林的 Precision – Recall 曲线和 ROC

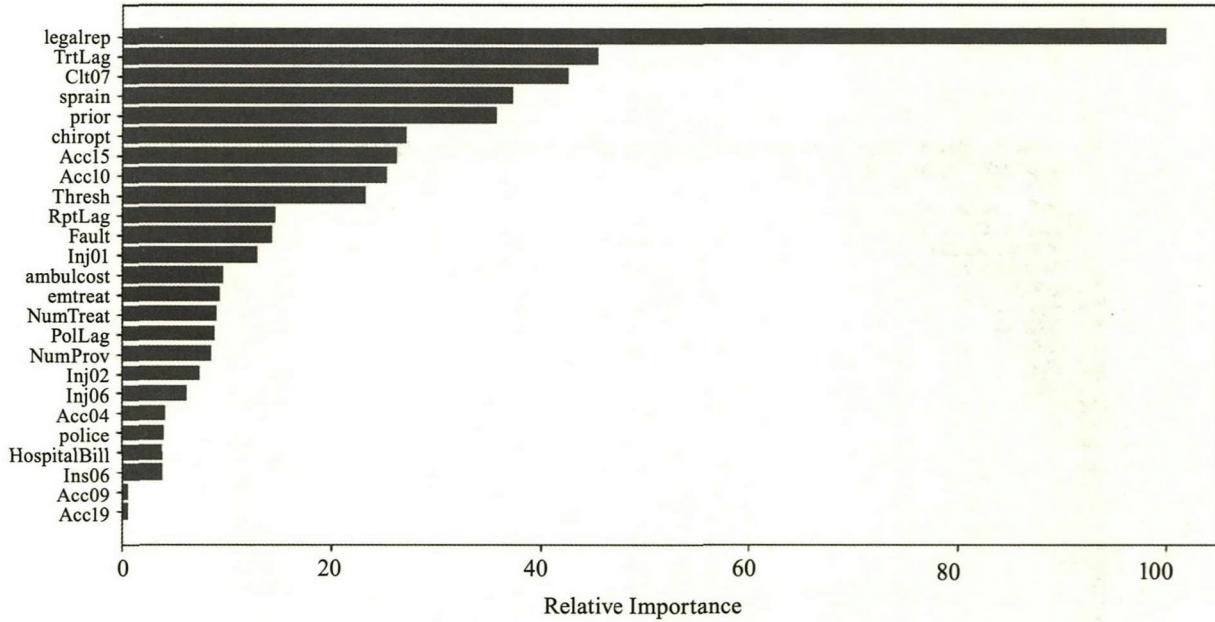


图 12 随机森林的特征重要度

由表 8 可以看出,如果不考虑 Random Subspace – 决策树、Random Patches – 决策树,随机森林和 Bagging – 决策树的 Accuracy 最高, Bagging – 决策树的 Precision 最高, 随机森林的 Recall、F1、AUC 最高, Bagging – 决策树的运行时间最短。若考虑 Random Subspace – 决策树、Random Patches – 决策树, 则 Random Patches – 决策树的 Accuracy 最高, Bagging – 决策树的 Precision 也是最高, Random Patches – 决策树的 Recall、F1 最高, 随机森林的 AUC 也是最高的, Bagging – 决策树的运行时间是最短的。由于随机森林也是 Random Patches 的一个基于决策树的算法特例, 因此在 Bagging、Random Subspace、Random Patches 三者之中, Random Patches 算法的得分最高, Bagging 的运行时间最短, 最适合 Bagging 集成方法的基学习器是决策树。

(四) 稳健性分析

从基学习器提升、抽样方法差异、样本选择和样本规模四个方面来看, Bagging 方法是一种稳健的欺诈识

别方法。本文通过同时在两种设备多次运行、使用交叉验证法、不设随机种子、使用不同的样本规模等多个方面验证了 Bagging 方法的稳健性。从基学习器提升的角度来看，本文在两种设备、多次对 Bagging 方法的保险欺诈识别效果进行检验，检验结果表明，对于已经达到性能上限的基学习器，Bagging 方法没有降低其预测表现；对于表现欠佳的基学习器，Bagging 方法可以提升其预测表现。即 Bagging 方法可以提升或保持（而不会降低）基学习器的预测表现。因此，从这个角度，Bagging 方法是一种稳健的保险欺诈识别方法。从抽样方法的角度来看，本文使用了 Bagging、Random Subspace、Random Patches 等抽样策略，但对于同一种基学习器，Bagging 方法的不同抽样策略对其保险欺诈识别效果的影响差异十分微小，因此，从抽样方法的角度，Bagging 方法是一种稳健的保险欺诈识别方法。从样本选择和样本规模角度来看，在进行 Accuracy、Precision、Recall、F1 的计算中，本文使用了 3 折交叉验证法，在两个设备多次进行模型的训练与预测，而每次都没有设定随机种子，因此每次抽样的样本都是不同的，但模型的结果表明，在多次预测中，模型的表现都非常稳定，各个得分几乎没有变化。而在进行 AUC 计算时，本文使用全样本进行估计，以评测模型的训练表现。在多次训练中，从 AUC 来看模型的训练表现也十分稳定。因此从样本选择和样本规模的角度，Bagging 方法是一种稳健的保险欺诈识别方法。

Bagging 各算法保险欺诈识别效果

表 8

	Accuracy	Precision	Recall	F1	AUC	Time(s)
Bagging – 决策树	0.97333	0.96667	0.94565	0.95604	0.98682	33.7423
Subspace – SVM	0.97333	0.96667	0.94565	0.95604	0.98911	562.083
Patches – 决策树	0.98	0.94444	0.98837	0.96591	0.98178	275.118
Subspace – BNB	0.92667	0.91111	0.85417	0.88172	0.94437	47.398
Patches – MNB	0.93	0.92222	0.85567	0.8877	0.94634	262.394
极端随机树	0.97	0.95556	0.94505	0.95028	0.98754	592.277
随机森林	0.97333	0.94444	0.96591	0.95506	0.9886	594.043

注：加粗表示含 Random Subspace – SVM、Random Patches – 决策树中最优的，斜体表示不含 Random Subspace – SVM、Random Patches – 决策树中表现最优的，加粗斜体表示两种情况下都是最优的。

限于保险欺诈数据的可得性，本文无力更换数据集对模型的稳健性进行检验，因此无法从数据集的角度分析 Bagging 方法在不同数据集上的稳健性。虽然我们可以推测相同的模型在相似的数据集上应当有相似的表现，但实际的表现仍需通过实证来检验，这需要保险实务界开放更多的数据以进行理论探索。

五、基学习器个数对算法表现的影响分析

Bagging 方法最重要的超参数莫过于基学习器个数，它对 Bagging 集成方法的模型表现有至关重要的影响，对基学习器个数的影响分析一方面使我们得以了解其对模型泛化能力的影响，一方面也为我们的调参指明了方向，从而为未来的大规模应用提高调参效率。

(一) 基学习器个数对 Bagging 算法保险欺诈识别效果的影响

由于 Bagging 算法提升最大的基学习器是决策树，因此本文以 Bagging – 决策树代表 Bagging 算法。

由图 13 可以看出，针对保险欺诈数据，基学习个数对 Bagging 算法的各项得分没有明显的趋势性影响。针对 Bagging 算法的基学习个数寻优只能由格点搜索或随机搜索获得，而在精度要求不高的情况下，可以简单地选择较少的基学习个数以控制模型的时间复杂度。

(二) 基学习器个数对 Random Subspace 算法保险欺诈识别效果的影响

由于 RandomSubspace 提升最大的基学习器是 BNB，因此本文以 RandomSubspace – BNB 分支算法代表 RandomSubspace 算法。

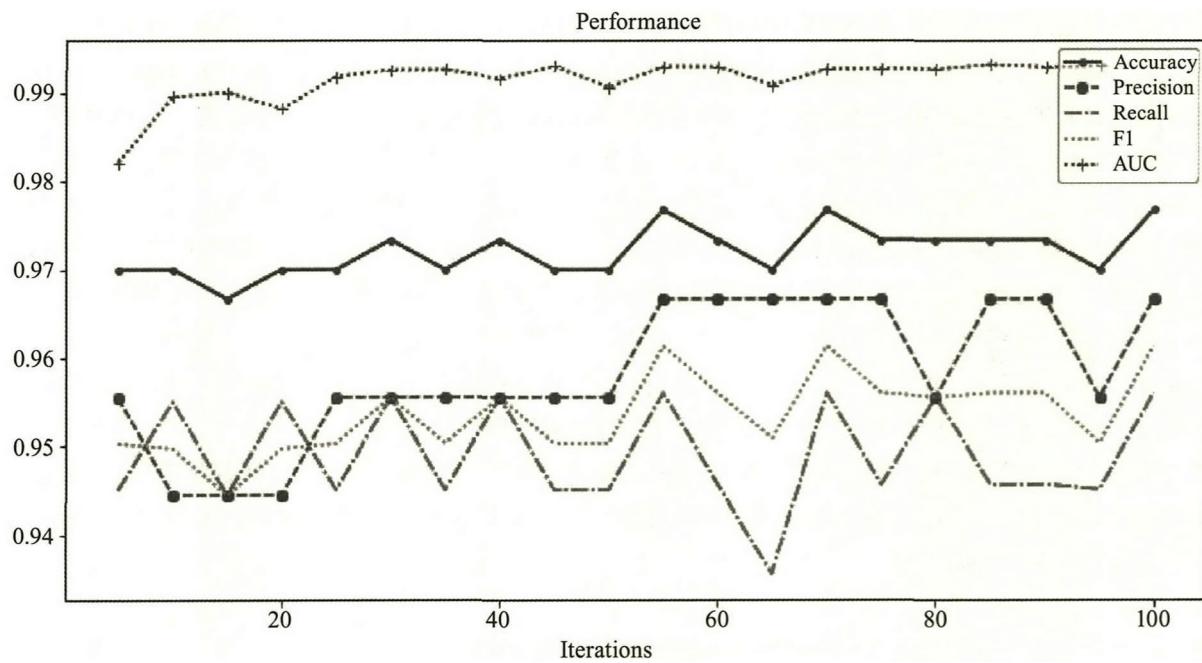


图 13 基学习器个数对 Bagging 算法保险欺诈识别效果的影响

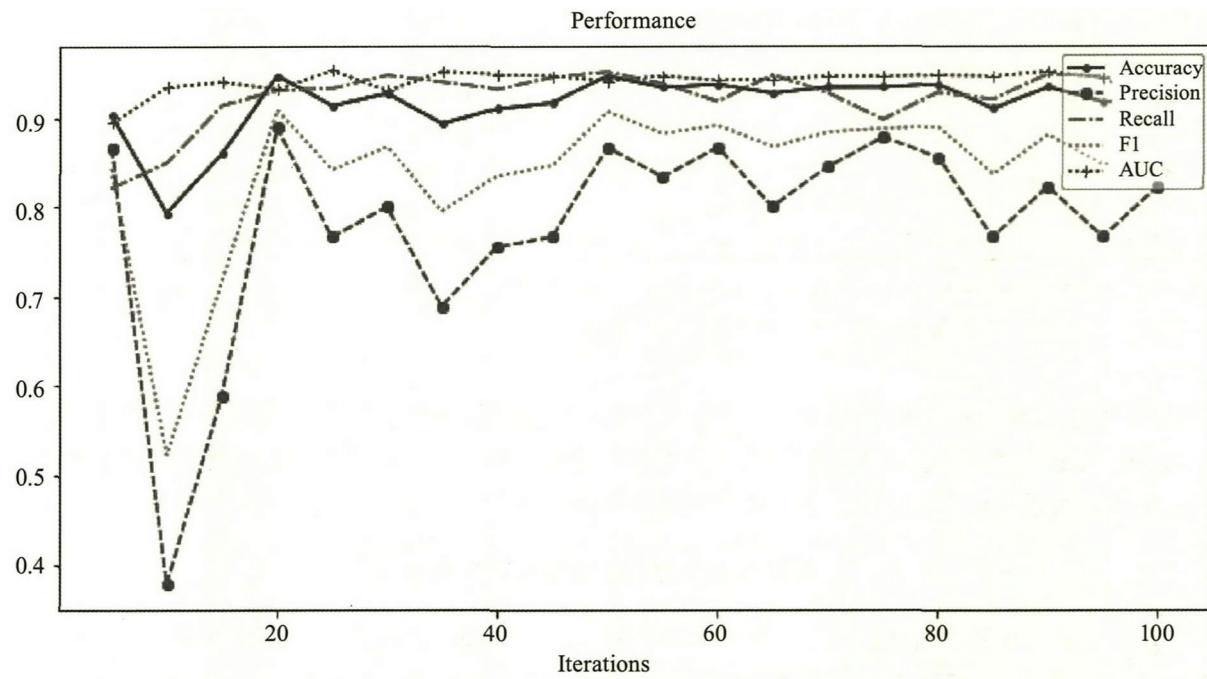


图 14 基学习器个数对 Random Subspace 算法保险欺诈识别效果的影响

由图 14 可以看出,基学习器个数对 RandomSubspace 算法各项得分的影响主要集中在基学习器个数较少的区间,在基学习器个数较少的区间,RandomSubspace 算法的 Precision、F1、Accuracy 变化非常剧烈,Recall 的变化较不明显,AUC 则几乎不受影响。因此,针对 RandomSubspace 的寻优应当也应当在基学习器个数较少的区间进行。

(三) 基学习器个数对 Random Patches 算法保险欺诈识别效果的影响

由图 15 可以看出,与 RandomSubspace 算法类似,基学习器个数对 RandomPatches 算法各项得分的影响主要集中在基学习器个数较少的区间,在基学习器个数较少的区间,除 AUC 外,其他各项得分受其影响十分

剧烈。因此,针对 Random Patches 的寻优应当也应当在基学习器个数较少的区间进行。

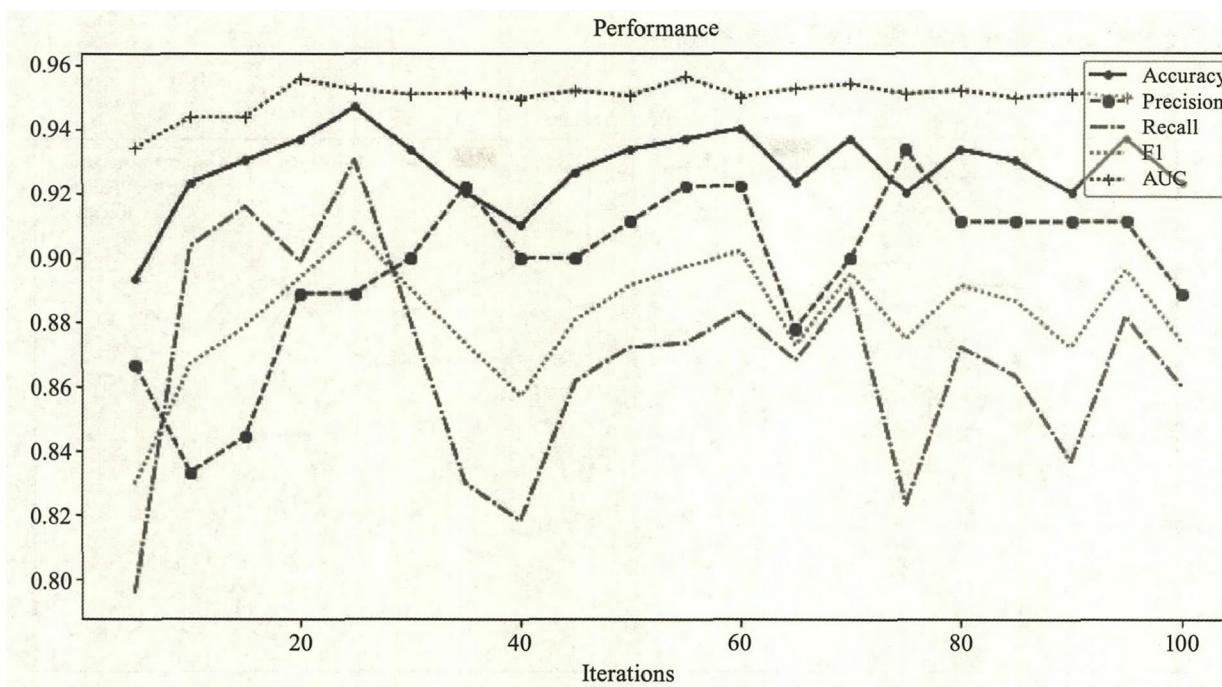


图 15 基学习器个数对 Random Patches 算法保险欺诈识别效果的影响

(四) 基学习器个数对极端随机树算法保险欺诈识别效果的影响

由图 16 可以看出,基学习个数仅对 AUC 有一个先上升后稳定的趋势性影响,而对 Accuracy、Precision、Recall、F1 则没有趋势性影响,所以整体来看,针对极端随机树算法的基学习个数寻优需要在个数较多的区间由格点搜索或随机搜索获得。

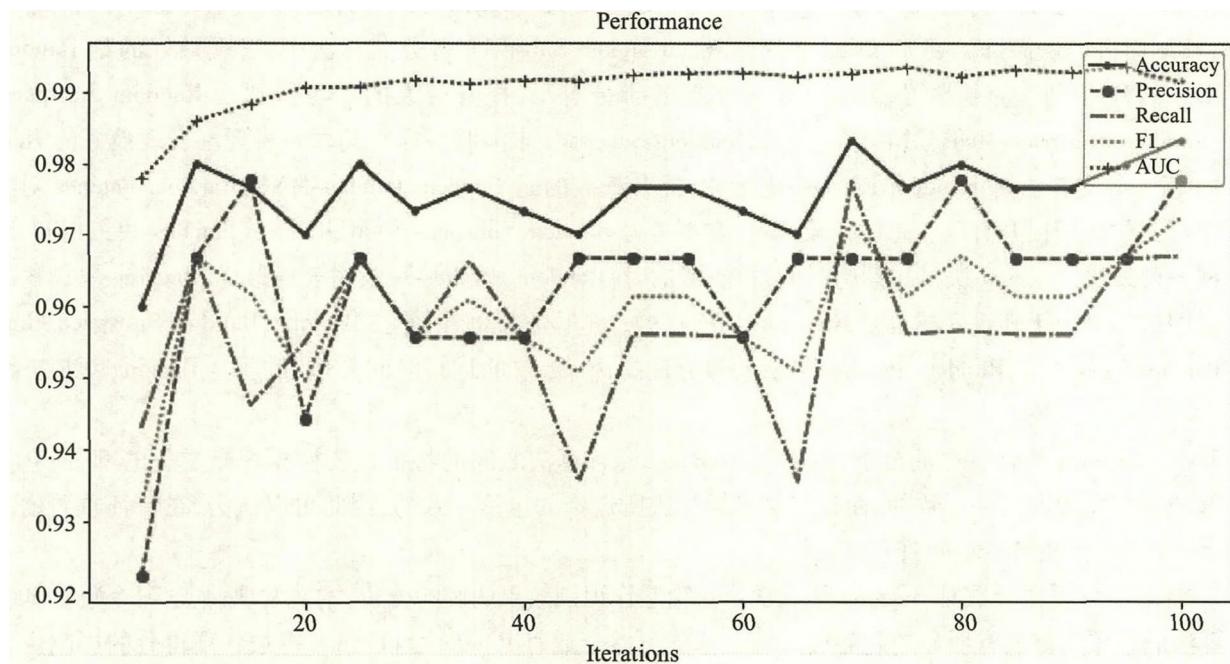


图 16 基学习器个数对极端随机树算法保险欺诈识别效果的影响

(五) 基学习器个数对随机森林算法保险欺诈识别效果的影响

由图 17 可以看出，基学习个数对随机森林算法的各项得分没有明显的趋势性影响，因此，针对随机森林的基学习个数寻优只能由格点搜索或随机搜索进行。

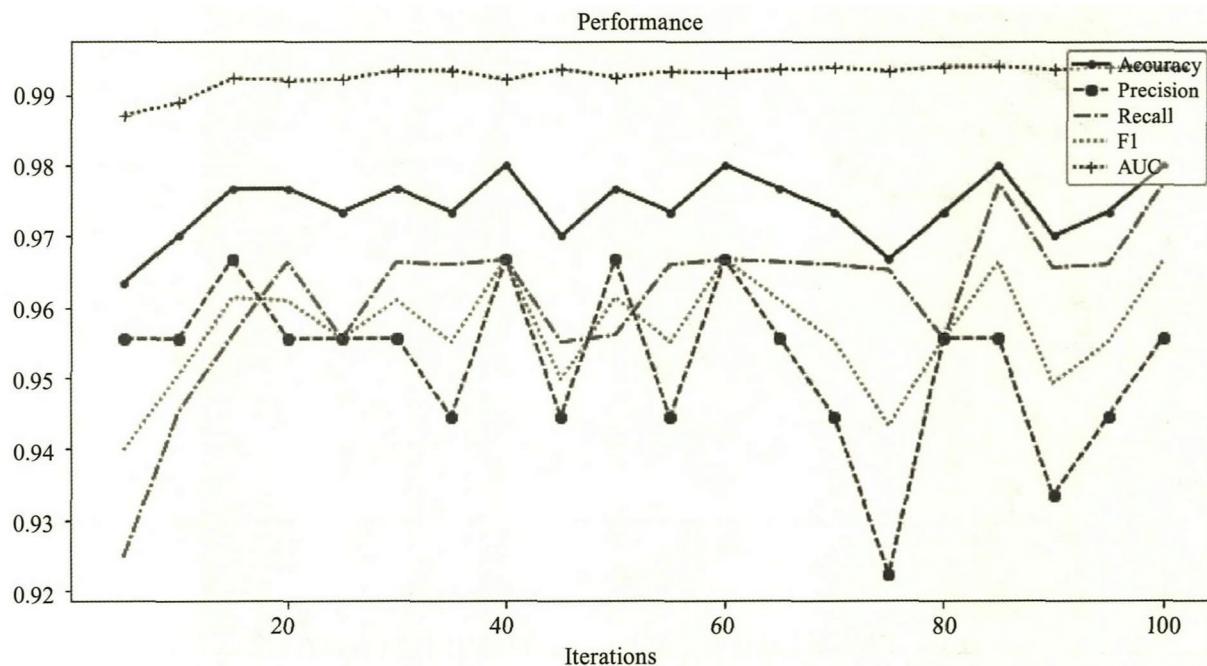


图 17 基学习器个数对随机森林算法保险欺诈识别效果的影响

六、结论与建议

根据以上分析，本文的主要结论如下：

1. Bagging 算法各分支算法中，表现最好、提升最大的是 Bagging – 决策树，运行时间最短的是 Bagging – MNB，整体而言表现最好的还是 Bagging – 决策树。Random Subspace 各分支算法中，表现最好的是 Random Subspace – SVM，但与作为基学习器的 SVM 相比提升十分有限，提升最大的学习器则是 Random Subspace – BNB、Random Subspace – MNB，对其他学习器，Random Subspace 未能明显提升其预测表现。表现最好的 Random Patches 分支算法是 Random Patches – 决策树，提升最大的是 Random Patches – MNB，Random Patches 对其他基学习器的表现提升有限。总体比较来看，在不考虑 Random Subspace – SVM、Random Patches – 决策树时，随机森林表现最优，Bagging – 决策树运行时间最短；考虑时，Random Patches – 决策树表现最优，Bagging – 决策树的运行时间最短。由于随机森林也是 Random Patches 的一个算法特例，因此在 Bagging、Random Subspace、Random Patches 三者之中，Random Patches 算法的得分最高，Bagging 的运行时间最短，最适合 Bagging 集成方法的基学习器是决策树。

2. 针对特征重要性排序的分析，本文发现：Bagging 各算法之间的特征重要性排序有重叠的部分，它们所选择的最重要的特征是一致的，而且与决策树算法所选择的也是一致的，因此可以认为，在特征重要度的选择上 Bagging 方法各算法间具有一致性。

3. 针对基学习器个数对 Bagging 集成学习保险欺诈识别效果的影响分析，本文发现：基学习个数对 Bagging 算法的各项得分没有明显的趋势性影响。基学习器个数对 Random Subspace 算法各项得分的影响主要集中在基学习器个数较少的区间，针对 Random Subspace 的寻优应当在基学习器个数较少的区间进行。基学习器个数对算法各项得分的影响主要集中在基学习器个数较少的区间，针对 Random Patches 的寻优应当

在基学习器个数较少的区间进行。基学习个数对极端随机树多数得分没有趋势性影响，基学习个数对随机森林算法的各项得分也没有明显的趋势性影响。

本文的研究同时可以得到如下结论：

1. 机器学习技术对识别保险欺诈有良好的效果。无论是作为基学习器的机器学习方法，还是 Bagging 方法，其识别保险欺诈的准确率都在相当的水平上。这是传统识别方法和人工识别方法无法比拟的。虽然保险欺诈的最终识别仍需要人工复检，但前期高识别率的筛查可以为此节约大量的人力成本和时间成本。因此保险公司可尽快着手应用机器学习技术进行保险欺诈识别。

2. 应当针对不同的识别场景采用不同的机器学习方法。一些识别保险欺诈的应用场景可能对精度要求高而对时间要求不高，另一些场景可能恰恰相反。不同的机器学习方法有这不同的识别精度和不同的识别时间，尤其是识别时间，其差异可能达到数十倍之多。不同的反欺诈场景需要对此予以取舍。

3. 基于决策树的机器学习方法在保险实务中具有更好的应用前景。不同于其他机器学习技术，基于决策树的机器学习方法不但识别效率高，而且能够提供一定的解释性，对被解释变量的重要性排序不仅可以为核保部门提供人工复查的理论依据，而且能够为销售人员提供营销支持，甚至为产品部门提供产品设计和定价支持，具有更丰富的应用场景。

可以预见，机器学习技术在保险实务中将有非常广阔的应用前景，在大数据时代充分运用机器学习技术，保险公司应当着手推进以下工作：首先，建设保险欺诈大数据工程，从而为保险大数据研究和应用提供足够的数据。然而数据的搜集和整理极为耗时，单个保险公司去建设是不现实的，而且保险欺诈是整个保险行业的问题，因此保险公司间应当共享欺诈数据，合力建设保险欺诈大数据工程。其次，在建设保险欺诈大数据工程过程中，保险业还应当着力建设保险欺诈特征工程，通过合理筛选重要特征，既能实现搜集整理数据效率的提升，同时也能实现算法运行速度的提升。最后，依托数据工程和特征工程，保险业可以合力构建和维护基于集成学习的反欺诈系统，不仅仅进行事后的欺诈识别，还要进行事前的欺诈预警，从而为保险公司的稳健可持续经营提供助力。

[参考文献]

- [1] 李聪. 中国健康保险欺诈的理论分析与实证研究 [D]. 2015 , 青岛大学 , 第 144 页.
- [2] 李连友, 林 源. 新型农村合作医疗保险欺诈风险度量实证研究 [J]. 中国软科学 , 2011 , (09) : 84 - 93.
- [3] 林 源, 李连友. 基于 PSD - LDA 模型的新农合欺诈风险测度实证研究. 财经理论与实践 [J]. 2014 , (05) : 18 - 23.
- [4] 刘 崇, 祝锡永. 基于 BP 神经网络的医保欺诈识别 [J]. 计算机系统应用 , 2018(06) : 34 - 39.
- [5] 刘坤坤. 车险保险欺诈识别和测量模型实证研究——基于广东省车险历史索赔数据 [J]. 暨南学报(哲学社会科学版) , 2012 , 34(08) : 89 - 93.
- [6] 王碧波. 反保险欺诈之对策与机制研究 [D]. 2012 , 南开大学 , 第 147 页.
- [7] 闫 春, 李亚琪, 孙海棠. 基于蚁群算法优化随机森林模型的汽车保险欺诈识别研究 [J]. 保险研究 , 2017 , (06) : 114 - 127.
- [8] 叶明华. 基于 BP 神经网络的保险欺诈识别研究——以中国机动车保险索赔为例 [J]. 保险研究 , 2011 , (03) : 79 - 86.
- [9] 喻 炜, 冯根福, 张文珺. 机动车辆保险欺诈检测系统及团伙识别研究 [J]. 保险研究 , 2017 , (02) : 63 - 73.
- [10] Bhowmik R. Detecting Auto Insurance Fraud by Data Mining Techniques [J]. Journal of Emerging Trends in Computing and Information Sciences , 2011 , 2(4) : 156 - 162.
- [11] Bisker J H , Dietrich B L , Ehrlich K , et al. Health Insurance Fraud Detection Using Social Network Analytics:

U. S. Patent Application 11/622,740 [P]. 2008 – 7 – 17.

- [12] Francis L. A. Application of Two Unsupervised Learning Techniques to Questionable Claims: Predit and Random forest [C]// Frees E. W. ,Meyers G. ,Derrig R. A. (eds) Predictive Modeling Applications in Actuarial Science: Case Studies in Insurance (Volume II) . New York: Cambridge University Press 2016: 180 – 207.
- [13] Kirlidog M ,Asuk C. A Fraud Detection Approach with Data Mining in Health Insurance [J]. Procedia – Social and Behavioral Sciences 2012 ,62: 989 – 994.
- [14] Peng Y ,Kou G ,Sabatka A ,et al. Application of Clustering Methods to Health Insurance Fraud Detection [C]//Service Systems and Service Management ,2006 International Conference on. IEEE ,2006 ,1: 116 – 120.
- [15] Li Y ,Yan C ,Liu W ,et al. A Principle Component Analysis – Based Random Forest with the Potential Nearest Neighbor Method for Automobile Insurance Fraud Identification [J]. Applied Soft Computing ,2018 ,70: 1000 – 1009.
- [16] Šubelj L ,Furlan Š ,Bajec M. An Expert System for Detecting Automobile Insurance Fraud Using Social Network Analysis [J]. Expert Systems with Applications 2011 ,38(1) : 1039 – 1052.
- [17] Stefano B ,Gisella F. Insurance Fraud Evaluation: A Fuzzy Expert System [C]//Fuzzy Systems ,2001. The 10th IEEE International Conference on. IEEE 2001 ,3: 1491 – 1494.
- [18] Wang Y ,Xu W. Leveraging Deep Learning with LDA – based Text Analytics to Detect Automobile Insurance Fraud [J]. Decision Support Systems 2018 ,105: 87 – 95.

Insurance Fraud Detection Based on Bagging Ensemble Learning

LI Xiu – fang ,HUANG Zhi – guo ,CHEN Xiao – wei

Abstract: Insurance fraud not only jeopardizes the normal operation of insurance companies ,but also increases the burden on policyholders and may even affect China’s financial stability. With the advent of the era of big data ,it is necessary to introduce revolutionary technology for insurance fraud detection. The Bagging ensemble method has become an optimal choice because it’s easy to adjust the model structure according to the amount of data ,easy to deploy ,controllable parameter space ,and support for parallel computing. The Bagging methodology mainly comprises Bagging algorithm ,Random Subspace algorithm ,and Random Patches algorithm ,and they can be combined with other base learners to form new branch algorithms and algorithm examples. Based on these algorithms ,the paper conducted empirical testing on insurance frauds ,the applicability of various algorithms and base learners ,and the impacts of the number of base learners on the performance of algorithms. It was found that ,for insurance fraud detection ,the Random Patches algorithm had the highest score and the Bagging had the shortest running time among the Bagging ,Random Subspace and Random Patches. Different algorithm should apply different base learner ,but in general ,among various base learners ,the best was decision tree for the Bagging ensemble method. The most important feature of the decision tree method was whether to entrust a lawyer. The number of base learners had different effects on the performance of different algorithms.

Key words: Bagging; insurance fraud; Extremely Randomized Trees; Random Forest

[编辑:施 敏]