维基百科
自由的百科全书

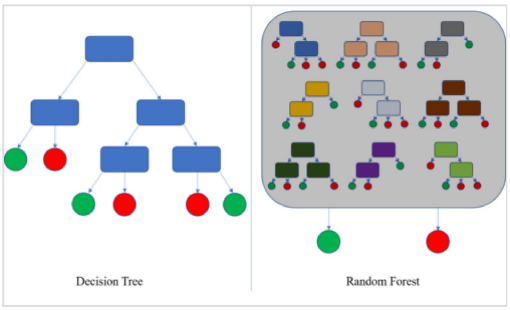
随机森林

在机器学习中，**随机森林**是一个包含多个决策树的分类器，并且其输出的类别是由个别树输出的类别的众数而定。

这个术语是1995年[1]由贝尔实验室的何天琴所提出的**随机决策森林**（**random decision forests**）而来的。[2][3]

然后Leo Breiman和Adele Cutler发展出推论出随机森林的算法。而"Random Forests"是他们的商标。

这个方法则是结合Breimans的“Bootstrap aggregating”想法和Ho的“random subspace method”以建造决策树的集合。



把随机树的聚合构建为随机森林的原理示意图。

历史

随机森林的引入最初是由华裔美国人何天琴于1995年[1]先提出的。[2]然后随机森林由Leo Breiman于2001年在一篇论文中提出的。[4]这篇文章描述了一种结合随机节点优化和bagging，利用类CART过程构建不相关树的森林的方法。此外，本文还结合了一些已知的、新颖的、构成了现代随机森林实践的基础成分，特别是

1. 使用out-of-bag误差来代替泛化误差
2. 通过排列度量变量的重要性

算法

预备：决策树学习

决策树是机器学习的常用方法。 Hastie等说：“树学习是如今最能满足于数据挖掘的方法，因为它在特征值的缩放和其他各种转换下保持不变，对无关特征是稳健的，而且能生成可被检查的模型。然而，它通常并不准确。”[5]

特别的，生长很深的树容易学习到高度不规则的模式，即过学习，在训练集上具有低偏差和高变异数的特点。随机森林是平均多个深决策树以降低变异数的一种方法，其中，决策树是在一个数据集上的不同部分进行训练的。[5]这是以偏差的小幅增加和一些可解释性的丧失为代价的，但是在最终的模型中通常会大大提高性能。

Bagging

随机森林训练算法把bagging的一般技术应用到树学习中。给定训练集$X = x_1, \ldots, x_n$和目标$Y = y_1, \ldots, y_n$，bagging方法重复（B次）从训练集中有放回地采样，然后在这些样本上训练树模型:

For $b = 1, \ldots, B$:

1. 从X和Y抽取n个训练样本（有放回抽样），分别记为X_b和Y_b。
2. 在X_b和Y_b上训练一个分类或回归树f_b。

在训练结束之后，对未知样本x的预测可以通过对x上所有单个回归树的预测求平均来实现：

$$\hat{f} = \frac{1}{B}\sum_{b=1}^{B} f_b(x')$$

或者在分类任务中选择多数投票的类别。

这种bagging方法在不增加偏置的情况下降低了方差，从而带来了更好的性能。这意味着，即使单个树模型的预测对训练集的噪声非常敏感，但对于多个树模型，只要这些树并不相关，这种情况就不会出现。简单地在同一个数据集上训练多个树模型会产生强相关的树模型（甚至是完全相同的树模型）。Bootstrap抽样是一种通过产生不同训练集从而降低树模型之间关联性的方法。

此外，x'上所有单个回归树的预测的标准差可以作为预测的不确定性的估计：

$$\sigma = \sqrt{\frac{\sum_{b=1}^{B}(f_b(x') - \hat{f})^2}{B-1}}.$$

样本或者树的数量**B**是一个自由参数。通常使用几百到几千棵树，这取决于训练集的大小和性质。使用交叉验证，或者透过观察out-of-bag误差（那些不包含x_i的抽样集合在样本x_i的平均预测误差），可以找到最优的**B**值。当一些树训练到一定程度之后，训练集和测试集的误差开始趋于平稳。

从 bagging 到随机森林

上面的过程描述了树的原始的 bagging 算法。随机森林与这个通用的方案只有一点不同:它使用一种改进的学习算法，在学习过程中的每次候选分裂中选择特征的随机子集。这个过程有时又被称为“特征 bagging”。这样做的原因是 bootstrap 抽样导致的树的相关性：如果有一些特征预测目标值的能力很强，那么这些特征就会被许多树所选择，这样就会导致树的强相关性。何天琴分析了不同条件下 bagging 和随机子空间投影对精度提高的影响。[3]

典型地，对于一个包含 p 个特征的分类问题，可以在每次划分时使用 $\sqrt{p}$ 个特征[5]:592。对于回归问题，作者推荐 $p/3$ 但不少于 5 个特征[5]:592。

极限树

再加上一个随机化步骤，就会得到**极限随机树**（*extremely randomized trees*），即极限树。与普通的随机森林相同，他们都是单个树的集成，但也有不同：首先，每棵树都使用整个学习样本进行了训练，其次，自上而下的划分是随机的。它并不计算每个特征的最优划分点（例如，基于信息熵或者基尼不纯度），而是随机选择划分点。该值是从特征经验范围内均匀随机选取的。在所有随机的划分点中，选择其中分数最高的作为结点的划分点。与普通的随机森林相似，可以指定每个节点要选择的特征的个数。该参数的默认值，对于分类问题，是$\sqrt{n}$，对于回归问题，是n，其中 n是模型的特征个数。[6]

性质

特征的重要性

随机森林天然可用来对回归或分类问题中变量的重要性进行排序。下面的技术来自Breiman的论文，R语言包randomForest包含它的实现。[7]

度量数据集 $\mathcal{D}_n = \{(X_i, Y_i)\}_{i=1}^n$的特征重要性的第一步是，使用训练集训练一个随机森林模型。在训练过程中记录下每个数据点的out-of-bag误差，然后在整个森林上进行平均。

为了度量第j个特征的重要性，第j个特征的值在训练数据中被打乱，并重新计算打乱后的数据的out-of-bag误差。则第j个特征的重要性分数可以通过计算打乱前后的out-of-bag误差的差值的平均来得到，这个分数通过计算这些差值的标准差进行标准化。

产生更大分数的特征比小分数的特征更重要。这种特征重要性的度量方法的统计定义由Zhu *et al.*[8]给出。

这种度量方法也有一些缺陷。对于包含不同取值个数的类别特征，随机森林更偏向于那些取值个数较多的特征，部分置换[9][10]、growing unbiased trees[11][12]可以用来解决这个问题。如果数据包含一些相互关联的特征组，那么更小的组更容易被选择。[13]

与最近邻算法的关系

Lin和Jeon在2002年指出了随机森林算法和K-近邻算法(k-NN)的关系。[14] 事实证明，这两种算法都可以被看作是所谓的“加权邻居的方案”。这些在数据集$\{(x_i, y_i)\}_{i=1}^n$上训练的模型通过查看一个点的邻居来计算一个新点x'的预测值$\hat{y}$，并且使用权重函数W对这些邻居进行加权:

$$\hat{y} = \sum_{i=1}^{n} W(x_i, x') \, y_i.$$

其中，$W(x_i, x')$是第i个点在同一棵树中相对于新的数据点x'的非负权重。对于任一特定的点x'，x_i的权重的和必须为1。权重函数设定如下:

- 对于k-NN算法，如果x_i是距离x'最近的k个点之一，则$W(x_i, x') = \frac{1}{k}$，否则为0。
- 对于树，如果x_i与x'属于同一个包含k'个点的叶结点，则$W(x_i, x') = \frac{1}{k'}$，否则为0。

因为森林平均了m棵树的预测，且这些树具有独立的权重函数W_j，故森林的预测值是:

$$\hat{y} = \frac{1}{m} \sum_{j=1}^{m} \sum_{i=1}^{n} W_j(x_i, x') \, y_i = \sum_{i=1}^{n} \left(\frac{1}{m} \sum_{j=1}^{m} W_j(x_i, x') \right) y_i.$$

上式表明了整个森林也采用了加权的邻居方案，其中的权重是各个树的平均。在这里，x'的邻居是那些在任一树中属于同一个叶节点的点x_i。只要x_i与x'在某棵树中属于同一个叶节点，x_i就是x'的邻居。

基于随机森林的非监督学习

作为构建的一部分，随机森林预测器自然会导致观测值之间的不相似性度量。还可以定义未标记数据之间的随机森林差异度量：其思想是构造一个随机森林预测器，将“观测”数据与适当生成的合成数据区分开来。[4][15] 观察到的数据是原始的未标记数据，合成数据是从参考分布中提取的。随机森林的不相似性度量之所以吸引人，是因为它能很好地处理混合变量类型，对输入变量的单调变换是不敏感的，而且在存在异常值的情况下度量结果依然可靠。由于其固有变量的选择，随机森林不相似性很容易处理大量的半连续变量。

学习算法

根据下列算法而建造每棵树：

1. 用N来表示训练用例（样本）的个数，M表示特征数目。
2. 输入特征数目m，用于确定决策树上一个节点的决策结果；其中m应远小于M。
3. 从N个训练用例（样本）中以有放回抽样的方式，取样N次，形成一个训练集（即bootstrap取样），并用未抽到的用例（样本）作预测，评估其误差。
4. 对于每一个节点，随机选择m个特征，决策树上每个节点的决定都是基于这些特征确定的。根据这m个特征，计算其最佳的分裂方式。
5. 每棵树都会完整成长而不会剪枝（Pruning，这有可能在建完一棵正常树状分类器后会被采用）。

优点

随机森林的优点有：

- 对于很多种资料，它可以产生高准确度的分类器。
- 它可以处理大量的输入变量。
- 它可以在决定类别时，评估变量的重要性。
- 在建造森林时，它可以在内部对于一般化后的误差产生不偏差的估计。
- 它包含一个好方法可以估计丢失的资料，并且，如果有很大一部分的资料丢失，仍可以维持准确度。
- 它提供一个实验方法，可以去侦测variable interactions。
- 对于不平衡的分类资料集来说，它可以平衡误差。
- 它计算各例中的亲近度，对于数据挖掘、侦测离群点（outlier）和将资料可视化非常有用。
- 使用上述。它可被延伸应用在未标记的资料上，这类资料通常是使用非监督式聚类。也可侦测偏离者和观看资料。
- 学习过程是很快速的。

开源实现

- The Original RF (http://www.stat.berkeley.edu/~breiman/RandomForests/cc_software.htm)（页面存档备份 (https://web.archive.org/web/20210225115642/http://www.stat.berkeley.edu/~breiman/RandomForests/cc_software.htm)，存于互联网档案馆） by Breiman and Cutler written in Fortran 77.
- ALGLIB (http://www.alglib.net/dataanalysis/decisionforest.php)（页面存档备份 (https://web.archive.org/web/20201109040906/http://www.alglib.net/dataanalysis/decisionforest.php)，存于互联网档案馆） contains a modification of the random forest in C#, C++, Pascal, VBA.
- party (http://cran.r-project.org/web/packages/party/index.html)（页面存档备份 (https://web.archive.org/web/20210224192518/http://cran.r-project.org/web/packages/party/index.html)，存于互联网档案馆） Implementation based on the conditional inference trees in R.
- randomForest (http://cran.r-project.org/web/packages/randomForest/index.html)（页面存档备份 (https://web.archive.org/web/20210308051225/http://cran.r-project.org/web/packages/randomForest/index.html)，存于互联网档案馆） for classification and regression in R.
- Python implementation (http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html)（页面存档备份 (https://web.archive.org/web/20210525051515/http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html)，存于互联网档案馆） with examples in scikit-learn.
- Orange data mining suite includes random forest learner and can visualize the trained forest.
- Matlab (https://code.google.com/p/randomforest-matlab)（页面存档备份 (https://web.archive.org/web/20160528051034/http://code.google.com/p/randomforest-matlab)，存于互联网档案馆） implementation.
- SQP (http://sqp.upf.edu)（页面存档备份 (https://web.archive.org/web/20210523083759/http://sqp.upf.edu/)，存于互联网档案馆） software uses random forest algorithm to predict the quality of survey questions, depending on formal and linguistic characteristics of the question.
- Weka RandomForest (http://weka.sourceforge.net/doc.dev/weka/classifiers/trees/RandomForest.html)（页面存档备份 (https://web.archive.org/web/20181215123831/http://weka.sourceforge.net/doc.dev/weka/classifiers/trees/RandomForest.html)，存于互联网档案馆） in Java library and GUI.
- ranger (https://github.com/imbs-hl/ranger)（页面存档备份 (https://web.archive.org/web/20210413221805/https://github.com/imbs-hl/ranger)，存于互联网档案馆） A C++ implementation of random forest for classification, regression, probability and survival. Includes interface for R.

参阅

- 机器学习
- 提升方法 - 机器学习方法
- 决策树学习 - 机器学习算法
- 集成学习——通过结合多个学习器来解决问题的一种机器学习范式
- 非参数统计
- 随机化算法 - 将一定程度的随机性作为其逻辑或程序的一部分的算法
- 梯度提升技术

参考文献

1. Tin Kam Ho. Random decision forests. Proceedings of 3rd International Conference on Document Analysis and Recognition (Montreal, Que., Canada: IEEE Comput. Soc. Press). 1995, **1**: 278–282 [2020-03-04]. ISBN 978-0-8186-7128-9. doi:10.1109/ICDAR.1995.598994. （原始内容存档于2021-03-03）.
2. Tin Kam Ho. The random subspace method for constructing decision forests. IEEE Transactions on Pattern Analysis and Machine Intelligence. Aug./1998, **20** (8): 832–844 [2020-03-04]. doi:10.1109/34.709601. （原始内容存档于2021-03-08）.
3. Ho, Tin Kam. A Data Complexity Analysis of Comparative Advantages of Decision Forest Constructors (PDF). Pattern Analysis and Applications. 2002: 102–112 [2019-02-16]. （原始内容 (PDF)存档于2016-04-17）.
4. RandomForest2001 (PDF). [2019-07-26]. （原始内容 (PDF)存档于2021-04-03）.
5. Hastie, Trevor; Tibshirani, Robert; Friedman, Jerome. The Elements of Statistical Learning 2nd. Springer. 2008 [2024-12-07]. ISBN 0-387-95284-5. （原始内容存档于2009-11-10）.
6. Geurts P, Ernst D, Wehenkel L. Extremely randomized trees (PDF). Machine Learning. 2006, **63**: 3–42 [2019-02-16]. doi:10.1007/s10994-006-6226-1. （原始内容 (PDF)存档于2017-10-31）.
7. Liaw A. Documentation for R package randomForest (PDF). 16 October 2012 [15 March 2013]. （原始内容 (PDF)存档于2021-03-19）.
8. Zhu R, Zeng D, Kosorok MR. Reinforcement Learning Trees. Journal of the American Statistical Association. 2015, **110** (512): 1770–1784. PMC 4760114. PMID 26903687. doi:10.1080/01621459.2015.1036994.
9. Deng,H.; Runger, G.; Tuv, E. Bias of importance measures for multi-valued attributes and solutions (PDF). Proceedings of the 21st International Conference on Artificial Neural Networks (ICANN): 293–300. 2011 [2019-02-16]. （原始内容 (PDF)存档于2018-08-09）.
10. Altmann A, Toloşi L, Sander O, Lengauer T. Permutation importance: a corrected feature importance measure. Bioinformatics. May 2010, **26** (10): 1340–7 [2019-02-16]. PMID 20385727. doi:10.1093/bioinformatics/btq134. （原始内容存档于2016-11-08）.

11. Strobl C, Boulesteix AL, Augustin T. Unbiased split selection for classification trees based on the Gini index (PDF). Computational Statistics & Data Analysis. 2007: 483–501 [2019-02-16]. （原始内容 (PDF)存档于2020-11-12）.
12. Painsky A, Rosset S. Cross-Validated Variable Selection in Tree-Based Methods Improves Predictive Performance. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2017, **39** (11): 2142–2153. PMID 28114007. arXiv:1512.03444. doi:10.1109/tpami.2016.2636831.
13. Tolosi L, Lengauer T. Classification with correlated features: unreliability of feature ranking and solutions. Bioinformatics. July 2011, **27** (14): 1986–94 [2019-02-16]. PMID 21576180. doi:10.1093/bioinformatics/btr300. （原始内容存档于2015-08-31）.
14. Lin, Yi; Jeon, Yongho. Random forests and adaptive nearest neighbors (技术报告). Technical Report No. 1055. University of Wisconsin. 2002 [2019-02-16]. （原始内容存档于2013-06-30）.
15. Shi, T., Horvath, S. Unsupervised Learning with Random Forest Predictors. Journal of Computational and Graphical Statistics. 2006, **15** (1): 118–138 . CiteSeerX 10.1.1.698.2365. JSTOR 27594168. doi:10.1198/106186006X94072.

外部链接

- （英文）Ho, Tin Kam (1995). "Random Decision Forest". Proc. of the 3rd Int'l Conf. on Document Analysis and Recognition, Montreal, Canada, August 14-18, 1995, 278-282 (https://web.archive.org/web/20080704141852/http://cm.bell-labs.com/cm/cs/who/tkh/papers/odt.pdf)（Preceding Work）
- （英文）Ho, Tin Kam (1998). "The Random Subspace Method for Constructing Decision Forests". IEEE Trans. on Pattern Analysis and Machine Intelligence 20 (8), 832-844 (https://web.archive.org/web/20070930204101/http://cm.bell-labs.com/cm/cs/who/tkh/papers/df.pdf)（Preceding Work）
- （英文）Deng, H; Runger, G; Tuv, Eugene (2011). Bias of importance measures for multi-valued attributes and solutions, Proceedings of the 21st International Conference on Artificial Neural Networks (ICANN2011) (https://web.archive.org/web/20110727072115/http://enpub.fulton.asu.edu/hdeng3/MultiICANN2011.pdf)
- （英文）Amit, Yali and Geman, Donald (1997) "Shape quantization and recognition with randomized trees". Neural Computation 9, 1545-1588. (http://www.cis.jhu.edu/publications/papers_in_database/GEMAN/shape.pdf)（页面存档备份 (https://web.archive.org/web/20180205094828/http://www.cis.jhu.edu/publications/papers_in_database/GEMAN/shape.pdf)，存于互联网档案馆）（Preceding work）
- （英文）Breiman, Leo "Looking Inside The Black Box". Wald Lecture II (https://web.archive.org/web/20081204092820/http://www.ics.uci.edu/~liang/seminars/win05/papers/wald2002-2.pdf)（Lecture）
- （英文）Breiman, Leo (2001). "Random Forests". Machine Learning 45 (1), 5-32 (http://www.springerlink.com/content/u0p06167n6173512/fulltext.pdf)（Original Article）

- （英文）Random Forest classifier description (https://web.archive.org/web/20080622230434/http://stat-www.berkeley.edu/users/breiman/RandomForests/cc_home.htm)（Site of Leo Breiman）
- （英文）Liaw, Andy & Wiener, Matthew "Classification and Regression by randomForest" R News (2002) Vol. 2/3 p. 18 (http://cran.r-project.org/doc/Rnews/Rnews_2002-3.pdf)（页面存档备份 (https://web.archive.org/web/20210307231334/http://cran.r-project.org/doc/Rnews/Rnews_2002-3.pdf)，存于互联网档案馆）（Discussion of the use of the random forest package for R）
- （英文）Ho, Tin Kam (2002). "A Data Complexity Analysis of Comparative Advantages of Decision Forest Constructors". Pattern Analysis and Applications 5, p. 102-112 (http://cm.bell-labs.com/cm/cs/who/tkh/papers/compare.pdf)（页面存档备份 (https://web.archive.org/web/20071001002851/http://cm.bell-labs.com/cm/cs/who/tkh/papers/compare.pdf)，存于互联网档案馆）（Comparison of bagging and random subspace method）
