摘要：

本文描述了一个R包Boruta，实现了一种寻找全部相关变量的新特征选择算法。该算法设计为一个随机森林分类算法的包装器。它迭代地删除了统计测试所证明的与随机探针不太相关的特征。Boruta包为算法提供了一个方便的接口。给出了算法的简要描述和应用实例。

关键词:特征选择、特征排序、随机森林。

介绍：

在机器学习方法的应用中，认为特性选择通常是一个重要的步骤，是有充分理由的。对于实际的模型构建来说，现代的数据集通常描述的变量太多。通常这些变量中的大多数与分类无关，而且显然它们的相关性事先不知道。处理过大的特性集有几个缺点。一个是纯技术性的——处理大型特性集会减慢算法的速度，占用太多的资源，而且很不方便。另一个更重要的是——当变量数量显著高于最优时，许多机器学习算法的准确率会下降(Kohavi和John 1997)。因此，选择可以给出最好的分类结果的小的(可能最小的)特性集，是出于实际的原因。这个问题,被称为minimaloptimal问题 (Nilsson, Pen˜a, Bj¨orkegren, and Tegn´er 2007),,已被深入研究，有许多算法被开发出来，以将特征集减少到可管理的大小。

然而，这个非常实际的目标掩盖了另一个非常有趣的问题——识别在某些情况下与分类有关的所有属性，查找全部相关属性，而不仅仅是非冗余的属性，可能这项工作本身就很有用。特别是，当一个人对理解与感兴趣的主题相关的机制感兴趣时，这几乎是必要的，而不仅仅是建立一个黑盒子预测模型。例如，在处理癌症环境下的基因表达测量的结果时，对所有与癌症相关的基因的识别对于完全理解这一过程是必要的，而一组最优的基因作为遗传标记可能更有用。Nilsson et al.(2007)给出了一个很好的讨论，说明了为什么找到全部相关属性是很重要的。

特征选择的全部相关问题都比通常的最小最优问题困难。：一个原因是我们不能依赖分类的准确性作为选择重要特征的标准(或者因为它不重要而拒绝)。在从特征集中去除特征后，分类精度的下降，足以声明特征的重要性，但缺乏这种效果并不足以说明它不重要。因此，需要另一个标准来声明重要或不重要的变量。此外，我们不能使用过滤方法，因为给定特征值与决策之间缺乏直接的相关性并不能证明该特征值与其他特征值一起不重要(Guyon和Elisseeff 2003)。因此，有一种算法被限制在包装器算法中，这比过滤器在计算上要求更高。

在包装器方法中，分类器被用作返回特征排序的黑盒，因此可以使用任何可以提供特征排序的分类器。由于实际原因，在这个问题中使用的分类器应该具有计算方面的效率和简单性，可能没有用户定义的参数。

当前的论文提出了一种算法，用于在R (R开发核心团队2010)包Boruta中查找信息系统中的所有相关特性(可从 http://CRAN.R-project.org/package= Boruta的综合R归档网络中获得)。该算法使用一种基于随机森林(Breiman 2001)分类器的包装方法(Boruta是斯拉夫神话中的森林之神)。该算法是Stoppiglia、Dreyfus、Dubois和Oussar(2003)提出的通过比较真实特征与随机探针的相关性来确定相关性的思想的扩展。最初，这个想法是在过滤方法的文中提出的，而在这里，它被用于包装器算法。在本文的其余部分中，首先对算法进行了简短的描述，然后给出了它在真实世界和人工数据集上的应用实例。

1. Boruta算法

Boruta算法是一个基于随机森林分类算法的包装器，该算法在R package randomForest (Liaw和Wiener 2002)中实现。随机森林分类算法是一种比较快速的分类算法，通常不需要调整参数就可以实现，并给出了特征重要性的数值估计。它是一种通过多无偏弱分类器-决策树的投票来进行分类的集成方法。这些树是在训练集的不同包装器样本上独立开发的。属性的重要性度量是由于对象间属性值的随机排列导致分类精度的损失。对森林中使用给定属性进行分类的所有树分别进行计算。然后计算精度损失的平均值和标准差。此外,Z得分用平均损失除以标准差的计算方法可以作为重要度。不幸的是，Z分数与随机森林算法所返回的特征重要性的统计意义并不直接相关，因为它的分布不是N(0,1) (Rudnicki, Kierczak, Koronacki，和Komorowski 2006)。然而，在Boruta，我们使用Z分数作为重要度量，因为它考虑了森林中树木之间平均准确度损失的波动。

由于我们不能直接使用Z分数来衡量重要性，所以我们需要一些外部参考来判断任何给定属性的重要性是否显著，也就是说，它是否可以从随机波动产生的重要性中辨别出来。此，我们设计了具有随机属性的信息系统。对于每个属性，我们都创建一个相应的“影子”属性，其值通过在对象之间移动原始属性的值来获得。然后，我们使用这个扩展系统的所有属性执行分类，并计算所有属性的重要性。

影子属性的重要性可以不为零，由于随机波动。因此，阴影属性的导入集被用作决定哪些属性是真正重要的参考。

由于随机森林分类器的随机性，重要性度量本身也有所不同。此外，它对信息系统中不重要的属性(也包括影子属性)的存在很敏感。此外，它还依赖于阴影属性的特定实现。因此，我们需要重复重新洗牌过程以获得统计有效的结果。

简而言之，Boruta基于形成随机森林分类器的相同思想，即在系统中加入随机性，从随机样本集合中收集结果，可以减少随机波动和相关性的误导影响。在这里，这个额外的随机性将为我们提供一个更清楚的视图，哪些属性是真正重要的。

Boruta算法包括以下步骤:

1。通过添加所有变量的副本来扩展信息系统(信息系统总是至少扩展5个影子属性，即使原始集合中的属性数小于5)。

2。调整添加的属性，以消除它们与响应的相关性。

1. 在扩展信息系统上运行一个随机森林分类器，并收集计算出的Z分数。
2. 找到阴影属性(MZSA)之间的最大Z分数，然后为每个得分高于MZSA的属性分配一个命中。
3. 对于未确定重要性的每个属性执行一个与MZSA相等的双面检验。
4. 将重要程度显著低于MZSA的属性视为“不重要”，并将其永久从信息系统中删除。
5. 认为重要性显著高于MZSA的属性为“重要”。
6. 删除所有阴影属性。
7. 重复此过程，直到为所有属性分配重要性，或者该算法已经达到先前设置的随机森林运行的极限。

在实践中，该算法前面有三个启动循环，限制较少的重要性标准。在程序开始时，当属性数较大时，引入启动循环以应对Z分数的高波动。在这些初始循环中，属性分别与第五、第三和第二最佳阴影属性进行比较;拒绝测试只在每一轮的最后进行，而确认测试根本不进行。

在实际案例中描述的过程的时间复杂度大约是O(P·N)，其中P和N分别是属性和对象的数量。对于大型数据集来说，这可能很耗时;尽管如此，这种努力对于产生统计上意义重大的相关特性选择是必不可少的。

为了说明Boruta算法的缩放特性，我们使用Madalon数据集进行了以下实验，这是一个人工数据集，这是NIPS2003中的一个问题。数据集包含有500个属性的2000个对象。我们制作了含有250 ，500 ，750，……,2000个对象的Madalon的样品。然后，对于每个子样本，我们创建了七个扩展集，分别包含500个、1000个、…、3500个作为均匀随机噪声被获得的表面属性。然后我们在64个测试集上使用Boruta进行标准的特征选择，并测量执行时间。实验结果如图1所示。随着属性的增加，我们可以看到几乎完美的线性扩展。另一方面，执行时间的增长速度要快于对象的数量，但差异并不大，对于大量的对象来说，它似乎收敛于线性扩展。

计时以CPU时间报告。使用最大数据集的数据,可以估计完成Boruta在单一核心的现代CPU上运行一个小时每一百万(属性×对象)所需的时间。

人们应该注意到，在许多情况下，特别是对于生物医学问题，计算时间只是收集数据所需时间的一小部分。还需要注意的是，运行“全部相关”特征选择算法的主要原因不是减少了计算时间(如果从非信息性属性中删除的数据集随后将被多次分析，则可以实现这一点)。主要原因是要找到所有与决策相关程度高于随机属性的属性。此外，虽然Boruta一般是一种序列算法，但底层的随机森林分类器是一项微不足道的并行任务，因此，如果使用随机森林算法的并行版本，Boruta甚至可以分布在数百个核上。

1. 使用Boruta包

Boruta算法在Boruta包中实现。

1. library("Boruta")

来自UCI机器学习库(Asuncion和Newman 2007)和mlbench包(Leisch and Dimitriadou 2010)的臭氧数据作为第一个例子:

R> library("mlbench")

R> data("Ozone")

R> Ozone <- na.omit(Ozone)

该算法由Boruta函数执行。对于其参数，应该使用公式或带有响应向量的预报数据框架指定模型；置信水平(建议保持默认)和随机森林运行的最大数量。

还可以提供mtry和ntree参数的值，这些值将传递给randomForest函数。通常使用默认的随机森林参数，它们在大多数情况下都是足够的，因为随机森林性能对其参数的依赖性较弱。如果不是这样，我们应该尝试找到mtry和ntree，以便随机森林分类器在OOB误差最小的情况下实现收敛。

将doTrace参数设置为1或2将使Boruta报告进程的进展；版本2有点啰嗦，也就是说，一旦属性决定被清除，它就会显示出来。

R> set.seed(1)

R> Boruta.Ozone <- Boruta(V4 ~ ., data = Ozone, doTrace = 2, ntree = 500)

Initial round 1: .......... 1 attributes rejected after this test: V2

Initial round 2: .......... 1 attributes rejected after this test: V3

Initial round 3: .......... Final round: .......... 8 attributes confirmed after this test: V1 V5 V7 V8 V9 V10 V11 V12 .... 1 attributes confirmed after this test: V13 .... 1 attributes rejected after this test: V6

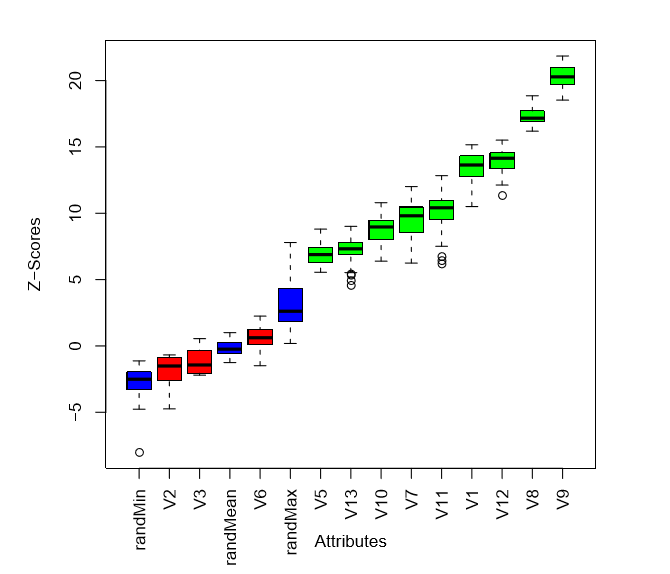
R> Boruta.Ozone

Boruta performed 48 randomForest runs in 2.540633 mins. 9 attributes confirmed important: V1 V5 V7 V8 V9 V10 V11 V12 V13 3 attributes confirmed unimportant: V2 V3 V6

臭氧集由12个属性组成;其中三个被拒绝，两个在第一轮之后，一个在最后一轮。其余属性表示为已确认。图2显示了Boruta运行期间属性之间的Z分数变化。使用Boruta对象的plot方法可以轻松生成:

1. plot(Boruta.Ozone)

图2:臭氧数据的Boruta结果图。蓝色方框表示阴影属性的最小、平均和最大Z值。红色和绿色方框分别表示拒绝属性和确认属性的Z分数。



可以看出，最重要的阴影属性的Z分数清楚地分离了重要的和非重要的属性。

此外，很明显，在单个随机森林运行中始终获得高重要性分数的属性被选择为重要属性。另一方面，我们可以观察到个体分数的相当大的变异性。在一次运行中，随机属性的最高分数要高于两个重要属性的最高重要性分数，以及五个重要属性的最低重要性分数。它清楚地表明，Boruta的结果通常比基于单个随机森林运行的特征选择方法产生的结果更稳定，这就是为什么需要多次迭代。

由于Boruta期间随机森林运行的数量受到maxrunning参数的限制，因此当仍然存在判断为既不被确认也不被拒绝的属性时，可以强制停止计算，从而最终标记为暂定属性。instance:

R> set.seed(1)

R> Boruta.Short <- Boruta(V4 ~ ., data = Ozone, maxRuns = 12)

R> Boruta.Short

Boruta performed 42 randomForest runs in 2.3612 mins. 8 attributes confirmed important: V1 V5 V7 V8 V9 V10 V11 V12 2 attributes confirmed unimportant: V2 V3 2 tentative attributes left: V6 V13

如果留下了暂定属性，应该考虑增加maxrun参数。然而，可能有一些属性非常重要，它们与MZSA非常接近，以至于Boruta无法对实际的随机森林运行数量做出预期的决定。因此Boruta包包含了一个 TentativeRoughFix 的功能，可以通过简单地将中间属性Z分数与最重要的影子属性Z分数的中值进行比较来弥补缺失的决策:

R> TentativeRoughFix(Boruta.Short)

Boruta performed 42 randomForest runs in 2.3612 mins. Tentatives roughfixed over 12 last randomForest runs. 9 attributes confirmed important: V1 V5 V7 V8 V9 V10 V11 V12 V13 3 attributes confirmed unimportant: V2 V3 V6

显然可以手动处理这些属性。

为了方便地将Boruta结果传递给其他分类器和工具，Boruta包包含了提取结果并将其转换为方便形式的函数。getConfirmedFormula和getNonRejectedFormula创建一个公式对象，该对象仅根据确认属性或确认属性和暂定属性分别定义一个模型：

R> getConfirmedFormula(Boruta.Ozone)

V4 ~ V1 + V5 + V7 + V8 + V9 + V10 + V11 + V12 + V13

attStats函数创建一个数据框架，其中包含每个属性的Z分数统计信息，以及该属性比最重要的影子属性更重要的随机森林运行的百分比:

1. attStats(Boruta.Ozone)

meanZ medianZ minZ maxZ normHits decision V1 13.3911279 13.6373356 10.505555 15.1610346 1.0000000 Confirmed V2 -2.0475252 -1.5112547 -4.741706 -0.6750894 0.0000000 Rejected V3 -1.2097874 -1.4335204 -2.202290 0.5520193 0.0000000 Rejected V5 6.9889240 6.8839769 5.552918 8.8074357 0.9166667 Confirmed V6 0.5866514 0.6179196 -1.491181 2.2507610 0.1250000 Rejected V7 9.4355872 9.8092537 6.244625 12.0112148 0.9791667 Confirmed V8 17.3302697 17.1651707 16.186920 18.8550455 1.0000000 Confirmed V9 20.3332547 20.2826539 18.530345 21.8499295 1.0000000 Confirmed V10 8.7124127 8.9674981 6.391154 10.7939586 0.9791667 Confirmed V11 10.0848916 10.4122110 6.179540 12.8348468 0.9583333 Confirmed V12 13.9761395 14.1462836 11.335510 15.5130497 1.0000000 Confirmed V13 7.1691008 7.3218887 4.561458 9.0149381 0.9166667 Confirmed

4. Example: Madelon data

Madelon是一个人工数据集，这是一个NIPS2003问题之一。(Guyon et al. 2005)数据集包含2000个对象，这些对象对应于一个5维超立方体的32个顶点上的点。每个顶点随机分配是两类:−1或+ 1,决定每个对象是一个类的顶点。500个属性按以下方式构造:其中5个是点的随机抖动坐标;其他15个是前5个的随机线性组合;最后，系统的其余部分是均匀随机噪声。任务是从系统中提取20个重要属性。

可以从UCI机器学习存储库(Asuncion和Newman 2007)获得Madelon数据(加载此数据集可能需要几分钟):

R> root <+ "http://archive.ics.uci.edu/ml/machine-learning-databases/madelon/MADELON/"

R> predictors <- read.table(paste(root, "madelon\_train.data", sep = ""))

R> decision <- read.table(paste(root, "madelon\_train.labels", sep = ""))

R> Madelon <- data.frame(predictors, decision = factor(decision[, 1]))

运行Boruta(执行可能需要几个小时):

R> set.seed(7777) R> Boruta.Madelon <- Boruta(decision ~ ., data = Madelon)

R> Boruta.Madelon

Boruta performed 51 randomForest runs in 1.861855 hours. 20 attributes confirmed important: V29 V49 V65 V106 V129 V154 V242 V282 V319 V337 V339 V379 V434 V443 V452 V454 V456 V473 V476 V494 480 attributes confirmed unimportant: V1 V2 V3 V4 V5 V6 V7 V8 V9 V10 V11 V12 V13 V14 V15 V16 V17 V18 V19 V20 V21 V22 V23 V24 V25 V26 V27 V28 (the rest of the output was omitted)

可以看到，我们已经获得了20个确认属性。plotZScore函数可视化了Boruta运行期间属性Z分数的演变:

R> plotZHistory(Boruta.Madelon)

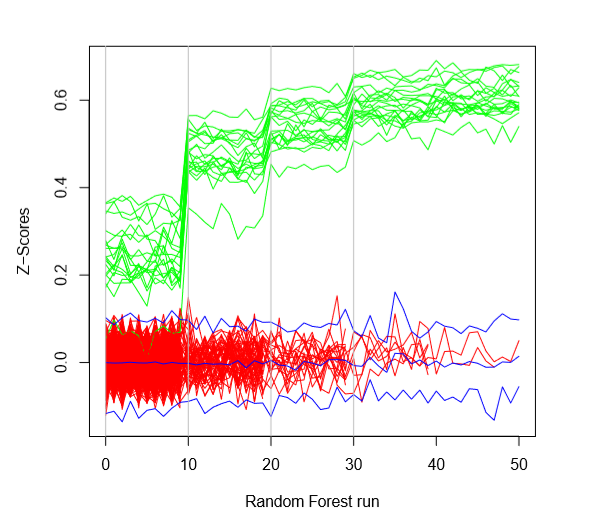


图3:Boruta运行时Z分数的演变。绿线对应于已确认的属性，红色表示拒绝的属性，蓝色分别表示最小、平均和最大阴影属性的重要性。灰色的线轮分开。

结果如图3所示。人们可能会注意到，连续去除随机噪声会增加重要属性的Z分数，并改善它们与不重要属性的分离;其中一个甚至在第一轮之后就从一组不重要的属性中“拉出”。同样，在某些情况下，不重要的属性可能会比最重要的影子属性获得更高的Z分数，这就是为什么我们需要多次随机森林运行才能得到统计上有意义的决策。

属性数量的减少是相当可观的(96%)。可以预期，由于消除了噪声，在减少的数据集上可以得到随机森林分类器的准确性的提高。

众所周知，特征选择过程会在结果模型中引入显著的偏差。例如Ambroise和McLachlan(2002)已经证明，在特征选择过程的帮助下，可以获得一个只使用非信息性属性并且在训练集上100%准确的分类器。显然，这样的分类器是无用的，并且在测试集上返回随机的答案。

因此，有必要检查Boruta是否能抵抗这种类型的错误。它是在交叉验证过程的帮助下实现的。将数据的一部分作为测试集放置，然后对剩余的数据集——训练集——执行完整的特征选择过程。最后利用训练集上获得的分类器对测试集中的对象进行分类，得到分类误差。这个过程重复了几次，以得到结果的变化性的估计。

Boruta执行一些随机森林运行，以获得重要属性和不相关属性之间统计上显著的划分。我们应该期望在单次RF运行中获得的排名应该与Boruta的排名非常相似。我们可以利用上面描述的交叉验证过程来检查这种情况是否属实。

Madelon数据被分成十次训练和测试集，分别包含90%和10%的对象。在每个训练集上运行Boruta，在每个训练集上生成三个随机森林分类器:第一个使用所有的属性，第二个分类器只使用Boruta选择的这些属性，第三个属性使用和Boruta找到的相同数量的属性，但是是被第一个训练所有属性的随机森林选择的最重要的那些属性。最后，收集了一个训练集上的OOB误差估计和所有分类器测试集上的误差。结果如表1所示。可以看到，OOB错误和测试集上的错误在减少的属性集上的随机森林运行都是一致的。这一观察结果通过t检验得到验证:

R> t.test(CV.Boruta$"Test conf.", CV.Boruta$"Test all", paired = TRUE)

Paired t-test

data: CV.Boruta$"Test conf." and CV.Boruta$"Test all" t = -24.2727, df = 9, p-value = 1.636e-09 alternative hypothesis: true difference in means is not equal to 0 95 percent confidence interval: -0.198962 -0.165038 sample estimates:

mean of the differences -0.182

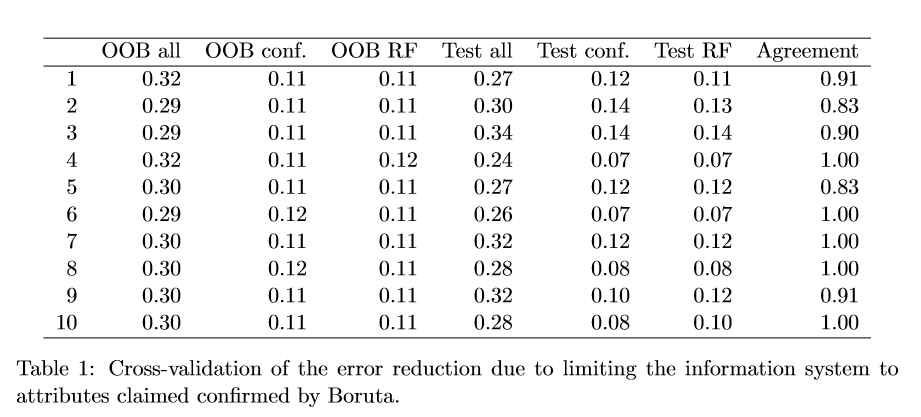


表1:由于将信息系统限制到由Boruta确认的属性而导致的错误减少的交叉验证。

正如我们所期望的，plain random forest提供的特性排名与Boruta的结果相当吻合。这就解释了为什么随机森林中简单的启发式特征选择过程，即选择一打左右的顶级评分属性，可以很好地获得良好的分类结果。然而，在处理更大、更复杂的集合时，这并不一定是必然的情况，在这些集合中，随机效应增加了随机森林重要性度量的可变性，从而破坏了特征排序的稳定性。

应该注意的是，Boruta是一个启发式过程，旨在查找所有相关属性，包括弱相关属性。在Nilsson等人(2007)之后，我们说当一个人可以找到一个属性子集而这个属性不是冗余的时候，属性就显得非常重要。Boruta中使用的启发式表示与决策变量显著相关的属性是相关的，这里的显著性意味着相关性高于随机生成的属性。显然，所有相关属性的集合可能包含高度相关但仍然是冗余的变量。同样，属性与决策的相关性并不意味着因果关系;当决策属性和描述性属性与其他变量独立关联时，可能会出现这种情况。Strobl、Hothorn和Zeileis(2009)给出了这样的例子。有兴趣在Boruta返回的结果中找到一组高度相关和不相关的属性的用户可以使用例如package party (Strobl et al. 2009)、caret (Kuhn 2008;Kuhn, Wing, Weston, Williams, Keefer和Engelhardt 2010)， varSelRF (Diaz-Uriarte 2007, 2010)或FSelector (Romanski 2009)进一步细化。

1. 总结

我们已经开发出了一种新的基于随机森林的特征选择方法Boruta，它提供了一个信息系统中重要和非重要属性的无偏和稳定的选择。由于迭代结构的存在，我们的方法可以同时处理这两个问题：随机森林重要性度量的波动性质和属性之间的相互作用。我们还演示了它在人工数据集上的有用性。该方法可用作R包。