大规模随机森林

随机森林是一类基于决策树的集成学习方法，在许多学习任务中表现出色，在最近的一项算法对比工作[1]中排名第一。随机森林是一类算法，有许多种不同的实现。在本次编程练习中，我们实现一种适用于大规模数据的方法。

[1] M. Fernandez-Delgado, E. Cernadas, S. Barro, and D. Amorim. Do we Need Hundreds of Classifiers to Solve Real World Classification Problems? Journal of Machine Learning Research, 15, 3133-3181, 2014.

算法

这里我假设各位已经学习完《机器学习》中线性分类器、决策树和集成学习部分内容。我们知道决策树递归的输入空间进行划分，生成树形分类规则。随机森林则是将决策树进行随机化，并训练多个随机化的决策树，将它们的输出整合作为随机森林的输出。

在标准决策树的学习中，在每一个节点上选择划分时，需要对所有特征上所有可能的划分点进行测试，从中挑选最好的划分。因此每一个节点的学习需要的计算量为：数据量x特征数量x每特征划分点数量。当数据庞大时较为耗时。

我们将实现的随机森林基于不同的决策树。该决策树每一个节点为一个线性分类器，由于我们并不需要在每个节点学习到最完美的划分，因此使用随机梯度方法来训练每个节点的线性分类器的学习使用随机梯度方法。同时，由于随机梯度方法带来的随机性，自然每次训练得到的决策树并不相同，因此可以训练多次，得到多个不同的决策树用于集成。

伪代码

输入：

D: 训练样本	m: 最少样本数	e: 分类器精度	S: 集成规模
d: 当前深度	$maxd$: 最大深度		

叶节点属性：

L,R: 左、右子节点	$\boldsymbol{c}$: 节点类别分布向量	w,b: 线性分类器

决策树生成递归代码：

BuildTree (D, m, e, d, $maxd$)

1 IF $|D| \leq m$ 或 D 中仅有一个类别的数据 或 $d \geq maxd$ THEN

2 L=R=*null*，$\boldsymbol{c}$ 为 D 中的类别分布

3 ENDIF

4 (w,b) = Linear(D, e)

5 DL = 符合 $w^\top x + b \geq 0$ 的 D 中样本，DR = 剩余样本

6 L = BuildTree(DL, m, e, d+1, $maxd$), R = BuildTree(DR, m, e, d+1, $maxd$)

7 RETURN

线性分类器训练：

Linear (D, e)

1 **随机选择 D 中的一个类别作为正类**($l(x)$ =+1)，**其他类别样本都作为负类** ($l(x)$ =-1)

2 **随机选择正类样本** x^+ **和 负类样本** x^-, **令** $w_0 = x^+ - x^-/||x^+ - x^-||$, $b_0 = (w_0^\top x^+ + w_0^\top x^-)/2$

3 FOR $t = 1$ TO T where T=$1/e^2$

4 **随机选取样本** x

5 IF $(w_{t-1}^\top x + b_{t-1}) \cdot l(x) < 0$ THEN

6 $w_t = w_{t-1} + \frac{0.1}{0.1+te^2} xl(x)$, $b_t = b_{t-1} + \frac{0.1}{0.1+te^2} l(x)$

7 ENDIF

8 IF (t % $|D|$ == 0) and $\|w_t - w_{t-|D|}\| \leq 0.001$ THEN BREAK

9 NEXT

10 RETURN (w_T, b_T)

决策树预测代码：

TestTree (x)

1 IF L==*null* THEN

2 RETURN $\boldsymbol{c}$

3 ELSEIF $w^\top x + b \geq 0$ THEN

4 RETURN L.TestTree(x)

5 ELSE

6 RETURN R.TestTree(x)

7 ENDIF

随机森林代码：

训练 (D, m, e, d, *maxd*, S)

1 FOR $i = 1$ TO S

2 tree$_i$ = BuildTree (D, m, e, d, *maxd*)

3 NEXT

预测 (x)

1 $\boldsymbol{c}$ **为0向量**

2 FOR $i = 1$ TO S

3 $\boldsymbol{c} = \boldsymbol{c}$ + tree$_i$.TestTree(x)

4 NEXT

3 RETURN $\boldsymbol{c}$ **中最大元素对应的类别**

代码实现

语言：可选择 java, c/c++, Matlab, Python 等你熟悉的语言实现。对于树形结构的生成，脚本语言（Matlab, Python等）通常速度较慢。可以使用现有数据读取实现以读取数据集（如Weka），但不得使用现有分类器实现。

数据：数据只考虑连续特征，离散特征的数据可直接当做连续特征处理。不考虑特征丢失值的处理。数据集可以使用 MNIST 数据集（数据: http://cis.jhu.edu/~sachin/digit/digit.html, WEKA: http://axon.cs.byu.edu/data/mnist/ ）

算法参数：随机森林中的决策树通常需要较大深度。可以设置 *maxd* 为无穷大，*m* 为 3，*e* 为 0.01，集成规模 *S* 为 20。

队列代替递归：当数据量较大时，基于递归的决策树生成方法容易发生栈溢出，因此请勿使用递归实现。决策树的生成和预测都要用队列实现。（基于队列的方法还有其他好处，当内存不足以装下整颗决策树时，基于队列的实现还可以扩展为将节点放在硬盘上；当有多个处理器时，容易扩展为并行树生成。这里都不作要求）。提示：可以先实现为递归，稍作改动即可实现为队列。

测试用例：请完成代码的同时写出测试用例，用以验证代码正确性。

提交

请在 ftp://lamda.nju.edu.cn/code 下创建自己的目录，目录名为“姓名-导师”，然后将代码上传到自己的目录中，不用压缩打包。