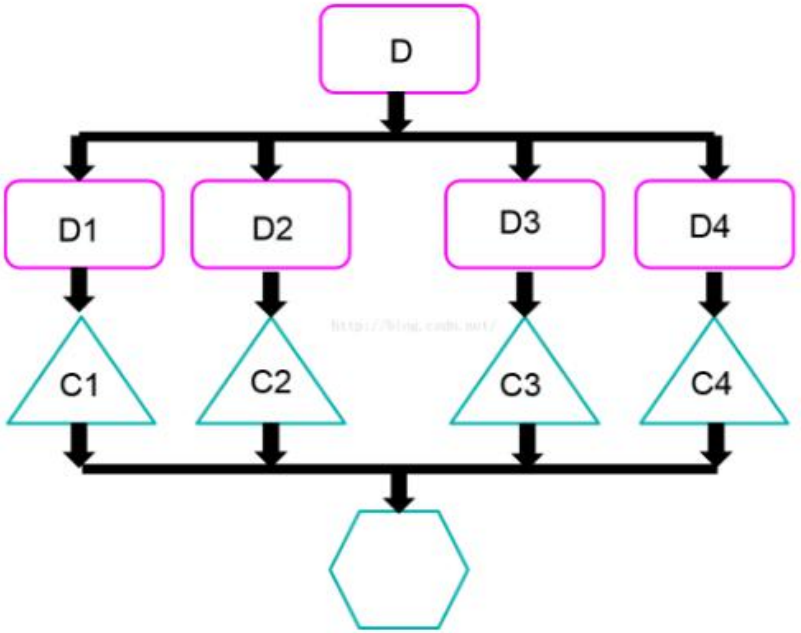
分类器组合方法

赵海臣

组合方法概述

- 通过聚集多个分类器的预测来提高分类准确率，这些技术称为组合(ensemble)或分类器组合(classifier combination)方法。
- 组合方法由训练数据构成一组基分类器(base classifier)，然后通过对每个基分类器的预测进行投票进行分类。



组合分类器使用条件

- 组合分类器的性能优于单个分类器必须满足两个必要的条件：
  - 基分类器之间应该是相互独立的
  - 基分类器应当好于随机猜想分类器(对于二分类，准确率应该高于50%)
- 实践上很难保证基分类器之间完全独立，尽管如此，轻微相关的情况下，组合方法仍然可以提高准确率。

构建组合分类器的方法

- 基本思想：在原始数据上构建多个分类器，然后在分类未知样本时聚集它们的预测结果。
  - 通过处理训练数据集(main)：
    - 根据某种抽样分布，通过对原始数据进行再抽样来得到多个训练集，使用每个训练集训练多个基分类器。
  - 通过处理输入特征(main)：
    - 通过选择输入特征的子集来形成每个训练集。
  - 通过处理类标号：
    - 将类标号随机划分成多个两个不相交的子集a1和a2，将训练数据转换为多个二值分类。通过投票来获得分数最高的类。
  - 通过处理学习算法：
    - 通过在算法的过程中加入随机性，得到不同的算法模型结果。

组合分类器分类结果

- 通过组合基分类器$C_i(x)$的预测来检验样本x进行分类：

$$C^*(x) = Vote(C_1(x), C_2(x), ..., C_k(x))$$

- 可以对单个预测值进行多数表决，或用基分类器的准确率对每个预测值进行加权来得到类标号。

组合方法的一般过程

- 训练多个分类器
  - 根据原数据集创建训练集
  - 由训练集构成基分类器
- 使用多个分类器预测同一个记录
  - 使用多数表决或准确率加权方式得到类标号

装袋bagging

- 装袋(bagging)又称自助聚集(boot strap aggregating)，是一种根据均匀概率分布从数据集中重复抽样(有放回的)的技术。
  - 每个自助样本集都和原数据集一样大；
  - 由于抽样过程是有放回的，因此一些样本可能在同一个训练数据集中出现多次，而其他一些却可能被忽略。
  - 一个自主样本集大约包含63%的原训练数据。

装袋Bagging步骤

- 生成N个样本集
  - 通过有放回抽样的方式抽取N个与原样本集大小一样的样本集
- 训练N个基分类器
  - 通过每个自助样本集分别训练一个基分类器
- 预测样本
  - 通过组合N个基分类器的结果，测试样本被指派到得票最高的类

装袋的适用场景

- 装袋技术主要通过降低基分类器的方差(不稳定性)来改善泛化误差：
  - 基分类器越不稳定(方差越大)，装袋效果越好
  - 若基分类器是稳定的，误差主要是由偏倚导致的，装袋可能降低分类器的性能，因为每个训练集的有效容量比原数据集大约小37%。

提升boosting

- 提升是一个迭代的过程，用来自适应地改变训练样本的分布，使得基分类器聚集在那些很难分的样本上。
  - 提升给每一个训练样本赋一个权值，而且可以在每一轮提升过程结束时自动地调整权值。
    - 增加被错误分类的样本的权值
    - 减小被正确分类的样本的权值
  - 这迫使分类器在随后的迭代中关注那些很难被分类的样本
  - *由于倾向于那些被错误分类的样本(可能是噪声)，提升boost很容易受到过拟合的影响。

AdaBoost

- 基本原理：前一个基本分类器分错的样本会得到加强，加权后的全体样本再次被用来训练下一个基本分类器。
  - AdaBoost步骤：
    - 初始化训练数据的权值分布。如果有N个样本，则每一个训练样本最开始时都被赋予相同的权值：1/N。
    - 训练弱分类器。具体训练过程中，如果某个样本点已经被准确地分类，那么在构造下一个训练集中，它的权值就被降低；相反，如果某个样本点没有被准确地分类，那么它的权值就得到提高。然后，权值更新过的样本集被用于训练下一个分类器，整个训练过程如此迭代地进行下去。
    - 将各个训练得到的弱分类器组合成强分类器。各个弱分类器的训练过程结束后，加大分类误差率小的弱分类器的权重，使其在最终的分类函数中起着较大的决定作用， 而降低分类误差率大的弱分类器的权重，使其在最终的分类函数中起着较小的决定作用。换言之，误差率低的弱分类器在最终分类器中占的权重较大，否则较小。

算法流程

- AdaBoost算法流程：
  - 首先，初始化训练数据的权值分布。每一个训练样本最开始时都被赋予相同的权值：1/N
  - 其次，进行多轮迭代
    - 使用具有权值分布Dm的训练数据集学习，得到基本分类器Gm(x)
    - 计算Gm(x)在训练数据集上的分类误差率
    - 计算Gm(x)的系数，am表示Gm(x)在最终分类器中的重要程度（目的：得到基本分类器在最终分类器中所占的权重）
    - 更新训练数据集的权值分布（目的：得到样本的新的权值分布），用于下一轮迭代
  - 最后，基于am权值组合各个弱分类器

随机森林Random Forest

- 专门为决策树分类器设计的组合方法，组合多棵决策树作出的预测，其中每棵决策树都是基于随机向量的一个独立集合的值产生的。
  - 特殊形式：使用决策树装袋是随机森林的特例，通过随机从原训练集中有放回地选取N个样本，将随机性加入构建模型的过程中。

随机森林的构建方法

- 每棵决策树都使用随机向量合并到树的增长过程中
  - 可以使用装袋技术提升训练集的随机性。
  - Forest-RI：随机选择F个(而不是所有)输入特征来对某一棵决策树的结点进行分裂，让树完全增长而不进行任何修剪，有助于减少结果树的偏倚。最后使用多数表决的方法组合预测。
  - Forest-RC：若原始特征数目太小，可以通过线性组合来创造新的特征，这些输入特征用区间[-1,1]上的均匀分布产生的系数进行线性组合，在每个节点产生F个随机组合，选择其中最好的来分裂节点。
  - 其他方式：每次分裂随机选择最好的分裂特征中的一个。

随机森林效果的因素

- 随机森林分类效果（错误率）与两个因素有关：
  - 森林中任意两棵树的相关性：相关性越大，错误率越大；
  - 森林中每棵树的分类能力：每棵树的分类能力越强，整个森林的错误率越低。
- 随机森林的参数F：
  - 减小特征选择个数F，树的相关性和分类能力也会相应的降低；增大F，两者也会随之增大。所以关键问题是如何选择最优的F（或者是范围），这也是随机森林唯一的一个参数。

The end