实战项目 1_数据挖掘算法改进

说明：本文系作者原创，他人不得抄袭！

本章是全文的重点章节之一，全面阐述了本文的重点工作内容：算法改进。首先提出算法改进原理、算法步骤，然后从算法准确率、运行时间对原始算法、改进算法进行综合对比分析，最后总结改进算法的优缺点与优化空间。

§1.1 k-NN 近邻算法

由 2.1 节分析可知，k-NN 算法最主要的时间开销在于：算法训练全部发生在算法分类时期，对于每一个新的待分实例，总是要基于整个数据集来计算所有样本到待测实例的欧氏距离，然后再进行排序，这是一种典型的消极学习方法。下面主要从距离计算、化消极学习方法为积极学习方法 2 方面来改进算法。

1、算法改进

（1）距离计算化简

原始 k-NN 算法基于欧氏距离公式计算样本间距离，公式如下：

$$Dist\ (\mathbf{X},\mathbf{Y}) = \sqrt{\sum_{i=1}^{n}\left(X_i - Y_i\right)^2} \tag{1}$$

分析公式可知：1)公式由先平方、再累加、后开方计算得来。

2）对于高纬度数据来说，计算复杂度高，资源开销大。

3）容易产生溢出现象：如果 2 个样本距离非常“近”，那么开方时容易产生下溢，造成距离结果为 0；

4）从算法目的来说，是基于距离计算来比较样本距离，从而选出最佳“近邻”。因为函数 f(x)=x**2(x>=0)与 f(x)=x**1/2 (x>=0)在[0,+&)都是严格单调递增函数，所以在[0,+&)上利用 f(x)=x 函数单调性比较 2 个数的大小与利用 f(x)=x**1/2 函数单调性比较 2 个数的大小具有等价作用，即比较结果总是相同的。同理又因为 f(x)=x(x>=0)和 f(x)=x**2(x>=0)在[0,+&)上也是严格单调递增的，所以基于 f(x)=x**2(x>=0)累加求和比较大小与基于 f(x)=x(x>=0)累加求和比较具有等价效果。因此，距离计算可简化为：先取绝对值、再累加。

（2）化消极学习方法为积极学习方法

k-NN 近邻算法是一种典型的消极学习方法，在算法的训练阶段只是简单的存储数据，而所有算法训练都在分类时完成，对于大数据量计算，无疑带来了巨大资源开销。因此，本文将 k-NN 算法分类阶段的分类工作转移到算法训练阶段，并且为距离增加索引排序，转化为“一次训练，多次使用用”的积极学习方法。

为距离增加索引排序，就必须有基准点。一般的论文是随机选取基准点，这样 k-NN 算法的效果很有可能随着基准点的不同而有所不同，即 k-NN 算法的结果是不稳定的。为了有效进行索引，就必须事先确定“好”的基准点。本文从维度（假设维度为 N）方面考虑，为了更好地“定位”待测实例的最佳近邻所在区域，选取的基准点可以是 N 个（至少 1 个），并且这 N 个基准点分别“独占”一个维度（即 N 个基准点向量相互垂直，这样可以最小化确定近邻所在区域）。以 2 维举例：在 0-1 标准化数据后，可以选取（0，0）和（0,1）作为基准点，这样用 2 个相互垂直的向量就可以将待测实例的近邻定位在一个“很小”的平面区域中，如下图所示：

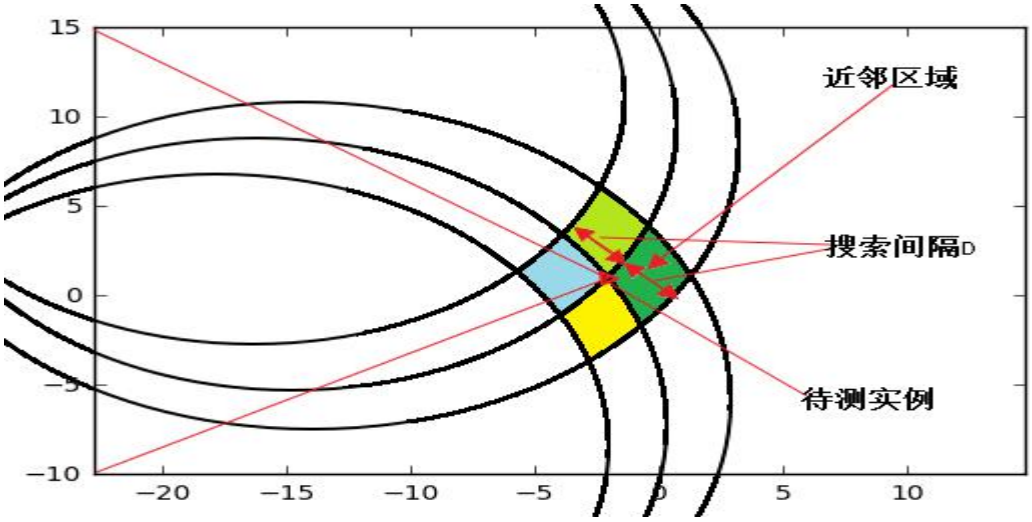

图 3-1 改进 k-NN 算法近邻区域确定（放大）示意图

式中，一个基准点为原点，坐标为 O(-22,-10)，另一个基准点坐标为 A(-22，15)。实际算法应用中，会将数据先进行 0-1 标准化处理，在二维空间中，选择（0，0）和（0,1）两个垂直向量来作为基准点。使用这 2 个基准点的好处是：

A.向量无需做相减运算

B.2 个向量相互垂直，可最小化确定待测实例近邻所在区域

（3）算法分类阶段

首先 0-1 标准化数据后，分别计算出 N 个基准点到待测实例的样本距离 R_i，i=1,2,...,N(N 为数据特征维度数)。又因为 0-1 标准化数据后，依据本文计算样本间距离平方和最大为 N（单位为 1），所以，给定用户输入参数 S（S 是将 N 均分的段数），搜索间隔计算为：D=N**0.5/S。然后，在排好序的 K 条索引链上，分别定位出距离区间[R_i-D, R_i+D](i=1,2,...,N)内的起始、末尾样本索引，接着查找出 N 条索引链上所有共同的样本点，最后统计出所有共同样本点的最频繁类别标号，作为待测实例的分类标号。

2、算法步骤

输入：数据集 dataSet（特征维度为 N）、分段参数 S

过程：1）数据预处理：处理缺失值、去除冗余特征和异常点、0-1 标准化

2）决定 N 个基准点

3）分别计算出所有样本到 N 个基准点的距离，并按照从小到大排序，得到 N 条距离索引链

4）分别计算出待测实例 X 到 N 个基准点的距离 R_i(i=1,2,...,N)

5）计算搜索间隔间隔 D=N**0.5/S

6）分别在 N 条索引链上定位出距离区间[R_i-D， R_i+D]范围内样本的起始、末尾样本索引

7）找出各索引链上起始、末尾索引号间内的所有共同样本点

8）统计出所有共同样本点的最频繁类别标号，返回该类别标号，作为待测实例的预测结果

输出：待测实例类别标签

3、结果对比分析

基于不同 UCI 数据集对原始、改进 k-NN 算法（统一取 2 个基准点）十字交叉验证，测试算法准确率、运行时间，结果如下图：

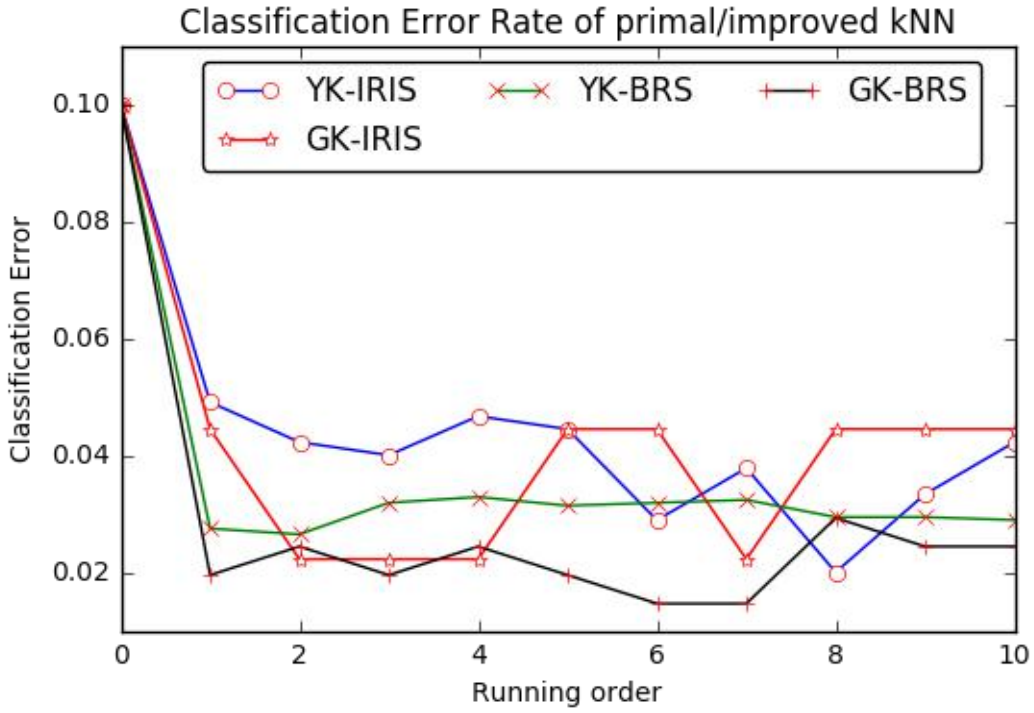

图 3-1 原始 kNN 算法、改进 kNN 算法分类错误率折线图

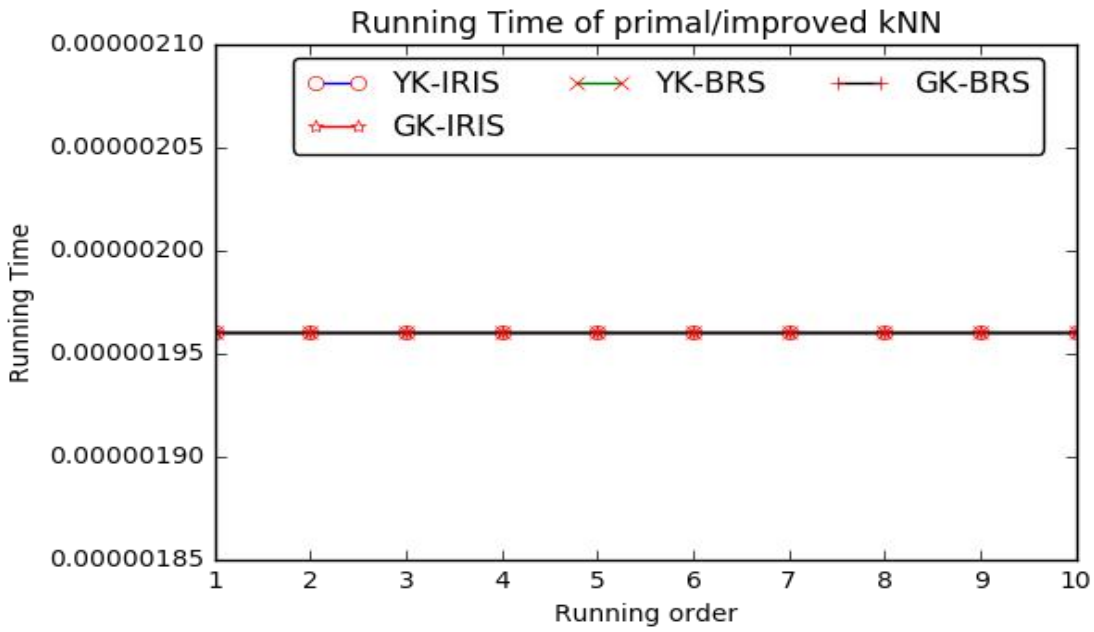

图 3-2 原始 kNN 算法、改进 kNN 算法分类错误率折线图

从图 3-1 可以看出，改进算法的准确率整体要比原始算法要好，且算法运行时间没有增加。综合说明，该改进 kNN 算法是有效的，具有一定的使用价值。

4、算法总结

优点：

1）改进算法不再基于 K 个近邻来预测待测实例类别，而是基于搜索间隔来定位待测实例的近邻所在区域；

2）改进 k-NN 算法是一种积极的学习方法，“一次训练多次使用”；

3）基准点个数可以小于特征维度，至少为一个。2 个基准点可以缩小待测实例的近邻定位区域，使得算法准确率更高。一般来说，2 个基准点足够使用；

4）基准点是可以事先确定的，消除随机确定的算法结果不稳定影响。

缺点：

1)对于大数据集，排序可能会带来很大的资源开销；

2)距离间隔即距离等分段数需要多次尝试才能获得较好的分类效果。但是，这比近邻数目 K 的确定还是更好确定、更符合我们的逻辑。

3)共同点的查找对于大数据集来说，也会需要很大的时间开销。如果能良好利用排序算法和并行机制的话，对于大数据集也会取得良好的分类效果；

4)改进算法对孤立点敏感，数据集不紧凑也会使得算法准确率有所下降。

§1.2 决策树算法

决策树算法 C4.5 在 2.3 节已经分析过，本章主要从优化算法公式、提高算法运行效率方面改进算法。

1、算法改进

C4.5 算法信息增益计算公式和属性分离信息计算公式可统一为：

$$Ent(X) = -\sum_{i=1}^{v} P_i \log_2 P_i = -\sum_{i=1}^{v} \frac{X_i}{S} \log_2 \frac{X_i}{S} \qquad (2)$$

式中，X 为特征，v 是特征 X 取值的总数，P_i 为特征 X 取值为 i 时的样本数在当前总样本中所占频率，S 为当前样本总数，X_i 为特征 X 取值为 i 时的样本数。

分析公式，不难发现：

1）公式结果由先相除、然后取对数、最后相乘计算而来。

2）除法运算容易发生下溢，而且大量反复的除法运算很消耗时间；

优化公式：

$$Ent(X) = -\sum_{i=1}^{v} P_i \log_2 P_i = -\sum_{i=1}^{v} \frac{X_i}{S} \log_2 \frac{X_i}{S} \qquad (3)$$

$$\Leftrightarrow -\sum_{i=1}^{v} \frac{X_i}{S}(\log_2 X_i - \log_2 S)$$

$$\Leftrightarrow -\sum_{i=1}^{v} (\frac{X_i}{S} \log_2 X_i - \frac{X_i}{S} \log_2 S)$$

$$\Leftrightarrow -\frac{1}{S}\sum_{i=1}^{v} X_i \log_2 X_i + \frac{1}{S}\sum_{i=1}^{v} X_i \log_2 S$$

$$\Leftrightarrow -\frac{1}{S}\sum_{i=1}^{v} X_i \log_2 X_i + \log_2 S$$

式中，$S = \sum_{i=1}^{v} X_i$，i=[1,2,...,v]

2、算法步骤

参照 C4.5 算法步骤

3、结果对比分析

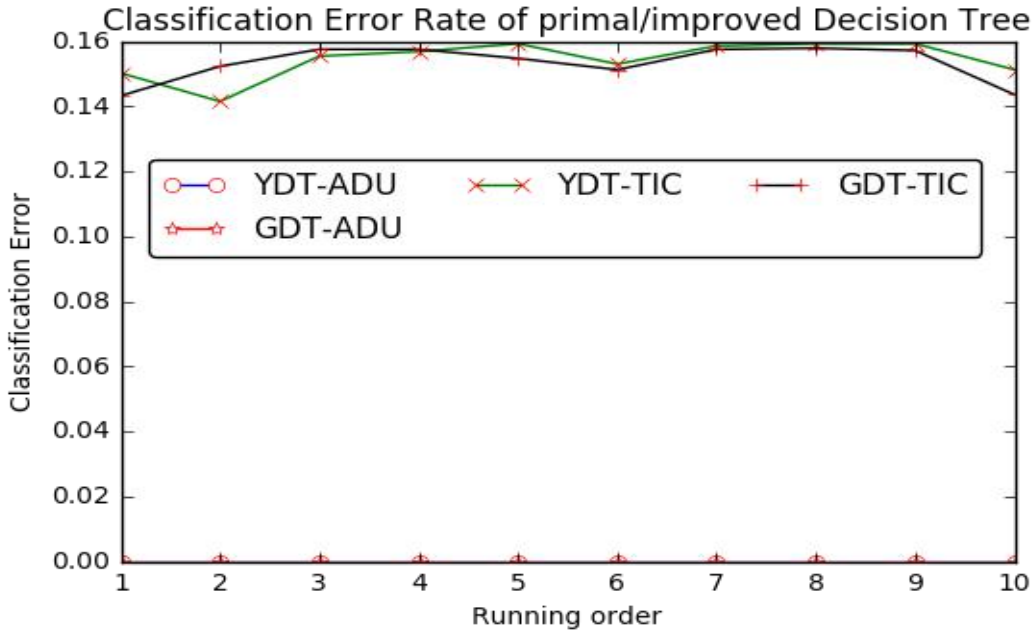

图 3-3 原始 C4.5 算法、简化 C4.5 算法分类错误率折线图

分析：从图 3-3 可以看出：两种实现方式等价的算法，其分类错误率也是几乎一致的，说明，简化的算法与原始算法等效。

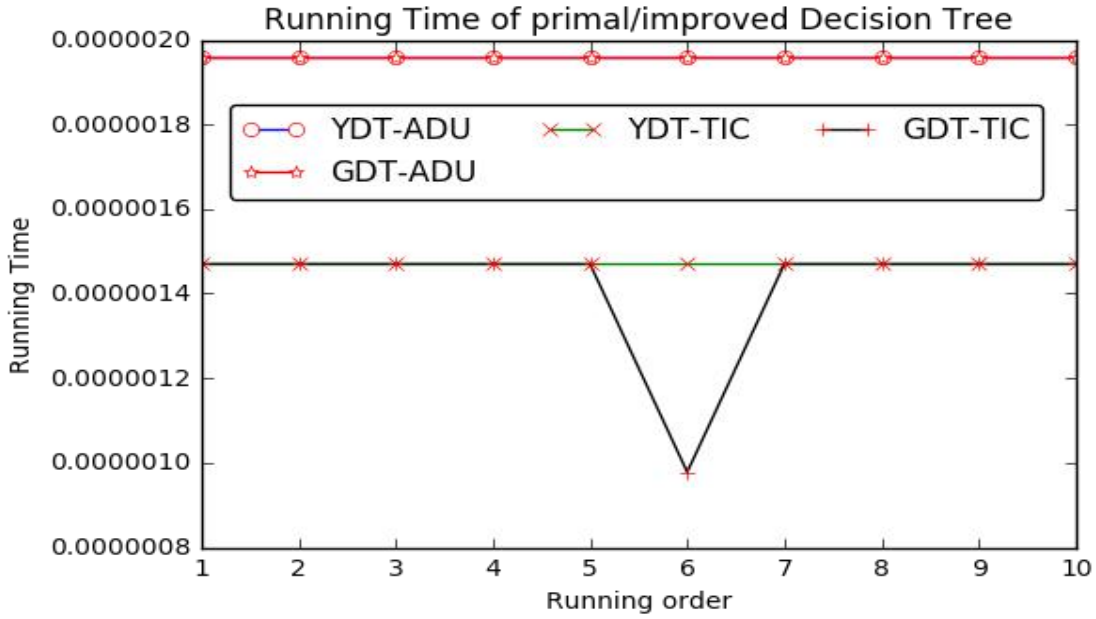

图 3-4 原始 C4.5 算法、简化 C4.5 算法运行时间折线图

分析：从图 3-4 可以看出：原始算法、简化算法的运行时间几乎一致，但是，简化算法有时会比原始算法运行效率更高，说明，消除算法的除法是有利于提高算法效率的。

4、算法总结

优点：1）消除公式中的除法运算，提高运算效率

2）公式运算结果更准确，更简化；

3）算法代码复用率高，提高算法效率；

缺点：同 C4.5 算法。

§1.3 本章小结

本章主要介绍 kNN 算法、决策树算法的改进原理、算法步骤、算法对比分析和优缺点总结这四部分内容，而且，从分析和优缺点可分析出，改进算法仍然有进一步的优化空间。