Weka 开发[15] ZeroR 源代码介绍（入门篇）

作者：屈伟/Koala++

以前写的 ID3 虽然比较简单，但是对于刚接触的人也许也不是那么简单，这次介绍 Weka 中默认的分类器 ZeroR，用这个入门应该是比较好的选择。

首先提一下，ZeroR 很多人以为是乱猜，实际是如果类别是离散值，就返回最有可能的类别，如果是连续值，则返回类别的平均值。

下面函数的前面两句话哪个分类器都有，就不说了。这个函数简单地让我不知道怎么讲了。m_Counts 如果是离散(Nominal)的类别，就把它初始化为一个有类别数大小的一维数组，如果是类别是连续(Numeric)值，那就是一个值。

下面 while 循环，这种写法是枚举数据集中的每一个样本，如果是离散值，m_Counts 相应的类别下标加上这个样本的权重（这里不用太深究到底什么是权重，你可以认为所有的样本权重都是 1），如果是连续值，m_Counts 加上类别值乘以这个样本的权重。

统计完每一个样本，如果是连续值，那么就用 m_Counts 除以总权重，其实这就是高中时学的加权平均的计算方法。如果是离散值，那就是类别出现最多(懒得去想 weight 的事了)的类别作为 m_ClassValue。

```
public void buildClassifier(Instances instances) throws Exception {
    // can classifier handle the data?
    getCapabilities().testWithFail(instances);

    // remove instances with missing class
    instances = new Instances(instances);
    instances.deleteWithMissingClass();

    double sumOfWeights = 0;

    m_Class = instances.classAttribute();
    m_ClassValue = 0;
    switch (instances.classAttribute().type()) {
    case Attribute.NUMERIC:
        m_Counts = null;
        break;
    case Attribute.NOMINAL:
        m_Counts = new double[instances.numClasses()];
        for (int i = 0; i < m_Counts.length; i++) {
            m_Counts[i] = 1;
        }
        sumOfWeights = instances.numClasses();
        break;
    }
    Enumeration enu = instances.enumerateInstances();
    while (enu.hasMoreElements()) {
```

```
        Instance instance = (Instance) enu.nextElement();
        if (!instance.classIsMissing()) {
            if (instances.classAttribute().isNominal()) {
                m_Counts[(int) instance.classValue()] +=
                        instance.weight();
            } else {
                m_ClassValue += instance.weight() *
                        instance.classValue();
            }
            sumOfWeights += instance.weight();
        }
    }
    if (instances.classAttribute().isNumeric()) {
        if (Utils.gr(sumOfWeights, 0)) {
            m_ClassValue /= sumOfWeights;
        }
    } else {
        m_ClassValue = Utils.maxIndex(m_Counts);
        Utils.normalize(m_Counts, sumOfWeights);
    }
}
```

分类一个样本，当然就是返回 m_ClassValue 值了（我希望这么简单的东西，你不至于还不知道是什么吧）：

```
public double classifyInstance(Instance instance) {
    return m_ClassValue;
}
```