Weka[2] Classifier 类

作者：Koala++/屈伟

这次介绍如何利用 weka 里的类对数据集进行分类，要对数据集进行分类，第一步要指定数据集中哪一列做为类别，如果这一步忘记了（事实上经常会忘记）会出现“Class index is negative (not set)!”这个错误，设置某一列为类别用 Instances 类的成员方法 setClassIndex，要设置最后一列为类别则可以用 Instances 类的 numAttributes()成员方法得到属性的个数再减 1。

然后选择分类器，比较常用的分类器有 J48，NaiveBayes，SMO（LibSVM 有 Java 版的，可以在 weka 中使用，但要设置路径），训练分类器使用 J48 的 buildClassifier（注意 J48 还有别的分类器它们都继承自 Classifier 类，使用方法都差不多），分类数据用 J48 类中的 classifyInstance 方法，例中使用的数据集为 contact-lenses.arff，分类结果为 2.0，结果为 2.0 的原因是：首先用文本编辑器打开数据集，有一行为@attribute contact-lenses {soft, hard, none}，而第一个样本为 young, myope, no, reduced, none，最后一列为类别，也就是 contact-lences 为类别，第一个样本的类别为 none，在属性说明中 none 为第二个所以为 2.0（从 0 开始数）。

```
package instanceTest;

import java.io.FileReader;

import weka.classifiers.trees.J48;
import weka.core.Instances;

public class ClassifierTest
{
    private Instances m_instances = null;

    public void getFileInstances( String fileName ) throws Exception
    {
        FileReader frData = new FileReader( fileName );
        m_instances = new Instances( frData );

        m_instances.setClassIndex( m_instances.numAttributes() - 1 );
    }

    public void classify() throws Exception
    {
        J48 classifier = new J48();
        //NaiveBayes classifier = new NaiveBayes();
        //SMO classifier = new SMO();

        classifier.buildClassifier( m_instances );
        System.out.println( classifier.classifyInstance( m_instances.instance( 0 ) ) );
    }

    public static void main( String[] args ) throws Exception
    {
        ClassifierTest ctest = new ClassifierTest();
```

```
        ctest.getFileInstances( "F://Program
            Files//Weka-3-4//data//contact-lenses.arff");
        ctest.classify();
    }
}
```