

## Weka[21] IBk 源代码分析

作者: Koala++/屈伟

如果你没有看上一篇 IB1, 请先看一下, 因为重复的内容我在这里不会介绍了。  
直接看 buildClassifier, 这里只列出在 IB1 中也没有出现的代码:

```
try {
    m NumClasses = instances.numClasses();
    m ClassType = instances.classAttribute().type();
} catch (Exception ex) {
    throw new Error("This should never be reached");
}
// Throw away initial instances until within the specified window size
if ((m WindowSize > 0) && (instances.numInstances() > m WindowSize)) {
    m Train = new Instances(m Train, m Train.numInstances()
        - m WindowSize, m WindowSize);
}

// Compute the number of attributes that contribute
// to each prediction
m NumAttributesUsed = 0.0;
for (int i = 0; i < m Train.numAttributes(); i++) {
    if ((i != m Train.classIndex())
        && (m Train.attribute(i).isNominal() || m Train
            .attribute(i).isNumeric())) {
        m NumAttributesUsed += 1.0;
    }
}

// Invalidate any currently cross-validation selected k
m_kNNValid = false;
```

IB1 中不关心 m\_NumClasses 是因为它就找一个邻居, 当然就一个值了。m\_WindowSize 是指用多少样本用于分类, 这里不是随机选择而是直接选前 m\_WindowSize 个。这里下面是看有多少属性参与预测。

KNN 也是一个可以增量学习的分器量, 下面看一下它的 updateClassifier 代码:

```
public void updateClassifier(Instance instance) throws Exception {

    if (m Train.equalHeaders(instance.dataset()) == false) {
        throw new Exception("Incompatible instance types");
    }
    if (instance.classIsMissing()) {
        return;
    }
    if (!m DontNormalize) {
        updateMinMax(instance);
    }
    m Train.add(instance);
    m_kNNValid = false;
    if ((m WindowSize > 0) && (m Train.numInstances() > m WindowSize)) {
        while (m Train.numInstances() > m WindowSize) {
            m Train.delete(0);
        }
    }
}
```

```
}
```

同样很简单，`updateMinMax`，如果超出窗口大小，把窗口中的第一个样本删除。