Java---- Machine- Learning

Published on *Java Machine Learning Library (Java-ML)* (http://java-ml.sourceforge.net)

By *Thomas Abeel*
Created *12/16/2008 - 12:02*

Clustering

This chapter will provide documentation on clustering algorithms, cluster evaluation measures and other topics related to clustering data. We assume that you are already familiar with the topics discussed in the Getting started [1] chapter.

Clustering basics

A clustering algorithm creates a division of the orginal dataset. In Java-ML this is done with the method *cluster* of the Clusterer interface.

Creating and running a clustering algorithm

```
. /* Load a dataset */
. Dataset data = FileHandler.loadDataset(new File [2]("iris.data"), 4, ",");
. /* Create a new instance of the KMeans algorithm, with no options
.    * specified. By default this will generate 4 clusters. */
. Clusterer km = new KMeans();
. /* Cluster the data, it will be returned as an array of data sets, with
.    * each dataset representing a cluster. */
. Dataset[] clusters = km.cluster(data);
```

[Documented source code] [3]

The code above will load the example iris data set. Next it creates an instance of the K-means algorithms and uses it to cluster the data. The results are returned in an array of Datasets where each Dataset represents a cluster.

Note that there is no guarantee that all original Instances will occur in the clusters or that each Instance occurs only once. Some algorithms allow overlapping clusters, some algorithms allow that 'noisy' datapoints are removed. This is algorithm specific and you can find more information on the API page for each algorithm.

Cluster evaluation

Java-ML provides a large number of cluster evaluation measures that are provided in the package net.sf.javaml.clustering.evaluation. All scores are measures for the quality of the clustering, i.e. how well it reflects the properties of the data. Mostly they try to quantify how well the data is separated in logical units by the clustering algorithm.

All scores implement the *double score(Dataset[]clusters)* method. This method returns a score for an array of datasets that is returned from a clustering algorithm. Typical usage is illustrated in the code snippet below.

```
. /* We load some data */
. Dataset data = FileHandler.loadDataset(new File [2]("iris.data"), 4, ",");
. /* We create a clustering algorithm, in this case the k-means
.  * algorithm with 4 clusters. */
. Clusterer km=new KMeans(4);
. /* We cluster the data */
. Dataset[] clusters = km.cluster(data);
. /* Create a measure for the cluster quality */
. ClusterEvaluation sse= new SumOfSquaredErrors();
. /* Measure the quality of the clustering */
. double score=sse.score(clusters);
```

[Documented source code] [4]

Weka clustering

Clustering algorithms from Weka can be accessed in Java-ML through the WekeClusterer bridge. This class makes it easy to use a clustering algorithm from Weka in Java-ML.

In the example below, we load the iris dataset, we create a clusterer from Weka (XMeans), we wrap it in the bridge and use the bridge to do the clustering.

```
. /* Load data */
. Dataset data = FileHandler.loadDataset(new File [2]("iris.data"), 4, ",");
. /* Create Weka classifier */
. XMeans xm = new XMeans();
. /* Wrap Weka clusterer in bridge */
. Clusterer jmlxm = new WekaClusterer(xm);
. /* Perform clustering */
. Dataset[] clusters = jmlxm.cluster(data);
. /* Output results */
. System [5].out.println(clusters.length);
```

[Documented source code] [6]

Source URL (retrieved on *07/23/2009 - 08:16*): http://java-ml.sourceforge.net/content/clustering

Links:
[1] http://java-ml.sourceforge.net/content/getting-started
[2] http://www.google.com/search?hl=en&q=allinurl:file java.sun.com&btnI=I'm Feeling Lucky
[3] http://java-ml.sourceforge.net/src/tutorials/clustering/TutorialKMeans.java
[4] http://java-ml.sourceforge.net/src/tutorials/clustering/TutorialClusterEvaluation.java
[5] http://www.google.com/search?hl=en&q=allinurl:system java.sun.com&btnI=I'm Feeling Lucky
[6] http://java-ml.sourceforge.net/src/tutorials/tools/TutorialWekaClusterer.java