

We propose a community project to offer reference implementations of published machine learning algorithms in a uniform, portable, readable and documented way.

What is Java-ML?

1. Machine learning techniques:
 - 18 Clustering algorithms
 - 15 Cluster evaluation functions
 - 3 Classifiers
 - 20 Distance functions
 - 4 Dataset filters
2. Common interface and modular design
3. Reference implementations
4. Well documented
5. Library style, no GUI
6. Open Source

Motivation

- Lack of a (good) machine learning library:
 - Reference implementations
 - Good, understandable documentation
 - With small, simple and straightforward to use API
- Existing libraries:
 - WEKA: is GUI oriented (>75% code) (alive)
 - MLC: 1997 latest source code
 - MLJ: 3 algorithms, 2002 (dead project)
 - YALE: is GUI oriented and is for experiment design
 - Cougar2: extension of WEKA and YALE (dead project)
- Provide a single place for machine learning algorithms

Code & Documentation

- **Wiki:** documentation, references
- **Forum:** interaction between developers, with end-users, ...
- **Bugtrackers:** keep track of bugs
- **Annotated source code:** java doc and explanation of the algorithm
- **Generated API:** from the source code
- **Changelog:** with Fisheye
- **SVN repository:** source code



Recent Changes

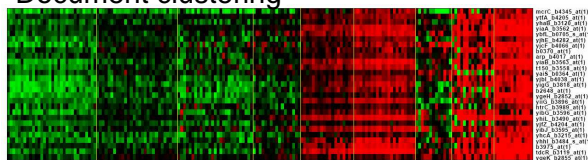
- 735 by ThomasAbeel 16 August 2007, 04:58:48 - 0503 (10) added support for distance
- 734 by ThomasAbeel 16 August 2007, 04:49:46 - 0503 (10) added some distance measures
- 733 by ThomasAbeel 16 August 2007, 04:13:11 - 0503 (10) added some distance measures

```

/**
 * Returns a new array with each position the clusterIndex in the dataset belongs.
 *
 * @param centroids the current set of cluster centroids, will be the new assignment
 * @param assignments the new assignment of all instances to the
 * @param output the cluster center, this will be modified
 */
private boolean recomputeCentroids(centroid[] centroids, int[] assignments, int[] output) {
    boolean changed = false;
    for (int i = 0; i < centroids.length; i++) {
        output[i] = new float[output.length];
        for (int j = 0; j < assignments.length; j++) {
            if (assignments[j] == i) {
                output[i].addDistance(centroids[i]);
            }
        }
        centroid[] newCentroids = DataSetTools.getCentroids(output, i, centroids);
        if (newCentroids[i].equals(centroids[i])) {
            changed = true;
        }
    }
    return changed;
}
    
```

Applications

- Gene clustering for gene family construction
- Gene expression profile clustering
- Core promoter clustering
- Document clustering



The present and future

- Website: <http://java-ml.sf.net>
- 70k lines of code
- Looking for:
 - algorithm implementations
 - developers and collaborators