PR1/2 Demo

Christian Permann

Faculty of Computer Science, University of Vienna,
Währinger Straße 29, 1090 Vienna

28.10.2018

Milestones P1

- Create a basic visualization tool. (done)
- Define interface for data manipulation. (done)
- Be compatible with ELKI. (in progress)
- Allow data import. (CSV available)
- Create data generation logic. (ELKI generator available, in progress)
- Possibly plan for dim-reduction for visualization and implement it. (PCA, etc)

The existing tool

The current iteration can visualize points, allows for generating points via the ELKI XML-based generator and was tested for performance with up to 300.000 points for a smooth experience.

Milestones P2

- Get to know ELKI. (in progress)
- Enable running (ELKI) algorithms from the custom tool.
- Properly save and display clustered data.
- Enable defining presets and running multiple clusterings.
- Allow for parameter space exploration within results.

The Tool

Demo

The Tool - Data Generation Interface

```
public interface IGenerator {
    JPanel getOptionsPanel();
    String getName();
    boolean canSimpleGenerate();
    boolean generate(PointContainer container);
    boolean canClickGenerate();
    boolean generate(double[] point,PointContainer container);
}
```

Figure 1: The interface for data manipulation

Possible Improvements on Clustervision

Clustervision ranks clustering results according to some quality measures but does not look further into the similarity between clusterings. Their ranking does think about not showing too similar clusterings at a high rank but the chosen similarity only looks at the labels, whereas it would be important to consider information like if two points, which were in the same cluster, are now in the same cluster as well. Clustering the cluster results with such a metric might be computationally expensive but may show more relevant differences in the results. Also defining a good distance metric for this difference seems non-trivial.

Data-sets

- real data-sets[1]
- well-known sets (benchmark sets)[2]
- generated sets with different distributions

ELKI Algorithms

- DBSCAN: epsilon, minpts
- GeneralizedDBSCAN: coremodel, corepred, npred
- LSDBC: alpha, k(knn)
- GriDBSCAN: epsilon, gridwidth, minpts
- OPTICSHeap: epsilon, minpts
- DeLiClu: minpts
- FastOPTICS: (Random projection) index, minpts
- KMeansLloyd: initializer, k, maxiter
- many many more... [3]


"Uci machine learning repository." https://archive.ics.uci.edu/ml/datasets.html?area=&att=&format=&numAtt=&numIns=&sort=nameUp&task=clu&type=&view=table.
Last-Accessed: 2018-10-28.


P. Fränti and S. Sieranoja, "K-means properties on six clustering benchmark datasets," 2018.


"Elki algorithms." https://elki-project.github.io/algorithms/.
Last-Accessed: 2018-10-28.