The Journal of Open Source Software

CluSim: a python package for calculating clustering similarity

Alexander J. Gates[1] **and Yong-Yeol Ahn**[2,3]

1 Department of Physics, Northeastern University, Boston, 02115, USA **2** Department of Informatics, Indiana University, Bloomington, 47408, USA **3** Program in Cognitive Science, Indiana University, Bloomington, 47408, USA

DOI: 10.21105/joss.01264

Software

- Review
- Repository
- Archive

Submitted: 21 January 2019
Published: 21 March 2019

Summary

Clustering is a primary method to reveal the structure of data (Jain, Murty, & Flynn, 1999). To understand, evaluate, and leverage data clusterings, we need to quantitatively compare them. Clustering comparison is the basis for method evaluation, consensus clustering, and tracking the temporal evolution of clusters, among many other tasks. For instance, the evaluation of a clustering method is usually achieved by comparing the method's result to a planted reference clustering, assuming that the more similar the method's solution is to the reference clustering, the better the method. Despite the importance of clustering comparison, no consensus has been reached for a standardized assessment; each similarity measure rewards and penalizes different criteria, sometimes producing contradictory conclusions.

Clustering similarity measures can be classified based on the cluster types: i) *partitions* that group elements into non-overlapping clusters, ii) *hierarchical clusterings* that group elements into a nested series of partitions (a.k.a. dendrogram), or iii) *overlapping clusterings* with elements belonging to multiple clusters. One approach to aid with the interpretation of the similarity score establishes a baseline in the context of a random ensemble of clusterings. Such a correction procedure requires two choices: *a model for random clusterings* and *how clusterings are drawn from the random model*. With few exceptions, similarity measures are only designed to compare clusterings of the same type, and the decisions required for the correction procedure are usually ignored or relegated to the status of technical trivialities (Gates & Ahn, 2017).

Here, we introduce *CluSim*, a python package providing a unified library of over 20 clustering similarity measures for partitions, dendrograms, and overlapping clusterings. To our knowledge, this package constitutes the first collection of clustering similarity measures for all three clustering types and extended access to random models of clusterings (Gates, Wood, Hetrick, & Ahn, 2018). We illustrate the use of the package through two examples: an evaluation of measure behavior with variation in 3 clustering properties (membership, cluster sizes, and number of clusters) and a clustering comparison of Gene Expression data in the context of different random models.

Examples

The basic class in the *CluSim* package is a *Clustering*, or an assignment of labeled elements (i.e. data points or network vertices) into clusters (the groups). *Hierarchical Clusterings* also contain a dendrogram, or more generally an acyclic graph, capturing the nested structure of the clusters. In *CluSim*, a *Clustering* can be instantiated from 7 different common formats, including full support for *scipy*, *scikit-learn*, and *dendropy* clustering formats (Pedregosa et al., 2011; Sukumaran & Holder, 2010).

CluSim provides more than 20 clustering similarity and distance measures for the comparison between two *Clusterings*. All similarity measures produce a score in the range $[0, 1]$, where 1 indicates identical clusterings and 0 indicates maximally dissimilar clusterings. See the online documentation for a detailed list and mathematical definitions of these similarity measures.

The clustering similarity measures presented here differ in how each evaluates the trade-offs between variation in three primary characteristics of clusterings: the grouping of elements into clusters, the number of clusters, and the size distribution of those clusters (Gates et al., 2018). To illustrate these trade-offs, we present three simple examples in Fig. 1. In the first example, 1,024 elements are grouped into 32 clusters of equal size and compared against a similar clustering with a fraction of the elements randomly exchanged between the clusters, keeping the same cluster sizes. As seen in Fig. 1**a**), all similarity measures decrease as the fraction of shuffled elements increases, but the measures differ on whether they can differentiate between clusterings that are completely random, or if there is a discontinuous jump in the similarity value. In the second example, 1,024 elements are grouped into 32 clusters of equal size and compared against a similar clustering with increasing cluster size skew-ness. As seen in Fig. 1**b**), some similarity measures decrease as the fraction cluster size heterogeneity increases, while others increase. Finally, in the third example, 1,024 elements are grouped into 8 clusters of equal size and compared against a similar clustering with an increasing number of equal-sized clusters. As seen in Fig. 1**c**), most similarity measures decrease as the number of clusters increases, but the normalized mutual information increases. For more details and an extended interpretation of these experiments, see the analysis in Gates et al (2019) (Gates et al., 2018). Ultimately, the practitioner should choose a clustering similarity measure that is sensitive to the relevant features of the clusterings for the problem at hand.

To facilitate comparisons within a set of clusterings, it is often argued to consider clustering similarity in the context of a random baseline (Gates & Ahn, 2017; Hubert & Arabie, 1985; Vinh, Epps, & Bailey, 2009). The *CluSim* package provides both analytic and statistical sampling methods for calculating such a correction for chance. Analytic solutions are available for the Rand index and Normalized Mutual Information using five random models: the permutation model, both one-sided and two-sided models for clusterings with a fixed number of clusters, and both one-sided and two-sided models for all random clusterings. See Gates & Ahn (2017) (Gates & Ahn, 2017) for detailed derivations and explanations of the differences between clustering random models. For all other similarity measures, the correction for chance is estimated by randomly sampling the random ensemble of *Clusterings* using the provided random Clustering generators.

A typical comparison using a correction for chance is illustrated in Fig. 2. Agglomerative Hierarchical Clustering was applied to gene expression data from (Souto, Costa, Araujo, Ludermir, & Schliep, 2008) and compared to the true classification of cancer types using the Rand Index (0.5, red). To determine if the Rand Index of 0.5 is indeed a good score, it is assessed relative to the distribution of pairwise comparisons amongst a sample of 100 random *Clusterings* from the Permutation model (blue, see [Hubert1985adjrand]) with mean Rand index of 0.44 (black). Thus, the naive assessment would conclude that Agglomerative Hierarchical Clustering has performed better than would be expected by chance. However, the more appropriate random model for this scenario is provided by the one-sided model with a Fixed Number of Clusters (see [Gates2017impact]), since Agglomerative Hierarchical Clustering fixes the number of clusters but not their sizes, and the comparison is made to a ground truth clustering. The distribution of pairwise comparisons amongst a sample of 100 random samples from this random model (blue)

The Journal of Open Source Software

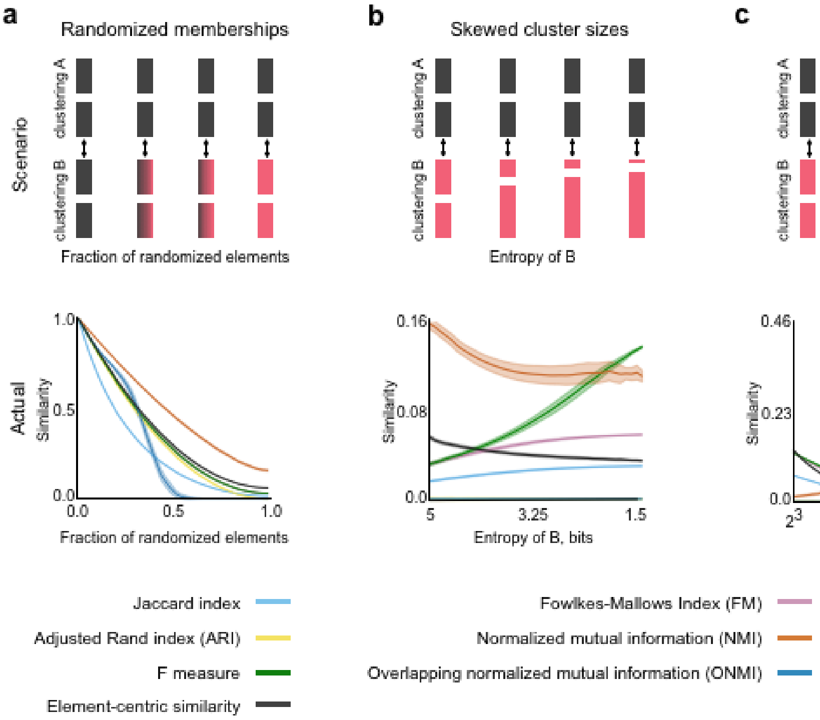

Figure 1: Three clustering similarity scenarios illustrate the trade-offs for clustering comparisons. 1,024 elements are assigned to clusters according to the following scenarios (**a**-**c**) and compared using the Jaccard index, adjusted Rand index, the F measure, normalized mutual information, overlapping normalized mutual information, and the element-centric similarity. All results are averaged over 100 runs; error bars denote one standard deviation. **a**, A clustering with 32 equal-sized clusters is compared to a randomized version of itself where elements are exchanged. **b**, A clustering with 32 equal-sized clusters is compared against clusterings with increasing cluster size skew-ness. **c**, A clustering with 8 equal-sized clusters is compared against a clustering with increasing number of clusters.

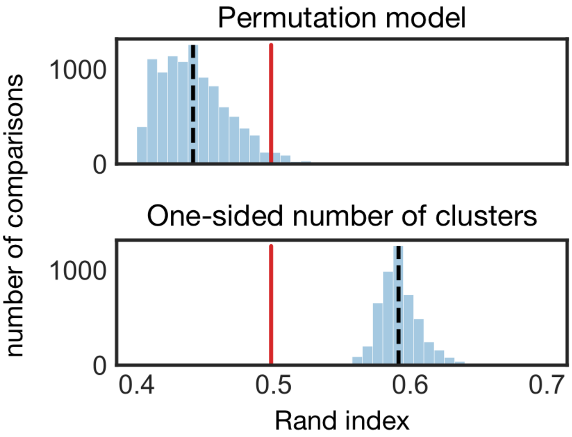

Figure 2: Evaluating clustering comparisons w.r.t. random models.. A comparison using the Rand Index between the classification of cancer types and clustering labels derived using Hierarchical Clustering on gene expression data (0.5, red). above, Pairwise comparisons between samples from the Permutation model (blue, see [Hubert1985adjrand]) with mean 0.44 (black). below, Pairwise comparisons between samples from the one-sided model with a Fixed Number of Clusters (blue, see [Gates2017impact]) with mean 0.59 (black). The Permutation model suggests Hierarchical Clustering is more similar to the ground truth than a random clustering, while the one-sized fixed number of clusterings model, the more appropriate model for this scenario, reveals that the result is less similar than random clusterings.

with a mean similarity of 0.59 (black), demonstrates that Agglomerative Hierarchical Clustering actually performed worse than if we had drawn a random clustering!

Acknowledgements

The authors would like to thank Ian Wood for thoughtful discussions, and Andre David for suggestions which significantly improved the presentation of this work.

References

Gates, A. J., & Ahn, Y.-Y. (2017). The impact of random models on clustering similarity. *Journal of Machine Learning Research*, *18*(87), 1–28. doi:10.1101/196840

Gates, A. J., Wood, I. B., Hetrick, W. P., & Ahn, Y.-Y. (2018). On comparing clusterings: An element-centric framework unifies overlaps and hierarchy.

Hubert, L., & Arabie, P. (1985). Comparing partitions. *Journal of Classification*, *2*(1), 193–218. doi:10.1007/BF01908075

Jain, A. K., Murty, M. N., & Flynn, P. J. (1999). Data clustering: A review. *ACM Computing Surveys (CSUR)*, *31*(3), 264–323. doi:10.1145/331499.331504

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., et al. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Souto, M. C. de, Costa, I. G., Araujo, D. S. de, Ludermir, T. B., & Schliep, A. (2008). Clustering cancer gene expression data: A comparative study. *BMC Bioinformatics*, *9*(1), 1. doi:10.1186/1471-2105-9-497

Sukumaran, J., & Holder, M. T. (2010). DendroPy: A python library for phylogenetic computing. *Bioinformatics*, *26*, 1569–1571. doi:10.1093/bioinformatics/btq228

Vinh, N. X., Epps, J., & Bailey, J. (2009). Information theoretic measures for clusterings comparison: Is a correction for chance necessary? In *Proceedings of the 26th annual international conference on machine learning* (pp. 1073–1080). ACM. doi:10.1145/1553374.1553511