Work 4

Carlos Jiménez, Sheena Lang, Zachary Parent, and Kacper Poniatowski

December 20, 2024

1 Abstract

This report presents an investigation into dimensionality reduction techniques and their impact on clustering performance, with a particular focus on Principal Component Analysis (PCA). We implement our own PCA algorithm and compare it against scikit-learn's implementation, evaluating both accuracy and computational efficiency.

The study utilizes two distinct datasets (Mushroom and Vowel) to assess the effectiveness of dimensionality reduction in conjunction with clustering algorithms, specifically Global K-Means and OPTICS. We explore various visualization techniques, including PCA and UMAP, to represent high-dimensional data in lower-dimensional spaces.

Our analysis demonstrates the trade-offs between dimensionality reduction and clustering performance, providing insights into optimal configurations for different data characteristics. The results show that our PCA implementation achieves comparable performance to scikit-learn's version, and that carefully selecting parameters for each stage of the pipeline can lead to better visualizations.

2 Background and Related Work

3 Methods

Dimensionality Reduction In this study, we implement our own version of Principal Component Analysis (PCA) as the primary dimensionality reduction technique. Our implementation follows the standard PCA algorithm, computing eigenvalues and eigenvectors of the covariance matrix to identify principal components. For validation and comparison purposes, we maintain a parallel implementation using scikit-learn's PCA [4], which serves as our baseline. This approach builds upon our previous work with clustering algorithms [3].

Clustering We employ two distinct clustering algorithms: Global K-Means [2] and OPTICS [1]. For Global K-Means, we use dataset-specific configurations: the Mushroom dataset uses `n_clusters=2`, `max_iterations=100`, and `tolerance=1e-3`, while the Vowel dataset uses `n_clusters=11`, `max_iterations=100`, and `tolerance=1e-4`. The OPTICS algorithm is similarly tuned with dataset-specific parameters: for Mushroom, we use `min_samples=10`, `min_cluster_size=5`, and `xi=0.1` with euclidean metric, while Vowel uses `min_samples=20`, `min_cluster_size=10`, and `xi=0.1` with manhattan metric.

Metrics To evaluate the effectiveness of our dimensionality reduction and clustering approaches, we employ several metrics. For clustering quality assessment, we use both internal metrics (Davies-Bouldin Index, Calinski-Harabasz Index) and external metrics (Adjusted Rand Index, F-Measure). Additionally, we measure the computational efficiency and accuracy of our PCA implementation against the scikit-learn baseline.

Visualization Our visualization strategy employs two main techniques: PCA and UMAP. While PCA serves both as a dimensionality reduction method and visualization tool, we specifically use it to project high-dimensional data onto 2D and 3D spaces for visual analysis. UMAP complements this by providing an alternative visualization approach, particularly useful for preserving local structure in the data. Both techniques are applied to visualize the original data distributions and the resulting cluster assignments.

4 Results and Analysis

Full results, in tabular and graphical form, are available in the appendix (Section 6).

References

[1] Mihael Ankerst, Markus M. Breunig, Hans-Peter Kriegel, and Jörg Sander. Optics: ordering points to identify the clustering structure. *SIGMOD Rec.*, 28(2):49–60, June 1999.

[2] Jakob J. Verbeek Artistidis Likas, Nikos Vlassis. The global k-means clustering algorithm. 12, 2001.

[3] Carlos Jiménez, Sheena Lang, Zachary Parent, and Kacper Poniatowski. Implementation, evaluation, and comparison of k-means and other clustering algorithms. 2024.

[4] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

6 Appendix