README: Wikipedia Article Clustering with EM Algorithm

By Manan Ambaliya(121118776)

Overview

This project clusters Wikipedia articles using a custom implementation of the Expectation-Maximization (EM) algorithm for Gaussian Mixture Models (GMMs) with diagonal covariance, leveraging TF-IDF representations of article text. It also features real-time convergence visualization (integrated into em_algorithm.py) and benchmarking for extra credit.

Files Included

- index_generator.py: Loads or creates the word index mapping from 4_map_index_to_word.json (helpful for interpretable output, stats, and debugging).

- em_algorithm.py: Main script for EM clustering, including real-time terminal visualization of convergence (using curses).

- output_formatter.py: Formats and summarizes EM results (top words, stats, etc.).

- visualizer.py: Generates ASCII word clouds for each cluster.

- benchmark_em.py: (Extra credit) Benchmarks custom EM vs. sklearn's GaussianMixture.

- people_wiki.csv: The dataset (Wikipedia articles).

- 4_map_index_to_word.json: Word index mapping.

- analysis_of_clustering_results.docx: Analysis of final clustering.

- technical_report_em_clustering.docx: Full technical report.

How to Run

1. Ensure people_wiki.csv and 4_map_index_to_word.json are in your working directory.

2. Install dependencies:

pip install numpy pandas scikit-learn

3. Run the word index generator to check or create word index mappings:

python index_generator.py

4. Run the EM algorithm with integrated real-time terminal visualization:

python em_algorithm.py

5. Generate stats and formatted outputs:

python output_formatter.py

6. Create ASCII word clouds:

python visualizer.py

7. (Extra Credit) Benchmark and compare with sklearn's GMM:

python benchmark_em.py

Outputs

- cluster_assignments.txt: Each article ID and its cluster assignment.

- cluster_stats.txt: Top 5 words in each cluster and their variances.

- em_parameters.txt: Final GMM parameters.

- convergence_log.txt: Log-likelihood per EM iteration.

- ascii_wordclouds.txt: ASCII word clouds per cluster.

Extra Credit Features

- Integrated Real-time EM convergence monitoring: Real-time visualization is directly implemented within em_algorithm.py using the curses library.

- Benchmarking suite: Compare speed, memory, and accuracy of custom EM versus sklearn's GaussianMixture.

Project Highlights

- Custom EM clustering for high-dimensional text data.

- Integrated live cluster convergence visualization for deeper understanding.
- Robust and transparent implementation, comparable to sklearn in accuracy.
- Full benchmarking suite and technical analysis for reproducibility and interpretation.