Agglomerative Likelihood Clustering

Lionel Yelibi* and Tim Gebbie†

Department of Statistical Sciences, University of Cape Town

Abstract

We present **Agglomerative Likelihood Clustering (ALC)**, a fast bottom-up clustering algorithm that maximizes the Giada–Marsili likelihood for correlation-based clustering of time-series. ALC replaces computationally expensive genetic search with a deterministic greedy merging scheme that evaluates likelihood gains for candidate merges. The method performs well on large synthetic datasets (up to 20k series) and is robust to noise using a bootstrap-resampling filter.

Motivation

- Fast, scalable clustering of dense correlation matrices (financial time-series, high-dim data).
- Avoid arbitrary thresholding and need to specify number of clusters.
- Competitive accuracy vs HDBSCAN and prior genetic algorithms (f-SPC) at reduced cost.

Giada–Marsili likelihood (key formulae)

Generative model (Noh ansatz):

$$x_i = g_{s_i}\eta_{s_i} + \sqrt{1-g_{s_i}^2}\,\epsilon_i.$$

Cluster log-likelihood for cluster s (single feature contribution):

$$L_c = \frac{1}{2}\sum_{s:n_s>1}\left[\ln\frac{n_s}{c_s} + (n_s-1)\ln\frac{n_s^2-n_s}{n_s^2-c_s}\right],$$

where n_s is cluster size and c_s the intra-cluster total correlation.
We use a per-cluster form when evaluating candidate merges:

$$L_c(s) = \frac{1}{2}\left[\ln\frac{n_s}{c_s} + (n_s-1)\ln\frac{n_s^2-n_s}{n_s^2-c_s}\right].$$

ALC: high-level algorithm

1. Initialize each object as its own singleton cluster.
2. For each cluster pair (i,j) compute $\Delta L_c = L_c(i\cup j) - (L_c(i) + L_c(j))$.
3. Greedily choose the merge with largest positive ΔL_c; update tracker and correlation sums.
4. Repeat until no positive ΔL_c exists.

Pseudocode (Appendix A in paper) is implemented efficiently to avoid recomputing full correlation matrices.

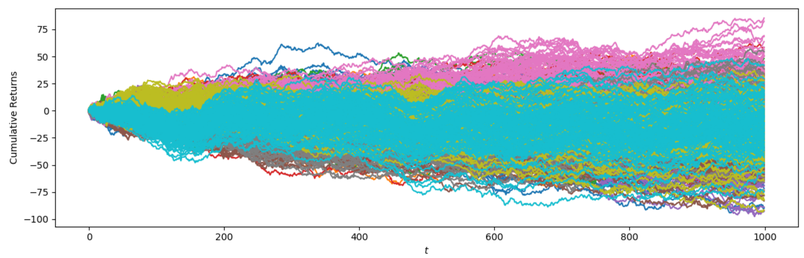
Bootstrap / Resampling (noise mitigation)

When $Q = D/N$ (quality) is low, ALC can be coupled with a resampling filter:

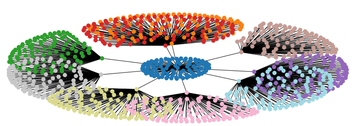
- Repeatedly sample subsets of size n such that $q = D/n \geq Q_{target}$.
- Cluster each subset; accumulate co-occurrence counts f_{ij}, d_{ij}.
- Form pairwise probability $p_{ij} = d_{ij}/f_{ij}$ and threshold to produce an adjacency matrix.

Synthetic experiments

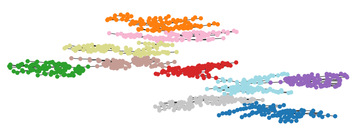
- Data: Student-t and Gaussian synthetic datasets with controllable intra-cluster coupling g_s.
- Sizes: experiments ranged N=500 to N=3000 (also up to 10k in scaling tests).
- Metrics: Adjusted Rand Index (ARI) vs ground truth, likelihood L_c, runtime scaling.



(a) **Simulated**: Cluster derived correlated time-series cumulative returns for 500 simulated assets over 1000 days.



(b) **Ground Truth**: The true correlation matrix MST



(c) **Estimated**: The estimated correlation matrix MST

Figure: Figure: illustrative synthetic results (replace with high-res figures).

Key numerical findings

- ALC achieves higher likelihoods than f-SPC and competitive ARI compared to HDBSCAN (Table IV).
- ALC runtime empirically $\mathcal{O}(N^{1.97})$ on synthetic data and $\mathcal{O}(N^{2.11})$ on real data — roughly quadratic.
- Bootstrap filtering (threshold $\omega \approx 0.75$) strongly improves ARI when signal is weak.

Hierarchical block correlations

- ALC prioritizes smaller dense clusters; HDBSCAN tends to merge into larger, sparser clusters.
- For nested-factor models, ALC recovers sub-clusters at higher noise tolerance (approx. 20

Algorithmic details & implementation notes

Tracker array: maintain cluster composition lists and incremental sums of pairwise correlations to compute c_s cheaply.
Merge selection: comprehensive search for best merge per iteration (greedy) — avoids stochastic path-dependence of genetic methods.
Complexity: naive pairwise merge evaluation is $O(N^2)$ per iteration but in practice optimized updates and early-stopping yield near-quadratic behaviour.

Recommendations for practitioners

- Use bootstrap filter when D/N is small (short time-series, many assets).
- For very large N (»10k), consider hybrid strategies: Louvain-like speedups or hierarchical pre-aggregation.
- Validate clusters with ARI / stability under resampling and inspect MST / UMAP visualizations.

Conclusions

ALC is a practical, likelihood-based clustering method that is parameter-light (no K) and competitive in quality and runtime with common alternatives. It is especially suited to correlation-based clustering of time-series and can be integrated with bootstrap resampling to mitigate estimation noise.

References (selected)

[1] Giada Marsili (2001,2002) – likelihood clustering.
Blatt et al. (1996,1997) – SPC.
Blondel et al. (2008) – Louvain.
McInnes et al. (2017) – HDBSCAN.
Full bibliography available in the original paper (Yelibi & Gebbie, 2021).

Questions? lionel.yelibi@alumni.uct.ac.za