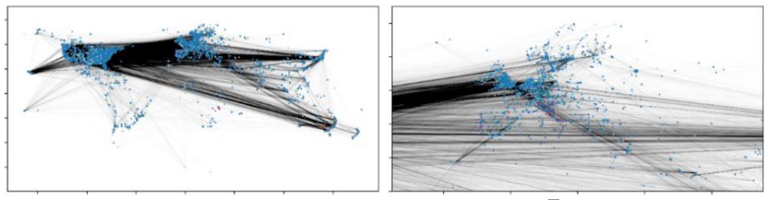
Adaptive Label Propagation with Entropy-Guided Weighting for Location-Aware Graph Clustering

UNIST CSE, Doyeol Oh (20211187)
https://github.com/ohdoyoel/unist_cse304_term_project

Introduction

- Traditional graph clustering (e.g., LP, Louvain) uses **only structure, often ignoring spatial coherence.**
- Location-based networks provide rich spatial signals via check-ins, but prior methods rely on **heavy models**.
- We propose **a lightweight, training-free method** that adaptively fuses structure and location using entropy-based weighting.
- Local label uncertainty guides **dynamic adjustment between topological and spatial similarity**.



Worldwide *Europe*

Limitation: Structure-only methods link distant users, ignoring local community boundaries.

Main Contributions

- 🧭 **Entropy-guided Weighting**
  : Balances structure and location using local label entropy.
- 📍 **Recent Check-in Only**
  : Uses latest location to avoid trajectory modeling.
- 📊 **Coherent Clustering**
  : Quantitative and visual gains on *Brightkite dataset*.
- ⚙️ **Training-free & Scalable**
  : No learning phase; fits low-resource settings.

Related Work

🧱 Key Limitations in Prior Work

- **Structure-only**: LP, Louvain ignore node context
- **GNN-based**: Require features, labels, and training
- **Location-aware**: Often need trajectories and end-to-end models

🌈 In contrast, our approach:

- **Adaptive LP**: Training-free, unsupervised, entropy-based fusion without labels

Method	Structure	Location	Training	Scalability
LP	✅	❌	❌	✅
Louvain	✅	❌	❌	✅
GNN	✅	✅	✅	❌
Adaptive LP	✅	✅	❌	✅

Problem Statement

Given a social graph $G = (V, E)$ and each user's most recent check-in location l_i, the goal is **to cluster users by combining structural proximity and spatial locality,** without only relying on node features or labels.

⚖️ Hybrid Similarity Function

To fuse structure and location, we define:

$$\mathrm{sim}_{ij} = \alpha_{ij} \cdot \mathrm{sim}_{\mathrm{str}}(i,j) + (1 - \alpha_{ij}) \cdot \mathrm{sim}_{\mathrm{geo}}(i,j)$$

- $\mathrm{sim}_{\mathrm{str}}(i,j)$: Jaccard similarity between neighbors
- $\mathrm{sim}_{\mathrm{geo}}(i,j)$: Cosine similarity of location vectors
- $\alpha_{ij} = \frac{(\alpha_i + \alpha_j)}{2}$: Weight based on entropy

🌡️ Entropy-Guided Weight

Each node i calculates local label entropy at each iter:

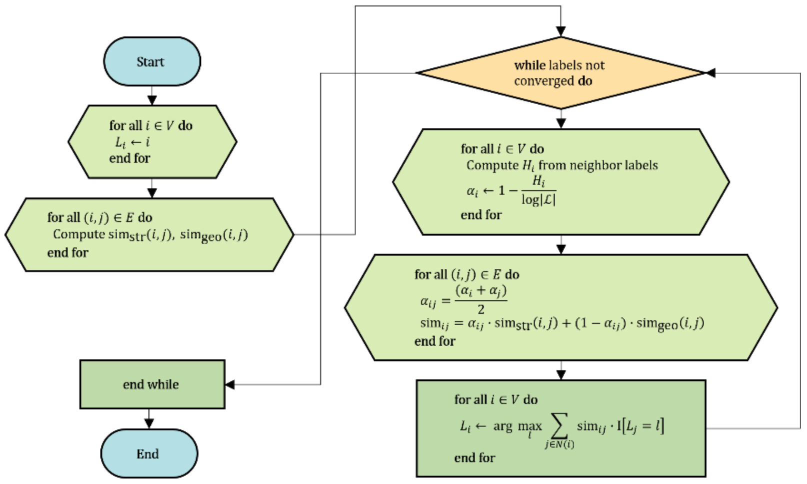
$$H_i = -\sum_{l \in \mathcal{L}} p_i(l) \log p_i(l), \qquad \alpha_i = 1 - \frac{H_i}{\log|\mathcal{L}|}$$

- Low entropy → Clear structural signal
  → Higher reliance on graph topology
- High entropy → Ambiguous structural signal
  → Shift focus to spatial similarity

Algorithm Steps

1. **Initialize**: Assign a unique label to each node
2. **Precompute**: Calculate pairwise structural similarity $\mathrm{sim}_{\mathrm{str}}(i,j)$ and spatial similarity $\mathrm{sim}_{\mathrm{geo}}(i,j)$
3. **For each node at every iteration**:
   - Compute entropy H_i over neighbor labels
   - Derive adaptive weight α_i and hybrid similarity sim_{ij}
   - Update label by weighted majority vote
4. **Repeat** until label assignments converge

$$L_i^{(t+1)} = \arg\max_{l \in \mathcal{L}} \sum_{j \in N(i)} \mathrm{sim}_{ij} \cdot \mathrm{I}\left[L_i^{(t)} = l\right]$$



Adaptive Label Propagation Algorithm Flow

Experiment

Dataset: Brightkite

- 58K users, 214K mutual edges
- 4.5M check-ins → only most recent used
- No trajectory modeling, but preserves spatial context

Baselines

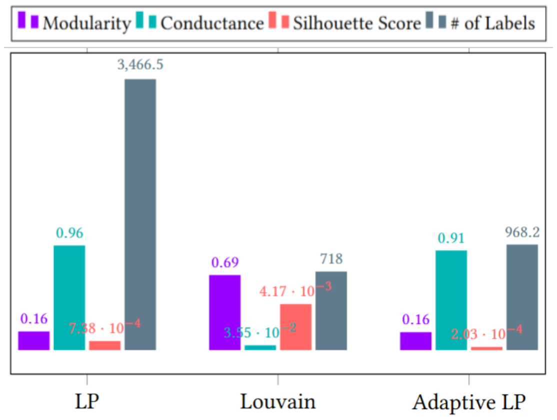
- **Label Propagation (LP)**: Structure-only label diffusion
- **Louvain**: Hierarchical modularity optimization

Metrics

- **Modularity**: Intra-cluster density vs. random chance
- **Conductance**: Sharpness of cluster boundaries
- **Silhouette Score**: Spatial compactness and separation
- **# of Labels**: Cluster granularity (over-/under-segmentation)

Result

All metrics are averaged over 10 independent runs to ensure consistency and reduce variance across methods.



Comparison of Averaged Clustering Metrics over 10 Runs

- **Modularity & Conductance**
  : ALP matches LP in modularity and conductance, but offers stronger inter-cluster connectivity with fewer isolated groups.
- **Spatial Coherence vs. Silhouette**
  : Though ALP records a lower silhouette score, it excels in producing geographically aligned clusters–showing its strength in balancing spatial and structural signals.
- **Cluster Granularity**
  : ALP finds a middle ground in cluster count, mitigating LP's excessive fragmentation (~3466 clusters) and Louvain's over-merging (~718 clusters) by maintaining moderate granularity (~968 clusters).

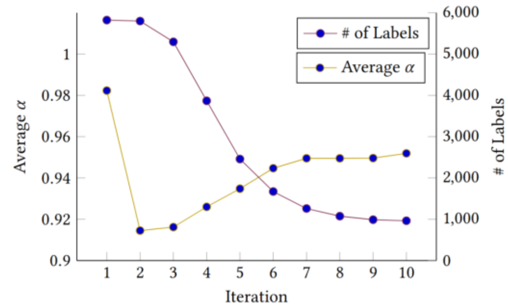
Computational Efficiency

Method	Runtime (s)	Memory (MB)
Label Propagation	306.0	29.3
Louvain	184.6	95.1
Adaptive Label Propagation	553.5	29.4

Averaged Runtime and Memory Usage over 10 Runs

- **Runtime**
  ALP requires more time (553.5s) due to iterative entropy and similarity computations.
- **Memory**
  Remains lightweight (29.4MB), comparable to LP and significantly more efficient than Louvain (95.1MB).
- **Insight**
  Suitable for large-scale or resource-constrained environments where memory and responsiveness matter.

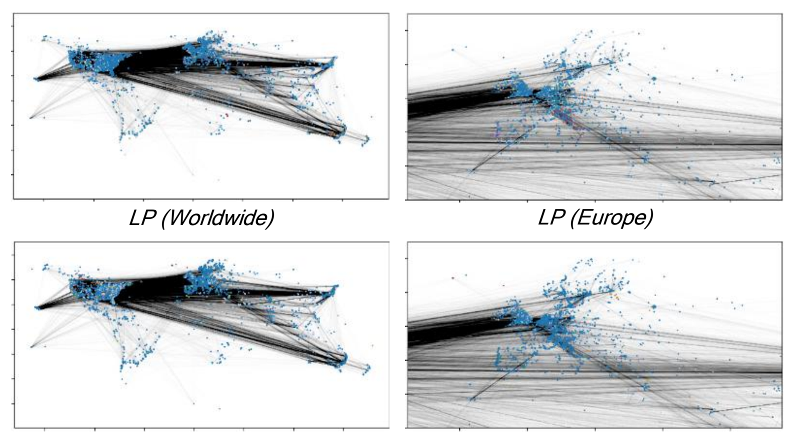
Adaptive Weighting Behavior



Evolution of # of Labels and Average α over 10 Runs

- **# of Labels**
  Drops rapidly from 5822 to 963, showing fast convergence.
- **Average α**
  Decreases early (favoring spatial similarity), then rises as structural confidence grows.
- **Insight**
  Confirms ALP's adaptive fusion of structure and location over time.

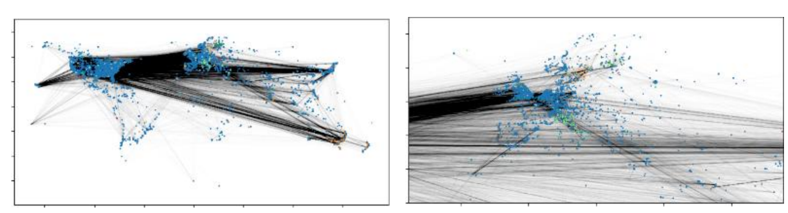
Spatial Coherence Visualization



LP (Worldwide) *LP (Europe)*

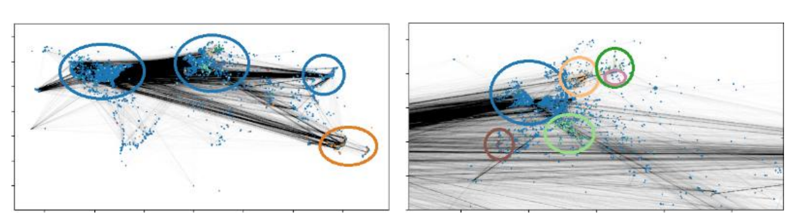
Louvain (Worldwide) *Louvain (Europe)*

LP / Louvain: Structure-only methods group distant users, producing fragmented clusters.



Adaptive LP (Worldwide) *Adaptive LP (Europe)*

ALP: Leverages recent check-ins + entropy-guided weighting to form compact, localized communities.



Adaptive LP (Worldwide), Marked *Adaptive LP (Europe), Marked*

Marked clusters:
Show ALP captures regional structures (cities) more accurately.

Conclusion

Proposed **a lightweight, unsupervised method that fuses structure and location via entropy-guided weighting**.

- **Strengths**: Interpretable, scalable, and spatially coherent without training.
- **Future work**: 1 Use full trajectory data for spatiotemporal clustering 2 Incorporate uncertainty metrics beyond entropy 3 Extend to streaming/dynamic social graphs