The Map Equation Goes Neural: Mapping Network Flows with Graph Neural Networks

NeurIPS 2024

Christopher Blöcker[1,2], Chester Tan[1], Ingo Scholtes[1,2]

[1]Chair of Machine Learning for Complex Networks
Center for Artificial Intelligence and Data Science
Julius-Maximilians-Universität Würzburg
Würzburg, Germany

[2]Data Analytics Group
Department of Informatics
University of Zurich
Zurich, Switzerland

Universität
Zürich[UZH]

too long; didn't listen

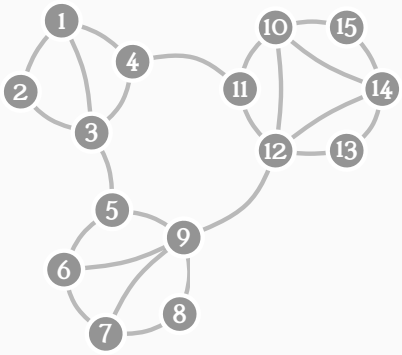
map equation + graph neural networks

Idea: leverage advances in deep learning for deep graph clustering with an information-theoretic objective function from network science – the map equation

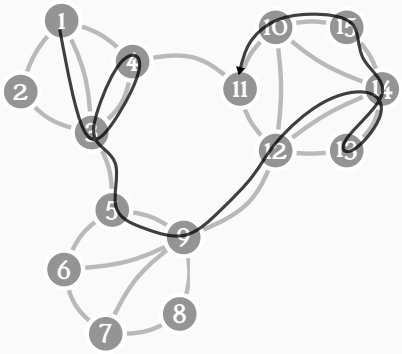
The Map Equation

→ Rosvall and Bergstrom; PNAS 2008

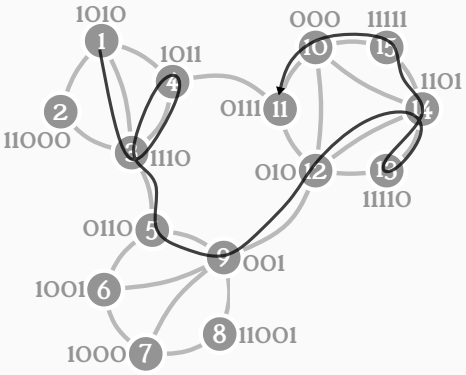
The Map Equation



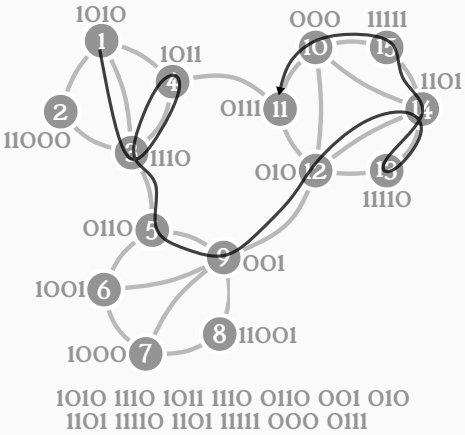
The Map Equation



The Map Equation



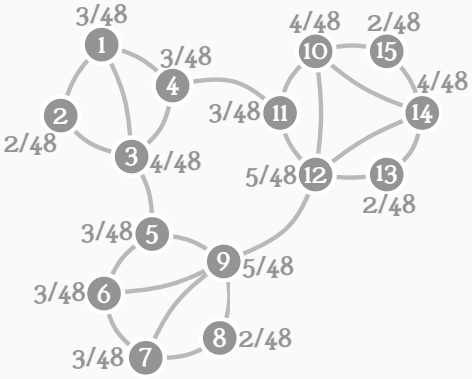
The Map Equation



Description length

$$H(P) = -\sum_{v \in V} p_v \log_2 p_v \text{ bits} \qquad (1)$$

The Map Equation



Description length

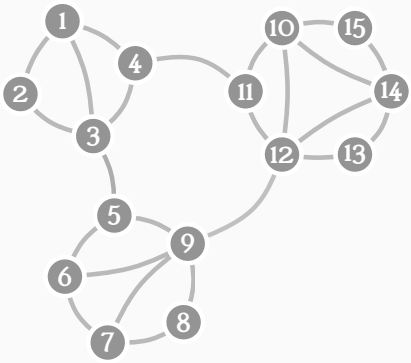
$$H(P) = -\sum_{v \in V} p_v \log_2 p_v \text{ bits} \tag{1}$$

Visit rates

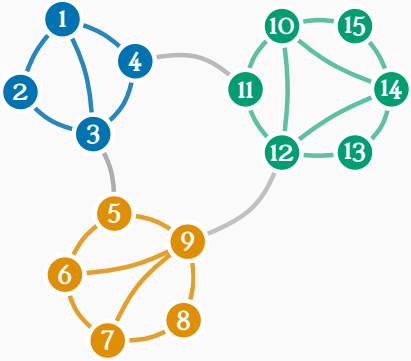
$$p_v = \sum_{u \in V} p_u t_{uv} \qquad t_{uv} = \frac{w_{uv}}{\sum_{v \in V} w_{uv}} \tag{2}$$

But we are interested in communities...

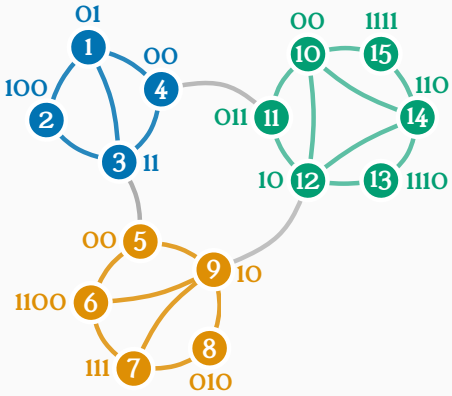
The Map Equation



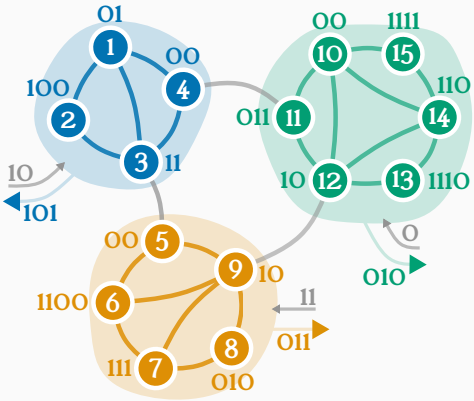
The Map Equation



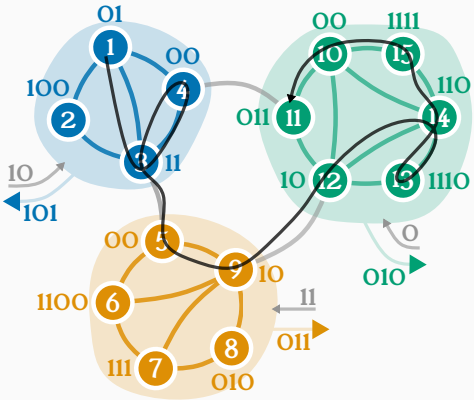
The Map Equation



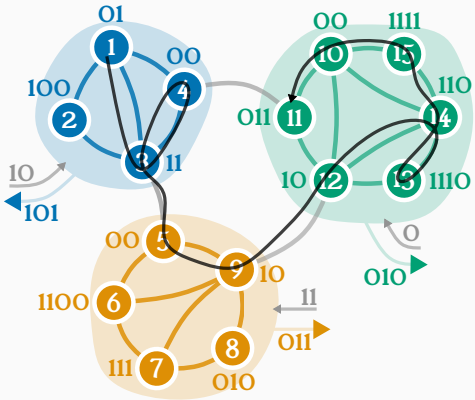
The Map Equation



The Map Equation



The Map Equation



1001 11 00 11 101 1100 10 011010
110 1110 110 1111 00 011

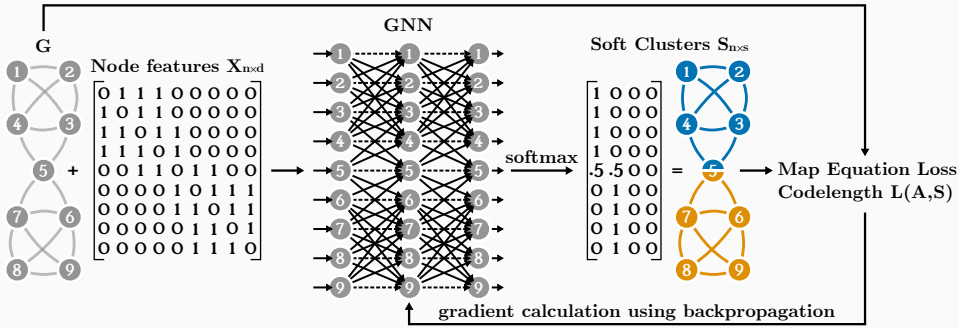
$$L(\mathrm{M}) = qH(Q) + \sum_{\mathrm{m}\in\mathrm{M}} p_{\mathrm{m}} H(P_{\mathrm{m}}) \rightarrow \min$$

Minimum Description Length (MDL) Principle

The Map Equation Goes Neural

The Map Equation Goes Neural

- We introduce a soft cluster assignment matrix for differentiability
  → overlapping modules
- We implement the differentiable map equation in tensor form
  → clustering loss for (G)NNs
- Optimisation with gradient descent means differentiating the codelength with respect to soft cluster assignments



The Map Equation Goes Neural

Compute visit rates from graph's transition matrix $\mathbf{T}$ and normalised node in-degrees $\mathbf{d}^{\text{in}}$

$$\mathbf{p}^{(t+1)} \leftarrow \frac{\alpha}{W_{\text{tot}}}\mathbf{d}^{\text{in}} + (1-\alpha)\,\mathbf{p}^{(t)}\mathbf{T} \qquad \text{and set} \qquad \mathbf{p}^{(0)} = \mathbf{d}^{\text{in}}$$

Compute flow matrix $\mathbf{F}$ and flows between clusters $\mathbf{C}$, given cluster assignments $\mathbf{S}$

$$\mathbf{F} = \frac{\alpha}{W_{\text{tot}}}\mathbf{A} + (1-\alpha)\,\text{diag}\,(\mathbf{p})\,\mathbf{T} \qquad \mathbf{C} = \mathbf{S}^{\top}\mathbf{F}\mathbf{S}$$

Define terms

$$q = 1 - \text{tr}\,(\mathbf{C}) \quad \mathbf{q}_{\text{m}} = \mathbf{C}\mathbf{1}_{\text{s}} - \text{diag}\,(\mathbf{C}) \quad \mathbf{m}_{\text{exit}} = (\mathbf{1}_{\text{s}}^{\top}\mathbf{C})^{\top} - \text{diag}\,(\mathbf{C}) \quad \mathbf{p}_{\text{m}} = \mathbf{q}_{\text{m}} + \mathbf{1}_{\text{s}}^{\top}\mathbf{C}$$

Assemble the map equation

$$L\,(\mathbf{A}, \mathbf{S}) = q \log_2 q - (\mathbf{q}_{\text{m}} \log_2 \mathbf{q}_{\text{m}})\,\mathbf{1}_{\text{s}} - (\mathbf{m}_{\text{exit}} \log_2 \mathbf{m}_{\text{exit}})\,\mathbf{1}_{\text{s}} - (\mathbf{p} \log_2 \mathbf{p})\,\mathbf{1}_{n} + (\mathbf{p}_{\text{m}} \log_2 \mathbf{p}_{\text{m}})\,\mathbf{1}_{\text{s}}$$

Evaluation

Evaluation – Setup

Datasets

- LFR networks with 1000 nodes, planted communities, and various mixing parameters
- Real networks from PyTorch Geometric, PyTorch Geometric Signed Directed, and Open Graph Benchmark (OGB)

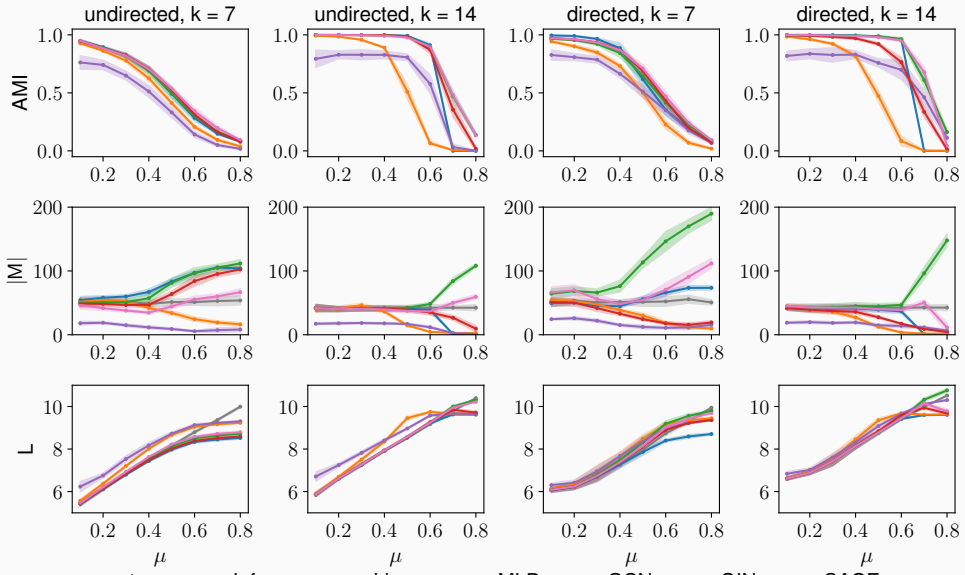
Community Detection

Neuromap with different (G)NN architectures

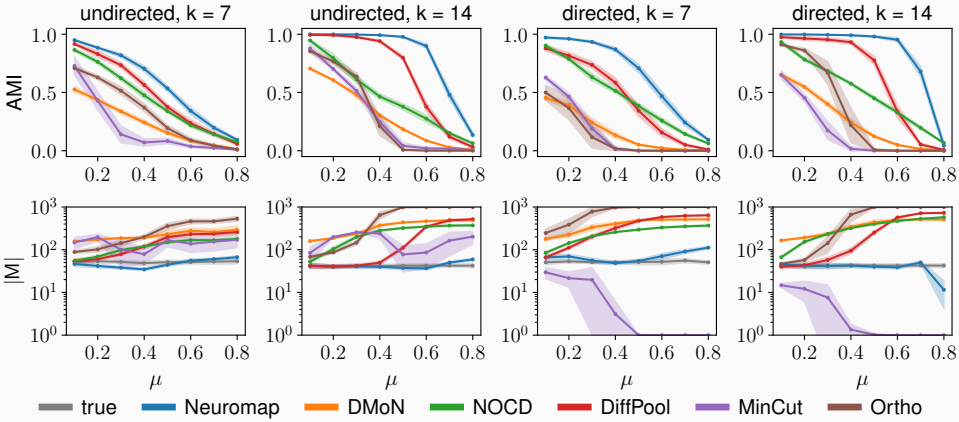
- Simple linear layer and MLP
- GCN, GIN, SAGE with 2 layers

Baselines: Infomap, DMoN, NOCD, DiffPool, MinCut, Ortho

Results – directed

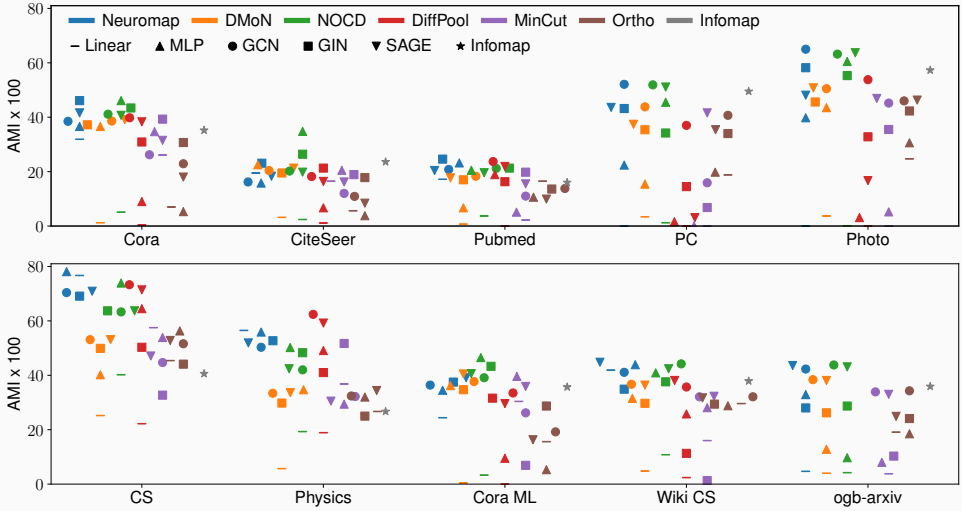


Results – directed



Here: SAGE, more architectures in the paper!

Results – real-world networks



Complete tabulated and additional results in the paper!

Conclusion

Conclusion

Motivation

Combine the best of both worlds: deep community detection with MDL

Our Approach

We make the map equation differentiable by introducing soft cluster assignments, and optimise it with (G)NNs and gradient descent.

Results

- Community-detection performance depends on the chosen (G)NN architecture
- Soft cluster assignments produce overlapping communities
- No explicit regularisastion needed, MDL is sufficient!
- Neuromap performs competitively against recent deep clustering approaches
- Additional benefit: enables fast experimentation with map equation adaptions!

Thank you for your attention!