Mapping Flows in Bipartite Networks

NetSci 2020

https://arxiv.org/abs/2007.01666

Christopher Blöcker and Martin Rosvall

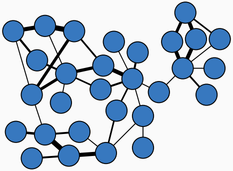
{christopher.blocker,martin.rosvall}@umu.se

Integrated Science Lab
Department of Physics
Umeå University

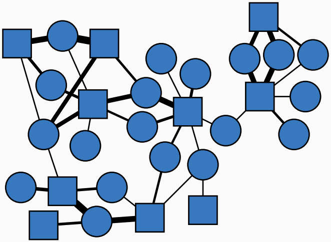
The Problem: Community Detection in Bipartite Networks

The Problem: Community Detection in Bipartite Networks

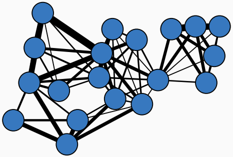
Treat as unipartite



⇐

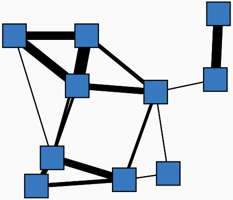


⇒



Unipartite projections

⇒



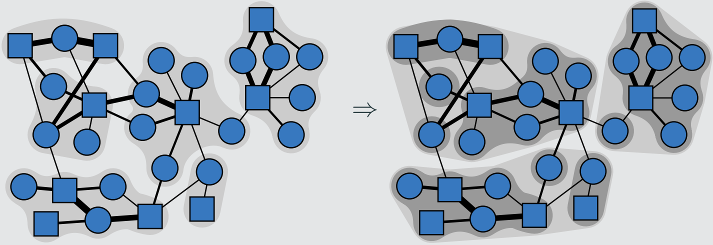
The Problem: Community Detection in Bipartite Networks

Our Solution

Teach the map equation to recognise bipartite networks.

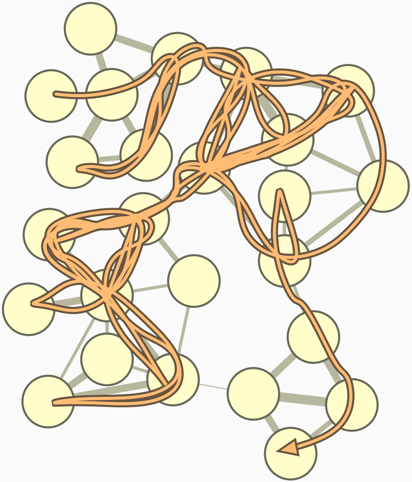
The Benefits

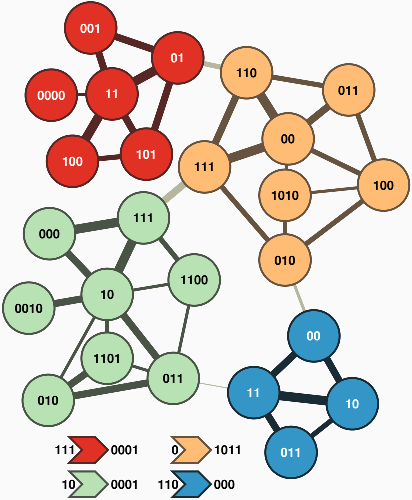
- We use all available data efficiently
- This improves the compression and we find more regularities
- We increase the resolution and explore different scales



Quick Overview:
The Map Equation Framework

The Map Equation Framework





$$L(M) = qH(\mathcal{Q}) + \sum_{m \in M} p_m H(\mathcal{P}_m)$$

Our Solution:
The Bipartite Map Equation

The Bipartite Map Equation

Idea

Reflect the bipartite network structure in the coding scheme.

Key insight

Random walks must alternate between node types.

$\rightarrow$ network flow is divided evenly between node types,

$\rightarrow$ one node type is visited in even steps, the other in odd steps,

$\rightarrow$ there are two random processes:

- X: the current node
- Y: the current node type
- then we can use Bayes' rule: $H(X|Y) = H(X) - H(Y) + H(Y|X)$

$$L(\mathsf{M}_1) = \underbrace{H(\mathcal{P})}_{H(X)} = \underbrace{1}_{H(Y)} + \underbrace{\frac{1}{2}H\left(\mathcal{P}^L\right) + \frac{1}{2}H\left(\mathcal{P}^R\right)}_{H(X|Y)}$$

$\rightarrow$ plug this into the map equation...

The Bipartite Map Equation

$$L(\mathsf{M}) = qH(\mathcal{Q}) + \sum_{\mathsf{m}\in\mathsf{M}} p_{\mathsf{m}} H(\mathcal{P}_{\mathsf{m}})$$

$$\downarrow$$

$$\begin{aligned} L_B(\mathsf{M}) = {} & q^L H\left(\mathcal{Q}^L\right) + \sum_{\mathsf{m}\in\mathsf{M}} p_{\mathsf{m}}^L H\left(\mathcal{P}_{\mathsf{m}}^L\right) \\ & + q^R H\left(\mathcal{Q}^R\right) + \sum_{\mathsf{m}\in\mathsf{M}} p_{\mathsf{m}}^R H\left(\mathcal{P}_{\mathsf{m}}^R\right) \end{aligned}$$

Problem

In sparse networks, knowing the node type comes close to knowing the exact node $\rightarrow$ encoding close to the entropy rate of the Markov process without identifying modular structure.

A key ingredient of the map equation is forgetting the right amount!

Our Solution 2.0:
The Bipartite Map Equation
with Varying Node-Type Memory

The Bipartite Map Equation with Varying Node-Type Memory

Idea

Reflect the bipartite network structure in the coding scheme **and forget node types at rate** α.

Then

- Random walks still alternate between node types
- Random walker "confuses" node types with probability α
  $\rightarrow$ new visit rates: $p_u \rightsquigarrow p_u^\alpha = ((1-\alpha)\,p_u, \alpha p_u)$ for left nodes u
  $p_v \rightsquigarrow p_v^\alpha = (\alpha p_v, (1-\alpha)\,p_v)$ for right nodes v
- Network flow is still evenly divided between node types!
- Before: $H(Y|X) = 0$
  Now: $H(Y|X) = H(\{\alpha, 1-\alpha\}) = H_\alpha$

$$L(\mathsf{M}_1) = \underbrace{H(\mathcal{P}_1)}_{H(X)} = \underbrace{1}_{H(Y)} - \underbrace{H_\alpha}_{H(Y|X)} + \underbrace{H(\mathcal{P}_1^\alpha)}_{H(X|Y)}$$

$\rightarrow$ plug this into the map equation...

The Bipartite Map Equation with Varying Node-Type Memory

$$L\left(\mathsf{M}\right) = qH\left(\mathcal{Q}\right) + \sum_{\mathsf{m}\in\mathsf{M}} p_{\mathsf{m}}H\left(\mathcal{P}_{\mathsf{m}}\right)$$

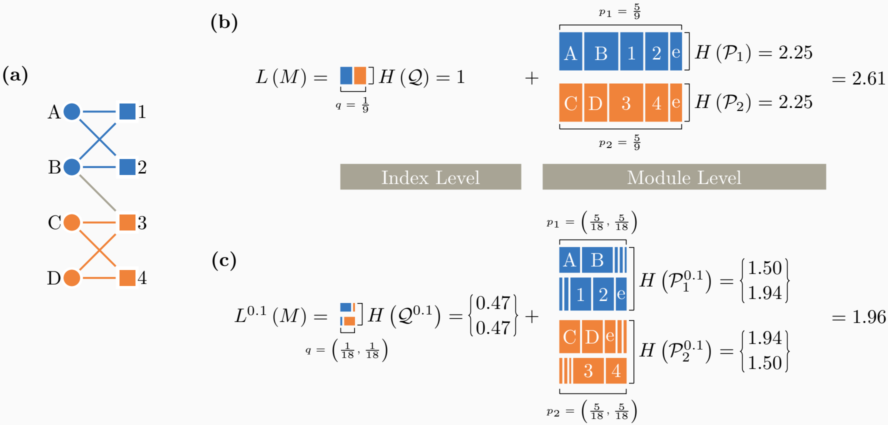
$$\downarrow$$

The Bipartite Map Equation with Varying Node-Type Memory

$$L^{\alpha}\left(\mathsf{M}\right) = q^{\alpha}H\left(\mathcal{Q}^{\alpha}\right) + \sum_{\mathsf{m}\in\mathsf{M}} p^{\alpha}_{\mathsf{m}}H\left(\mathcal{P}^{\alpha}_{\mathsf{m}}\right)$$

For $\alpha = \frac{1}{2}$, we recover the standard map equation!

The Bipartite Map Equation with Varying Node-Type Memory



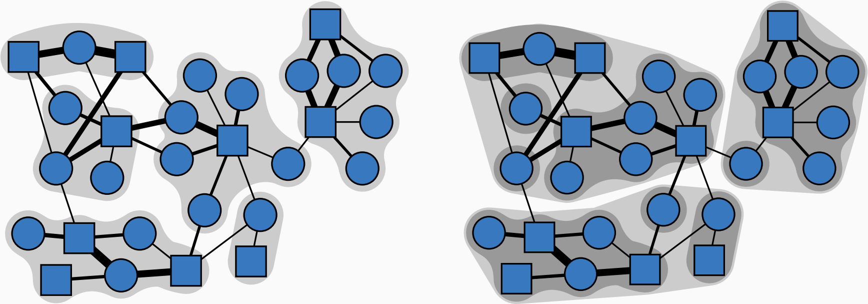
Results

Results

The Bipartite Map Equation with Varying Node-Type Memory

$$L^{\alpha}(\mathsf{M}) = q^{\alpha} H(\mathcal{Q}^{\alpha}) + \sum_{\mathsf{m} \in \mathsf{M}} p_{\mathsf{m}}^{\alpha} H(\mathcal{P}_{\mathsf{m}}^{\alpha})$$

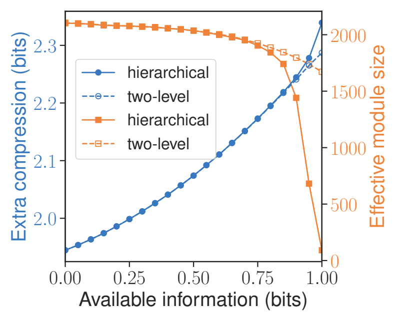
$\alpha = \frac{1}{2}$ $\qquad$ $\alpha = \frac{1}{6}$



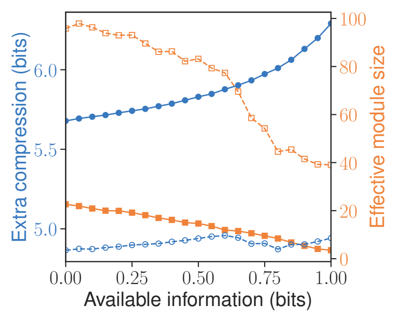
$\Rightarrow$ we can change the resolution and explore different scales!

Results

(a) Last.fm user-song network.



(b) IMDb actor-movie network.



Extra compression: compared to one-module partition

Available information (bits) $= 1 - H_\alpha$

Effective module size $= 2^{H(S)}$

Conclusion

Results

Problem

Most community detection methods address unipartite networks.

Idea

Extend the map equation framework to reflect the regularities in bipartite networks.

Benefits

We use node-type information efficiently and increase the compression. Further, by adjusting the resolution, we explore community structures at different scales.

Acknowledgements

https://arxiv.org/abs/2007.01666

This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.

We would like to thank Leto Peel, Vincenzo Nicosia, and Jelena Smiljanić for discussions that helped to improve this paper.

Martin Rosvall was supported by the Swedish Research Council, Grant No. 2016-00796.