Memory and Mesoscopic Structures in Diffusion Processes

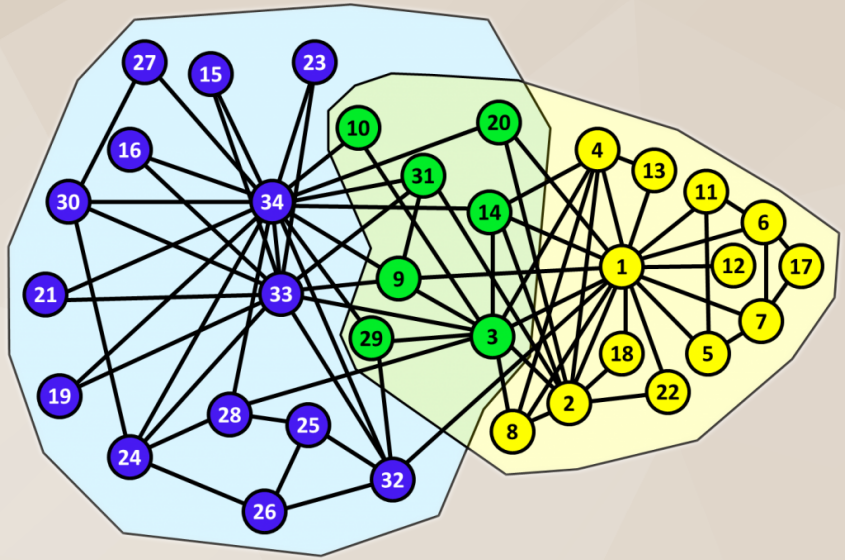
Mauro Faccin
Jean-Charles Delvenne
ICTEAM, Université Catholique de Louvain, Belgique

Netsci-X 2016 - Wrakłow

Partitioning Problem

Why should we care

- simple description
- system model
- mesoscopic structures



Good partitioning is the one leading to an interesting reduced model

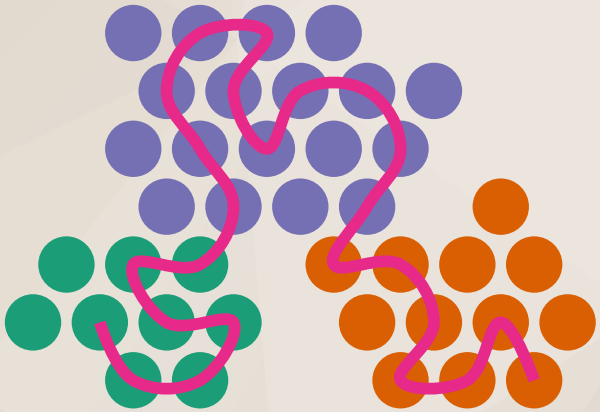
Projected Markov Chain

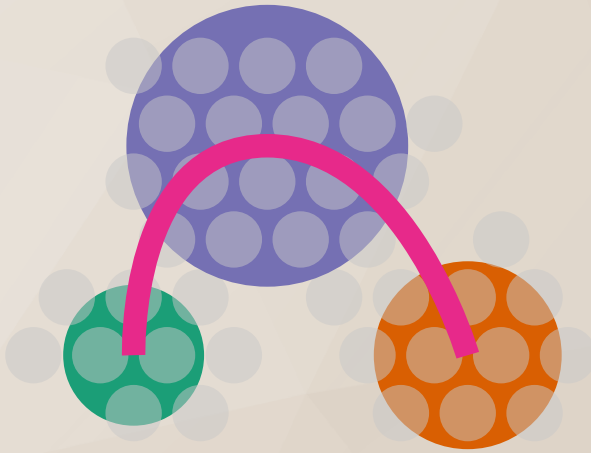
Markov Chain

$\ldots, \mathbf{x}_{\text{past}}, \mathbf{x}_{\text{now}}, \mathbf{x}_{\text{future}}, \ldots$

Projection

$\ldots, \mathbf{Y}_{\text{past}}, \mathbf{Y}_{\text{now}}, \mathbf{Y}_{\text{future}}, \ldots$

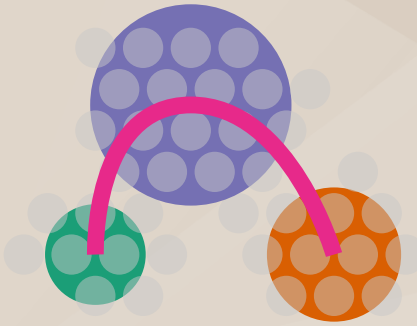




Recipe

Projected Markov Chain:

$$\ldots, \mathbf{Y}_{\text{past}}, \mathbf{Y}_{\text{now}}, \mathbf{Y}_{\text{future}}, \ldots$$



Maximize:
$$\mathbf{I}(\mathbf{Y}_{\text{future}}; \mathbf{Y}_{\text{past}})$$

amount of information flowing from *past* to *future*.

Minimize:
$$\mathbf{I}(\mathbf{Y}_{\text{future}}; \mathbf{Y}_{\text{past}} | \mathbf{Y}_{\text{now}})$$

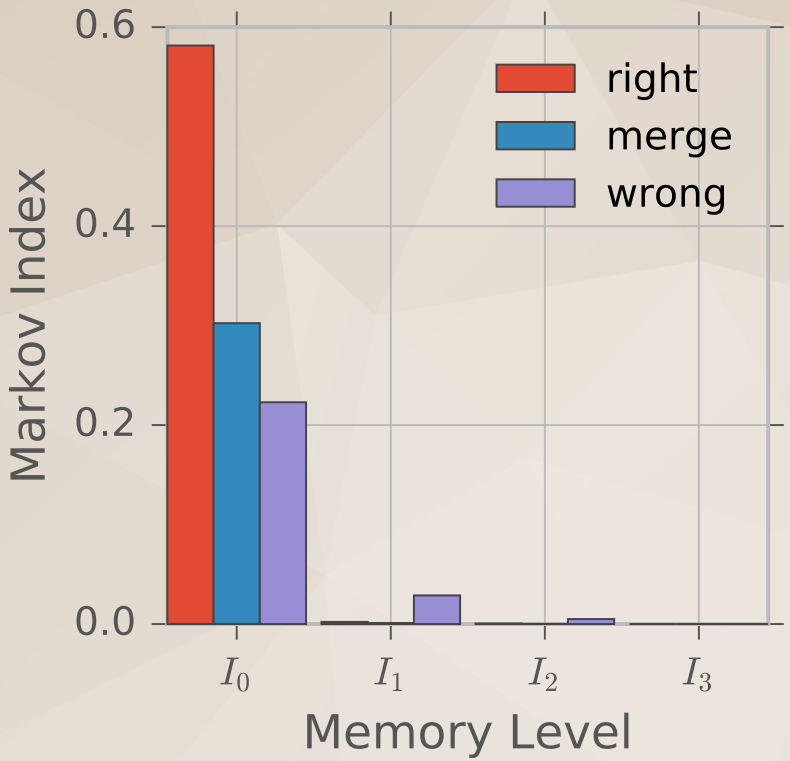
higher order memory embedded into the process.

$$\mathbf{I}(\mathbf{Y}_{\text{future}}; \mathbf{Y}_{\text{past}}) \leq \mathbf{I}(\mathbf{x}_{\text{future}}; \mathbf{x}_{\text{past}})$$

$$\mathbf{I}(\mathbf{Y}_{\text{future}}; \mathbf{Y}_{\text{past}} | \mathbf{Y}_{\text{now}}) \geq \mathbf{I}(\mathbf{x}_{\text{future}}; \mathbf{x}_{\text{past}} | \mathbf{x}_{\text{now}}) = 0$$

Def. Markov Index: $$\mathbf{I_k} = \mathbf{I}(\mathbf{Y}_{\text{future}}; \mathbf{Y}_{\text{past}} | \underbrace{\mathbf{Y_1}, \ldots, \mathbf{Y_k}}_{\mathbf{Y}_{\text{now}}})$$

Example: Scale-Free Communities



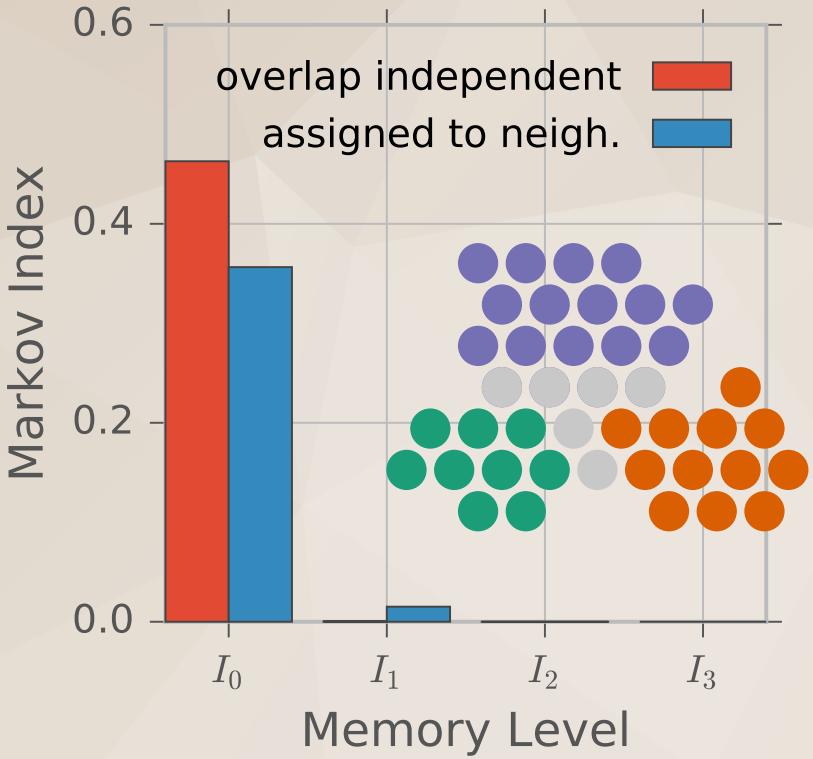
right: same partitions for building
merge: two partitions merged
wrong: a number of nodes mis-assigned

High $\mathbf{l_0}$ and low higher order memory effects. Keeps most information.

If I merge two partitions I *lose* useful information.

Wrong node assignation can lead to unwanted memory effects.

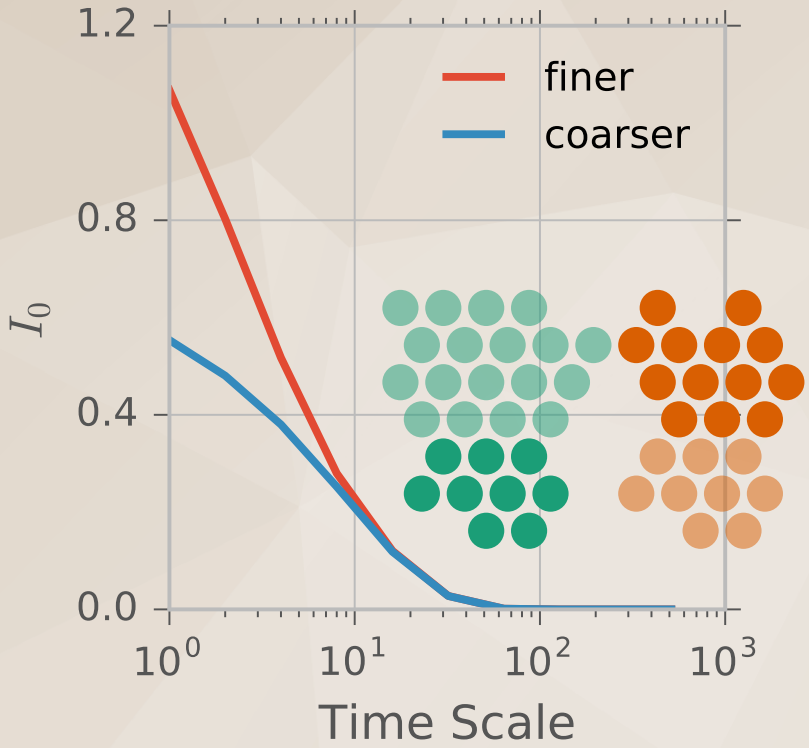
Example: Overlapping Partitions



The overlapping area is best assigned to an independent partition

Trying to assign to neighbour partitions leads to unwanted memory effects

Example: Hierarchies



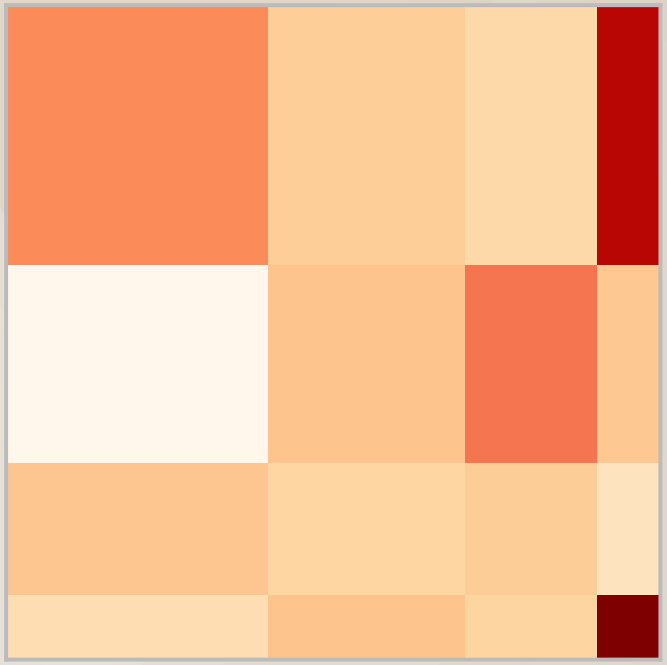
Time sampling:

A more detailed partitioning leads to higher first order Markov Index ($\mathbf{I_0}$)

With time sampling, gain on $\mathbf{I_0}$ is negligible

Example: Block Models

Adjacency Matrix



The approach generalizes to other types of mesoscopic structures as block models, core-periphery structures…

Questions?

Outline:

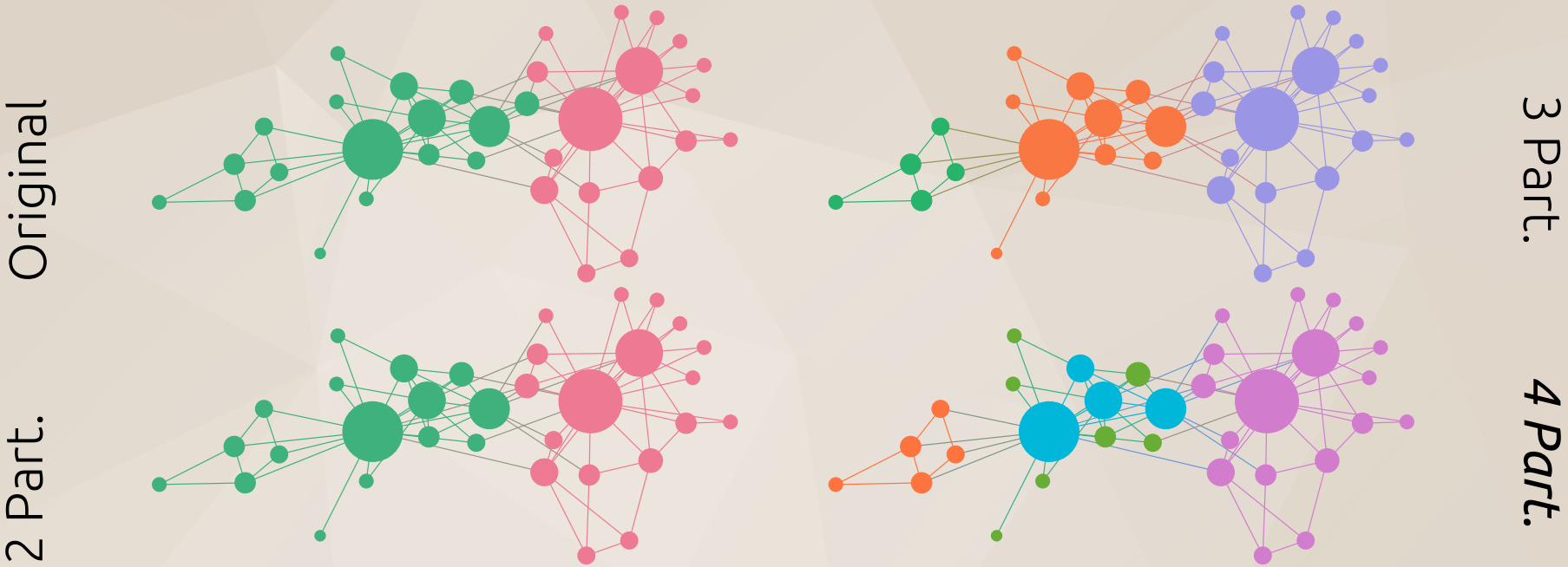
- Partitioning based on system dynamics,
- Generalize to overlapping partitions, hierarchical structures, block structures …
- Auto-select partition number, proper time-scale

Joint work with:
JC Delvenne
@ ICTEAM and BigData Group, UCLouvain.

Good partitioning is the one leading to an interesting reduced model

Too late for the KCC Trophy? Dammit!

Simplistic approch:
hierarchical algorithm with cost function $\mathbf{l_1} - \mathbf{l_0}$.



Only one node is mis-assigned.

This is the best partitioning, blue/green has core/periphery structure.