Clustering and Time Scales in Markov Chains

Jean-Charles Delvenne (Université catholique de Louvain)
Sophia Yaliraki (Imperial College London)
Mauricio Barahona (Imperial College London)

What is a good clustering?

- Non-overlapping clusterings
- Quantify the quality of a clustering:
  - MinCut
  - Normalised Cut (Shi-Malik 00)
  - (α,ε)-clustering (Kannan, Vempala, Vetta 00)
  - Modularity (Newman-Girvan 04)
  - Local modularity (Muff, Rao, Caflisch 05)
  - Density (Delling, Gaertler, Görke, Nikoloski, Wagner 07)
  - K-means for various kernels (Fouss ; Latapy Pons)
  - Potts model clustering (Reichardt, Bornholdt 04)
  - Fitness function (Fortunato. Lancichinetti, Kertész 08)
  - etc.

Graphs as Markov chains

- Nodes = state of the Markov chain
- Edges = possible transition
- Probability of transition = 1/outdegree
- If undirected, then
  - stationary proba of node = degree / $\sum$ degrees
  - every path of length 1 (=edge) has same proba
- In general:
  - weighted graph = Markov chain
  - weight = energy

Clustering of Markov chains

Interpret graph measures:

- For a 2-way clustering of nodes $V = V_1 \cup V_2$
  - MinCut $=$ Prob[V_1 V_2] + Prob[V_2 V_1]
  - NormalisedCut= MinCut (1/Prob[V_1] $+$ 1/Prob[V_2]))
  - Conductance $=$ MinCut / min(Prob[V_1] , Prob[V_2])
  - Modularity $= 2$ Prob[V_1] Prob[V_2] □– MinCut
  - Modularity $= 2$ (1/Prob[V_1] $+$ 1/Prob[V_2])$^{-1}$ – MinCut

- For a *k*-way clustering:
  - MinCut $= 1 - \sum_i$ Prob [V_i V_i]
  - Modularity $= 1 -$ (MinCut + $\sum_i$ Prob[V_i] ^{2})

Clustering of Markov chains

- MinCut gives unbalanced clusterings
- Others =MinCut + bias towards balanced clusterings
- Modularity originally defined for unweighted graphs.
- Here, defined on Markov chains (=weighted graphs)
- All make use of paths of length one

  = short term behaviour.
- What about long term behaviour?
- For NCut with Markov Chain: Zhou, Huang, Schölkopf 05

Clustering and random walks

Algorithms on Markov chains:

- MCL (van Dongen 00)
- Euclidean Commute distance (Fouss, Pirotte, Saerens 04)
- Walktrap (Latapy, Pons 06)
- Etc.

Clustering and physics

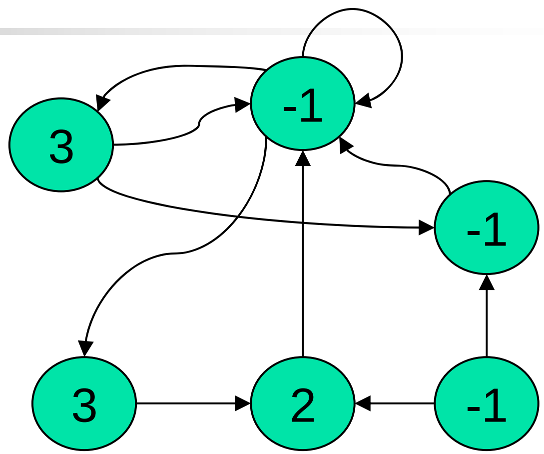
- Intuition from physics:
  - Quasi-equilibrium (e.g. glass):
    - medium time scale is observed
  - Cluster = quasi-equilibrium at some time scale
  - Time scale = resolution parameter of clustering:
    - short time = many clusters
    - long time = few clusters

- Quality of clustering <u>at a given time scale</u>

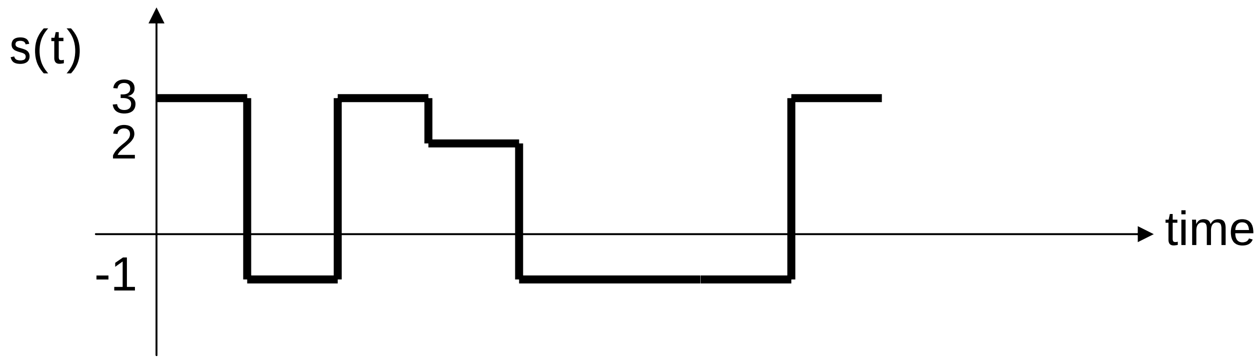
Stability: definition and properties

Functions on Markov chain

- Assign a real to every state:

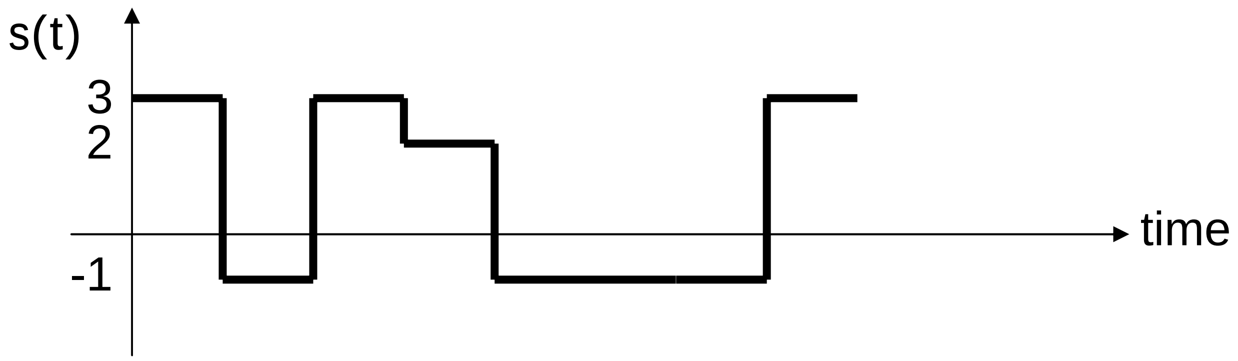


- A random walk generates a signal (=sequence of reals): e.g., 3, -1, 3, 2, -1, -1, -1, 3,...



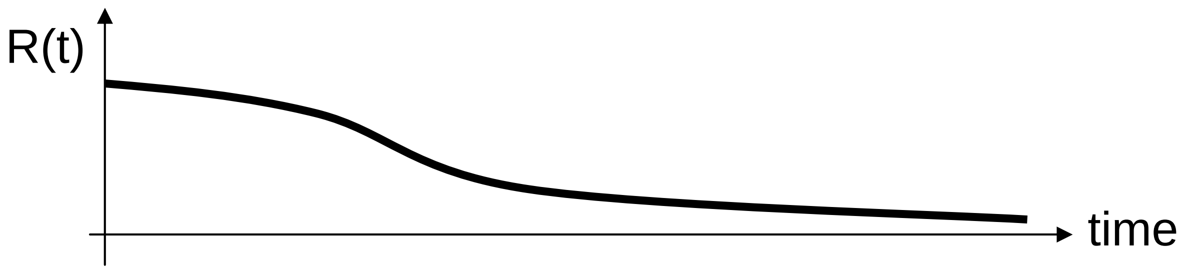
Functions on Markov chain

- Random signal s(t) not Markovian in general



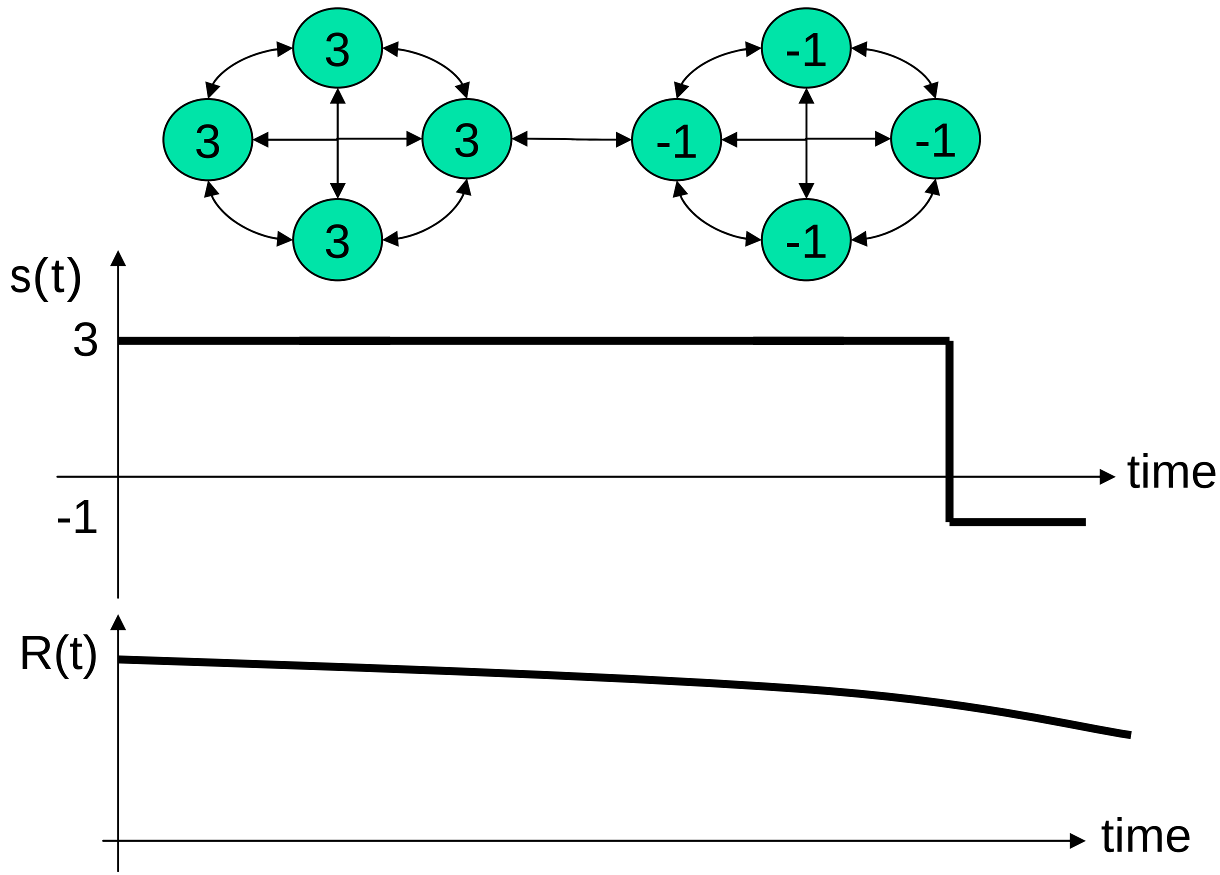
- Autocovariance function:

$$R(t) = \mathbf{E}(s(0)s(t)) - \mathbf{E}(s(0))\mathbf{E}(s(t))$$



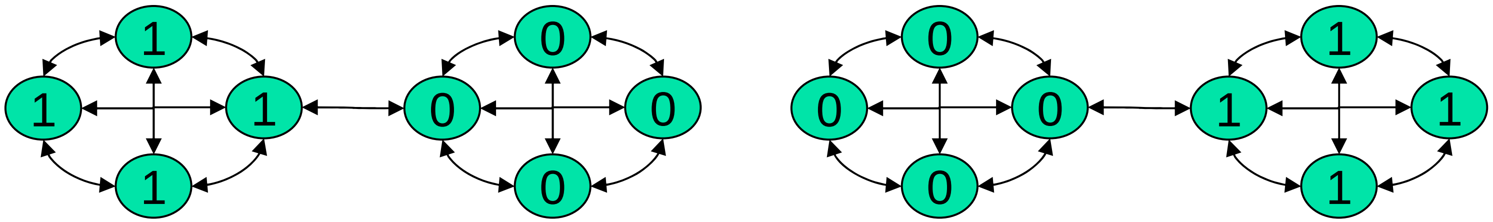
Functions on Markov chains

If the function is constant on clusters:



Functions and clusterings

- Good clustering = any function constant on clusters:
  - has slow signals = low frequencies
  - has autocovariance with slow decay
  - is slowly forgotten
- It is enough to check basis functions:



Stability of a clustering

- Stability = sum of autocovariance for all basis functions
- Stability(t) = $\sum_i h_i^T(\Pi - \pi\pi^T)M^t h_i$

- π = stationary distribution (=normalised degrees)
- $\Pi = \mathrm{diag}(\pi)$
- $h_i^T = (0\ldots011\ldots10\ldots0)$
  (basis function = characteristic function of cluster)
- M is matrix of transition probabilities
  (=normalised adjacency matrix)
- Computation up to t = $\mathcal{O}(t.|\mathrm{edges}|)$

The best clustering

- We want to maximise stability
- At every time a different clustering may be optimal.
- From physical intuition, we expect:
  - high time=few clusters
  - low times=many clusters
  - A hierarchy across time

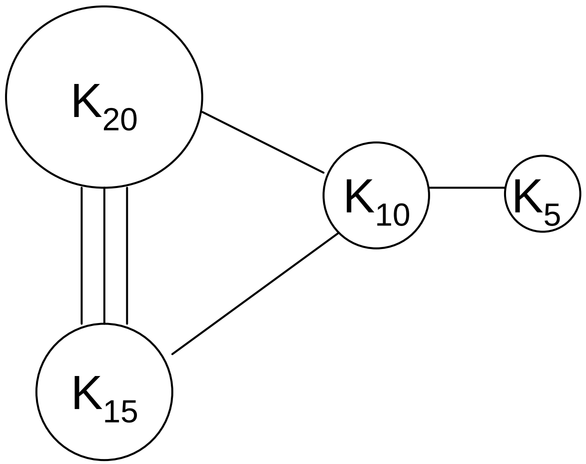
Stability, modularity, and spectral clustering

- Stability(0) is optimal for finest clustering
- Stability(1) is modularity
- Stability(∞) is optimal for 2-way clustering = sign of normalised Fiedler vector.

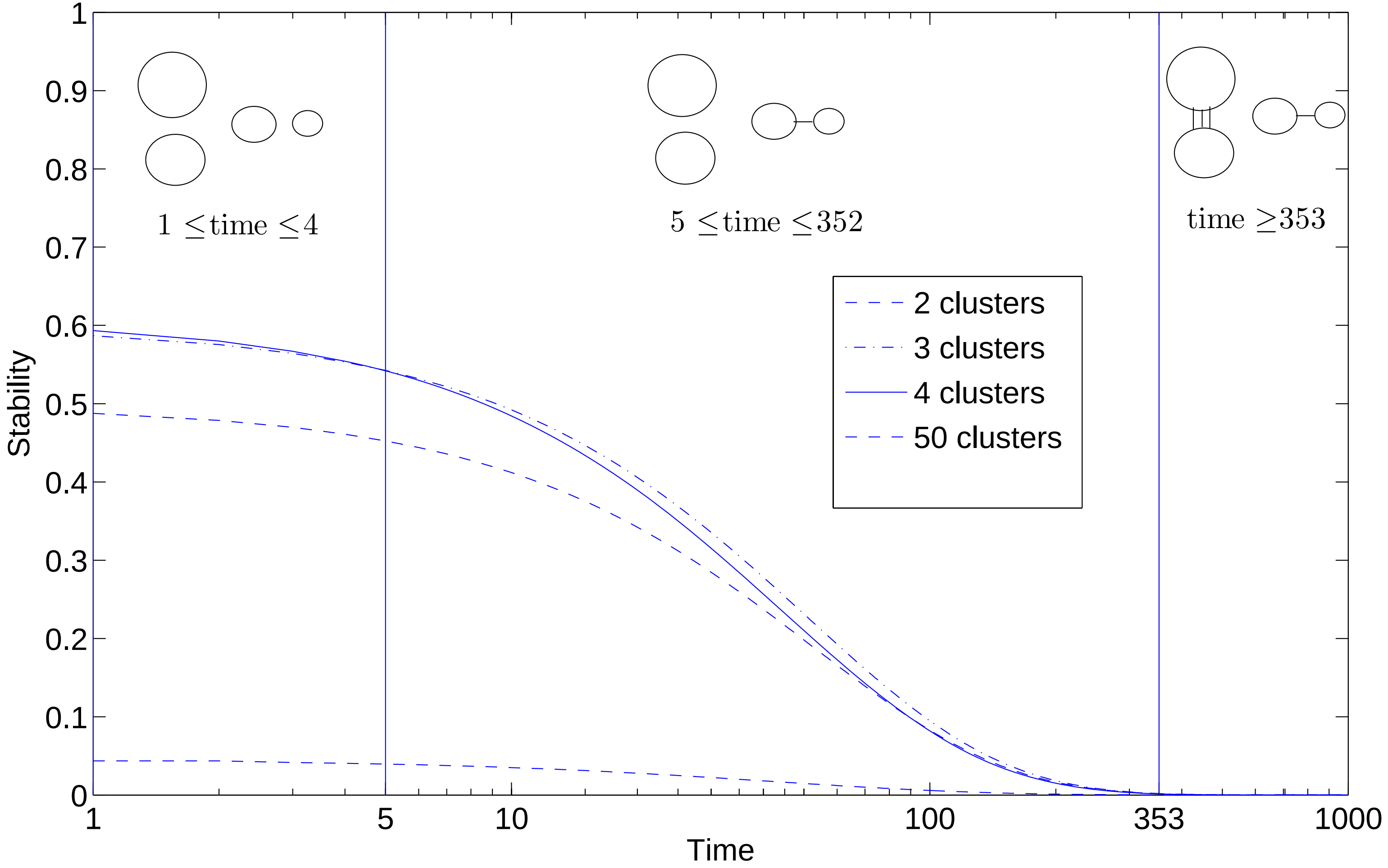
- Natural definition of modularity for Markov chains
- Spectral clustering= exact algorithm for largest times scales
- Sequence of optimal clusterings = hierachy?

Example

Graph with obvious clusters:



Example



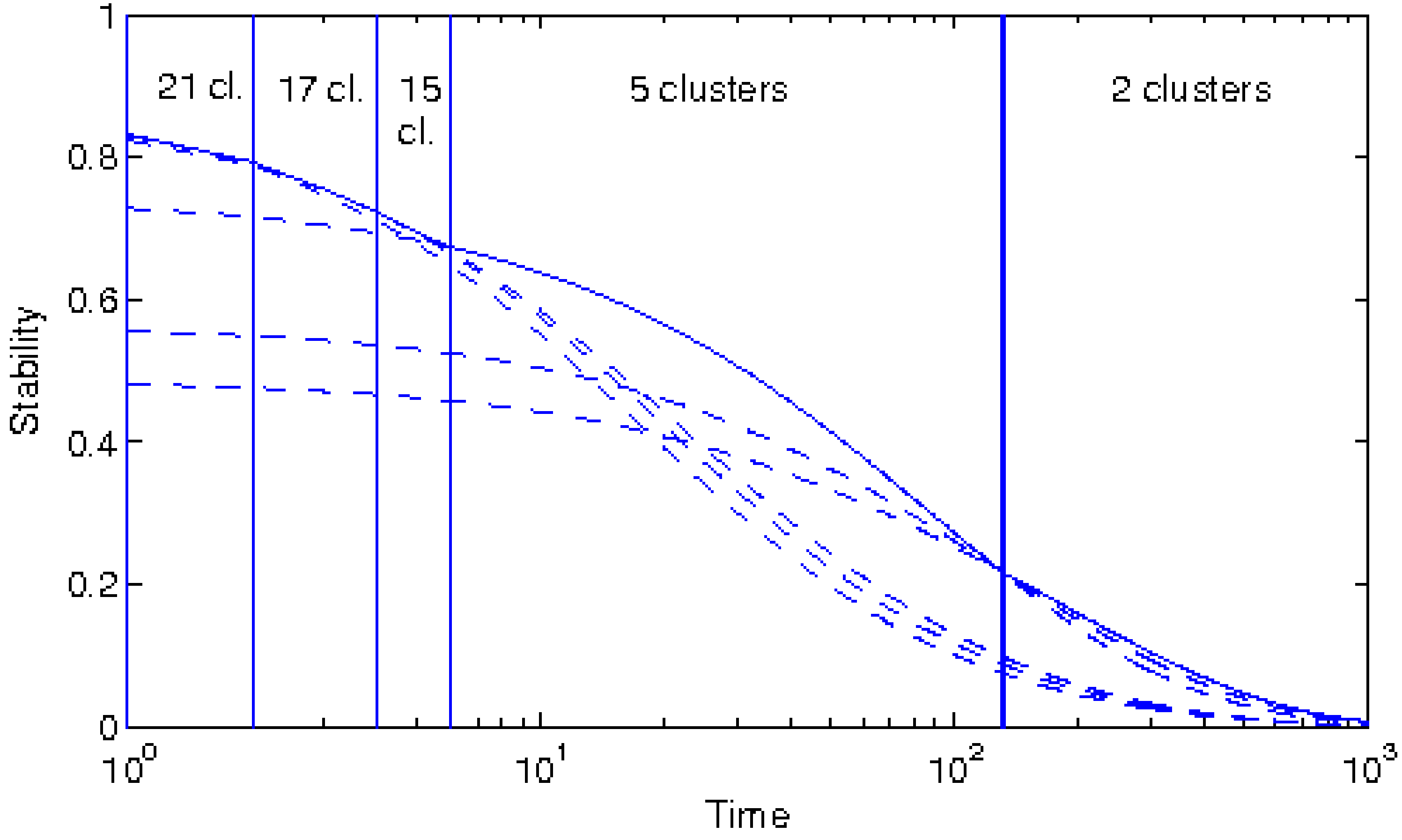
1 ≤time ≤4
5 ≤time ≤352
time ≥353
2 clusters
3 clusters
4 clusters
50 clusters
Stability
Time
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
1
5
10
100
353
1000

Applications

Application I: Hierarchical Clustering

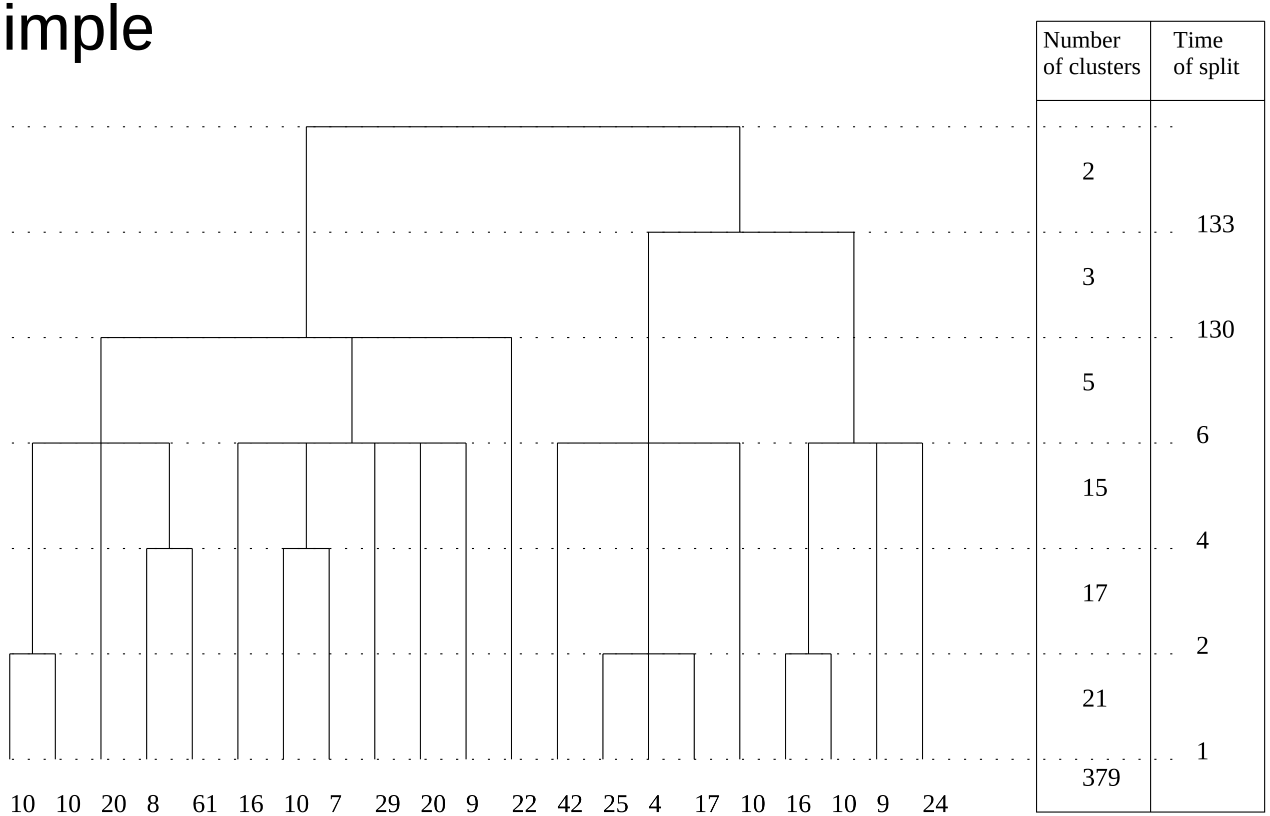
- A 379-node graph (collaborations in network science, collected by Mark Newman),
- Apply KVV algorithm: *k*-way clustering by spectral method
- Compute *k*-way clustering for every *k*
- Rank them according to stability for every time
- Discard those that are never optimal

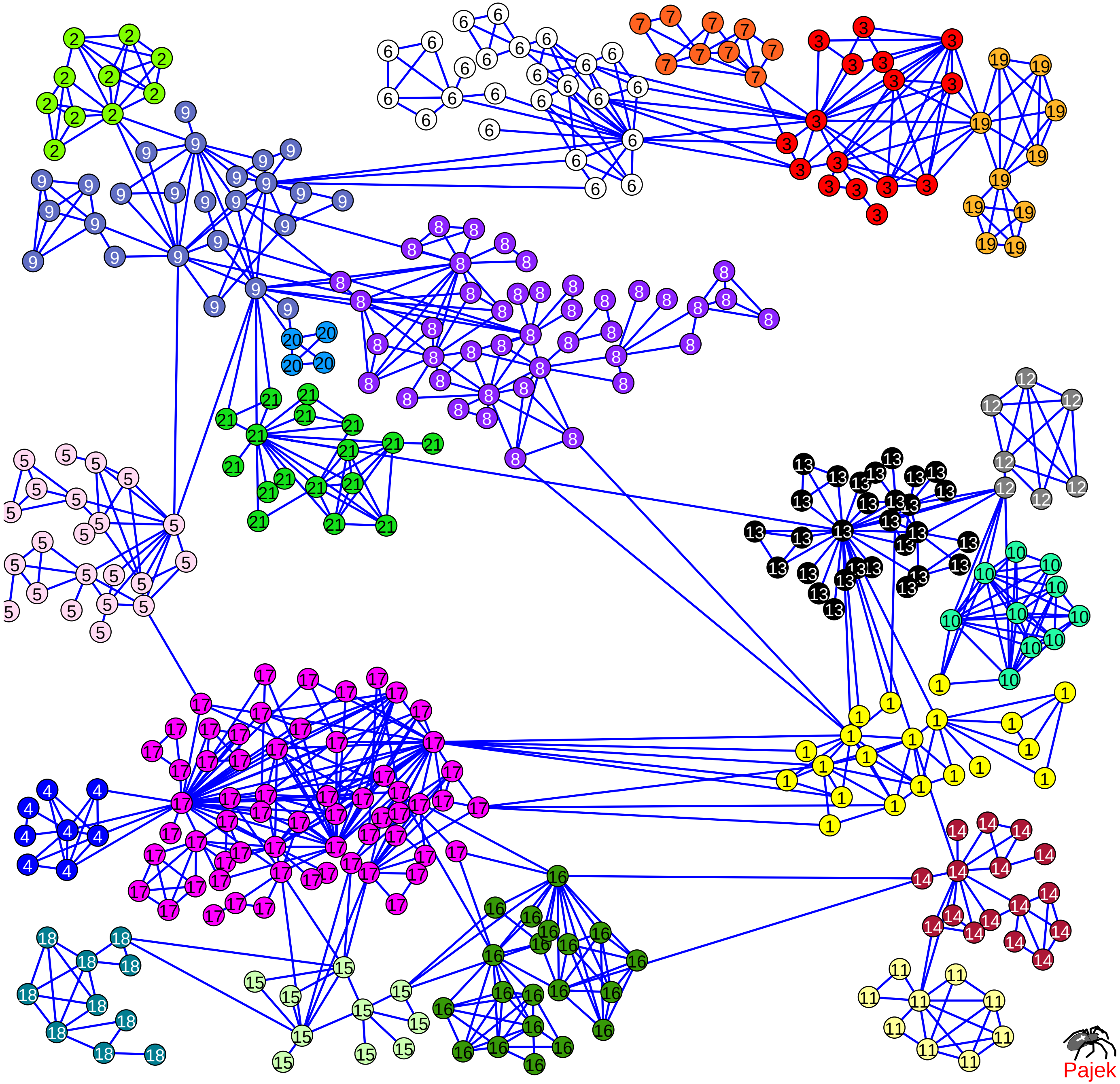
Stability curve



Dendrogram

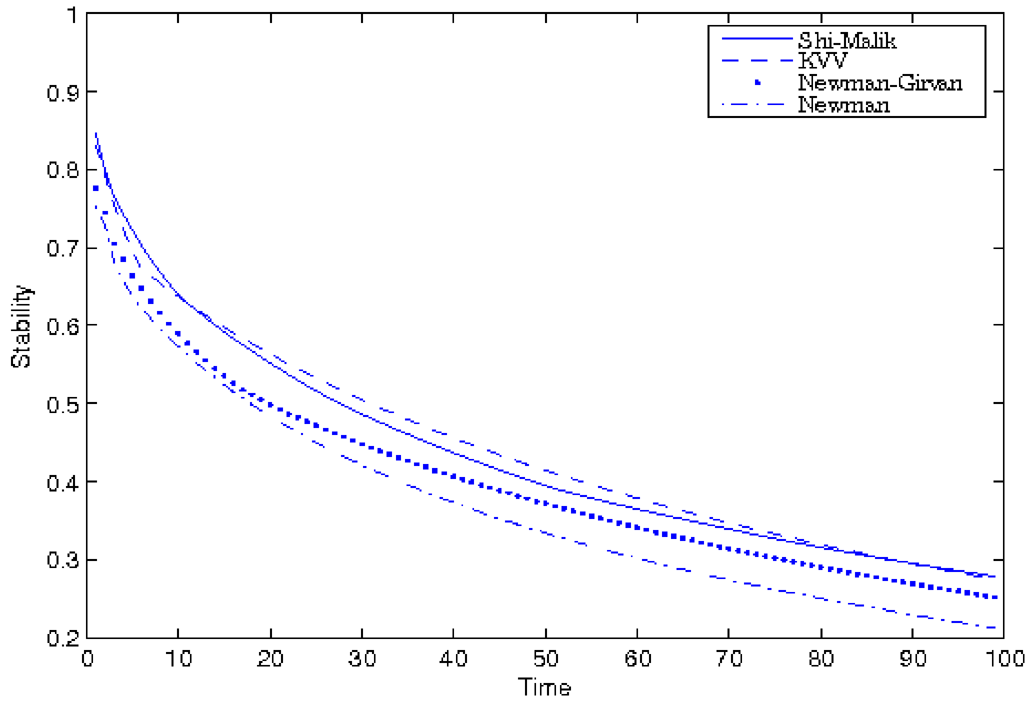
- Not a binary tree
- Simple




Pajek

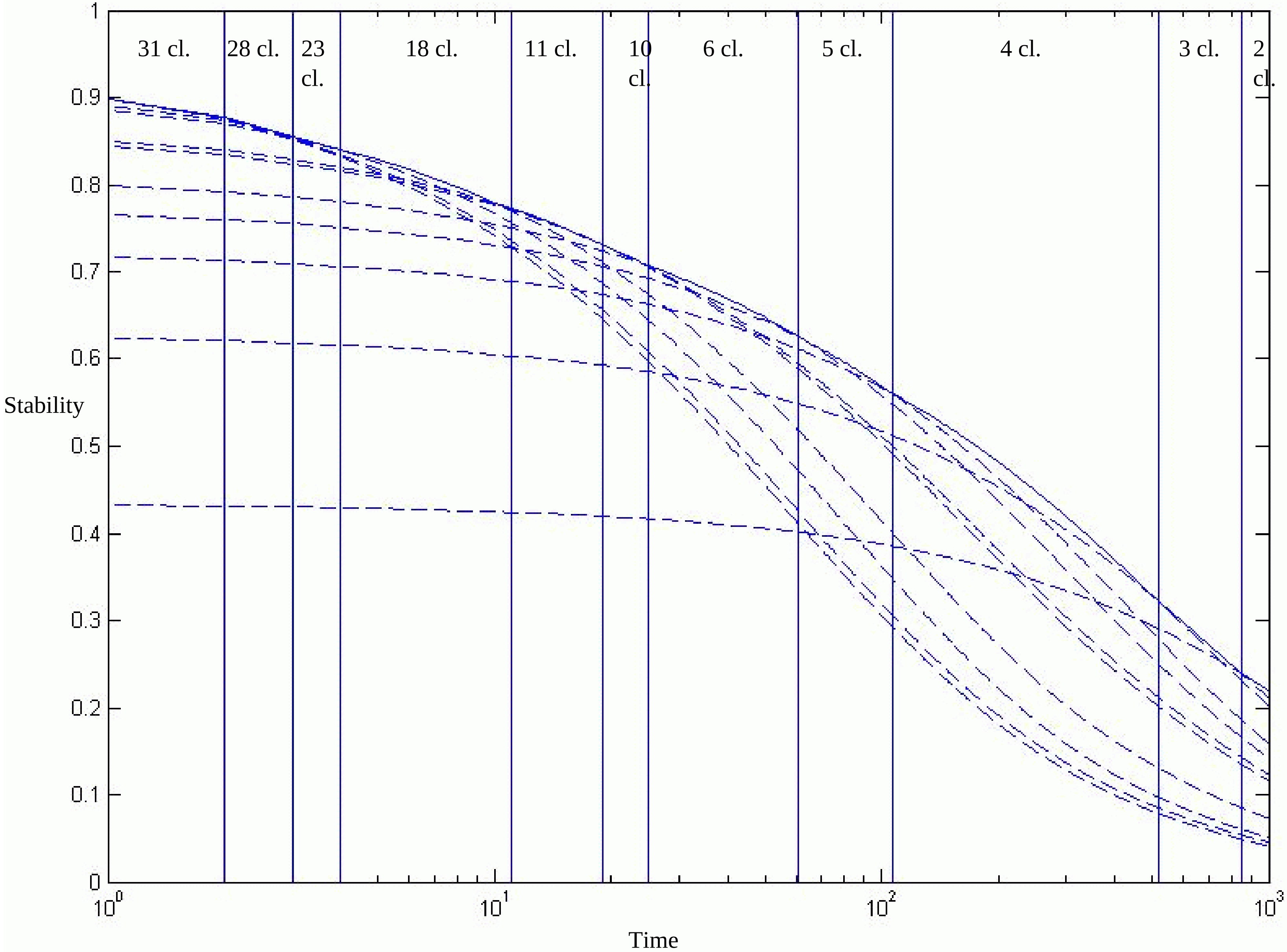
Application II:
Compare Clustering Algorithms

- A 379-node graph
- Apply several *k*-way clustering algorithms (for all *k*).
- Result: Performance can vary with time scale.

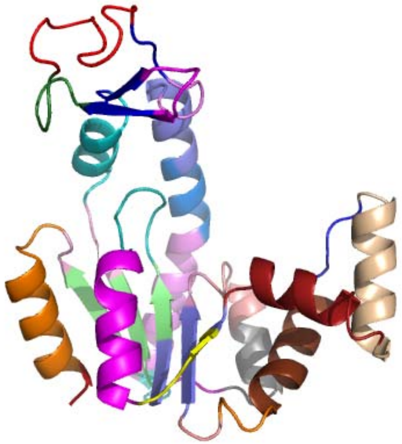


Application III: Model Reduction of Proteins

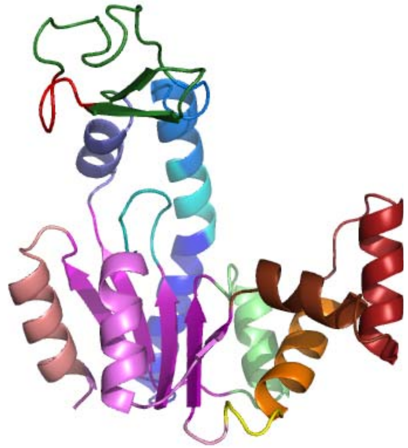
- Protein = graph of atoms and chemical bonds
- Dynamic modelling (e.g., mass-spring system)
- How to reduce the number of degrees of freedom: find rigid clusters.
- As a quick approximation: find clusters with Shi-Malik algorithm (= *k*-way clustering with spectral method).
- Intuition: a group of atoms with many bonds is more likely to be rigid.
- Compute *k*-way clusterings for all *k.*
- Rank the clusterings for every time.
- Most significant = largest time scales



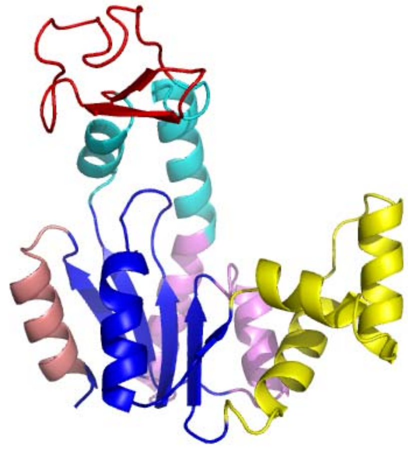
31 cl.
28 cl.
23 cl.
18 cl.
11 cl.
10 cl.
6 cl.
5 cl.
4 cl.
3 cl.
2 cl.
Stability
Time
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
10^0
10^1
10^2
10^3



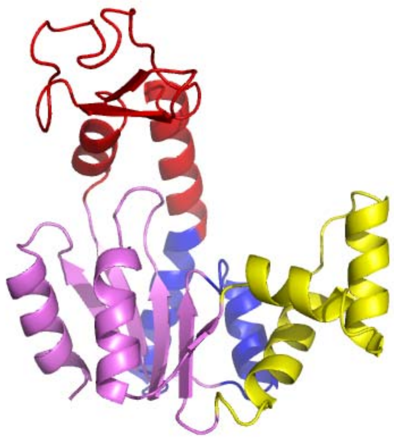
31 clusters (time=1)



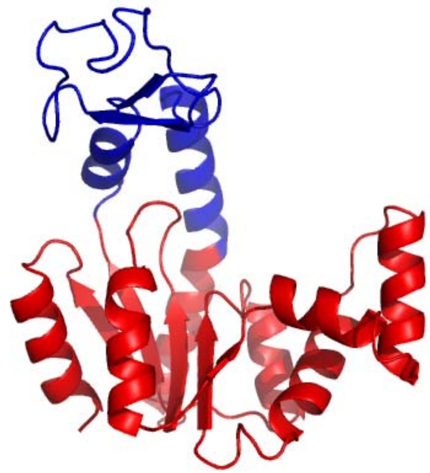
18 clusters (4 ≤ time ≤ 10)



6 clusters (25 ≤ time ≤ 60)



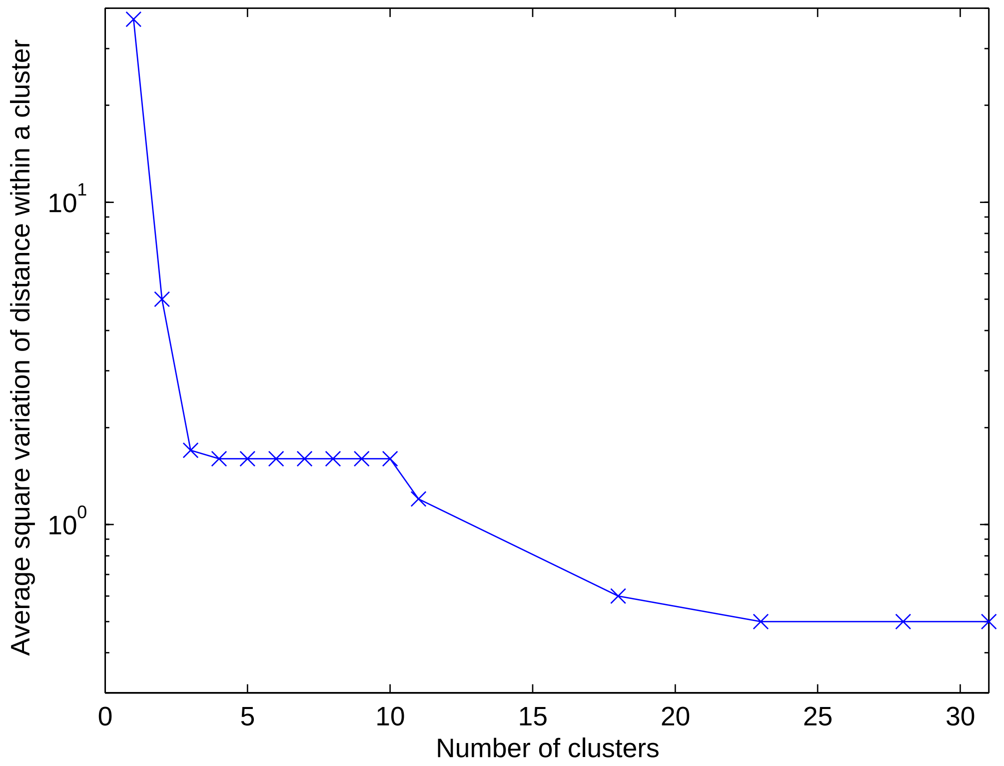
4 clusters (107 ≤ time ≤ 159)



2 clusters (time ≥ 852)

Change of distance within clusters

- To assess quality of clusters: change of intra-cluster distances between two different configurations



Extensions

Almost bipartite graphs

- Stability not always decreasing with time.

- Example: almost bipartite: oscillations
- Bipartite graph= « anti-clustering» = square root of clustering

- If we are only interested in clusterings:

  Average two consecutive times to kill oscillations.

Fractional times

- Fortunato 07: interesting clusterings finer than given by modularity.
- How to see between time=0 and time=1?
- Fractional powers if nonnegative spectrum
- Interpolate between $I = M^0$ and M^1
- Stability(t)= $\sum_i h_i^T(\Pi - \pi\pi^T)((1-t)I + tM)h_i$
- Close to continuous-time process:
$$(1-t)I + tM \approx \exp(t(M-I))$$
- Similar to Arenas, Fernandez, Gomez 07 (=adding self-loops of weight *r*)

Overlapping clusterings

- Membership function:
  - E.g., node *i* belongs to cluster 1 (60%) and cluster 3 (40%)

- Characteristic functions of clusters:
  - Stability(t)= $\sum_i h_i^T(\Pi - \pi\pi^T)M^t h_i$
  - $h_i \geq 0$; $\sum_i h_i = \mathbf{1}$

Conclusions

- Stability of clustering = autocovariance
- Applies to reversible Markov chains, undirected graphs
- Optimal clustering depends on time scale
- Bridge between modularity (time=1) and spectral clustering (time=∞)

- Applications:
  - comparing algorithms
  - creating simple hierachies
  - find significant time scales