Graph partitioning algorithms

Leonid E. Zhukov

School of Data Analysis and Artificial Intelligence
Department of Computer Science
National Research University Higher School of Economics

Network Science

NATIONAL RESEARCH UNIVERSITY

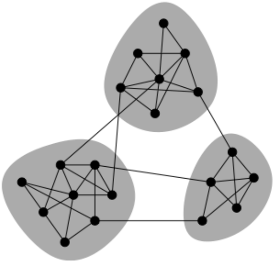
Lecture outline

1. Graph paritioning
   - Metrics
   - Algorithms
2. Spectral optimization
   - Min cut
   - Normalized cut
   - Modularity maximization
3. Multilevel spectral

Network communities

Definition

Network communities are groups of vertices such that vertices inside the group connected with many more edges than between groups.



- Graph partitioning problem

Graph partitioning

Combinatorial problem:

- Number of ways to divide network of n nodes in 2 groups (bi-partition):

$$\frac{n!}{n_1!n_2!}, \qquad n = n_1 + n_2$$

- Dividing into k non-empty groups (Stirling numbers of the second kind)

$$S(n,k) = \frac{1}{k!}\sum_{j=0}^{n}(-1)^j C_k^j (k-j)^n$$

- Number of all possible partitions (n-th Bell number):

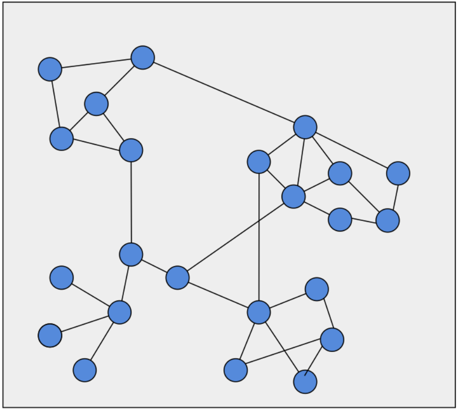
$$B_n = \sum_{k=1}^{n} S(n,k)$$

$B_{20} = 5,832,742,205,057$

Community detection

- Consider only sparse graphs $m \ll n^2$
- Each community should be connected
- Combinatorial optimization problem:
  - optimization criterion
  - optimization method
- Exact solution NP-hard
  (bi-partition: $n = n_1 + n_2, \;\; n!/(n_1!n_2!)$ combinations)
- Solved by greedy, approximate algorithms or heuristics
- Recursive top-down 2-way partition, multiway partiton
- Balanced class partition vs communities

Graph cut



Optimization criterion: graph cut

Graph $G(E, V)$ partition: $V = V_1 + V_2$

- Graph cut

$$Q = cut(V_1, V_2) = \sum_{i \in V_1, j \in V_2} e_{ij}$$

- Ratio cut:

$$Q = \frac{cut(V_1, V_2)}{||V_1||} + \frac{cut(V_1, V_2)}{||V_2||}$$

- Normalized cut:

$$Q = \frac{cut(V_1, V_2)}{Vol(V_1)} + \frac{cut(V_1, V_2)}{Vol(V_2)}$$

- Quotient cut (conductance):

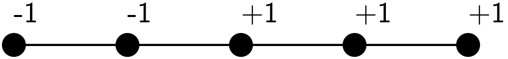
$$Q = \frac{cut(V_1, V_2)}{\min\left(Vol(V_1), Vol(V_2)\right)}$$

where: $Vol(V_1) = \sum_{i \in V_1, j \in V} e_{ij} = \sum_{i \in V_1} k_i$

Optimization methods

- Greedy optimization:
  – Local search [Kernighan and Lin, 1970], [Fidducia and Mettheyses, 1982]
- Approximate optimization:
  – Spectral graph partitioning [M. Fiedler, 1972], [Pothen et al 1990], [Shi and Malik, 2000]
  – Multicommodity flow [Leighton and Rao, 1988]
- Heuristics algorithms:
  – Multilevel graph partitioning (METIS) [G. Karypis, Kumar 1998]
- Randomized algorithms:
  –Randomized min cut [D. Karger, 1993]

- Let $V = V^+ + V^-$ be partitioning of the nodes
- Let $\mathbf{s} = \{+1, -1, +1, ... -1, +1\}^T$ - indicator vector



$$s(i) = \begin{cases} +1: & \text{if} \quad v(i) \in V^+ \\ -1: & \text{if} \quad v(i) \in V^- \end{cases}$$

- Number of edges, connecting V^+ and V^-

$$cut(V^+, V^-) = \frac{1}{4} \sum_{e(i,j)} (s(i) - s(j))^2 = \frac{1}{8} \sum_{i,j} A_{ij} (s(i) - s(j))^2 =$$

$$= \frac{1}{4} \sum_{i,j} (k_i \delta_{ij} s(i)^2 - A_{ij} s(i) s(j)) = \frac{1}{4} \sum_{i,j} (k_i \delta_{ij} - A_{ij}) s(i) s(j)$$

$$cut(V^+, V^-) = \frac{1}{4} \sum_{i,j} (D_{ij} - A_{ij}) s(i) s(j)$$

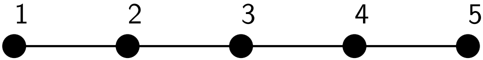
Graph cuts

- Graph Laplacian: $\mathbf{L}_{ij} = \mathbf{D}_{ij} - \mathbf{A}_{ij}$, where $\mathbf{D}_{ii} = diag(k_i)$

$$\mathbf{L}_{ij} = \begin{cases} k(i)\,, & \textit{if } \; i = j \\ -1\,, & \textit{if } \exists \; e(i,j) \\ 0\,, & \textit{otherwise} \end{cases}$$

- Laplacian matrix 5x5:

$$\mathbf{L} = \begin{pmatrix} 1 & -1 & & & \\ -1 & 2 & -1 & & \\ & -1 & 2 & -1 & \\ & & -1 & 2 & -1 \\ & & & -1 & 1 \end{pmatrix}$$



Graph cuts

- Graph Laplacian: $\mathbf{L} = \mathbf{D} - \mathbf{A}$
- Graph cut:
$$Q(\mathbf{s}) = \frac{\mathbf{s}^T \mathbf{L} \mathbf{s}}{4}$$
- Minimal cut:
$$\min_{\mathbf{s}} Q(\mathbf{s})$$
- Balanced cut constraint:
$$\sum_i s(i) = 0$$
- Integer minimization problem, exact solution is NP-hard!

Spectral method - relaxation

- Discrete problem $\rightarrow$ continuous problem
- Discrete problem: find

$$\min_{\mathbf{s}}(\frac{1}{4}\mathbf{s}^T\mathbf{L}\mathbf{s})$$

  under constraints: $s(i) = \pm 1$, $\sum_i s(i) = 0$;
- Relaxation - continuous problem: find

$$\min_{\mathbf{x}}(\frac{1}{4}\mathbf{x}^T\mathbf{L}\mathbf{x})$$

  under constraints: $\sum_i x(i)^2 = n$, $\sum_i x(i) = 0$
- Given $x(i)$, round them up by $s(i) = sign(x(i))$
- Exact constraint satisfies relaxed equation, but not other way around!

Spectral method - computations

- Constraint optimization problem (Lagrange multipliers):

$$Q(\mathbf{x}) = \frac{1}{4}\mathbf{x}^T\mathbf{L}\mathbf{x} - \lambda(\mathbf{x}^T\mathbf{x} - n), \quad \mathbf{x}^T\mathbf{e} = 0$$

- Eigenvalue problem:

$$\mathbf{L}\mathbf{x} = \lambda\mathbf{x}, \quad \mathbf{x} \perp \mathbf{e}$$

- Solution:

$$Q(\mathbf{x_i}) = \frac{n}{4}\lambda_i$$

- First (smallest) eigenvector:

$$\mathbf{L}\mathbf{e} = 0, \quad \lambda = 0, \quad \mathbf{x_1} = \mathbf{e}$$

- Looking for the second smallest eigenvalue/eigenvector λ_2 and $\mathbf{x}_2$
- Minimization of Rayleigh-Ritz quotient:

$$\min_{\mathbf{x}\perp\mathbf{x}_1}\left(\frac{\mathbf{x}^T\mathbf{L}\mathbf{x}}{\mathbf{x}^T\mathbf{x}}\right)$$

Spectral graph theory

- $\lambda_1 = 0$
- Number of $\lambda_i = 0$ equal to the number of connected components

- $0 \leq \lambda_2 \leq 2$
  $\lambda_2 = 0$, disconnected graph
  $\lambda_2 = 1$, totally connected

- Graph diameter (longest shortest path)

$$D(G) >= \frac{4}{n\lambda_2}$$

Spectral graph partitioning algorithm

Algorithm: Spectral graph partitioning - normalized cuts

Input: adjacency matrix $\mathbf{A}$

Output: class indicator vector $\mathbf{s}$

compute $\mathbf{D} = diag(deg(\mathbf{A}))$;

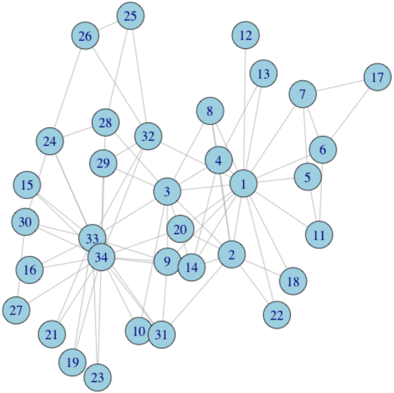
compute $\mathbf{L} = \mathbf{D} - \mathbf{A}$;

solve for second smallest eigenvector:
min cut: $\mathbf{Lx} = \lambda \mathbf{x}$;

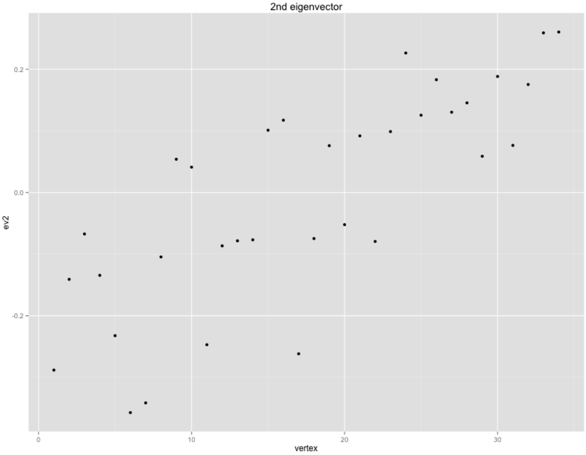
normalized cut : $\mathbf{Lx} = \lambda \mathbf{Dx}$;

set $\mathbf{s} = sign(\mathbf{x}_2)$

Example

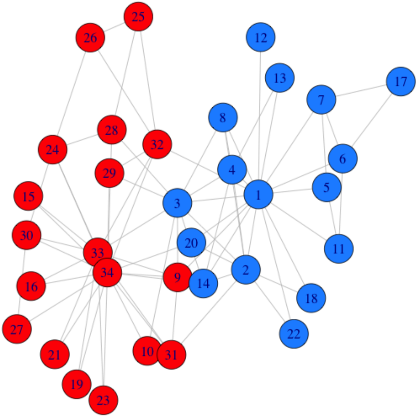

25
26
12
13
17
7
8
28
32
24
6
4
29
5
1
15
3
30
20
11
33
34
2
9
14
16
18
27
22
10
31
21
19
23

Example

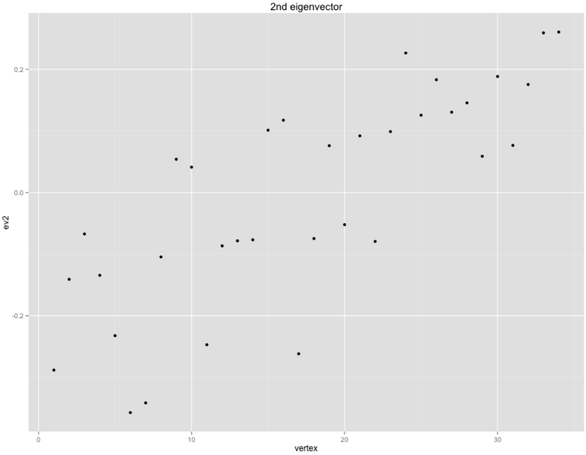


Eigenvalues: $\lambda_1 = 0$, $\lambda_2 = 0.2$, $\lambda_3 = 0.25$ ...

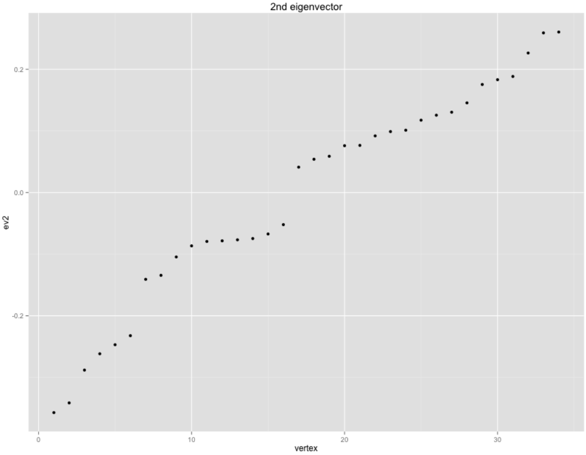
Example


25
26
12
13
17
7
8
28
32
24
6
4
29
5
1
15
3
30
20
11
33
2
34
9
14
16
18
27
22
21
10
31
19
23

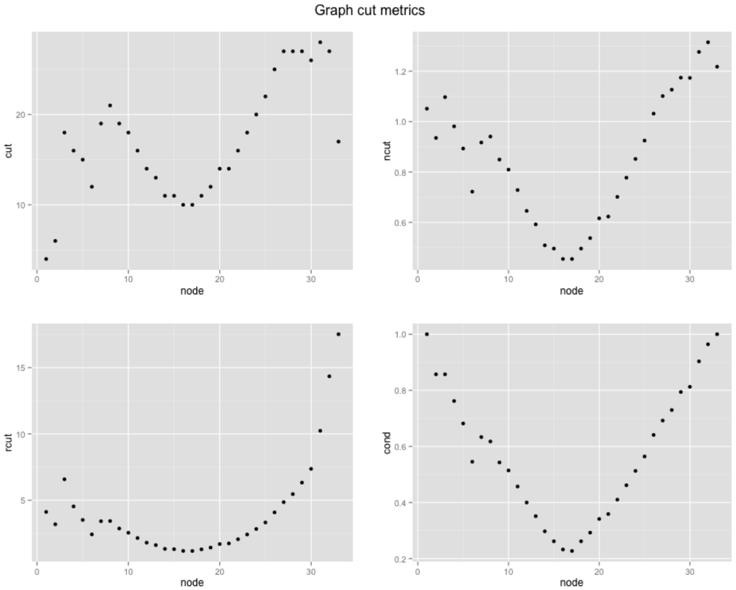
Example



Spectral ordering



Cut metrics


Graph cut metrics
cut
ncut
rcut
cond
node
node
node
node
20
10
1.2
1.0
0.8
0.6
15
10
5
1.0
0.8
0.6
0.4
0.2
0
10
20
30

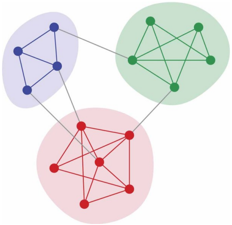
Optimization criterion: modularity

- Modularity:

$$Q = \frac{1}{2m} \sum_{ij} \left(A_{ij} - \frac{k_i k_j}{2m} \right) \delta(c_i, c_j)$$

where n_c - number of classes and

$$\delta(c_i, c_j) = \begin{cases} 1: & \text{if } \ c_i = c_j \\ 0: & \text{if } \ c_i \neq c_j \end{cases}$$ - kronecker delta



[Maximization!]

Spectral modularity maximization

- Direct modularity maximization for bi-partitioning, [Newman, 2006]
- Let two classes $c_1 = V^+$, $c_2 = V^-$, indicator variable $s = \pm 1$

$$\delta(c_i, c_j) = \frac{1}{2}(s_i s_j + 1)$$

- Modularity

$$Q = \frac{1}{4m} \sum_{ij} \left(A_{ij} - \frac{k_i k_j}{2m} \right) (s_i s_j + 1) = \frac{1}{4m} \sum_{i,j} B_{ij} s_i s_j$$

where

$$B_{ij} = A_{ij} - \frac{k_i k_j}{2m}$$

M. Newman, 2006

Spectral modularity maximization

- Qudratic form:

$$Q(\mathbf{s}) = \frac{1}{4m}\mathbf{s}^T\mathbf{B}\mathbf{s}$$

- Integer optimization - NP, relaxation $s \to x$, $x \in R$
- Keep norm $||x||^2 = \sum_i x_i^2 = \mathbf{x}^T\mathbf{x} = n$
- Quadratic optimization

$$Q(\mathbf{x}) = \frac{1}{4m}\mathbf{x}^T\mathbf{B}\mathbf{x} - \lambda(\mathbf{x}^T\mathbf{x} - n)$$

- Eigenvector problem

$$\mathbf{B}\mathbf{x}_i = \lambda_i\mathbf{x}_i$$

- Approximate modularity

$$Q(\mathbf{x_i}) = \frac{n}{4m}\lambda_i$$

- Modularity maximization - largest $\lambda = \lambda_{max}$

Spectral modularity maximization

- Can't choose $\mathbf{s} = \mathbf{x}$k, can select optimal $\mathbf{s}$
- Decompose in the basis: $\mathbf{s} = \sum_j a_j \mathbf{x_j}$, where $a_j = \mathbf{x}_j^T \mathbf{s}$
- Modularity

$$Q(\mathbf{s}) = \frac{1}{4m}\mathbf{s}^T\mathbf{B}\mathbf{s} = \frac{1}{4m}\sum_i (\mathbf{x}_i^T\mathbf{s})^2\lambda_i$$

- $\max Q(\mathbf{s})$ reached when $\lambda_1 = \lambda_{max}$ and max $\mathbf{x}_1^T\mathbf{s} = \sum_j x_{1j}s_j$
- Choose $\mathbf{s}$ as close as possible to $\mathbf{x}$, i.e. $\max_s(\mathbf{s}^T\mathbf{x}) = \max_{s_i}\sum s_i x_i$:
  $s_i = +1$, if $x_i > 0$
  $s_i = -1$, if $x_i < 0$
- Choose $\mathbf{s} \parallel \mathbf{x}_1$, $\mathbf{s} = sign(\mathbf{x}_1)$

Modularity maximization

Algorithm: Spectral moduldarity maximization: two-way parition

Input: adjacency matrix $\mathbf{A}$

Output: class indicator vector $\mathbf{s}$

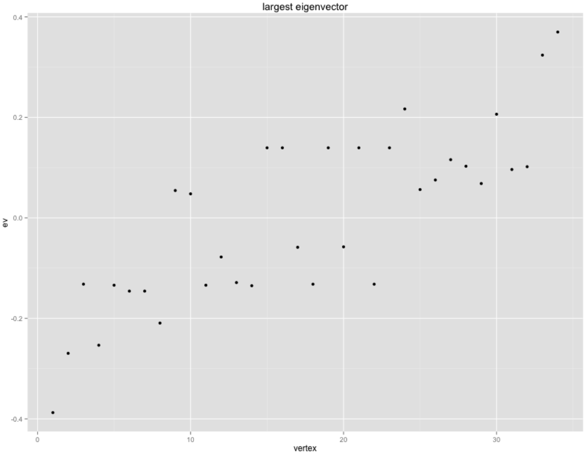
compute $\mathbf{k} = deg(\mathbf{A})$;

compute $\mathbf{B} = \mathbf{A} - \frac{1}{2m}\mathbf{k}\mathbf{k}^T$;

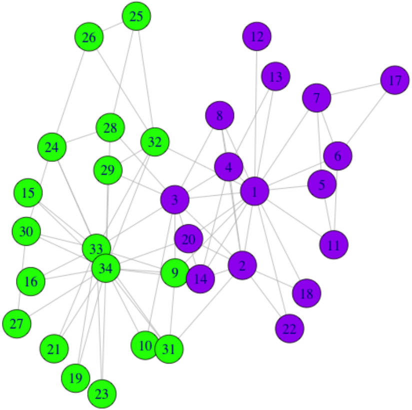
solve for maximal eigenvector $\mathbf{B}\mathbf{x} = \lambda\mathbf{x}$;

set $\mathbf{s} = sign(\mathbf{x}_{max})$

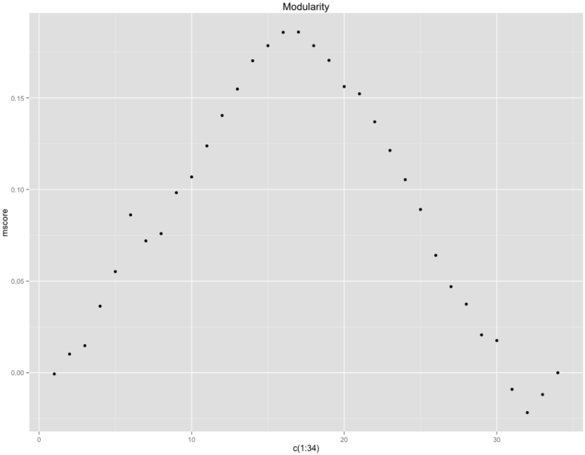
Example



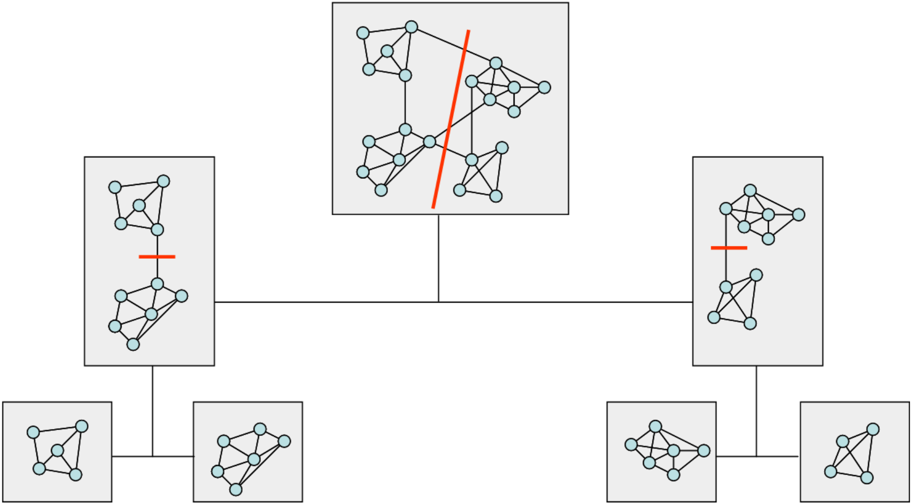
Example


25
26
12
13
17
7
8
28
32
24
6
4
29
5
1
15
3
30
20
11
33
2
34
9
14
16
18
27
22
10
31
21
19
23

Example

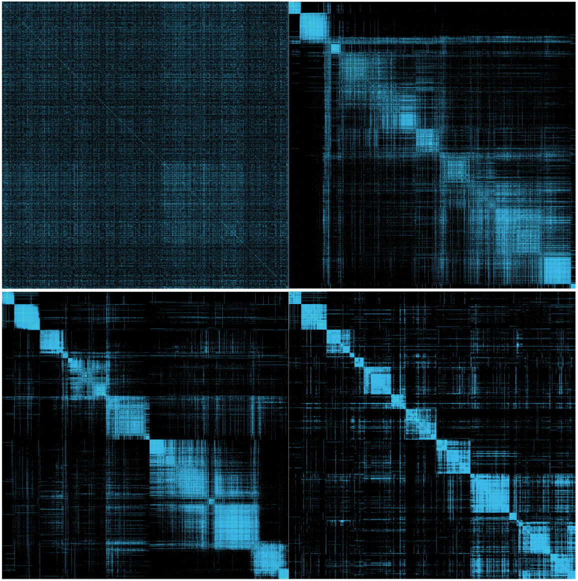


Multilevel spectral



recursive partitioning

Multilevel spectral



References

- M. Fiedler. Algebraic connectivity of graphs, Czech. Math. J, 23, pp 298-305, 1973
- A. Pothen, H. Simon and K. Liou. Partitioning sparse matrices with eigenvectors of graphs, SIAM Journal of Matrix Analysis, 11, pp 430-452, 1990
- Bruce Hendrickson and Robert Leland. A Multilevel Algorithm for Partitioning Graphs, Sandia National Laboratories, 1995
- Jianbo Shi and Jitendra Malik. Normalized Cuts and Image Segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, N 8, pp 888-905, 2000
- M.E.J. Newman. Modularity and community structure in networks. PNAS Vol. 103, N 23, pp 8577-8582, 2006
- B. Good, Y.-A. de Montjoye, A. Clauset. Performance of modularity maximization in practical contexts, Physical Review E 81, 046106, 2010