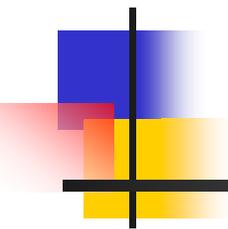
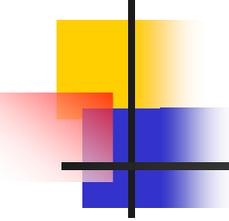


On Search, Ranking, and Matchmaking in Information Networks



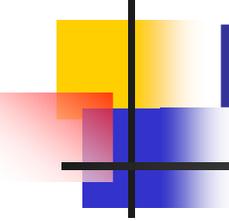
Sergei Maslov

Brookhaven National Laboratory



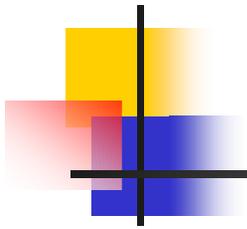
Information networks: WWW and beyond

- First part of my talk: 10^{10} webpages in the world: need to search and rank!!
- Second part: opinion networks
 - WWW can be thought of as a network of opinions (hyperlinks – positive votes)
 - Our choices and opinions on products, services and each other – a much larger opinion network!
 - Very incomplete (sparse) → one could use intelligent “matchmaking” to match users to new products or each other



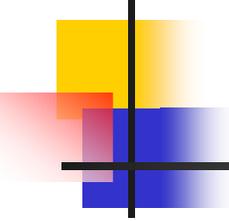
Ranking webpages

- Assign an “importance factor” G_i to every webpage
- Given a keyword (say “jaguar”) find all the pages that have it in their text and display them in the order of descending G_i .
- One solution still used in scientific publishing is $G_i = K_{in}(i)$ (the number of incoming links), but:
 - Too democratic: It doesn’t take into account the importance of nodes sending links
 - Easy to trick and artificially boost the ranking



How Google works

- Google's recipe (circa 1998) is to simulate the behavior of **many virtual "random surfers"**
- **PageRank**: G_i ~ the **number of virtual hits** the page gets. It is also ~ the **steady state** number of random surfers at a given page
- Popular pages send more surfers your way → PageRank ~ K_{in} weighted by the **popularity** of a source of each hyperlink
- Surfers get bored following links → with probability **$\alpha=0.15$** at any timestep a surfer jumps to a randomly selected page (not following any hyperlinks)
- Last rule also solves the **ergodicity problem**



Mathematics of the Google

- To calculate the PageRank Google solves a self-consistent Eq.:

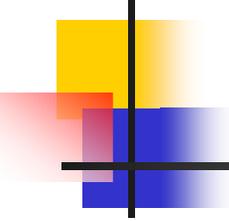
$$G_i \sim \sum_{j \rightarrow i} G_j / K_{\text{out}}(j)$$

- To account for random jumps:

$$\begin{aligned} G_i &= (1-\alpha) \sum_{j \rightarrow i} G_j / K_{\text{out}}(j) + \alpha \sum_j G_j / N \\ &= (1-\alpha) \sum_{j \rightarrow i} G_j / K_{\text{out}}(j) + \alpha \end{aligned}$$

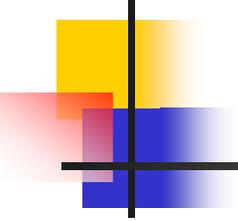
(uses normalization: $\langle G \rangle = \sum_j G_j / N = 1$)

- Pages with $K_{\text{out}}(j) = 0$ are removed



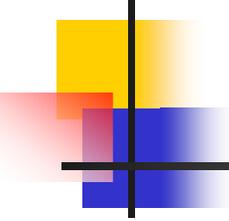
Matrix formulation

- Equivalent to finding the **principal eigenvector** (with $\lambda=1$) of the matrix $(1-\alpha) \mathbf{T} + \alpha \mathbf{U}$, where $T_{ij} = 1/K_{\text{out}}(j)$ if $j \rightarrow i$ and 0 otherwise, and $U_{ij} = 1/N$
- Could be **easily solved iteratively** by starting with $G_i^{(0)} = 1$ and repeating $G^{(n+1)} = (1-\alpha) \mathbf{T} G^{(n)} + \alpha$
- All $G_i > \alpha$



How Communities in the WWW influence the Google ranking

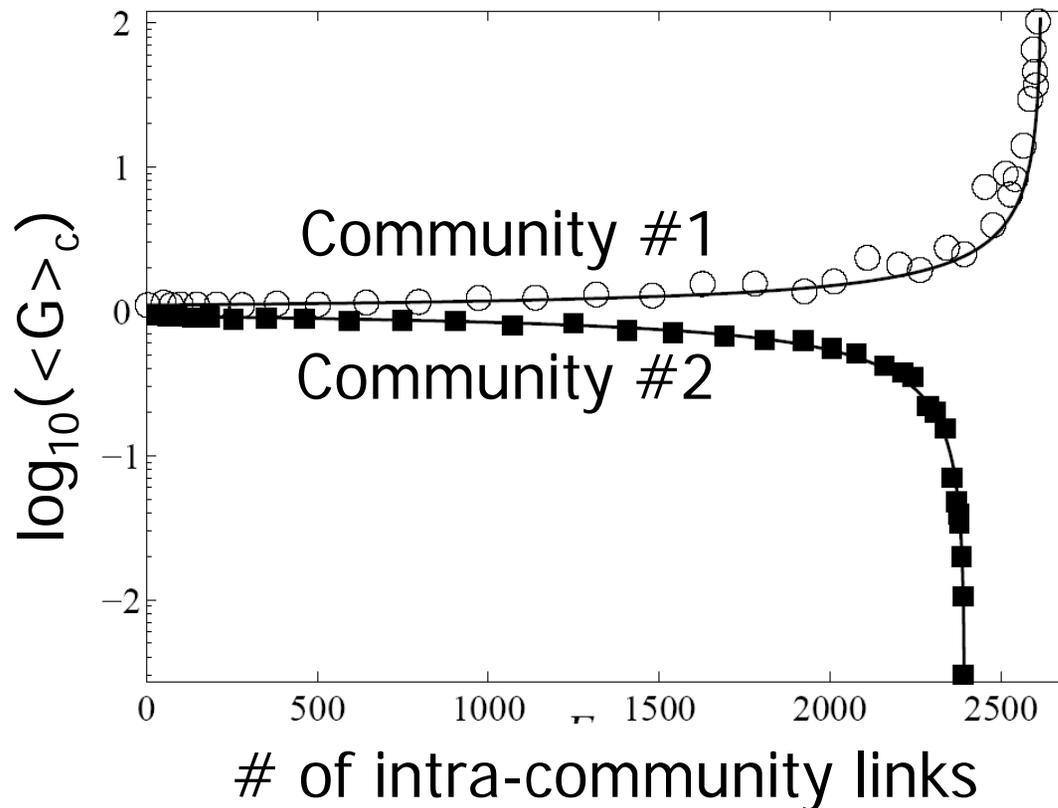
H. Xie, K.-K. Yan, SM, cond-mat/0409087



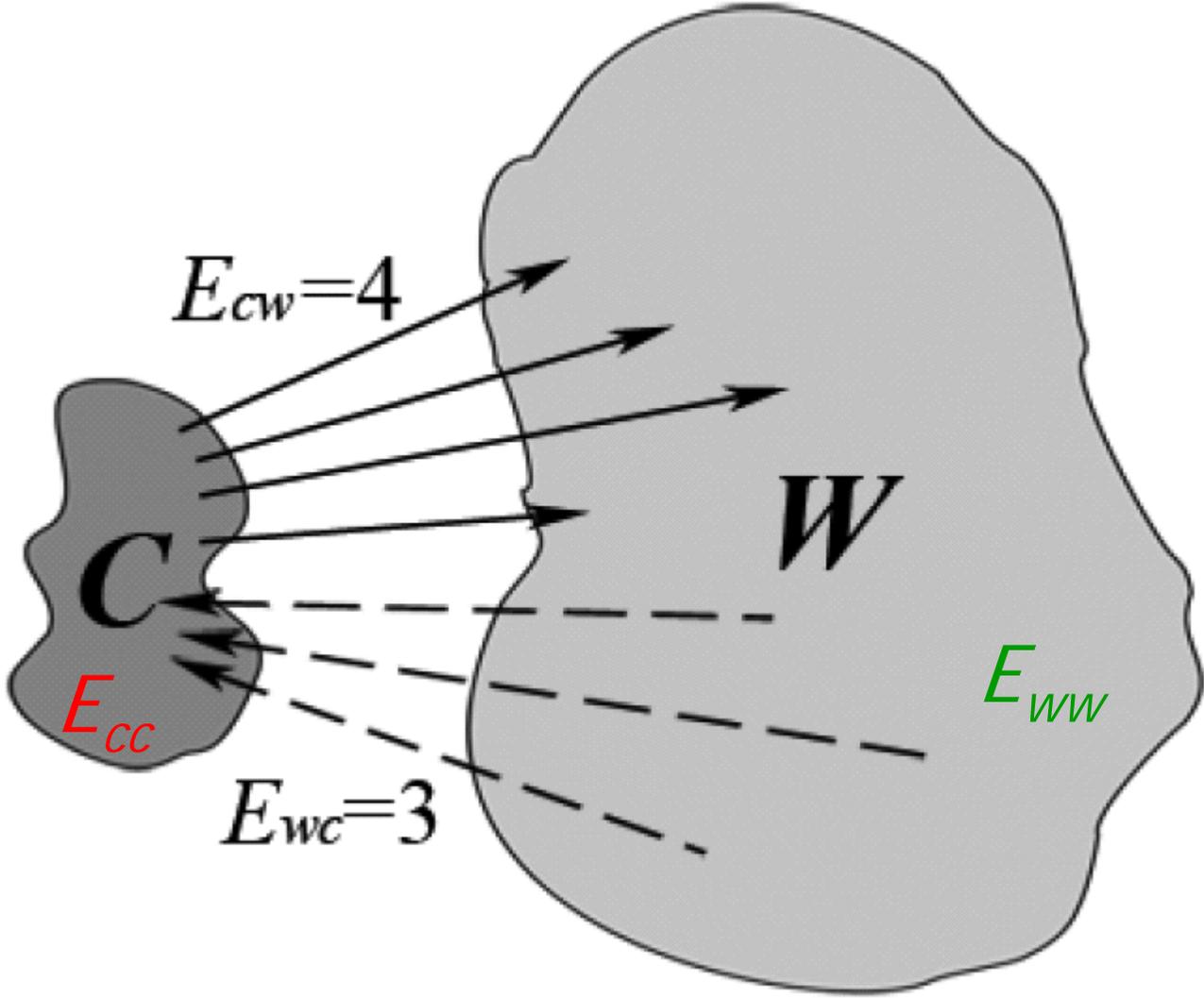
How do WWW communities influence their average G_i ?

- Pages in a web-community preferentially link to each other. Examples:
 - Pages from the same organization (e.g. SFI)
 - Pages devoted to a common topic (e.g. physics)
 - Pages in the same geographical location (e.g. Santa Fe)
- Naïve argument: **communities** tend to “**trap**” random surfers to spend more time inside them
→ they should **increase** the Google ranking of individual webpages in the community

Test of a naïve argument



- Naïve argument is **wrong!**
- The effect could go **either way**



- G_c – average Google rank of pages in the community; $G_w=1$ – in the outside world
- $E_{cw} G_c / \langle K_{out} \rangle_c$ – current from C to W
- It must be equal to:
 $E_{wc} G_w / \langle K_{out} \rangle_w$ – current from W to C

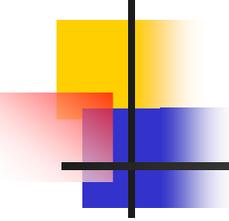
$$\frac{G_c}{G_w} = \frac{E_{wc}}{E_{cw}} \cdot \frac{\langle K_{out} \rangle_c}{\langle K_{out} \rangle_w}$$

- Thus G_c depends on the ratio between E_{cw} and E_{wc} – the number of edges (hyperlinks) between the community and the world

Balancing currents for nonzero α

- $J_{cw} = (1 - \alpha) E_{cw} G_c / \langle K_{out} \rangle_c + \alpha G_c N_c$
– current from C to W
- It must be equal to:
 $J_{cw} = (1 - \alpha) E_{wc} G_w / \langle K_{out} \rangle_w + \alpha G_w N_w (N_c / N_w)$
– current from W to C

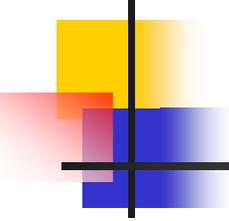
$$G_c = \frac{(1 - \alpha) \frac{E_{wc}}{N_c \langle K_{out} \rangle_w} + \alpha}{(1 - \alpha) \frac{E_{cw}}{N_c \langle K_{out} \rangle_c} + \alpha} = \frac{(1 - \alpha) \frac{E_{wc}}{E_{wc}^{(random)}} + \alpha}{(1 - \alpha) \frac{E_{cw}}{E_{cw}^{(random)}} + \alpha}$$



What are the consequences?

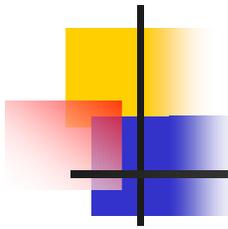
$$G_c = \frac{(1 - \alpha) \frac{E_{wc}}{E_{wc}^{(random)}} + \alpha}{(1 - \alpha) \frac{E_{cw}}{E_{cw}^{(random)}} + \alpha}$$

- For **very isolated communities** ($E_{cw}/E_{cw}^{(r)} < \alpha$ and $E_{wc}/E_{wc}^{(r)} < \alpha$) one has $G_c = 1$. Their Google rank is **decoupled** from the outside world!
- Overall range: $\alpha < G_c < 1/\alpha$



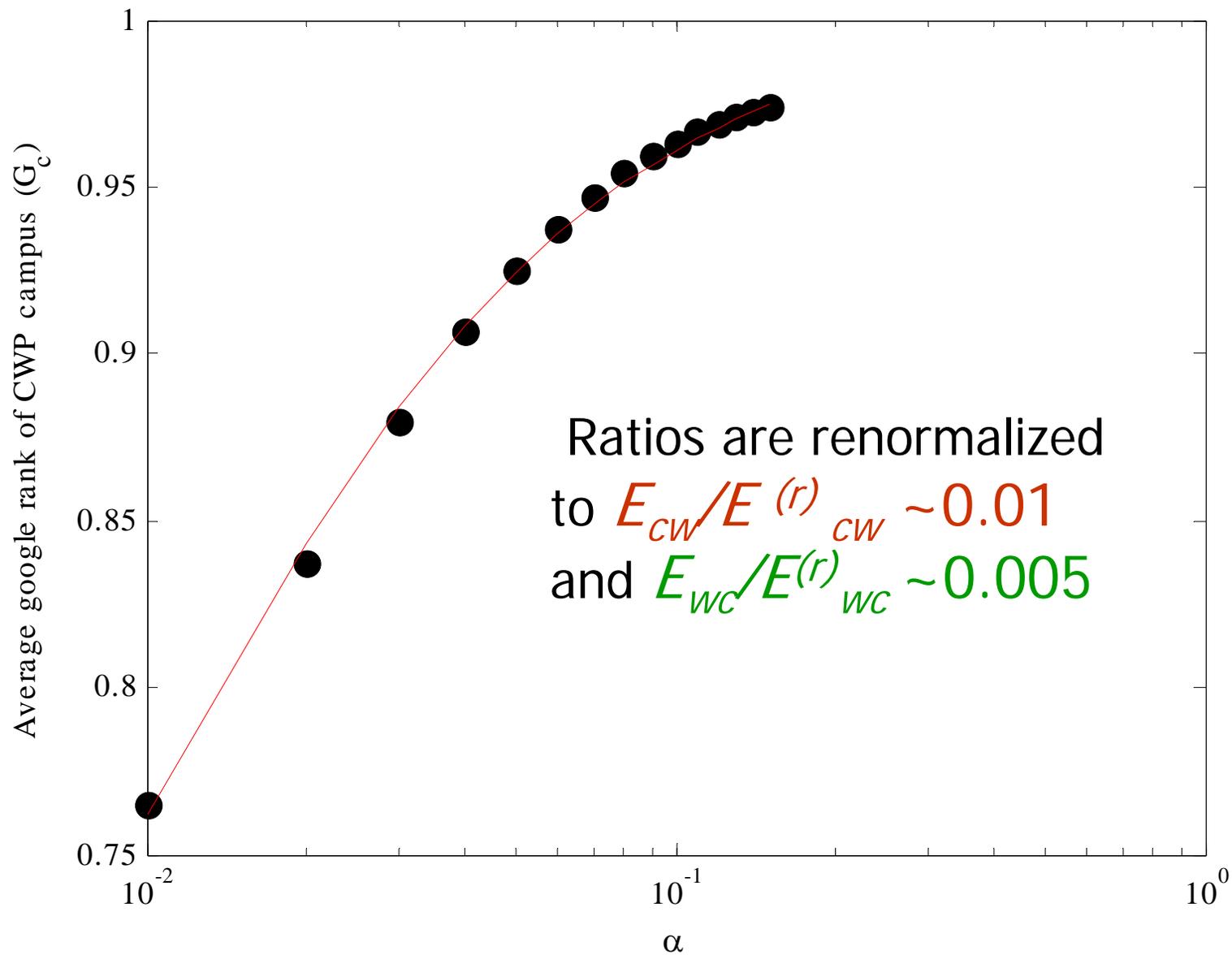
WWW - the empirical data

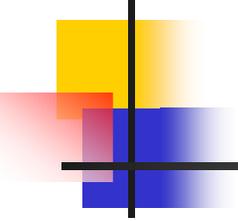
- We have data for **~10 US universities** (+ all UK and Australian Universities)
- Looked closely at **Long Island University**
 - 4 large campuses
 - 45,000 webpages and 160,000 hyperlinks
 - After removing $K_{out}=0$ left with ~15,000 webpages and 90,000 links
 - Can do a mini-Google PageRank on this set alone



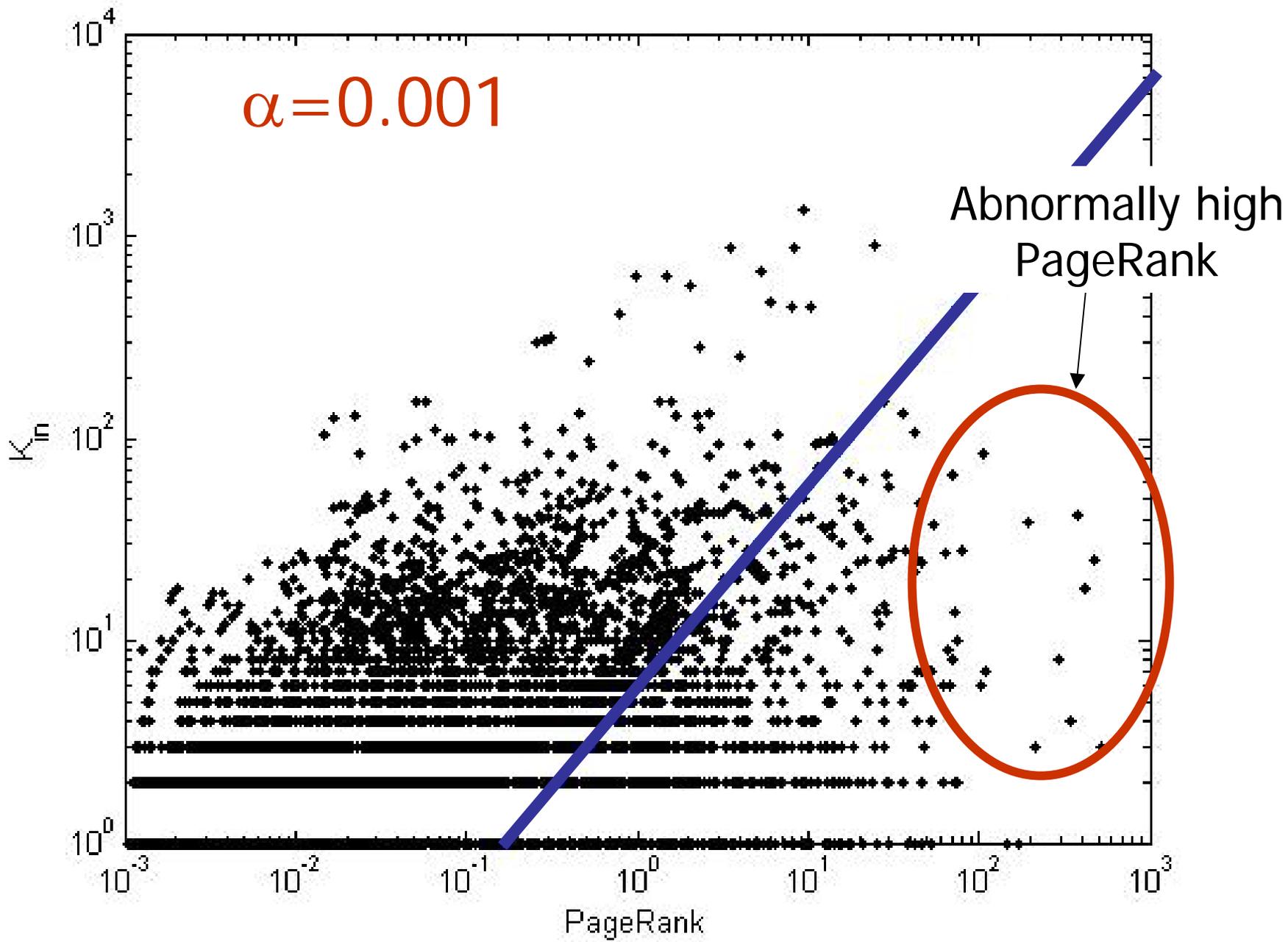
LIU communities

- LI University has 4 campuses. We looked at one of them (CWP Campus)
- $E_{CW} = 1393$; $E^{(r)}_{CW} \cong 16,000$;
 $E_{CW}/E^{(r)}_{CW} \sim 0.09 < \alpha = 0.15$
- $E_{WC} = 336$; $E^{(r)}_{WC} \cong 12,500$;
 $E_{WC}/E^{(r)}_{WC} \sim 0.03 < \alpha = 0.15$
- This community should be decoupled from the outside world



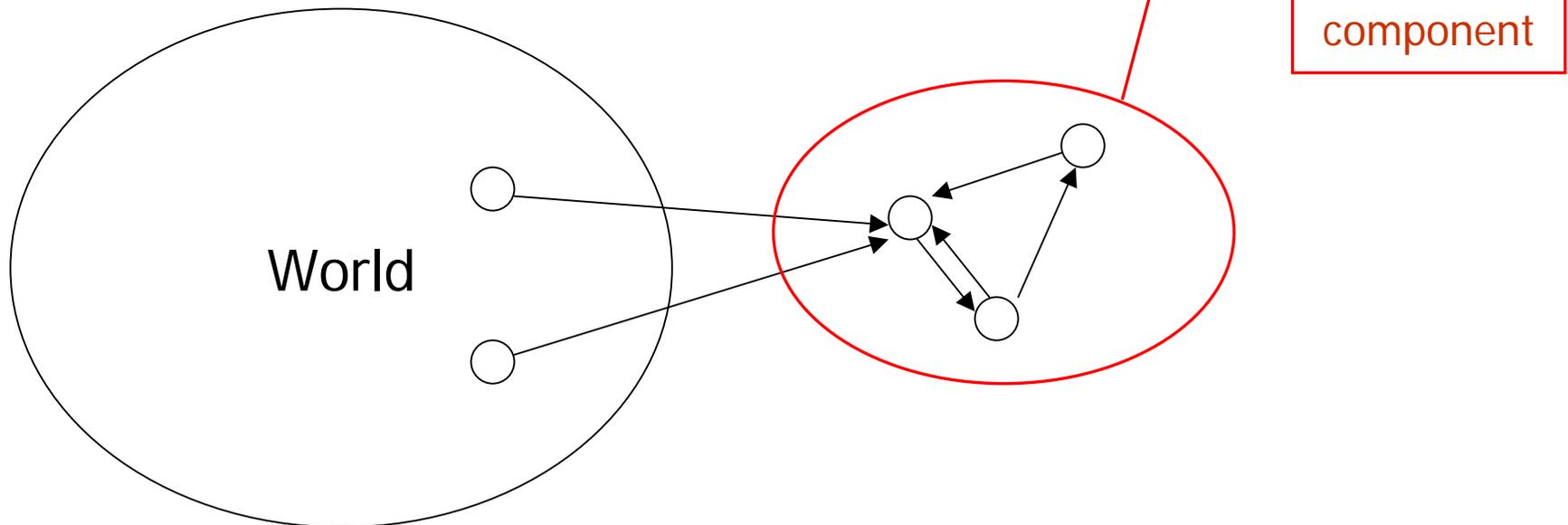


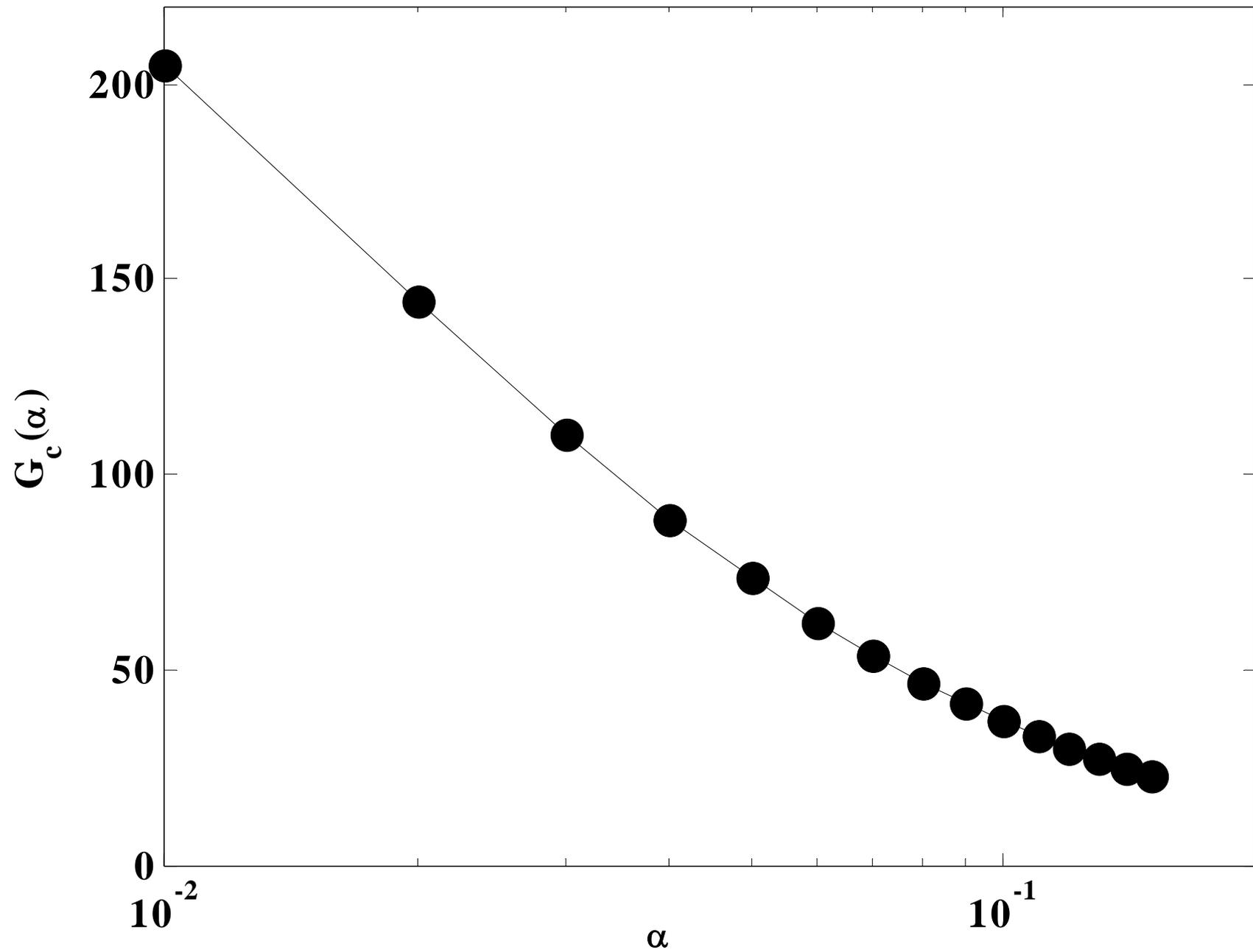
But: the community effect
could be also strong!

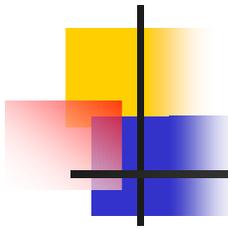


Top PageRank LIU websites for $\alpha=0.001$ don't make sense

- #1 www.cwpost.liu.edu/cwis/cwp/edu/edleader/higher_ed/hear.html
- #5 .../higher_ed/index.html
- #9 .../higher_ed/courses.html

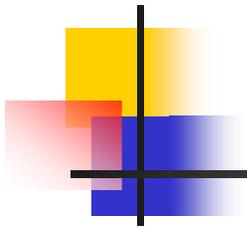






Collaborators and postdoc info:

- Collaborators:
 - Huafeng Xie – City University of NY
 - Koon-Kiu Yan - Stony Brook U.
- Looking for a **postdoc** to work in my group at **Brookhaven National Laboratory in New York** starting **Fall/Winter 2005 or even 2006**
 - Topics:
 - Large-scale properties of (mostly) **bionetworks** (partially supported by a NIH/NSF grant with Ariadne Genomics)
 - Internet/Google/Opinion networks
 - E-mail CV and 3 letters of recommendation to: **maslov@bnl.gov**; See **www.cmth.bnl.gov/~maslov**

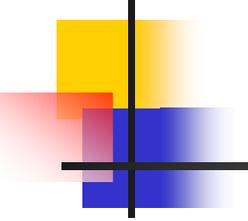


Part 2:

Opinion networks

"Extracting Hidden Information from Knowledge Networks", S. Maslov, and Y-C. Zhang, Phys. Rev. Lett. (2001).

"Exploring an opinion network for taste prediction: an empirical study", M. Blattner, Y.-C. Zhang, and S. Maslov, in preparation.

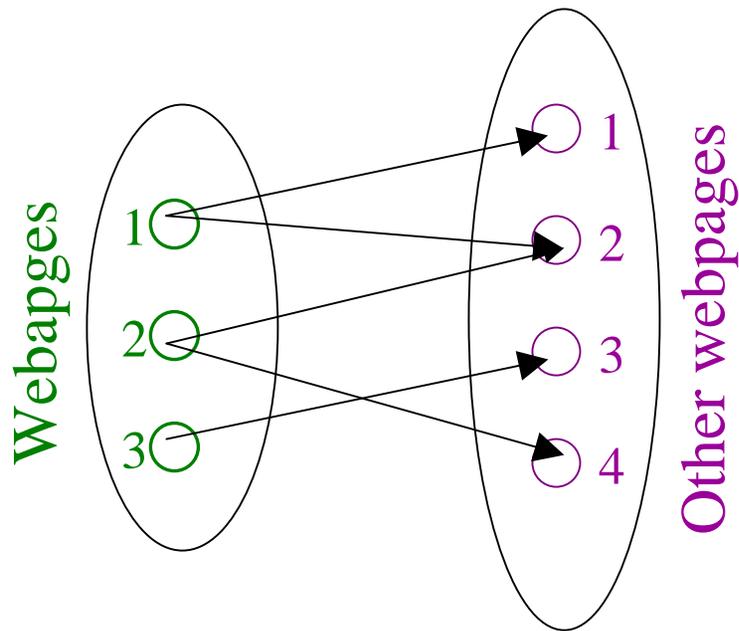


Predicting customers' tastes from their opinions on products

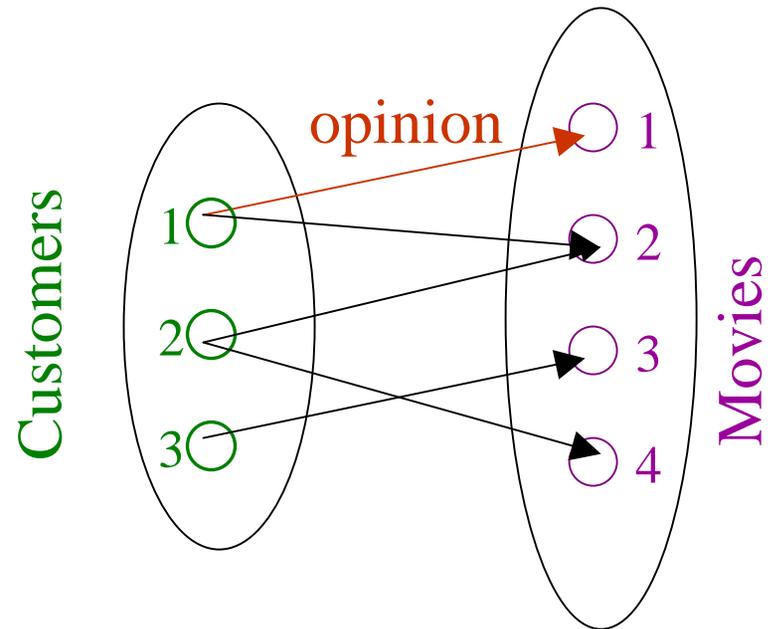
- Each of us has **personal tastes**
- **Information** about them is contained in our **opinions** on products
- **Matchmaking**: opinions of customers with tastes **similar to mine** could be used to forecast my opinions on untested products
- Internet allows to do it on **large scale** (see amazon.com and many others)

Opinion networks

WWW

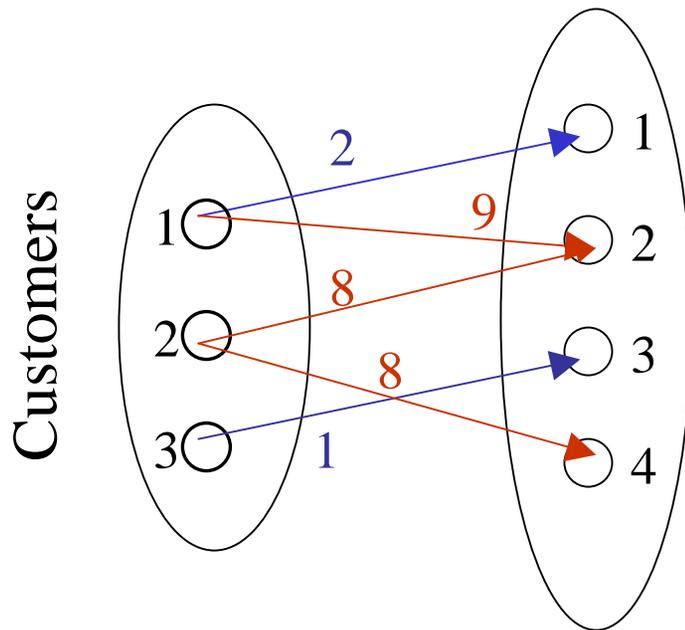


Opinions of movie-goers
on movies



Storing opinions

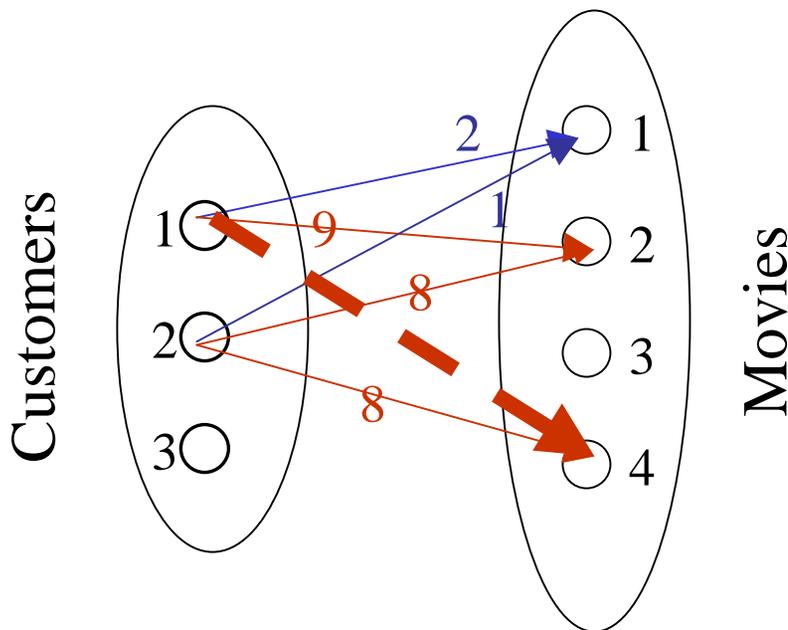
Network of opinions



Matrix of opinions Ω_{IJ}

X	X	X	2	9	?	?
X	X	X	?	8	?	8
X	X	X	?	?	1	?
2	?	?	X	X	X	X
9	8	?	X	X	X	X
?	?	1	X	X	X	X
?	8	?	X	X	X	X

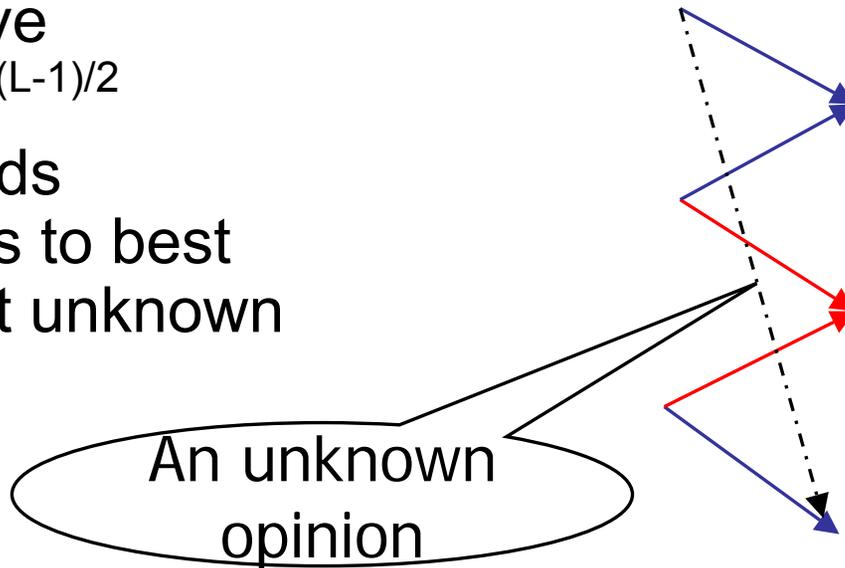
Using correlations to reconstruct customer's tastes



- Similar opinions \Rightarrow similar tastes
- Simplest model:
 - Movie-goers \Rightarrow M-dimensional vector of tastes \mathbf{T}_i
 - Movies \Rightarrow M-dimensional vector of features \mathbf{F}_j
 - Opinions \Rightarrow scalar product:
$$\Omega_{ij} = \mathbf{T}_i \cdot \mathbf{F}_j$$

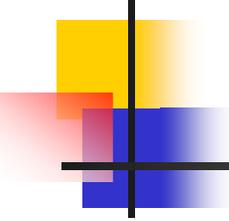
Loop correlation

- Predictive power $1/M^{(L-1)/2}$
- One needs many loops to best reconstruct unknown opinions



$L=5$ known opinions:

Predictive power of an unknown opinion is $1/M^2$

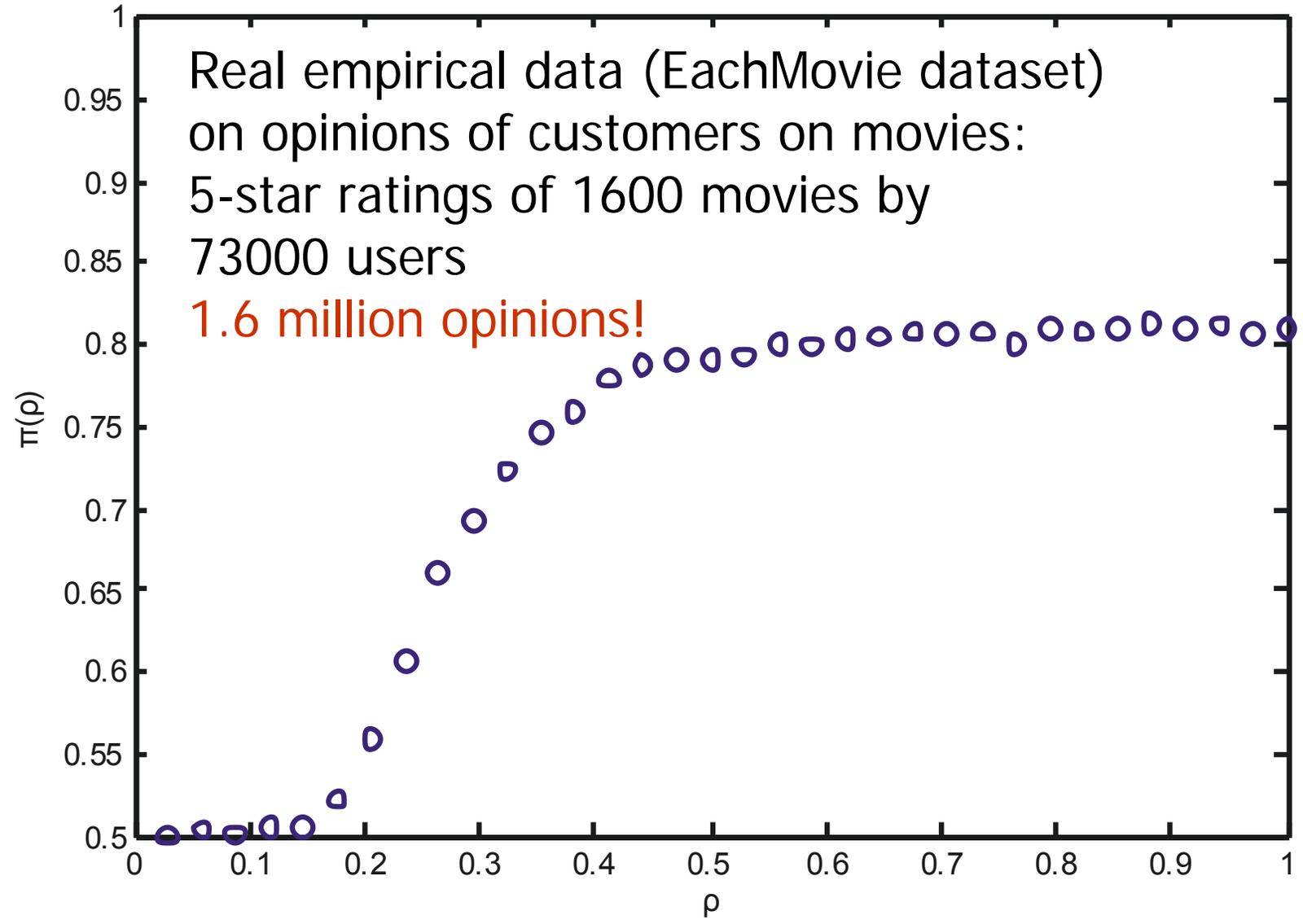


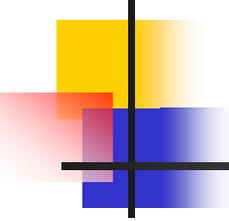
Main parameter: density of edges

- The **larger** is the **density of edges p** the **easier** is the prediction
- At $p_1 \approx 1/N$ ($N = N_{\text{customers}} + N_{\text{movies}}$) macroscopic **prediction becomes possible**.
Nodes are connected but vectors \mathbf{T}_i and \mathbf{F}_j are not fixed: **ordinary percolation** threshold
- At $p_2 \approx 2M/N > p_1$ all tastes and features (\mathbf{T}_i and \mathbf{F}_j) can be uniquely reconstructed: **rigidity percolation** threshold

Real empirical data (EachMovie dataset)
on opinions of customers on movies:
5-star ratings of 1600 movies by
73000 users

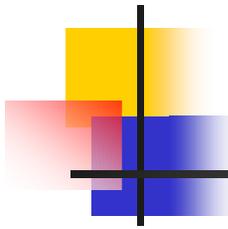
1.6 million opinions!





Spectral properties of Ω

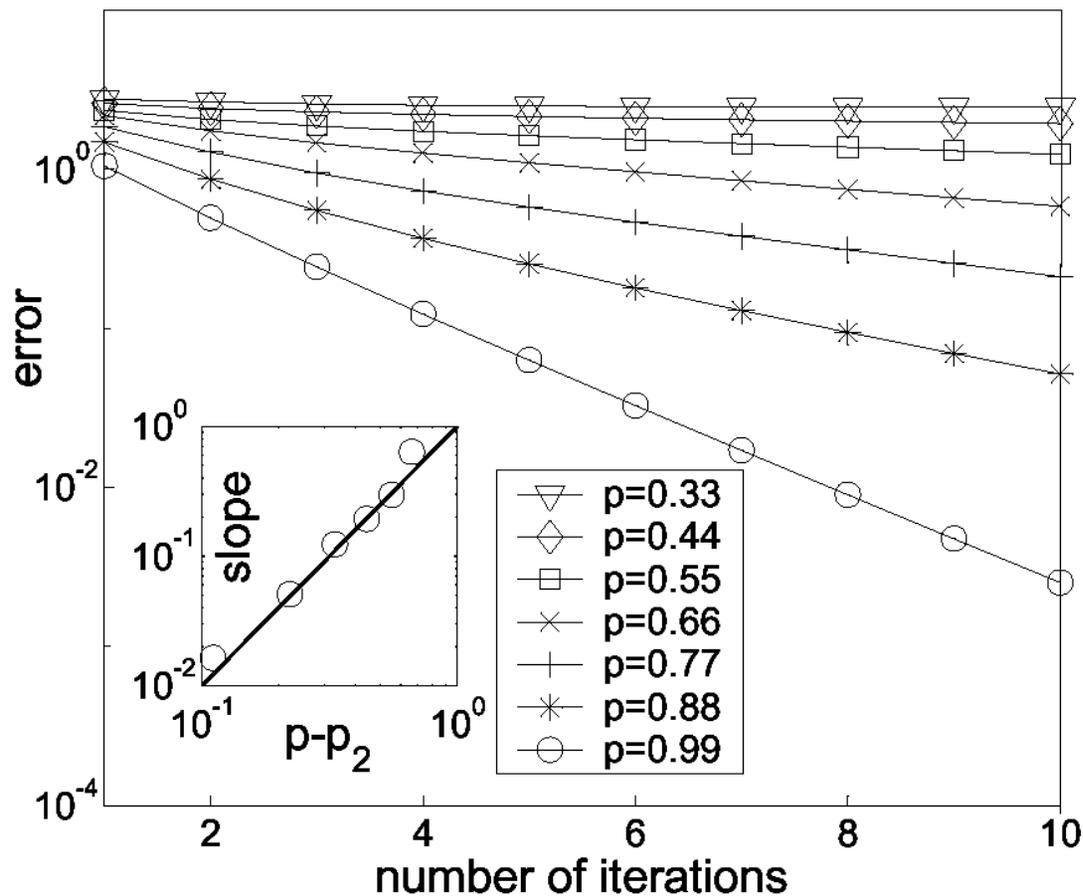
- For $M < N$ the matrix Ω_{ij} has $N-M$ zero eigenvalues and M positive ones: $\Omega = R \bullet R^+$.
- Using SVD one can “diagonalize” $R = U \bullet D \bullet V^+$ such that matrices V and U are orthogonal $V^+ \bullet V = 1$, $U \bullet U^+ = 1$, and D is diagonal.
Then $\Omega = U \bullet D^2 \bullet U^+$
- The amount of information contained in Ω : $NM - M(M-1)/2 \ll N(N-1)/2$ - the # of off-diagonal elements



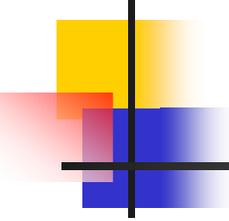
Recursive algorithm for the prediction of unknown opinions

1. Start with Ω_0 where all unknown elements are filled with $\langle \Omega \rangle$ (zero in our case)
2. Diagonalize and **keep only M largest** eigenvalues and eigenvectors
3. In the resulting truncated matrix Ω'_0 replace all **known** elements with their exact values and go to step 1

Convergence of the algorithm

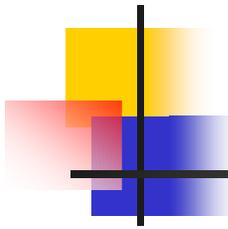


- Above p_2 the algorithm exponentially converges to the exact values of unknown elements
- The rate of convergence scales as $(p-p_2)^2$



Reality check: sources of errors

- Customers are not rational!
 $\Omega_{IJ} = \mathbf{r}_I \cdot \mathbf{b}_J + \Omega_{IJ}^{\text{(idiosyncrasy)}}$
- Opinions are delivered to the matchmaker through a narrow channel:
 - Binary channel $\mathbf{S}_{IJ} = \text{sign}(\Omega_{IJ})$: 1 or 0 (liked or not)
 - Experience rated on a scale 1 to 5 or 1 to 10 at best
- If number of edges K , and size N are large, while M is small these errors could be reduced

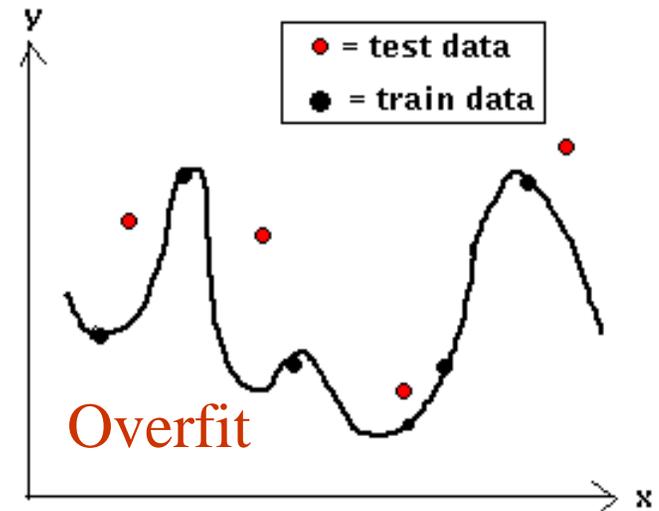
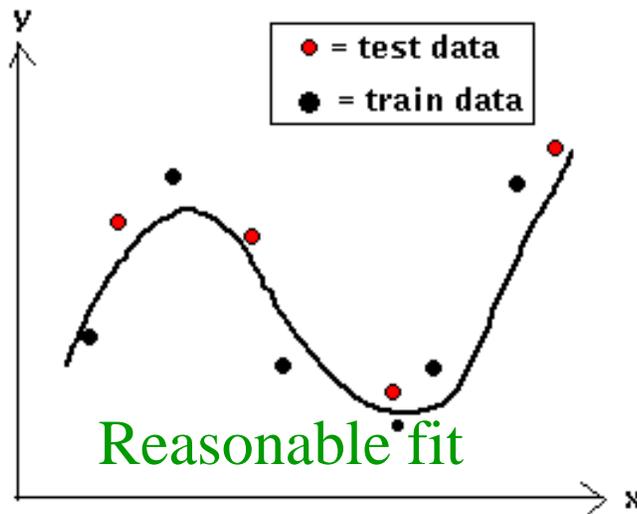


How to determine M ?

- In **real** systems M is **not fixed**: there are always finer and **finer details** of tastes
- Given the number of known opinions K one should choose $M_{\text{eff}} \leq K / (N_{\text{readers}} + N_{\text{books}})$ so that systems are below the second transition $p_2 \Rightarrow$ tastes should be determined **hierarchically**

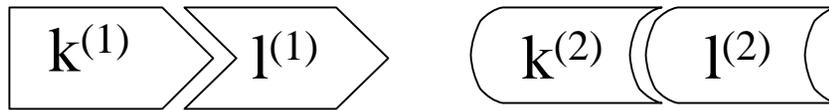
Avoid overfitting

- Divide known votes into **training** and **test sets**
- Select M_{eff} so that to avoid **overfitting !!!**

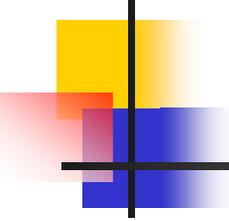


Knowledge networks in biology

- Interacting biomolecules: key and lock principle

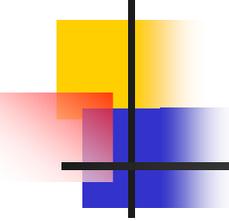


- Matrix of interactions (binding energies) $\Omega_{IJ} = \mathbf{k}_I \bullet \mathbf{l}_J + \mathbf{l}_I \bullet \mathbf{k}_J$
- Matchmaker (bioinformatics researcher) tries to guess yet unknown interactions based on the pattern of known ones
- Many experiments measure $S_{IJ} = \theta(\Omega_{IJ} - \Omega_{th})$



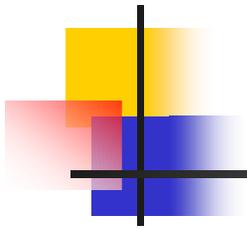
Collaborators:

- Yi-Cheng Zhang – U. of Fribourg
- Marcel Blattner – U. of Fribourg

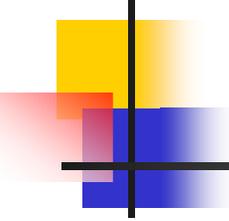


Postdoc position

- Looking for a **postdoc** to work in my group at **Brookhaven National Laboratory in New York** starting **Fall 2005**
- Topic - large-scale properties of (mostly) **bionetworks** (partially supported by a NIH/NSF grant with Ariadne Genomics)
- E-mail CV and 3 letters of recommendation to: **maslov@bnl.gov**
- See **www.cmth.bnl.gov/~maslov**



THE END



Information networks

- Why the research into properties of **complex networks** is so **active lately**?
- **Biology**: lots of **large-scale experimental data** is generated in the last 10 years: most of it is on the level of networks
- The explosive growth of **information networks** (WWW and the Internet) is what fuels it all (directly or indirectly)!

$$E_{cc} + E_{wc} = N_c \langle K_{in} \rangle_c$$

$$E_{cc} + E_{cw} = N_c \langle K_{out} \rangle_c;$$

$$\frac{G_c}{G_w} = \left(\frac{\langle K_{in} \rangle_c N_c - E_{cc}}{\langle K_{out} \rangle_c N_c - E_{cc}} \right) \cdot \frac{\langle K_{out} \rangle_c}{\langle K_{out} \rangle_w}$$

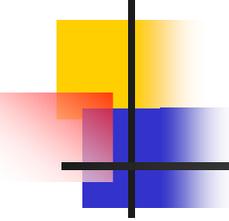
$$J_{wc} = (1 - \alpha)G_w E_{wc} / \langle K_{out} \rangle_w + \alpha G_w N_c$$

$$J_{cw} = (1 - \alpha)G_c E_{cw} / \langle K_{out} \rangle_c + \alpha G_c N_c$$

$$E_{cw}^* = E_{cw}(1 - \alpha) + N_c \langle K_{out} \rangle_c \alpha$$

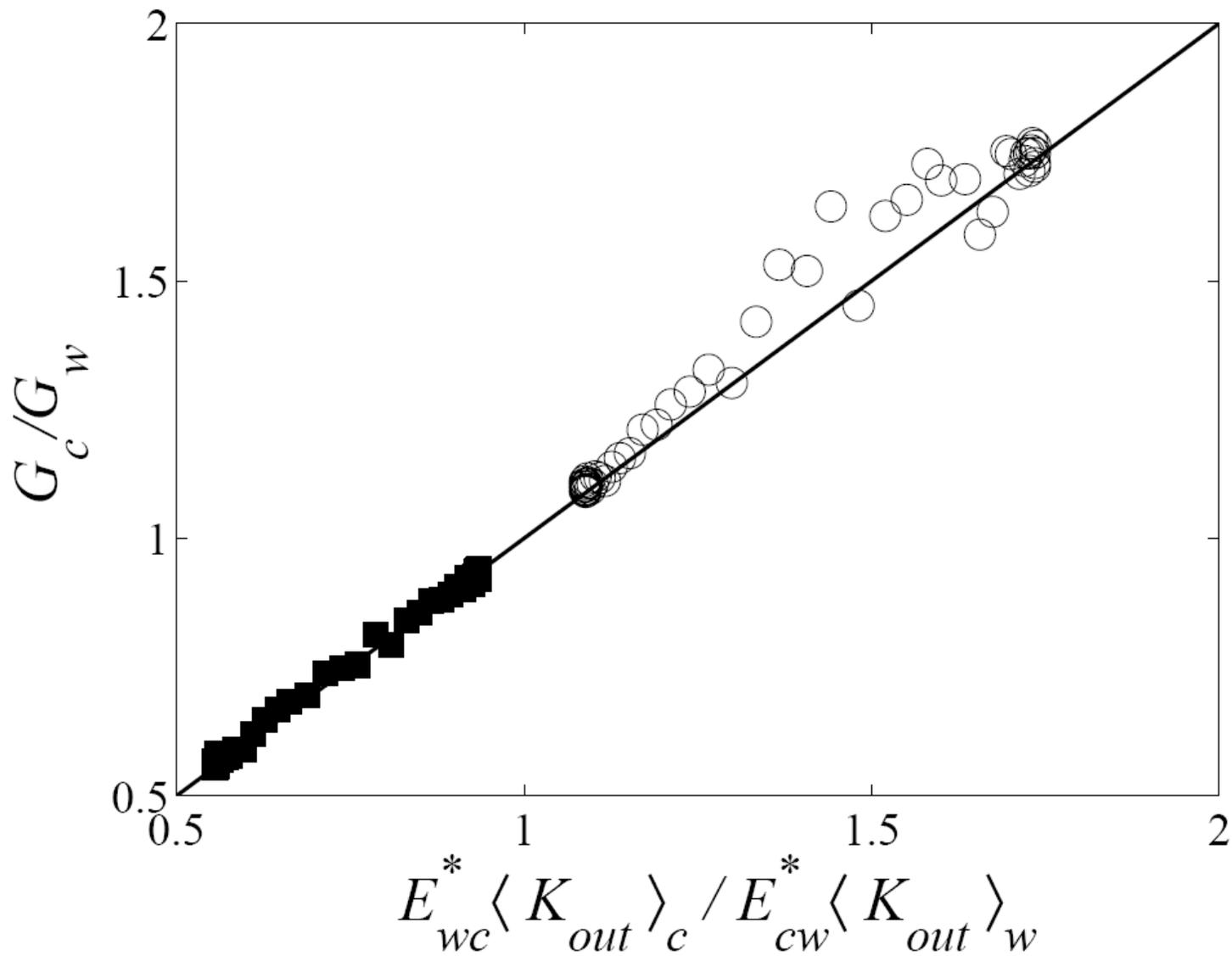
$$E_{wc}^* = E_{wc}(1 - \alpha) + N_c \langle K_{out} \rangle_w \alpha$$

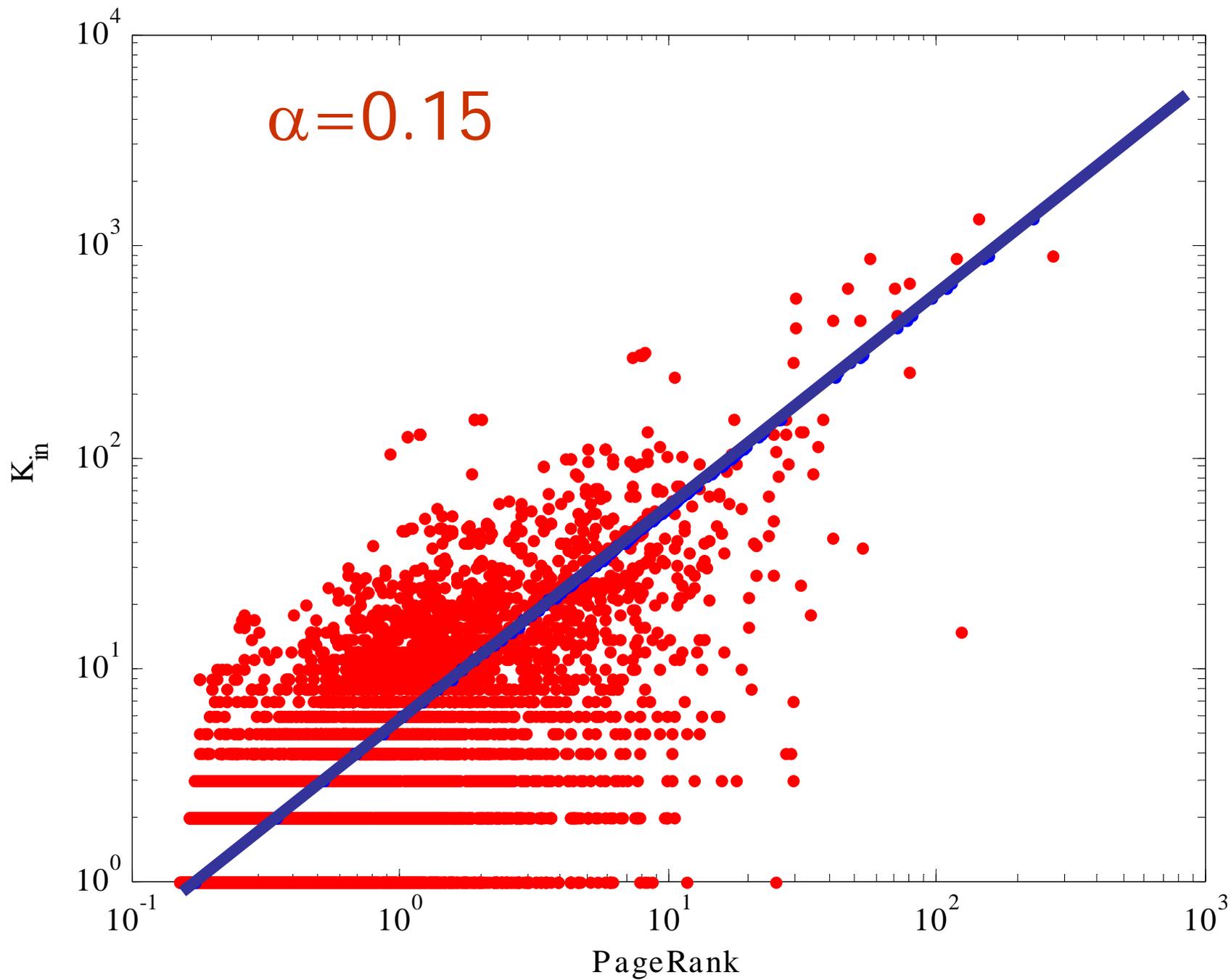
$$\frac{G_c}{G_w} = \frac{E_{wc}^*}{E_{cw}^*} \cdot \frac{\langle K_{out} \rangle_c}{\langle K_{out} \rangle_w}$$

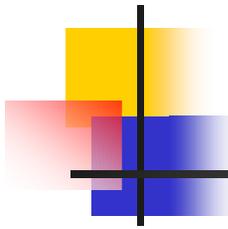


Analysis

- Derived for $\alpha=0$
- Uses a strong **mean field approximation** that nodes that send current to and from the community have average G_i for the outside world ($G_w=1$) and community (G_c)
- In a true community both E_{cW} and E_{wC} are **smaller** than in **randomized network** but the effect depends on the **competition** between them

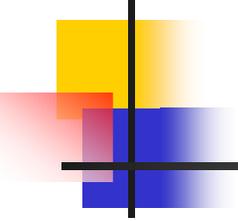






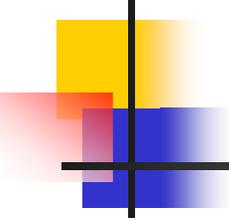
Networks with artificial communities

- To test we generate a scale-free network with an **artificial community** of N_c pre-selected nodes
- Use **Metropolis Algorithm** with $H = -(\# \text{ of intra-community nodes})$ and some **inverse temperature** β
- Detailed balance:
$$E_{cw}E_{wc} = E_{cc}E_{ww}e^{-\beta}$$



Modules in networks and how to detect them using the Random walks/diffusion

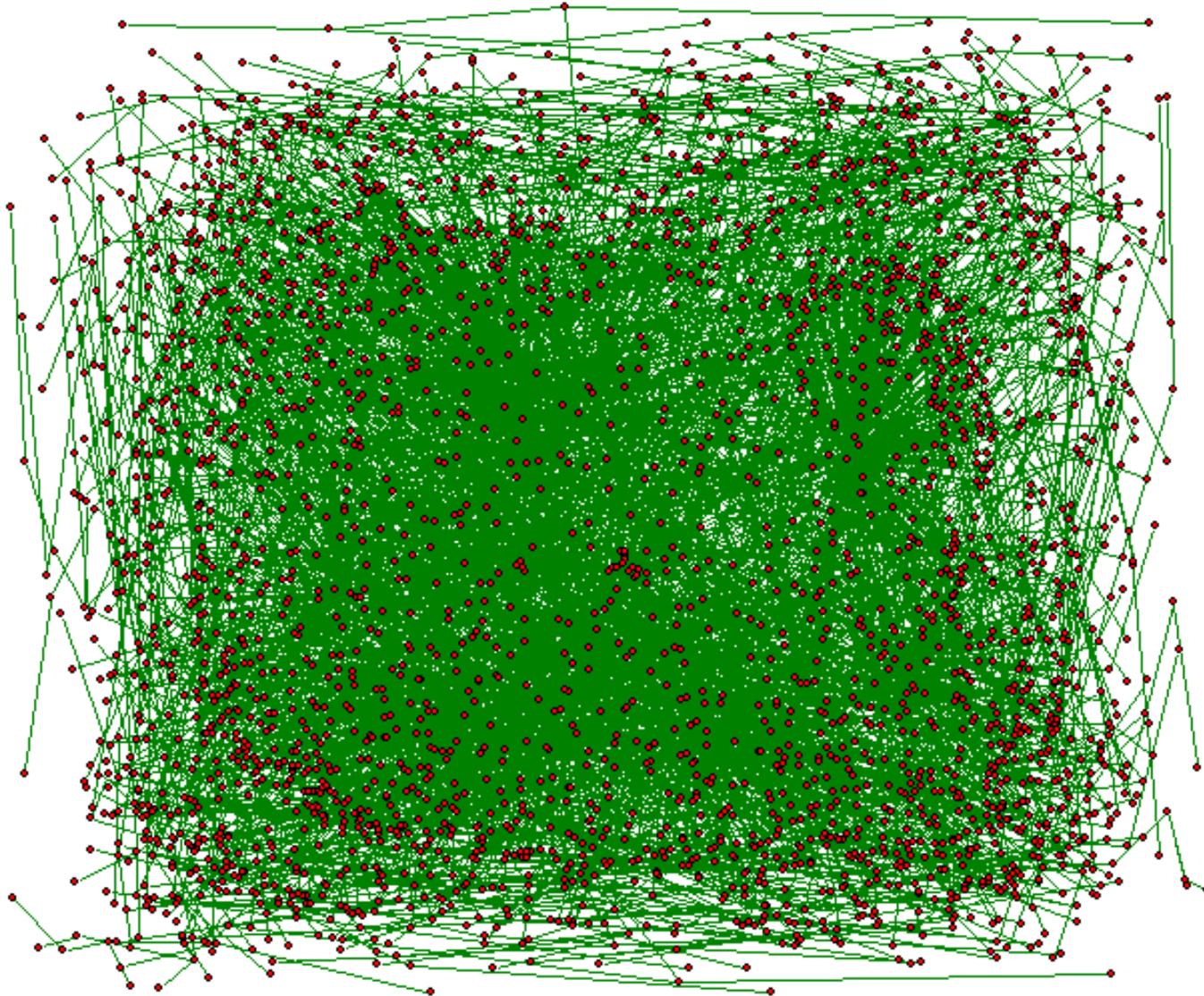
K. Eriksen, I. Simonsen, SM, K. Sneppen, PRL (2003)

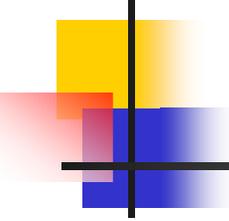


What is a module?

- Nodes in a given module (or community group or functional unit) tend to connect with other nodes in the same module
 - Biology: proteins of the same function or sub-cellular localization
 - WWW – websites on a common topic
 - Internet – geography or organization (e.g. military)

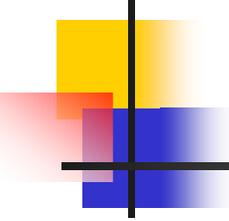
Do you see any modules here?





Random walkers on a network

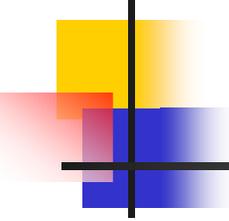
- Study the behavior of many **VIRTUAL** random walkers on a network
- At each time step each random walker steps on a randomly selected neighbor
- They equilibrate to a steady state
 $n_i \sim k_i$ (solid state physics: $n_i = \text{const}$)
- Slow modes allow to detect **modules** and **extreme edges**



Matrix formalism

$$n_i(t+1) = \sum_j \hat{T}_{ij} n_j(t)$$

$$\hat{T}_{ij} = \begin{cases} 1/K_j & \text{if } j \leftrightarrow i \\ 0 & \text{otherwise} \end{cases}$$

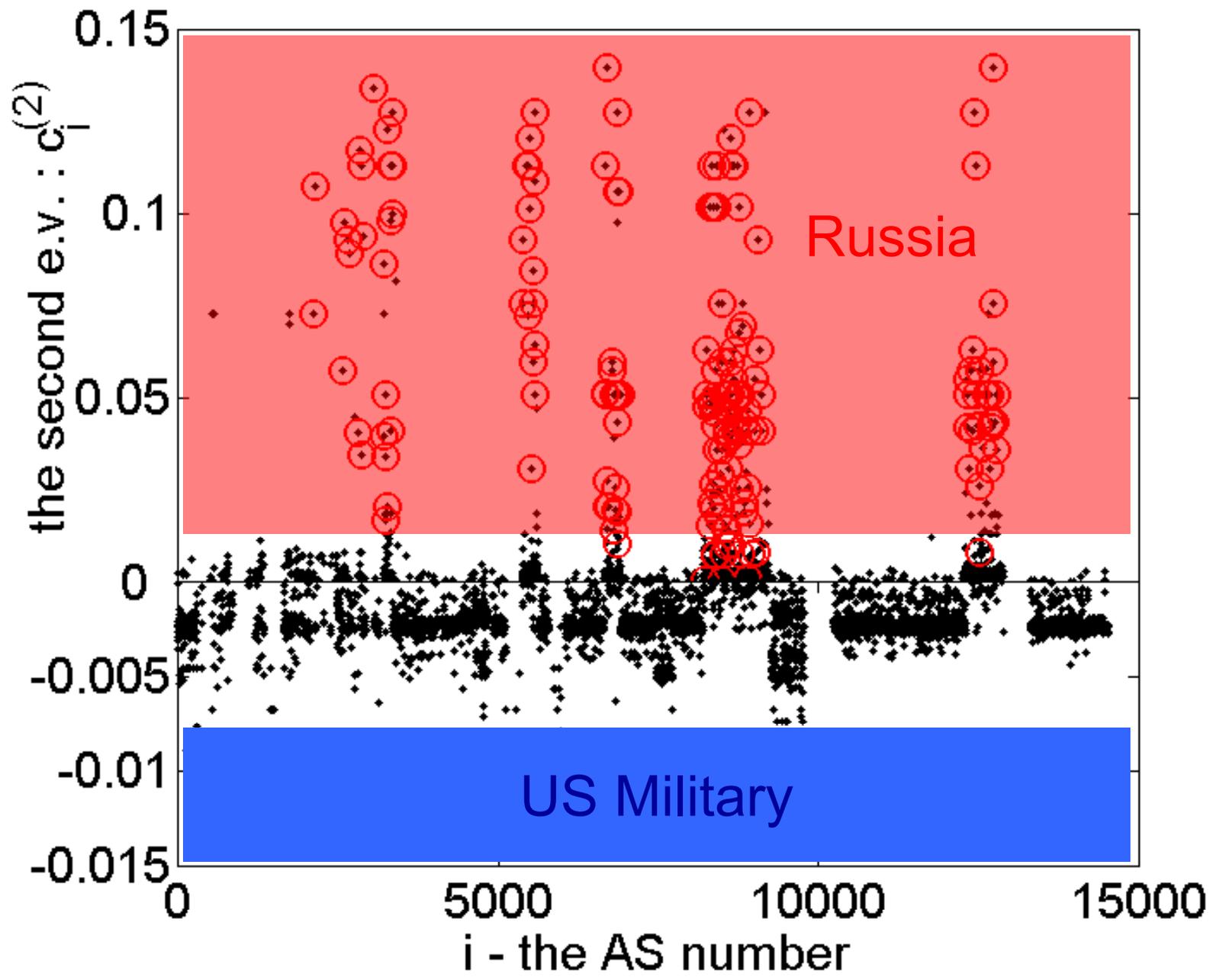


Eigenvectors of the transfer matrix T_{ij}

$$\lambda^{(\alpha)} v_i^{(\alpha)} = \sum_j \hat{T}_{ij} v_j^{(\alpha)}$$

$$n_i(t) = \left(\lambda^{(\alpha)} \right)^t v_i^{(\alpha)}$$

$$-1 \leq \lambda^{(\alpha)} \leq 1$$



2 0.9626 RU RU RU RU CA RU RU
?? ?? US US US US ??
(US Department of Defence)

3 0.9561 ?? FR FR FR ?? FR ??
RU RU RU ?? ?? RU ??

4 0.9523 US ?? US ?? ?? ?? ?? (US Navy)
NZ NZ NZ NZ NZ NZ NZ

5. 0.9474 KR KR KR KR KR ?? KR
UA UA UA UA UA UA UA

