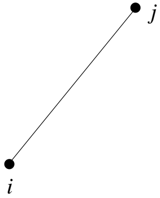
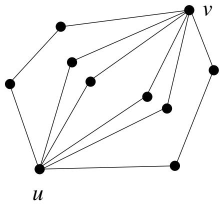
Algebraic Distance on Graphs

based on paper in SISC 2011

Introduction to Network Science, Spring 14

How to model a connectivity between graph vertices?

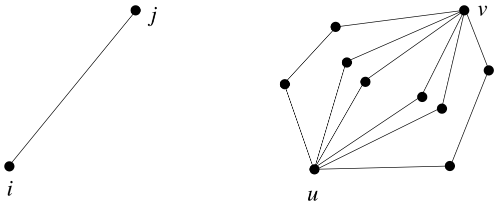




Possible problems in (hyper)graph models:

- unweighted edges;
- edges with almost identical weights: $0.999 \approx^{?} 1.001$;
- incomplete set of edges.

How to model a connectivity between graph vertices?



Possible problems in (hyper)graph models open questions like

- how to break ties?
- should we choose a heaviest edge?
- should we match a disconnected pair?

How can one measure a connectivity?

Some existing approaches

- Shortest path
- All/some (weighted) indirect paths
- Spectral approaches
- Flow network capacity based approaches
- Random-walk approaches: commute time, first-passage time, etc. (Fouss, Pirotte, Renders, Saerens, ...)
- Speed of convergence of the compatible relaxation from AMG (Brandt, Ron, Livne, ...)
- Probabilistic interpretation of a diffusion (Nadler, Lafon, Coifman, Kevrekidis, ...)
- Minimization of effective resistance of a graph (Ghosh, Boyd, Saberi, ...)

Stationary iterative relaxation

Simulation process that shows which pair of vertices tends to be 'more connected' than other.

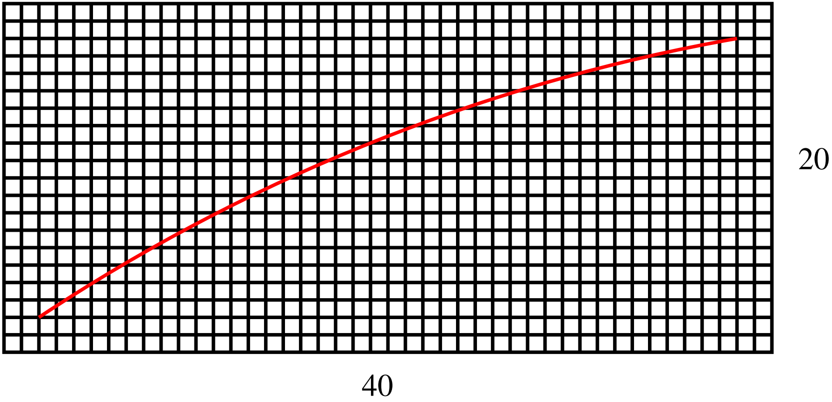
1. $\forall\ i \in V$ define $x_i = rand()$
2. Do k times step 3
3. $\quad\quad \forall\ i \in V\ x_i^k = (1-\omega)x_i^{k-1} + \omega \sum_j w_{ij} x_j^{k-1} / \sum_{ij} w_{ij}$

Conjecture

If $|x_i - x_j| > |x_u - x_v|$ then the local connectivity between u and v is **stronger** *than that between i and j.*

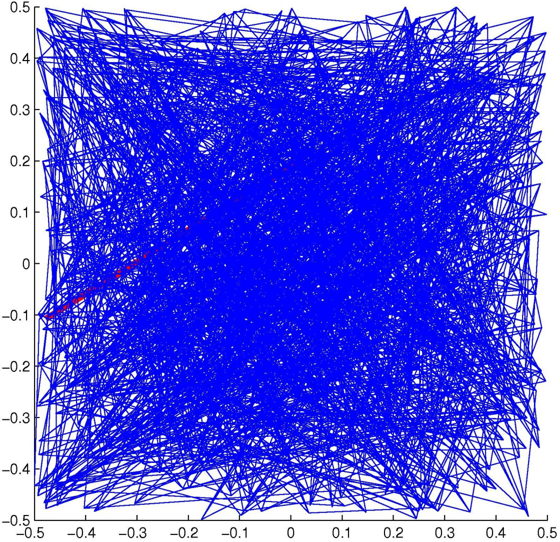
We will call $s_{ij}^{(k)} = |x_i - x_j|$ *the algebraic distance* between i and j after k iterations.

Toy example: graph mesh 20x40+diagonal

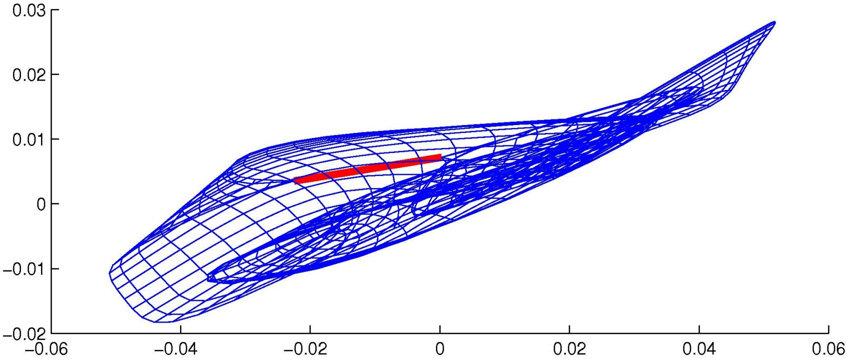


edge weights: red=2, black=1

Mesh 20x40+diagonal, random 2D initialization



Mesh 20x40+diagonal, after 15 iterations of JOR



Stationary iterative relaxation

Rewrite the iterative process as $x^{(k+1)} = Hx^{(k)}$, where H:

$$H_{GS} = (D-L)^{-1}U, \qquad H_{SOR} = (D/\omega - L)^{-1}\left((1/\omega - 1)D + U\right),$$
$$H_{JAC} = D^{-1}(L+U), \qquad H_{JOR} = (D/\omega)^{-1}\left((1/\omega - 1)D + L + U\right).$$

Definition

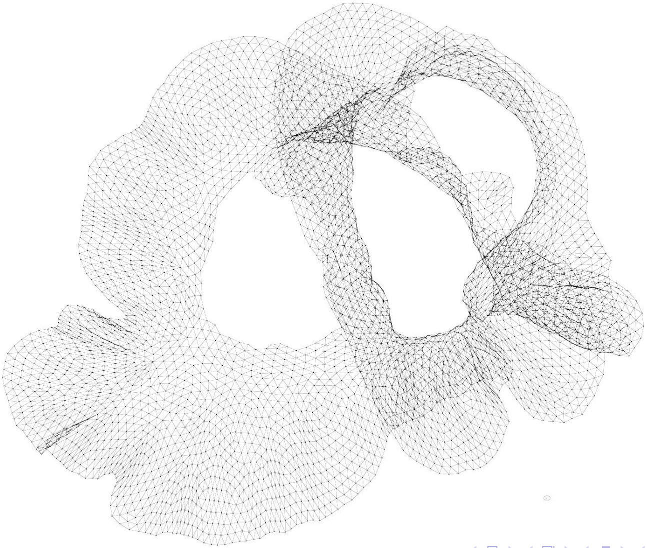
Extended p-normed algebraic distance *between i and j after k iterations $x^{(k+1)} = Hx^{(k)}$ on R random initializations*

$$\rho_{ij}^{(k)} := \left(\sum_{r=1}^{R} |x_i^{(k,r)} - x_j^{(k,r)}|^p\right)^{1/p}$$

Algebraic distance is inspired by **Bootstrap Algebraic Multigrid**

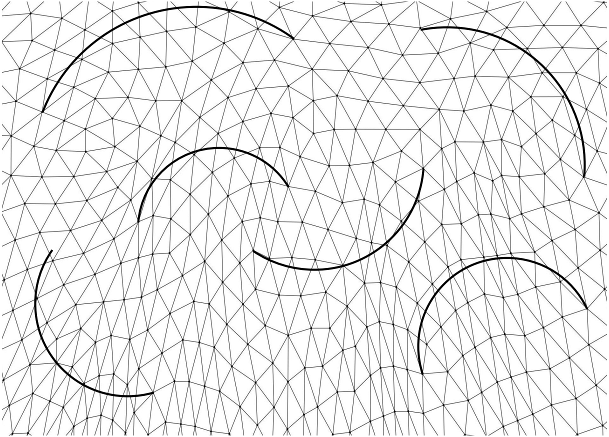
Example: airfoil - finite element graph, $|E| \approx 13000$

For every edge ij there exist a path $i - k - j$

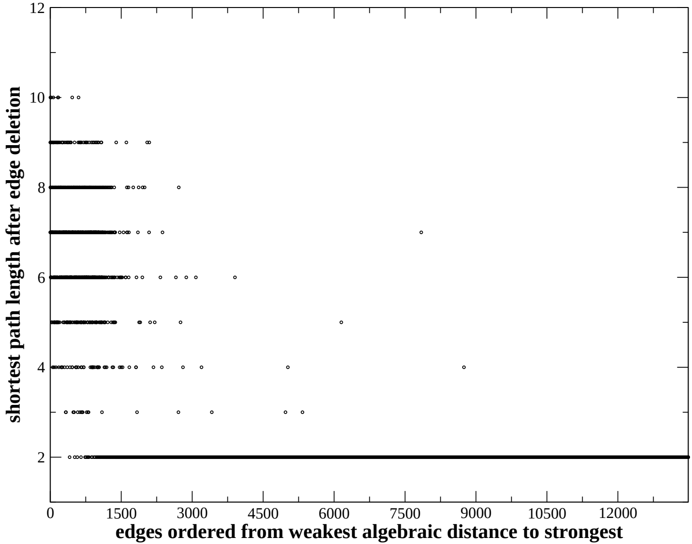


Input: airfoil + 1500 random edges

- Add 1500 edges such that for every new edge ij, the second shortest path between i and j, p_2, has length $2 < |p_2| < 11$
- Calculate $\rho_{ij}^{(k)}$, $R = 5$, $k = 15$, $p = 2$



Shortest paths after edge deletion



Analysis and Model

Chen, S, ”Algebraic Distance on Graphs”, 2011

- convergence properties of H
- how to choose $x^{(0)}$
- properties of early iterations
- special focus on JOR
- define ”**Mutually Reinforcing Model**” of graph vertices and their neighborhoods
- describe this model using JOR

Theorem

Given a connected graph, let $(\mu_i, \hat{v}_i)$ be the eigen-pairs of (L, D), labeled in nondecreasing order of the eigenvalues, and assume that $\mu_2 \neq \mu_3 \neq \mu_{n-1} \neq \mu_n$. Unless $\omega = 2/(\mu_2 + \mu_n)$, $\hat{s}_{ij}^{(k)}$ will always converge to a limit $|(e_i - e_j)^T \xi|$ in the order $O(\theta^k)$, for some ξ and $0 < \theta < 1$.

(i) *If* $0 < \omega < \frac{2}{(\mu_3+\mu_n)}$, *then* $\xi \in \mathrm{span}\{\hat{v}_2\}$ *and* $\theta = \frac{1-\omega\mu_3}{1-\omega\mu_2}$;

(ii) *If* $\frac{2}{(\mu_3+\mu_n)} \leq \omega < \frac{2}{(\mu_2+\mu_n)}$, *then* $\xi \in \mathrm{span}\{\hat{v}_2\}$ *and* $\theta = -\frac{1-\omega\mu_n}{1-\omega\mu_2}$;

(iii) *If* $\frac{2}{(\mu_2+\mu_n)} < \omega < \min\{\frac{2}{(\mu_2+\mu_{n-1})}, \frac{2}{\mu_n}\}$, *then* $\xi \in \mathrm{span}\{\hat{v}_n\}$ *and* $\theta = -\frac{1-\omega\mu_2}{1-\omega\mu_n}$;

(iv) *If* $\frac{2}{(\mu_2+\mu_{n-1}}) \leq \omega < \frac{2}{\mu_n}$, *then* $\xi \in \mathrm{span}\{\hat{v}_n\}$ *and* $\theta = \frac{1-\omega\mu_{n-1}}{1-\omega\mu_n}$.

Theorem

Given a graph, let $(\mu_i, \hat{v}_i)$ *be the eigen-pairs of* (L, D)*, labeled in nondecreasing order of the eigenvalues. Denote* $\hat{V} = [\hat{v}_1, \ldots, \hat{v}_n]$*. Let* $x^{(0)}$ *be the initial vector of the JOR process, and let* $a = \hat{V}^{-1}x^{(0)}$ *with* $a_1 \neq 0$*. If the following two conditions are satisfied:*

$$1 - \omega\mu_n \geq 0 \quad \textit{and} \quad f_k := \frac{\alpha {r_k}^{2k}(1 - r_k)^2}{1 + \alpha {r_k}^{2k}(1 + r_k)^2} \leq \frac{1}{\kappa},$$

where $\alpha = \left(\sum_{i \neq 1} a_i^2\right) / \left(4a_1^2\right)$*,* r_k *is the unique root at* $[0, 1]$ *of*

$$2\alpha r^{2k+2} + 2\alpha r^{2k+1} + (k+1)r - k = 0\ ,$$

$$\textbf{then } 1 - \left\langle \frac{x^{(k)}}{\left\|x^{(k)}\right\|}, \frac{x^{(k+1)}}{\left\|x^{(k+1)}\right\|} \right\rangle^2 \leq \frac{4cond(D)f_k}{(1 + cond(D)f_k)^2}.$$

Analysis

Sketch of Theorems

- *We cannot use H_{JAC} for bipartite components. Other iteration matrices are convergent with particular ω.*
- *JOR: Given a connected graph, let $(\mu_i, \hat{v}_i)$ be the eigen-pairs of the matrix pencil (L, D). The normalized algebraic distance will converge either to $\text{span}\{\hat{v}_2\}$ or $\text{span}\{\hat{v}_n\}$.*
- *JOR: Usually, the convergence will be slow. For example, in many cases it will be $O\left(\left|\frac{\sigma_3}{\sigma_2}\right|^k\right)$, where σ_i is an eigenvalue of H_{JOR}.*
- *JOR:* **However**, *after small number of iterations (k), $x^{(k)}$ will be very close to $x^{(k+1)}$.*

Interpretation

A mutually reinforcing environment:

- Everybody is influenced by its neighbors:

$$x_i = \mu x_i + \sum_j \left(\frac{w_{ij}}{\sum_k w_{ik}} \right) x_j.$$

- $0 \leq \mu \leq 1$:
  - When μ is close to zero, the environment plays a major role.
  - When μ is close to one, the entities are stubborn.
- μ is a property of the entire environment.
- Two entities x_i and x_j are close/similar if

$$\left| x_i - x_j \right| \text{ is small.}$$

Interpretation

Matrix form of the model

$$x = \mu x + D^{-1}Wx,$$

or

$$Lx = \mu Dx \qquad (0 \leq \mu \leq 1).$$

Possibilities:

- $\mu = 0$, $x = \mathbf{1}$. A strong reinforcing environment, but no discriminating power.
- $\mu = \mu_2$, $x = \hat{v}_2$. Good. (μ_2 usually close to zero.)

The limit of the scaled algebraic distance $\hat{s}_{ij}^{(k)}$ exactly meets the second possibility.

$$\hat{s}_{ij}^{(k)} \to \left| (e_i - e_j)^T \hat{v}_2 \right| .$$

Interpretation

At an early stage of the iterations (assuming that the iterates are normalized):

$$x^{(k)} \approx x^{(k+1)} = \frac{H_{JOR}\ x^{(k)}}{\text{normalization}} \approx \frac{(I - \omega D^{-1} L) x^{(k)}}{1 - \omega \mu_2}.$$

Simplified to

$$x^{(k)} \approx \mu_2 x^{(k)} + D^{-1} W x^{(k)}.$$

This means that $x^{(k)}$ approximately satisfies the model.

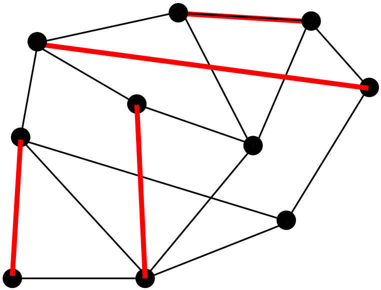
Conclusion: $\hat{v}_2$ is good. $x^{(k)}$ is also good. They both fit the mutually reinforcing model.

Applications

Maximum weighted matching problem

- Graph $G = (V, E)$
- Weighting function on edges $w : E \rightarrow \mathbb{R}^+$
- Matching: $M \subseteq E$ with no incident edges.
- $w(M) = \sum_{ij \in M} w_{ij}$
- Maximum weighted matching: $M', \forall M \; w(M') \geq w(M)$

Methods: textbook greedy algorithm; path growing algorithm [DrakeHougardy03]

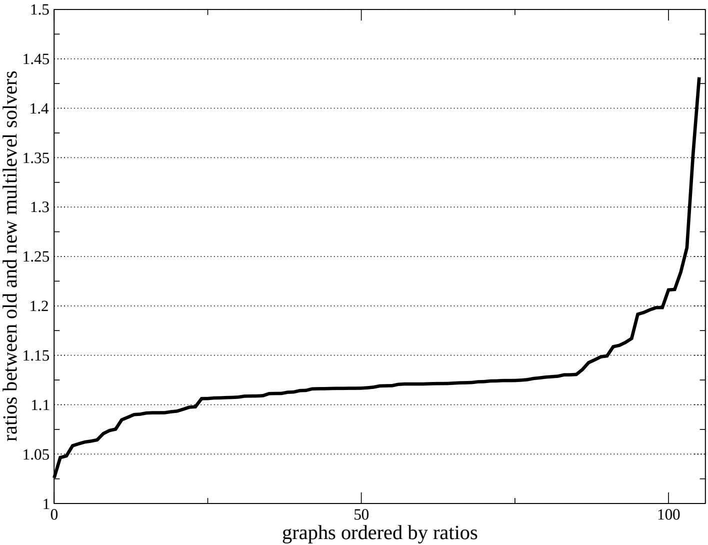


Heuristic for weighted matching problem: GREEDY+

Preprocessing:
Input: Graph G
Output: edge weights s'_{ij}
For all edges $ij \in E$ calculate $\rho_{ij}^{(k)}$ for some $k,\ R$ and p
For all nodes $i \in V$ define $a_i = \sum_{ij \in E} 1/\rho_{ij}^{(k)}$
For all edges $ij \in E$ define $s'_{ij} = a_i/\delta_i + a_j/\delta_j$

GREEDY algorithm:
Input: Graph G with new edge weights s'_{ij}
Output: weight of matching M with original edge weights
$M \leftarrow \emptyset$
while $E \neq \emptyset$ **do**
  add the lightest edge $e \in E$ to M
  remove e and all its incident edges from E
end

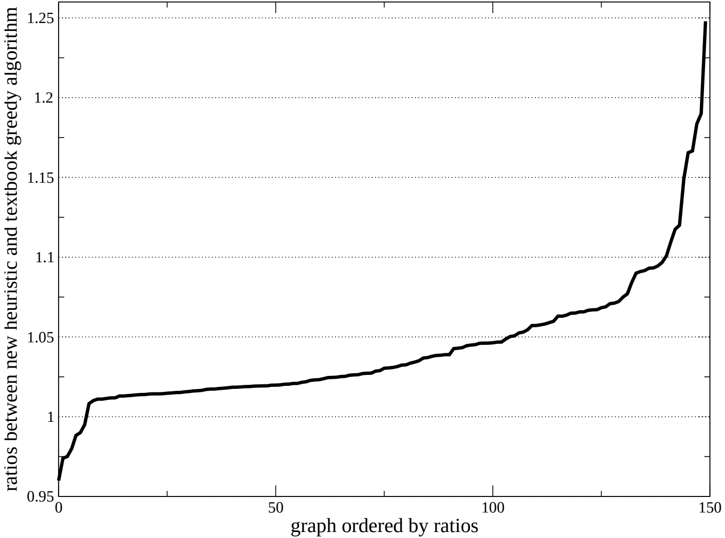
Experimental results: weighted matching problem



Preprocessing:

Input: Graph G
Output: node weights a_i
For all edges $ij \in E$ calculate $\rho_{ij}^{(k)}$ for some $k,\ R$ and p
For all nodes $i \in V$ define $a_i = \sum_{ij \in E} 1/\rho_{ij}^{(k)}$
For all edges $ij \in E$ define $\rho'_{ij} = \rho'_{ij}/(a_i + a_j)$
For all nodes $i \in V$ redefine $a_i = \sum_{ij \in E} \rho'_{ij}$
Sort V by a_i and output its increasing order

Experimental results: maximum independent set

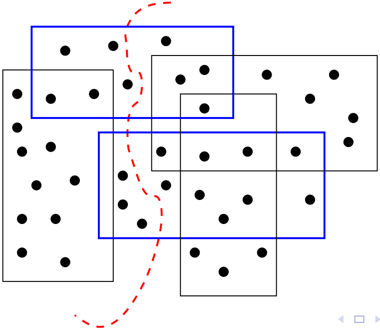


(Hyper)graph k-partitioning

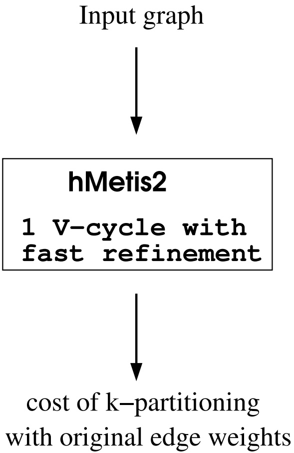
Given a hypergraph $\mathcal{H} = (\mathcal{V}, \mathcal{E})$

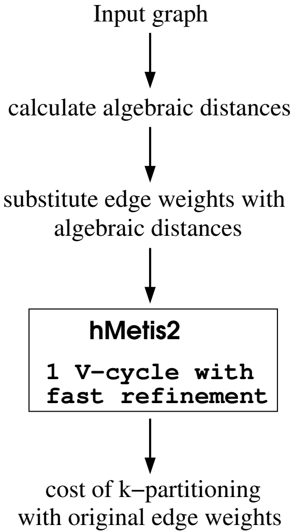
$$\text{minimize} \sum_{\substack{h \in \mathcal{E} \text{ s.t. } \exists i,j \in h \text{ and} \\ i \in \pi_p \Rightarrow j \notin \pi_p}} w_h$$

$$\text{such that } \forall p \in \{1, \dots, k\}, |\pi_p| \leq (1 + \alpha) \cdot \frac{|V|}{k}$$

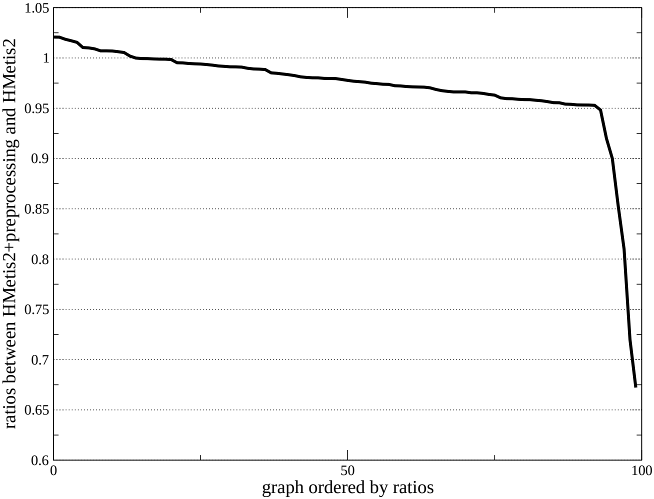


(Hyper)graph partitioning





Experimental results: 2-partitioning

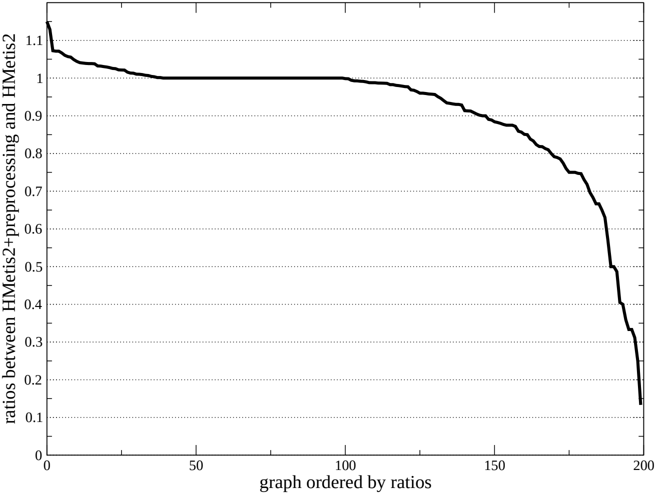


Algebraic distance for hypergraphs

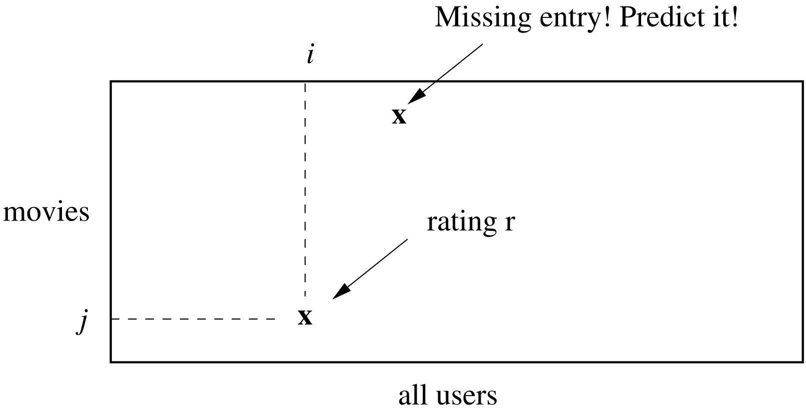
Preprocessing:
Input: Hypergraph $\mathcal{H}, k = 20, R = 10$
Output: weights $s_h^{(k)}$
$G = (V, E) \leftarrow$ bipartite graph model of $\mathcal{H}$
Create R initial vectors $x^{(0,r)}$
for $r = 1, \ldots, R$ **do**
    for *m=1,...,k* **do**
        $x_i^{(m,r)} \leftarrow \sum_j w_{ij} x_j^{(m-1,r)} / \sum_j w_{ij}, \ \forall i$
    end
end
$s_h^{(k)} \leftarrow \sum_r \max_{i,j \in h} |x_i^{(k,r)} - x_j^{(k,r)}|, \ \forall h \in \mathcal{E}$

Algebraic distance on hypergraph $:= s_h^{(k)}$

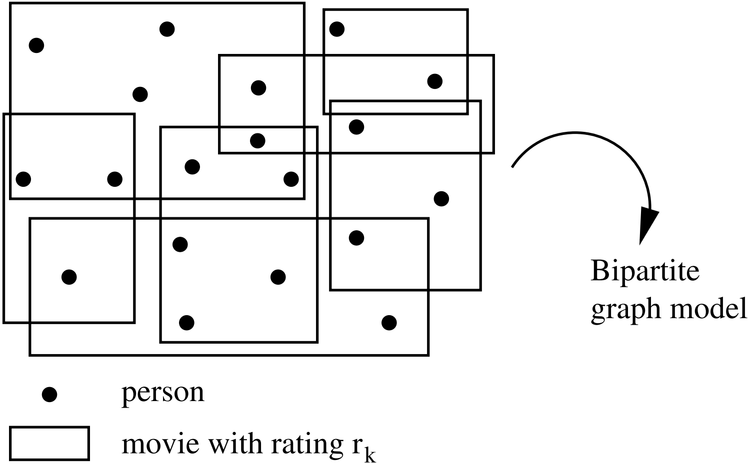
Experimental results: 2-partitioning of hypergraphs



Recommendation systems: Netflix problem



Recommendation systems: hypergraph model



Algebraic distance for recommendation systems

Preprocessing:
Input: Hypergraph $\mathcal{H}$
Output: weights $s_h^{(k)}$
$G = (V, E) \leftarrow$ bipartite graph model of $\mathcal{H}$
Calculate algebraic distances for movies
Introduce them as new weights for hyperedges (for example, scale the matrix columns)

Algebraic distance for recommendation systems

Measure of success is the root mean square error
(More or less) all SVD-based methods perform similarly on the Netflix database, RMSE$\approx$0.90-0.92

Algebraic distance for recommendation systems

Measure of success is the root mean square error
(More or less) all SVD-based methods perform similarly on the Netflix database, RMSE$\approx$0.90-0.92

- Remove 85% of the data

Algebraic distance for recommendation systems

Measure of success is the root mean square error
(More or less) all SVD-based methods perform similarly on the Netflix database, RMSE$\approx$0.90-0.92

- Remove 85% of the data
- SVD-based methods with **linear** combination of latent factors, RMSE$>$1.00

Algebraic distance for recommendation systems

Measure of success is the root mean square error
(More or less) all SVD-based methods perform similarly on the Netflix database, RMSE$\approx$0.90-0.92

- Remove 85% of the data
- SVD-based methods with **linear** combination of latent factors, RMSE$>$1.00
- SVD-based methods with **high-order polynomial** combination of latent factors, RMSE$\approx$0.92-0.94
  Roderick, S, "High-order Polynomial Interpolation for Predicting Decisions", 2009

Algebraic distance for recommendation systems

Measure of success is the root mean square error
(More or less) all SVD-based methods perform similarly on the Netflix database, RMSE$\approx$0.90-0.92

- Remove 85% of the data
- SVD-based methods with **linear** combination of latent factors, RMSE$>$1.00
- SVD-based methods with **high-order polynomial** combination of latent factors, RMSE$\approx$0.92-0.94
  Roderick, S, "High-order Polynomial Interpolation for Predicting Decisions", 2009
- **Algebraic distance + SVD-based methods** with **high-order polynomial** combination of latent factors, RMSE$\approx$0.90-0.92