Scalable Network Analysis

Inderjit S. Dhillon
University of Texas at Austin
COMPUTERSCIENCE

COMAD, Ahmedabad, India
Dec 20, 2013

Outline

- Unstructured Data - Scale & Diversity
  - Evolving Networks
- Machine Learning Problems arising in Networks
  - Recommender Systems
  - Link Prediction
  - Sign Prediction
- Formulation as Missing Value Estimation
- Scalable Algorithms
  - NOMAD: Distributed matrix completion algorithm
- Results on Applications
- Conclusions

Structured Data

- Data organized into fields: Relational databases, spreadsheets, XML
- Highly optimized for storage & retrieval (e.g. using SQL)

Structured Data

- Focus is on data format, efficient storage & search
- Less or no uncertainty in *semantics*: e.g. businesses know the fields of the data

Unstructured Data

Modern data is unstructured and diverse

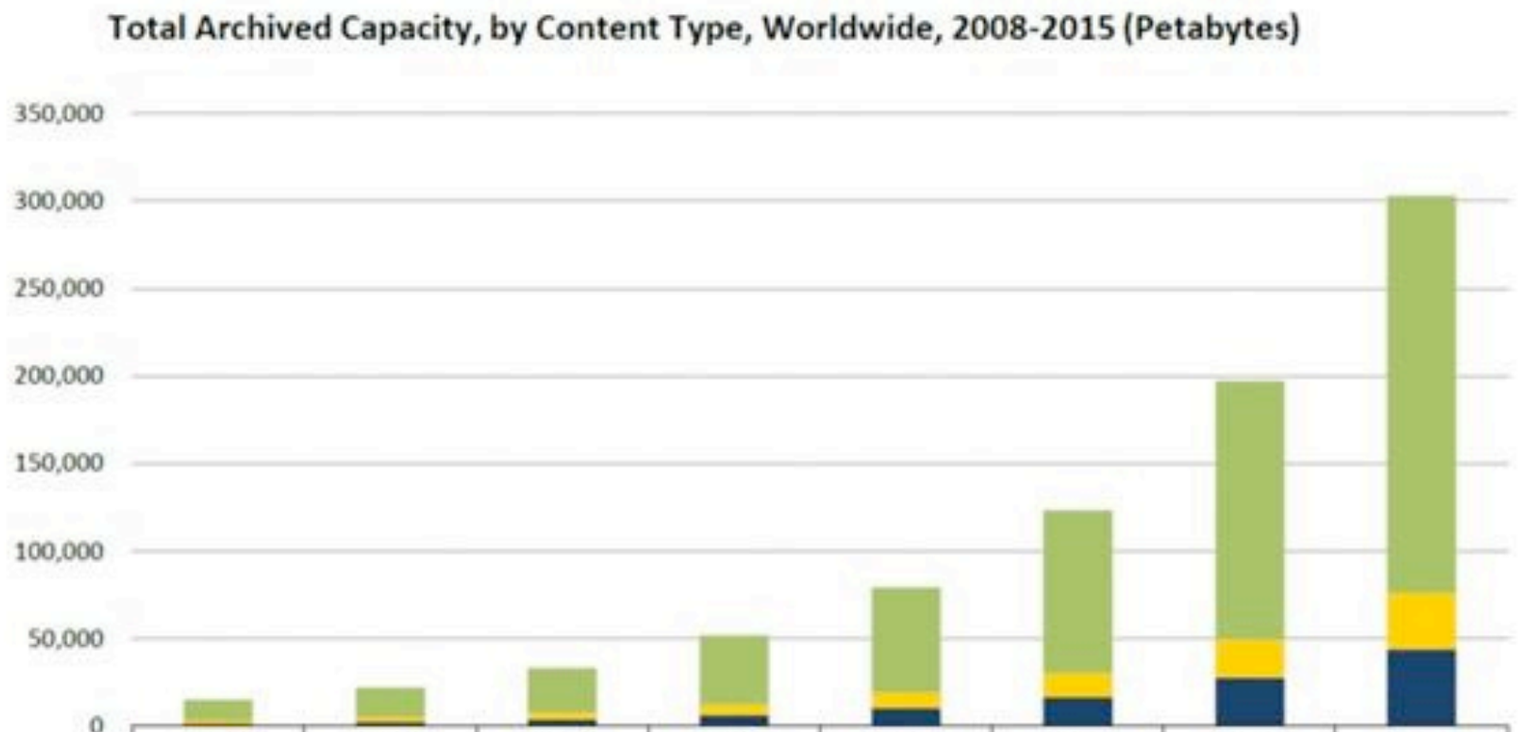
Networks, Graphs

Text

Images, Videos

Unstructured Data

Much greater growth rate



	2008	2009	2010	2011	2012	2013	2014	2015
Unstructured	11,430	16,737	25,127	39,237	59,600	92,536	147,885	226,716
Database	1,952	2,782	4,065	6,179	9,140	13,824	21,532	32,188
E-mail	1,652	2,552	4,025	6,575	10,411	16,796	27,817	44,091

Source: Enterprise Strategy Group, 2010.

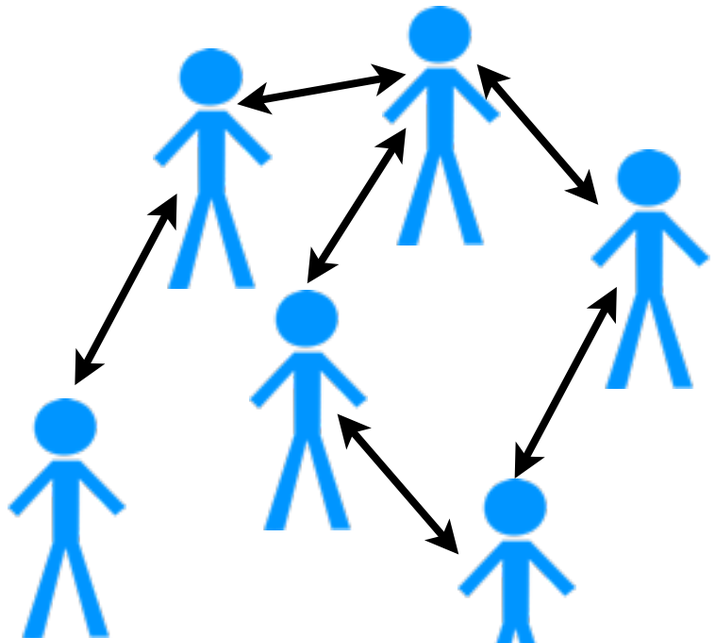
Unstructured Data

- Dynamic aspects of unstructured data:
  - Constantly evolving
  - Uncertainties abound: *What* should I ask of the data?
  - Seek *insights*
- Heterogeneity renders traditional database models inadequate

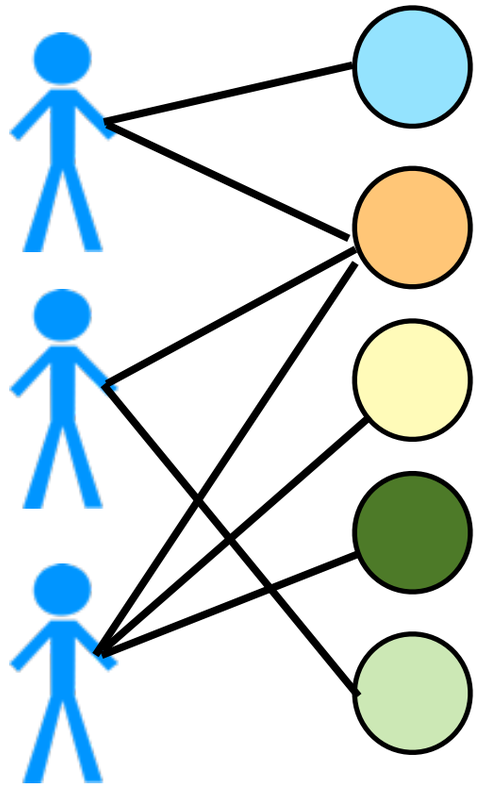
Unstructured Data

- Buzzwords - “Big Data” & “Data Science”
- *Machine Learning*: Predictive models for data
- *Engineering perspective:* Scale matters

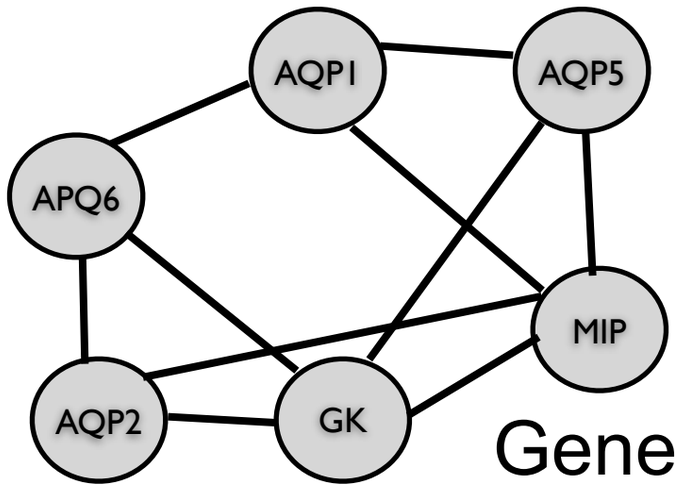
Network Graphs



Social networks
(Friendship)



Bipartite networks
(Membership, Ratings, etc.)



Gene networks
(Functional interaction)

Graph Evolution

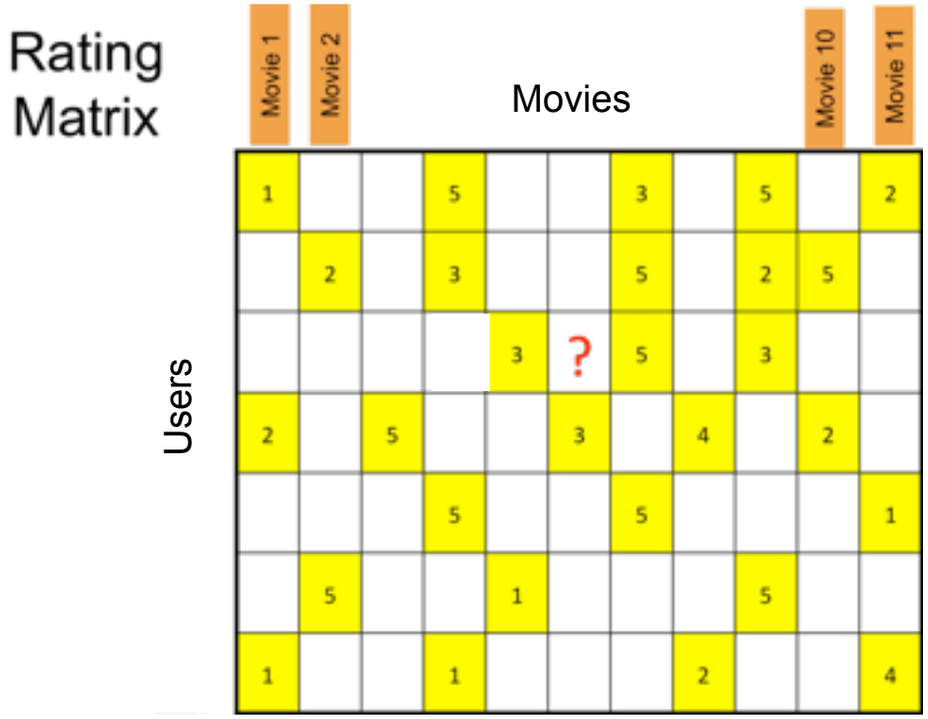
- Social networks are highly dynamic
  - Constantly grow, change quickly over time
    - Users arrive/leave, relationships form/dissolve
- Understanding graph evolution is important

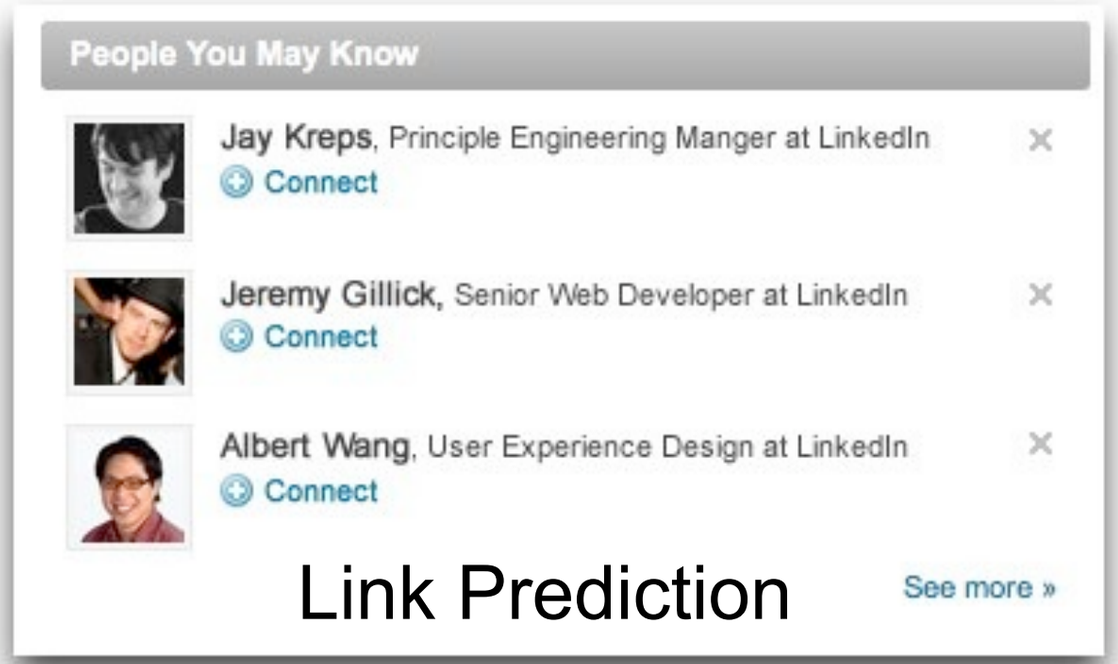
Graphs meet Machine Learning

- Network analysis: Understanding structure & evolution of networks
- Formulate predictive problems on the adjacency matrix of the graph
- Confluence of graph theory & machine learning

Scalable Network Analysis

Recommender Systems

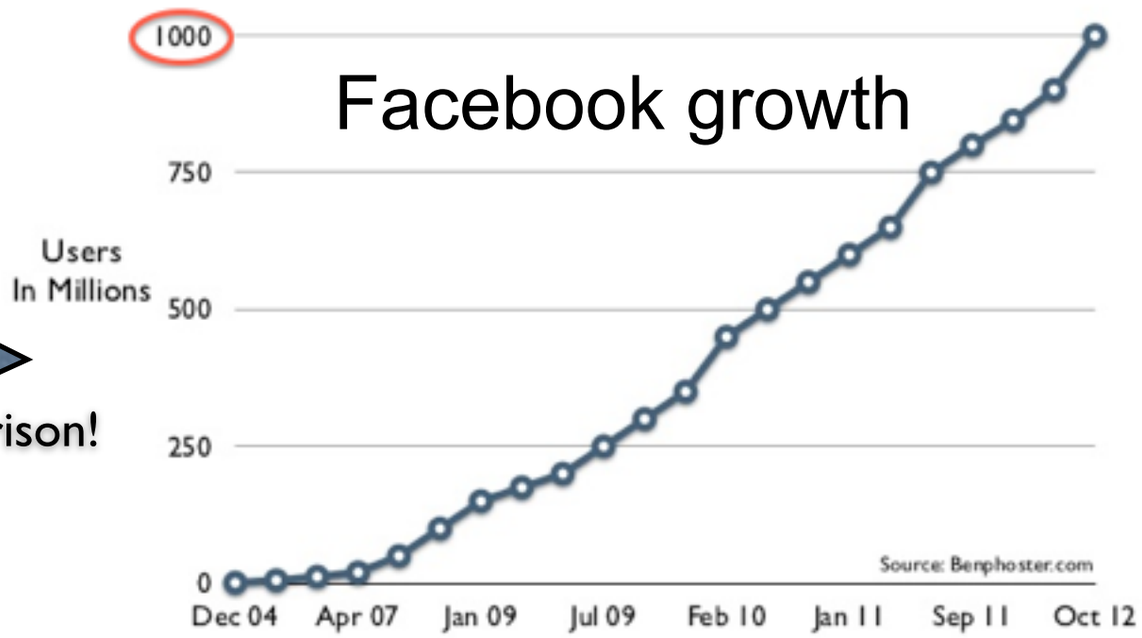




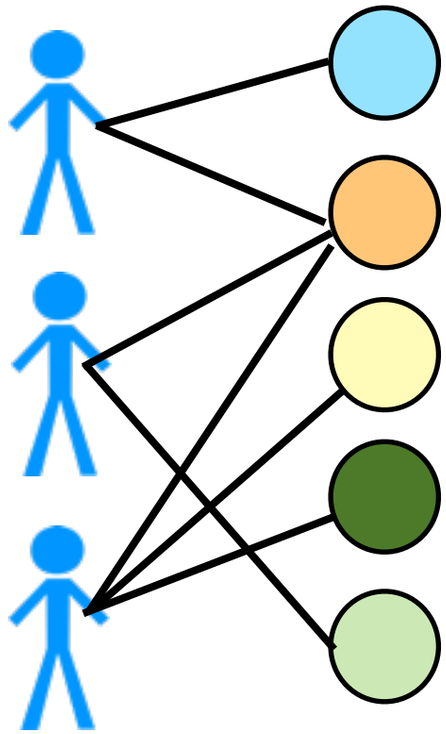
Link Prediction

Netflix problem:
100M ratings,
0.5M users,
20K movies

A Toy Problem In Comparison!



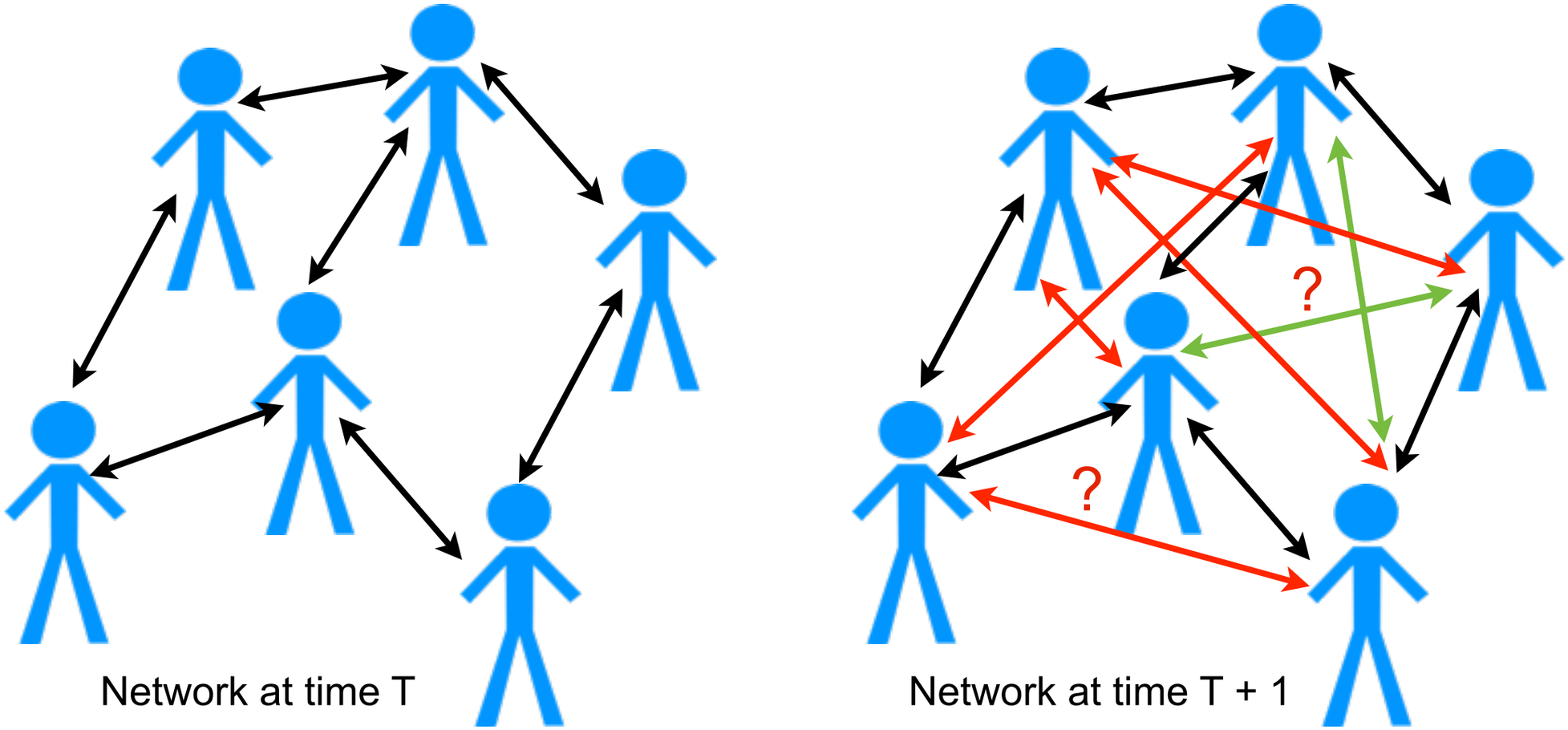
Recommender Systems



Rating Matrix

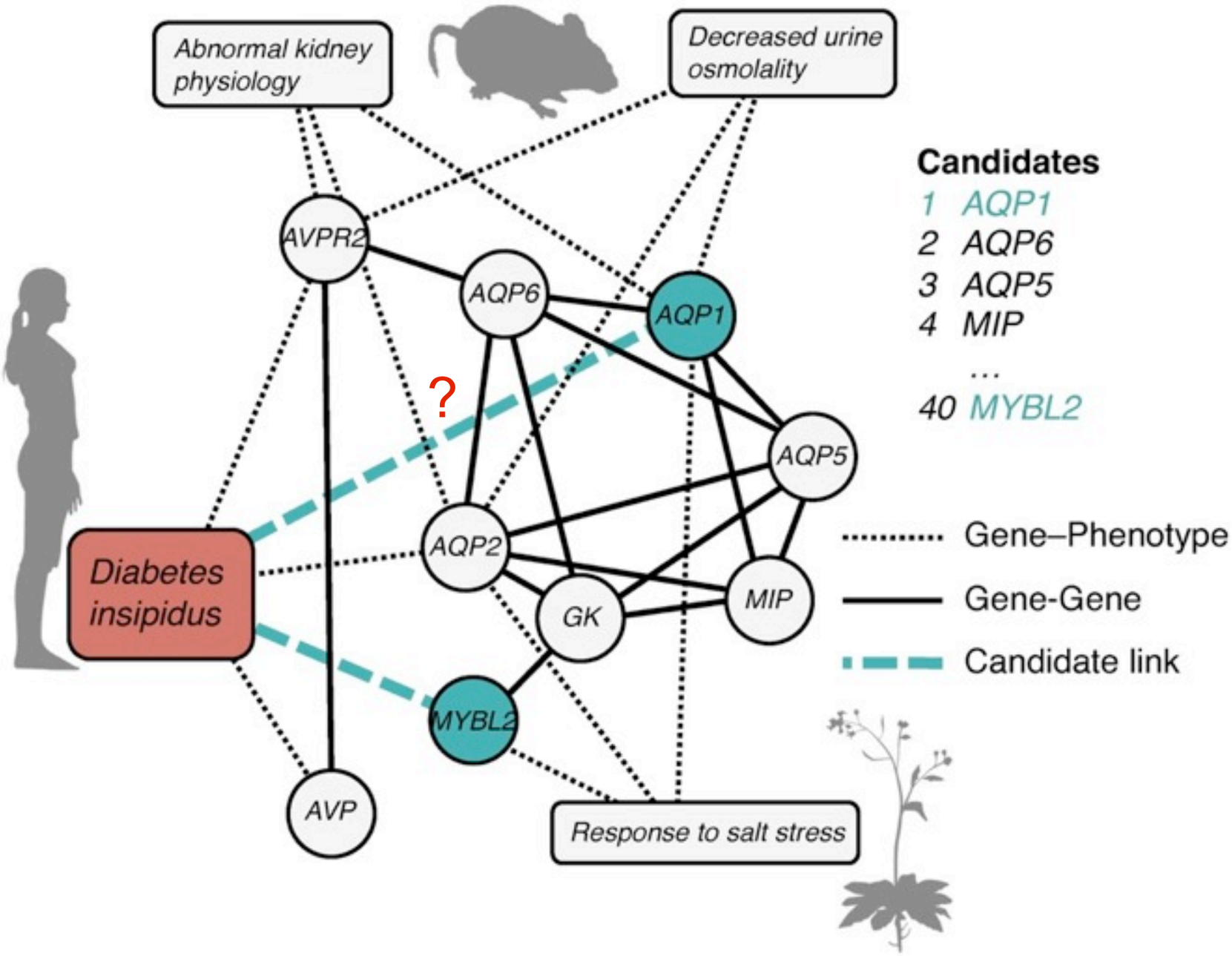
	Movie 1	Movie 2			Movies				Movie 10	Movie 11
	1			5			3		5		2
		2		3			5		2	5	
					3	?	5		3		
Users	2		5			3		4		2	
				5			5				1
		5			1				5		
	1			1				2			4

Link Prediction in Social Networks

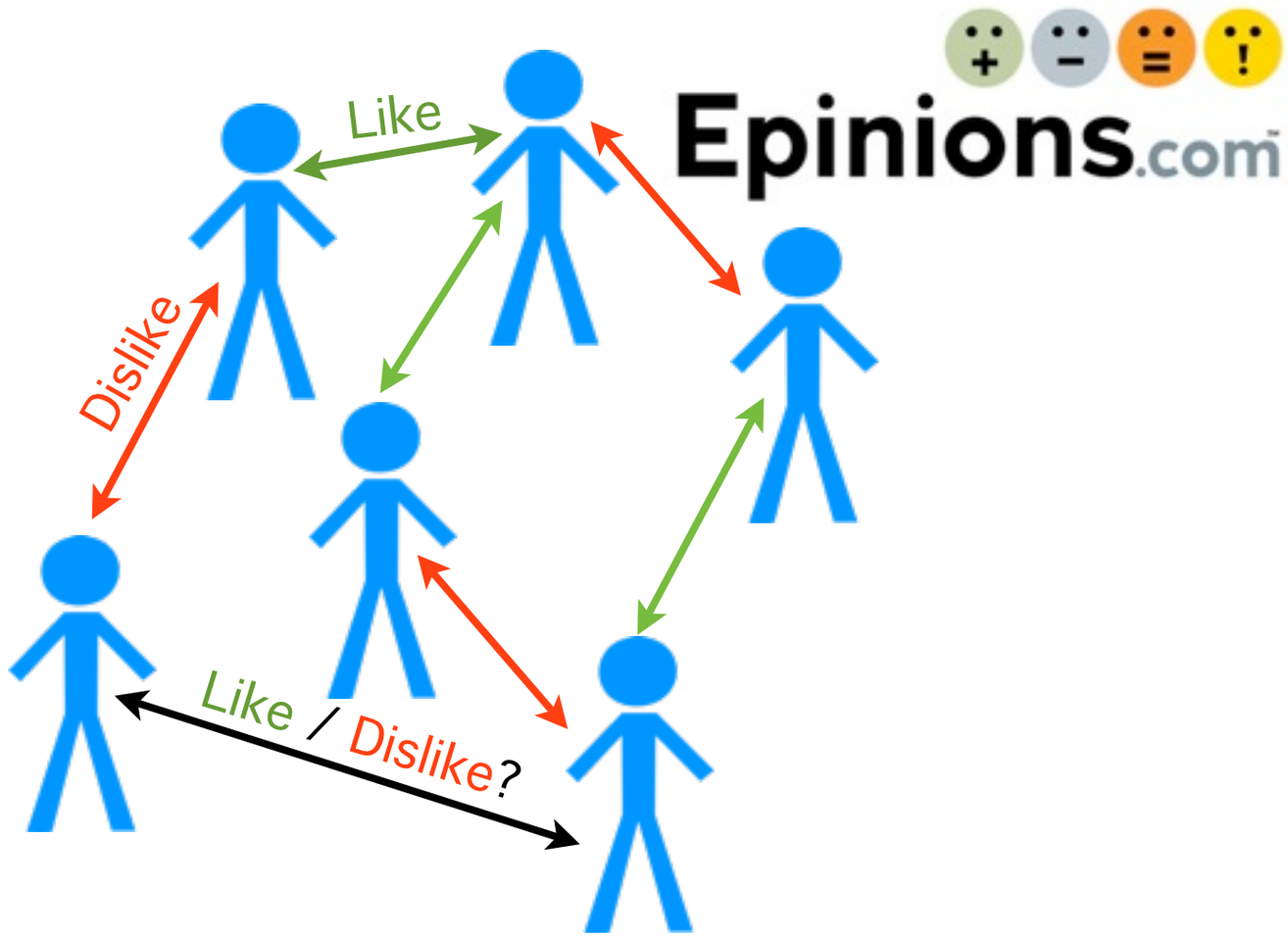


- Problem: Infer *missing* relationships from a given snapshot of the network

Predicting gene-disease links



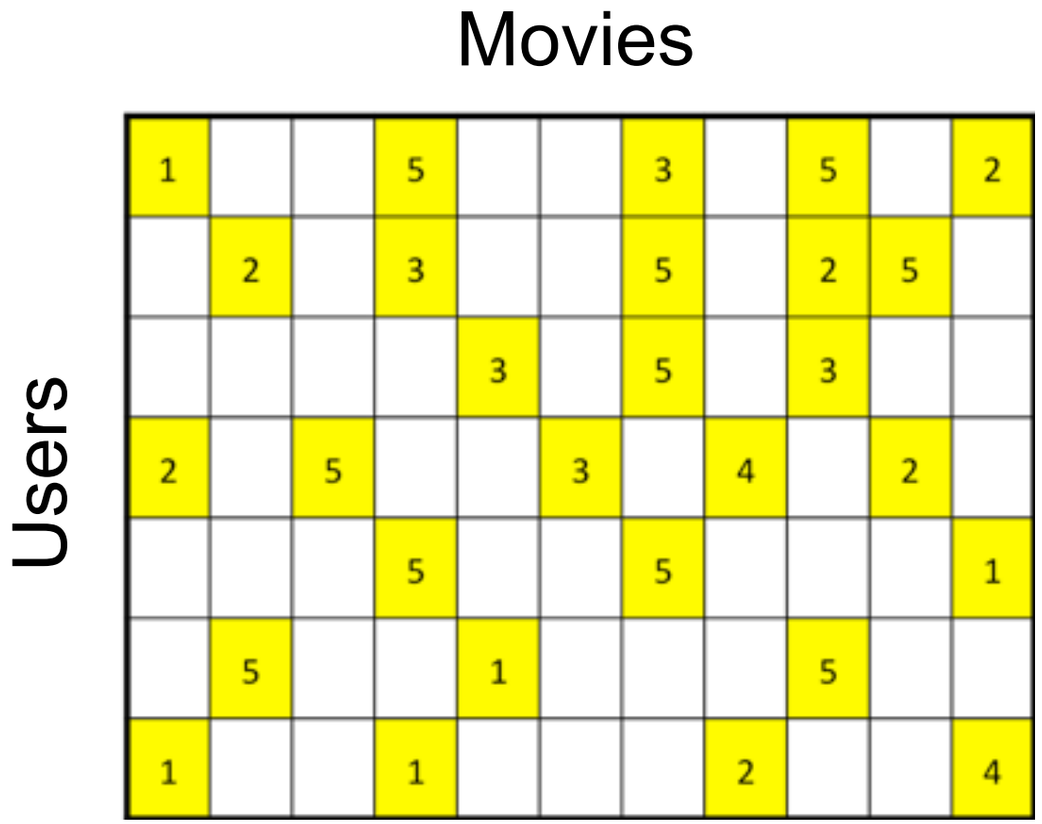
Signed Social Networks



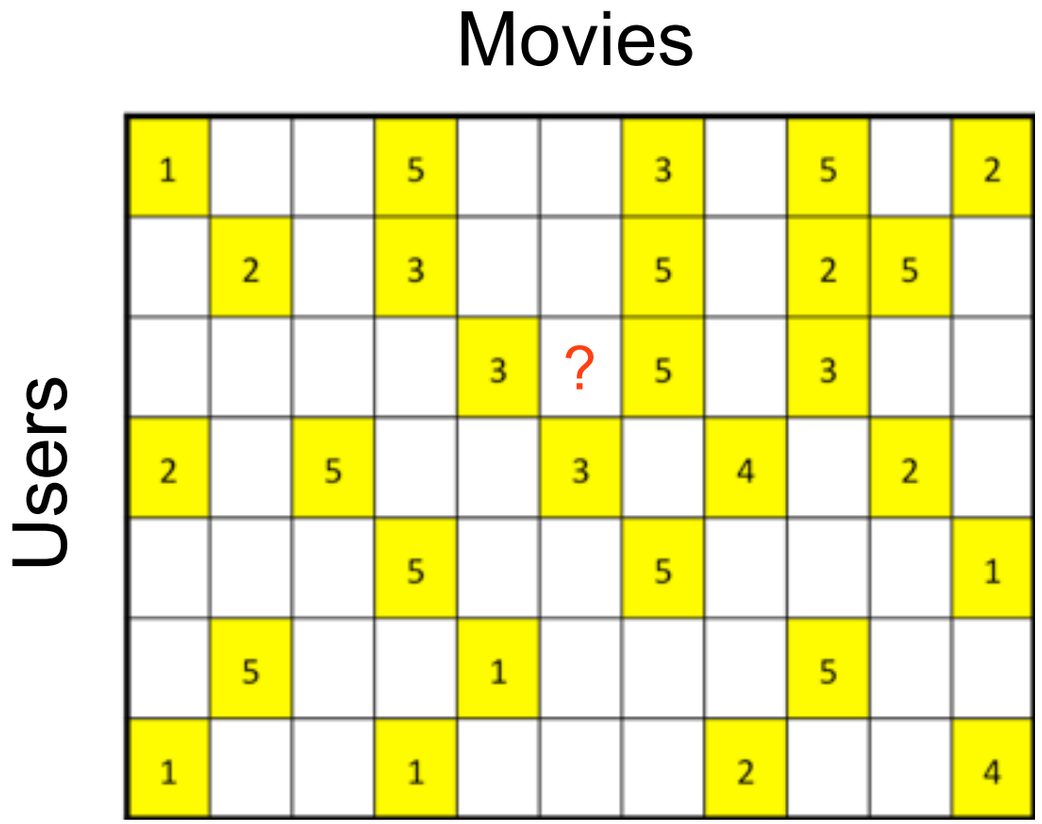
- Sign Prediction Problem: Given a snapshot of the signed social network, predict the signs of missing edges

Formulation as Missing Value Estimation

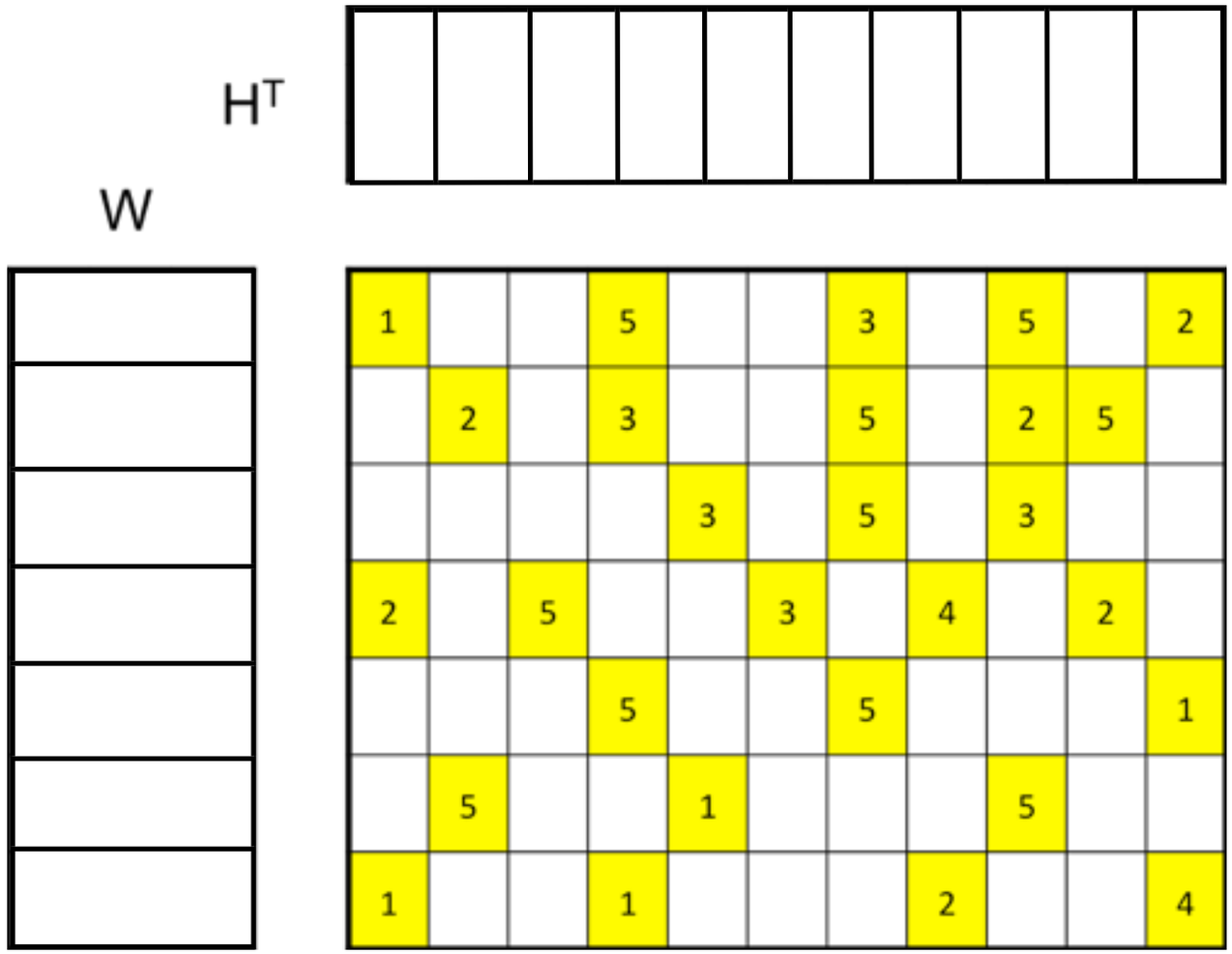
Low-rank Matrix Completion



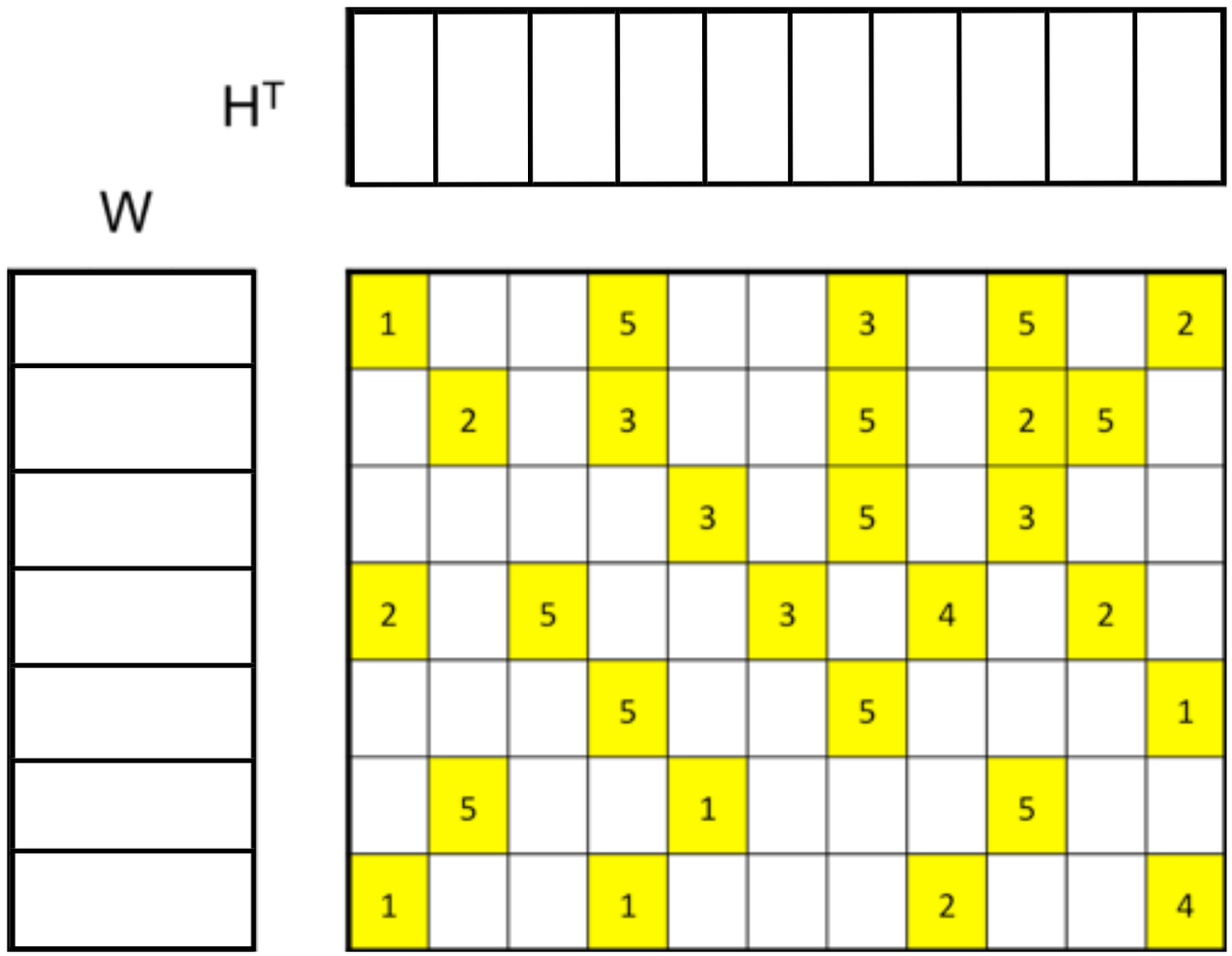
Low-rank Matrix Completion



Low-rank Matrix Completion

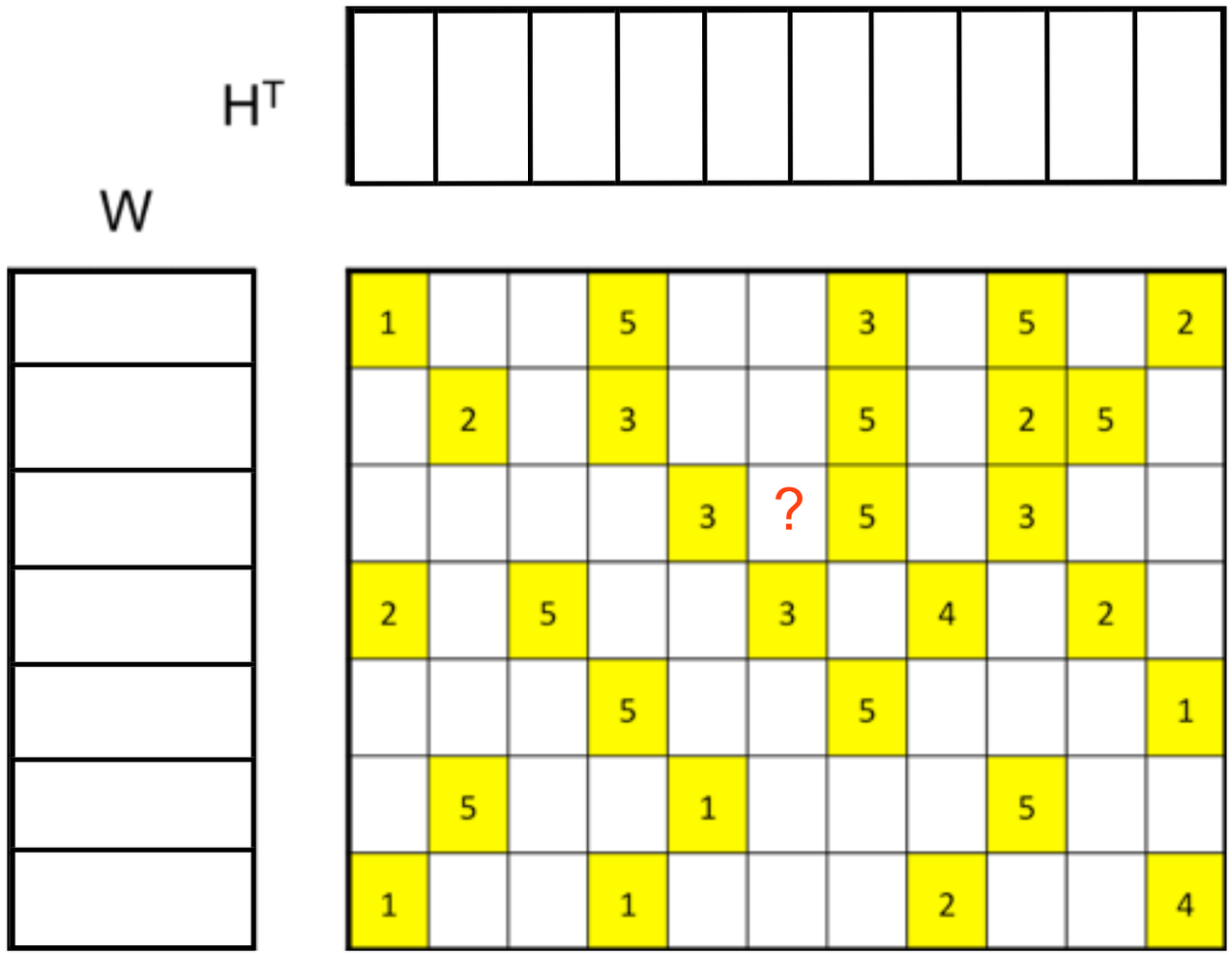


Low-rank Matrix Completion



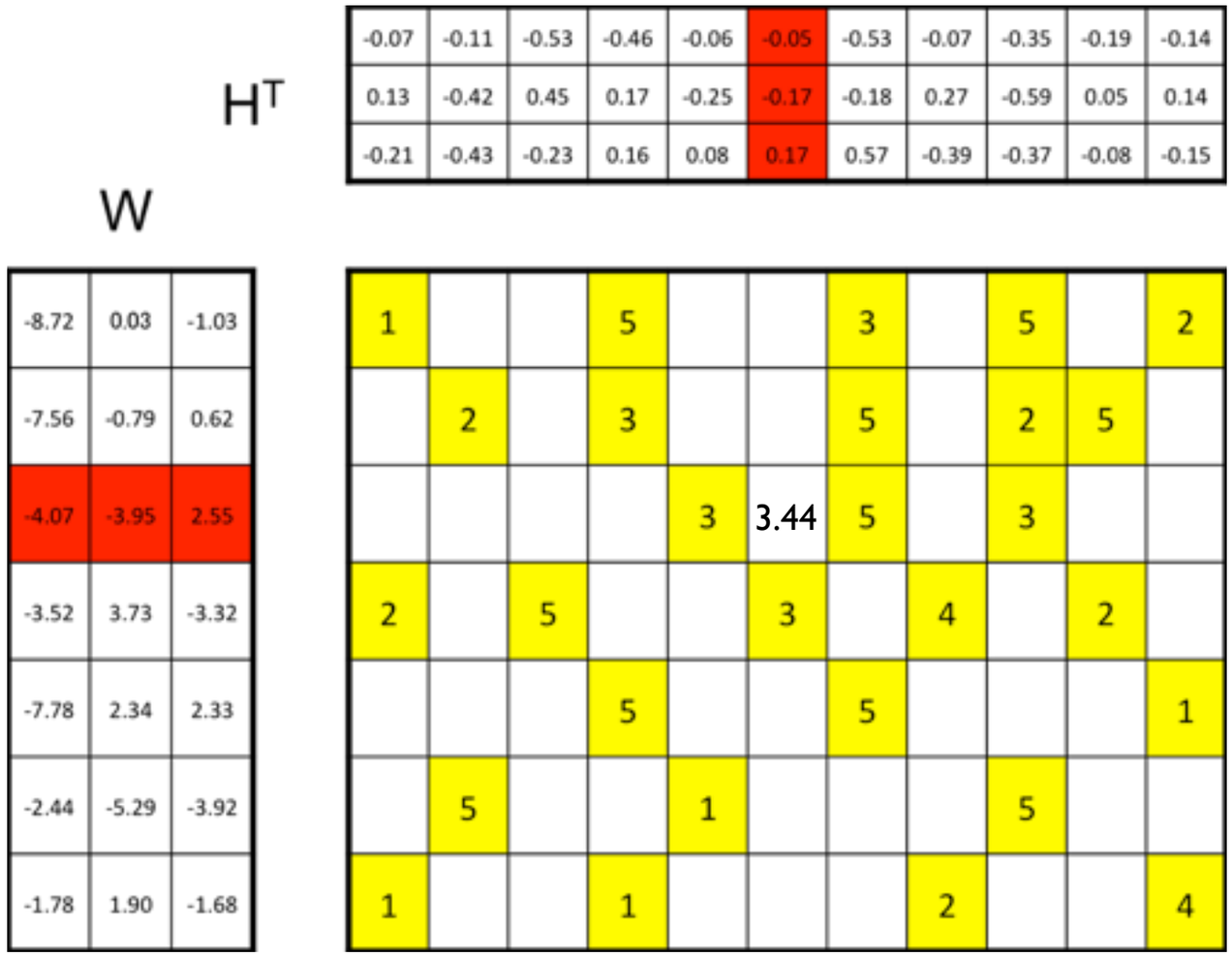
$$\min_{W \in \mathbb{R}^{m \times k}, H \in \mathbb{R}^{n \times k}} \sum_{(i,j) \in \Omega} (A_{ij} - W_i H_j^T)^2 + \lambda(\|W\|_F^2 + \|H\|_F^2)$$

Low-rank Matrix Completion



$$\min_{W \in \mathbb{R}^{m \times k}, H \in \mathbb{R}^{n \times k}} \sum_{(i,j) \in \Omega} (A_{ij} - W_i H_j^T)^2 + \lambda(\|W\|_F^2 + \|H\|_F^2)$$

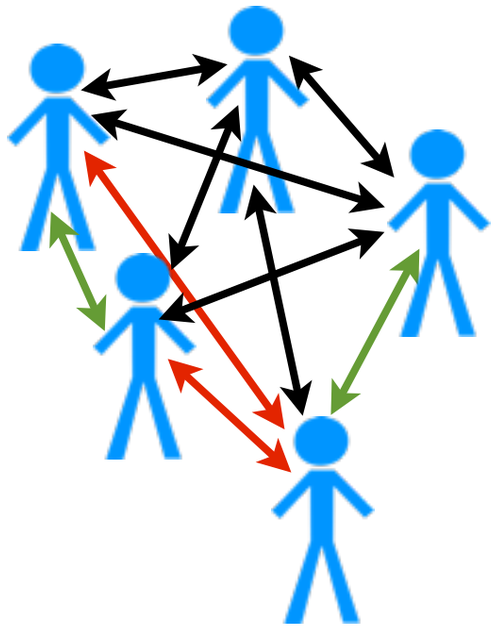
Low-rank Matrix Completion



$$\min_{W \in \mathbb{R}^{m \times k}, H \in \mathbb{R}^{n \times k}} \sum_{(i,j) \in \Omega} (A_{ij} - W_i H_j^T)^2 + \lambda(\|W\|_F^2 + \|H\|_F^2)$$

Link Prediction

- Can be posed as matrix completion problem



$$A^{(t)} = \begin{bmatrix} . & 1 & 1 & . & . \\ 1 & . & 1 & 1 & 1 \\ 1 & 1 & . & . & 1 \\ . & 1 & . & . & . \\ . & 1 & 1 & . & . \end{bmatrix} \approx \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

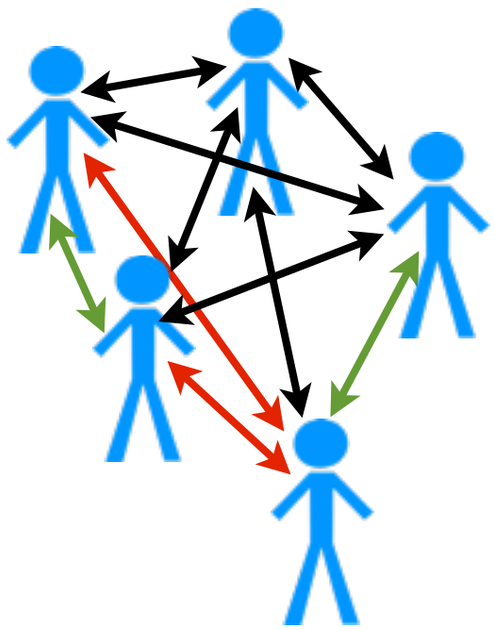
- Issue: Only positive relationships are observed

Test Link	Score
(4,5)	1
(1,4)	1
(3,4)	1
(1,5)	1

Link Prediction

- Formulate Biased Matrix Completion Problem:

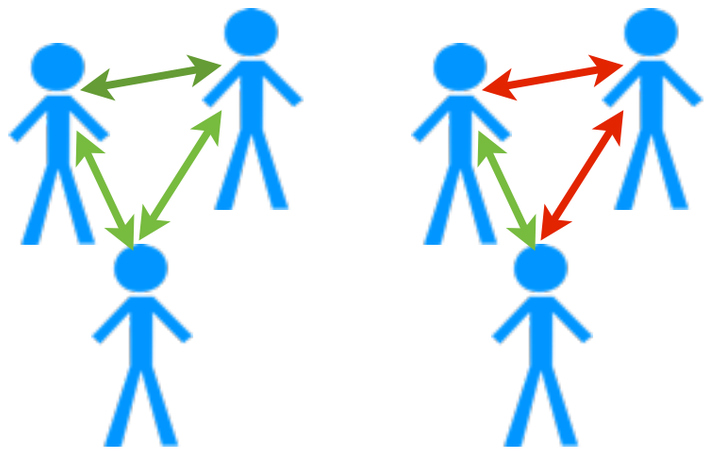
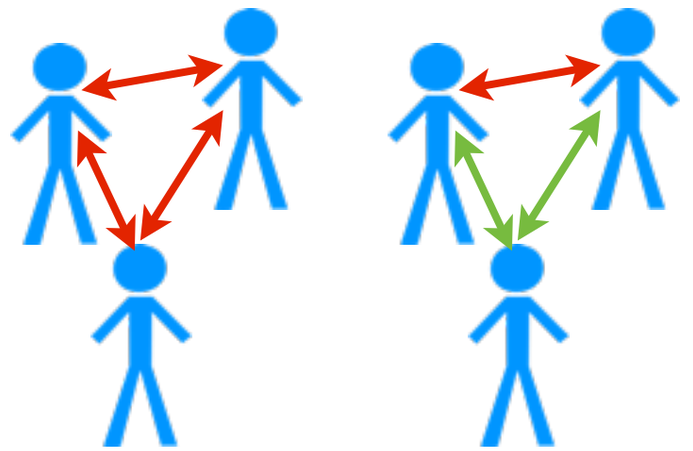
$$\min_{W \in \mathbb{R}^{n \times k}, H \in \mathbb{R}^{n \times k}} \sum_{(i,j) \in \Omega} (A_{ij} - W_i H_j^T)^2 + \alpha \sum_{(i,j) \notin \Omega} (A_{ij} - W_i H_j^T)^2 + \lambda(\|W\|_F^2 + \|H\|_F^2)$$



$$A^{(t)} = \begin{bmatrix} . & 1 & 1 & . & . \\ 1 & . & 1 & 1 & 1 \\ 1 & 1 & . & . & 1 \\ . & 1 & . & . & . \\ . & 1 & 1 & . & . \end{bmatrix} \approx \begin{bmatrix} 0.85 \\ 1.09 \\ 0.97 \\ 0.69 \\ 0.85 \end{bmatrix} \begin{bmatrix} 0.85 & 1.09 & 0.97 & 0.69 & 0.85 \end{bmatrix}$$

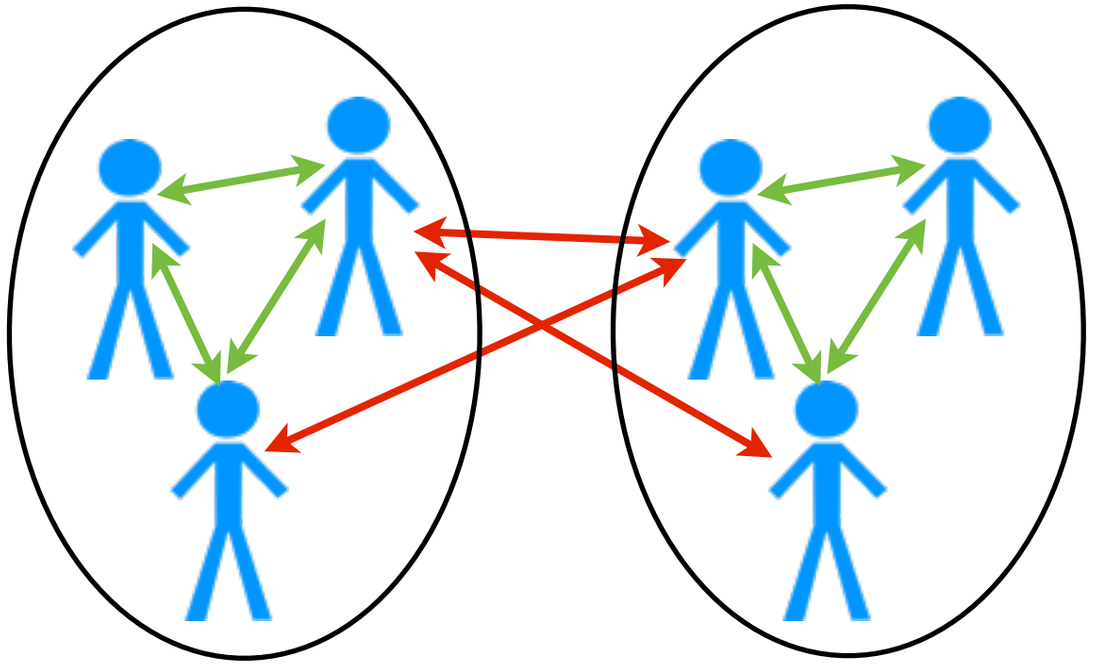
Test Link	Score
(4,5)	0.59
(1,4)	0.59
(3,4)	0.67
(1,5)	0.72

Signed Social Networks

Balanced	Not balanced
 Friend of a friend is a friend Enemy of an enemy is a friend	

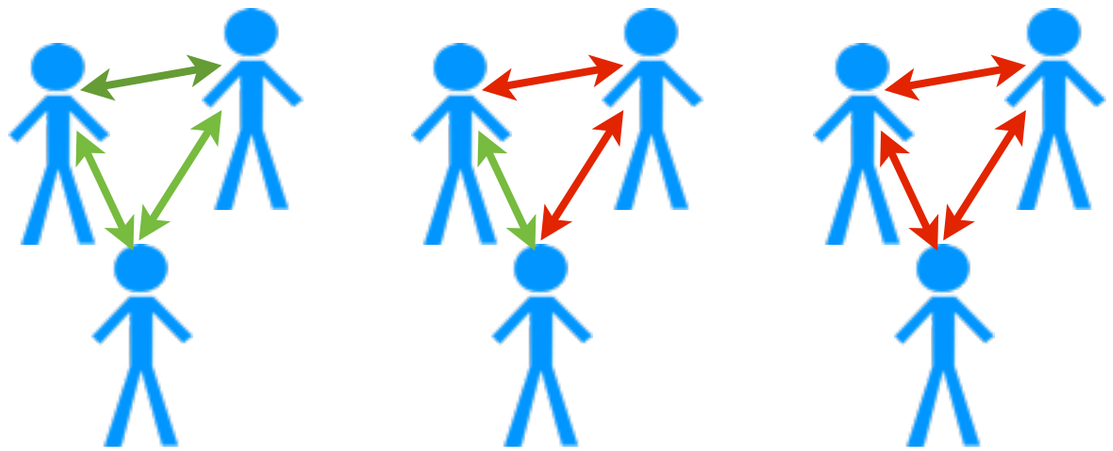
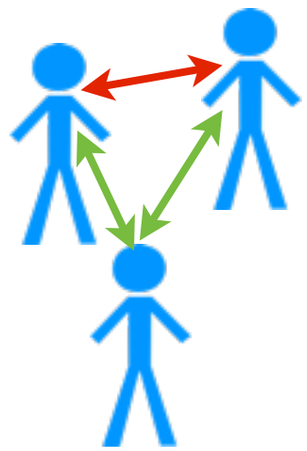
- Social Balance [Harary,1953]:
  - In real-world signed networks, triangles tend to be *balanced*

Signed Social Networks



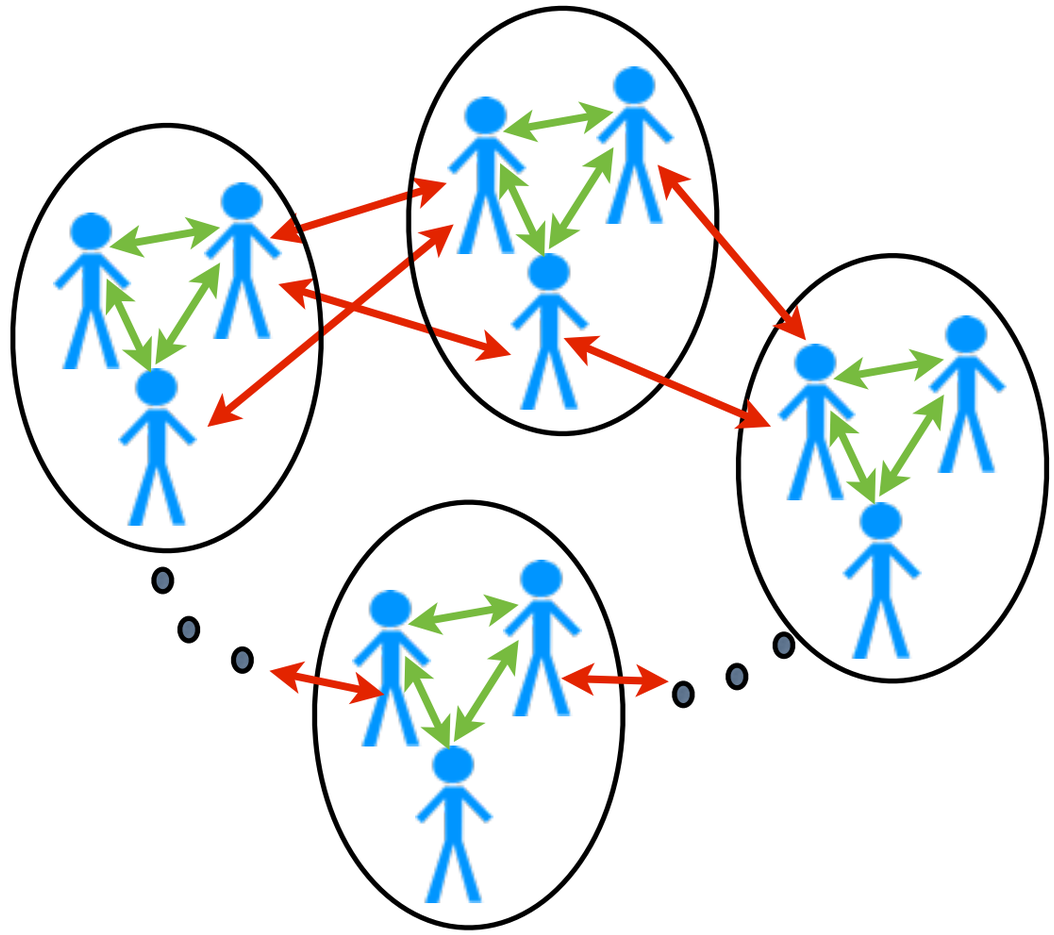
Theorem: All triangles in a network are balanced if and only if there exist two antagonistic groups.

Signed Social Networks

Weakly Balanced	Not balanced
	

- Relaxation: Weak balance
- Allow triangles with all negative edges

Signed Social Networks



Theorem: All triangles in a network are *weakly* balanced if and only if there exist *k* antagonistic groups.

Sign Prediction

Theorem: A *k*-weakly balanced signed network has rank at most *k.*

- Sign inference can be posed as low-rank matrix completion

Theorem: If there are no “small” groups, the underlying network can be *exactly* recovered, under certain conditions.

K. Chiang *et al*. ***Prediction and Clustering in Signed Networks: A Local to Global Perspective.*** To appear in JMLR.

Scalable Algorithms

Stochastic Gradient

- Sample random index (i,j) and update corresponding factors:

$$\mathbf{w}_i \leftarrow \mathbf{w}_i - \eta((A_{ij} - \mathbf{w}_i^T \mathbf{h}_j)\mathbf{h}_j + \lambda \mathbf{w}_i)$$

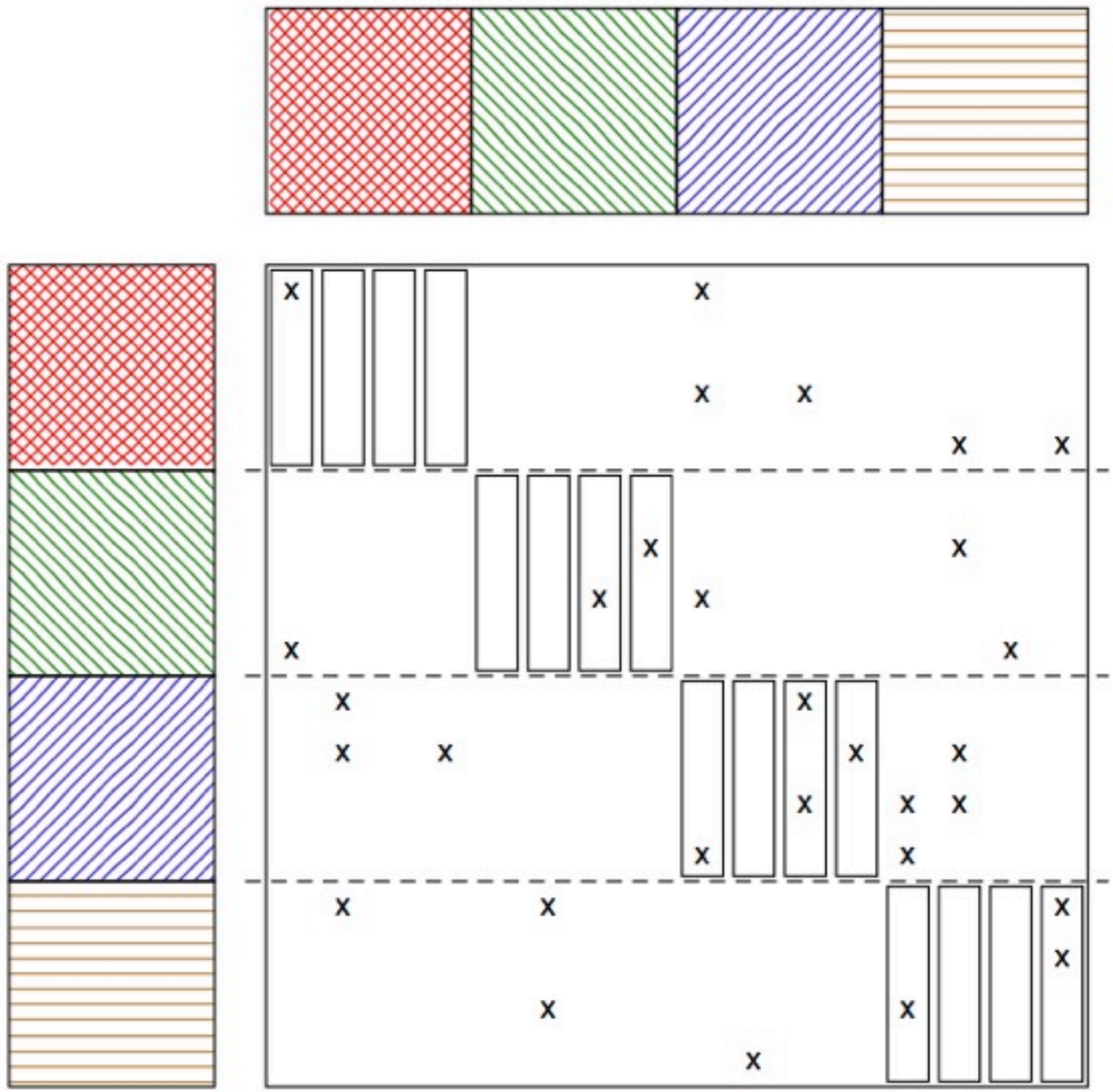
$$\mathbf{h}_j \leftarrow \mathbf{h}_j - \eta((A_{ij} - \mathbf{w}_i^T \mathbf{h}_j)\mathbf{w}_i + \lambda \mathbf{h}_j)$$

- Time per update $O(k)$
- Effective for very large-scale problems

Distributed Stochastic Gradient Descent (DSGD) [Gemulla *et al*. KDD 2011]

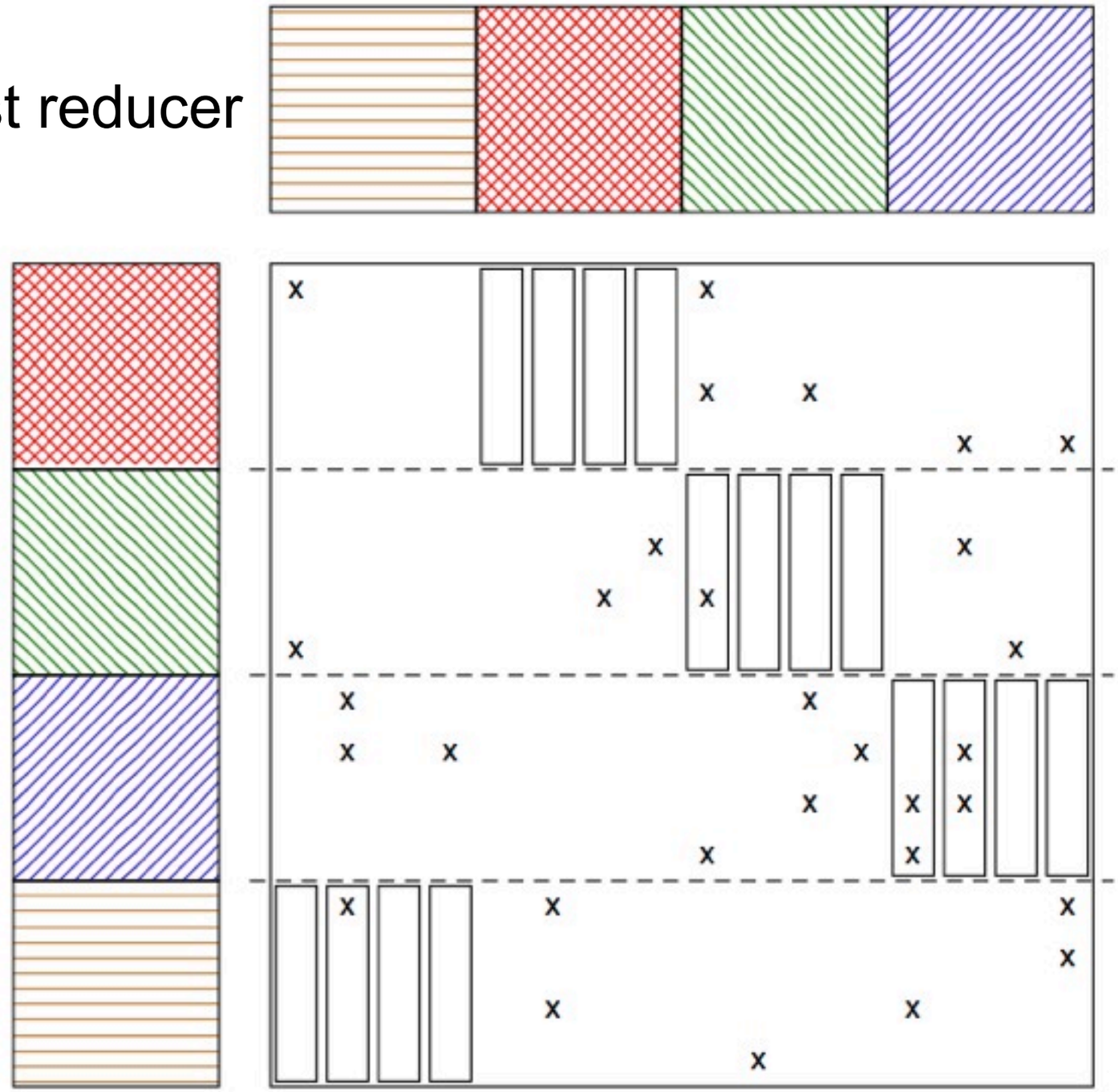
- Decoupled updates
- Easy to parallelize
- But communication & computation are interleaved

DSGD

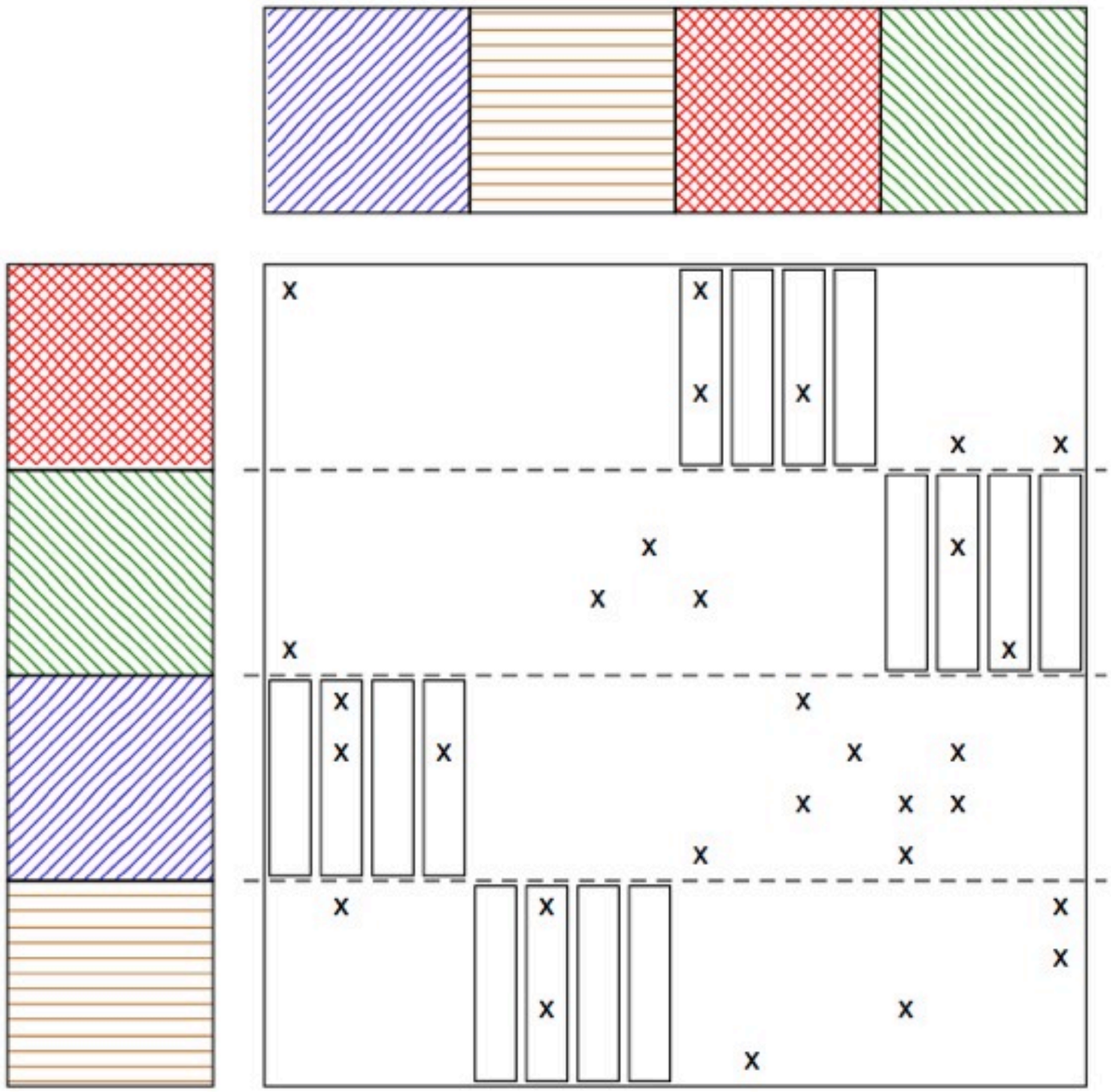


DSGD

Curse of the last reducer



DSGD



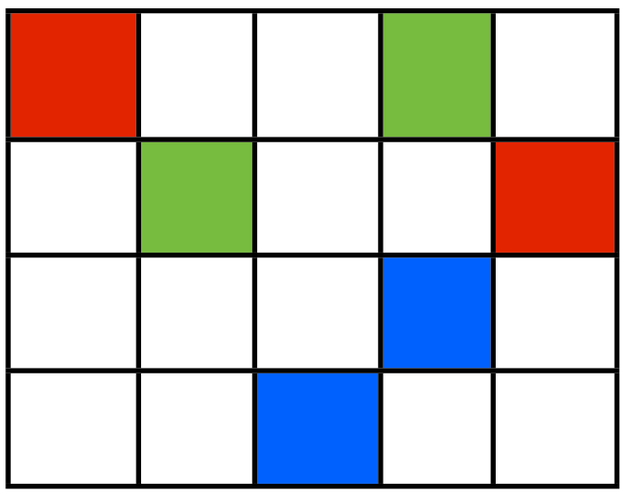
NOMAD

- **Goal**: Keep CPU & network simultaneously busy.

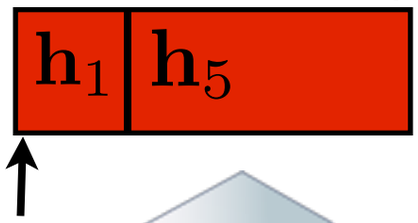
Non-locking
stOchastic
Multi-machine algorithm for
Asynchronous &
Decentralized matrix factorization

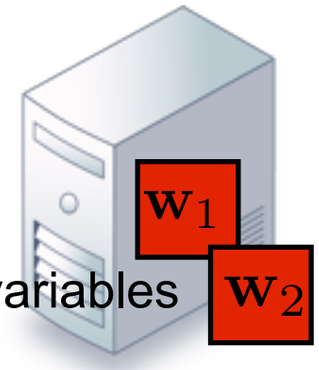
- Asynchronous distributed solution.

NOMAD

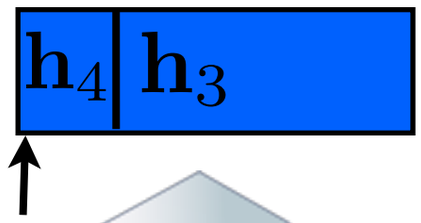


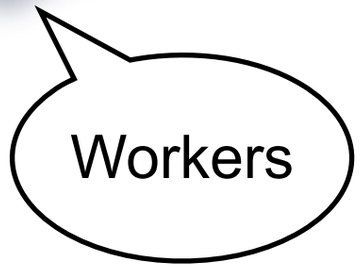
Nomadic variables queue

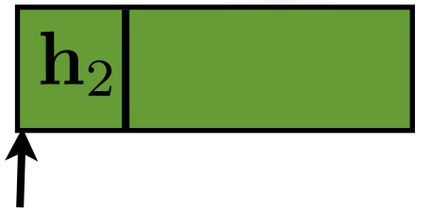




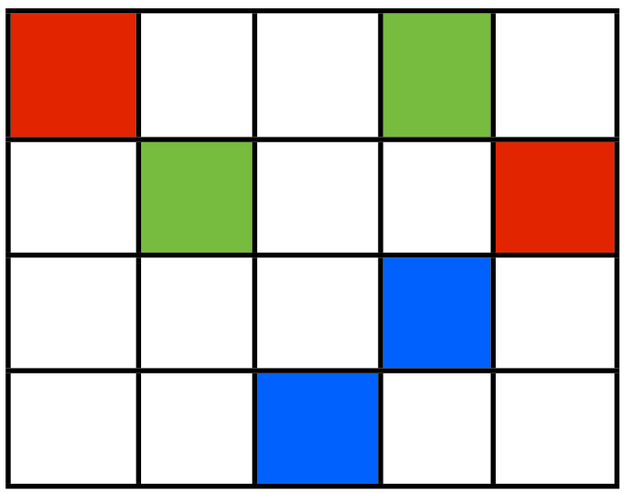
Native variables

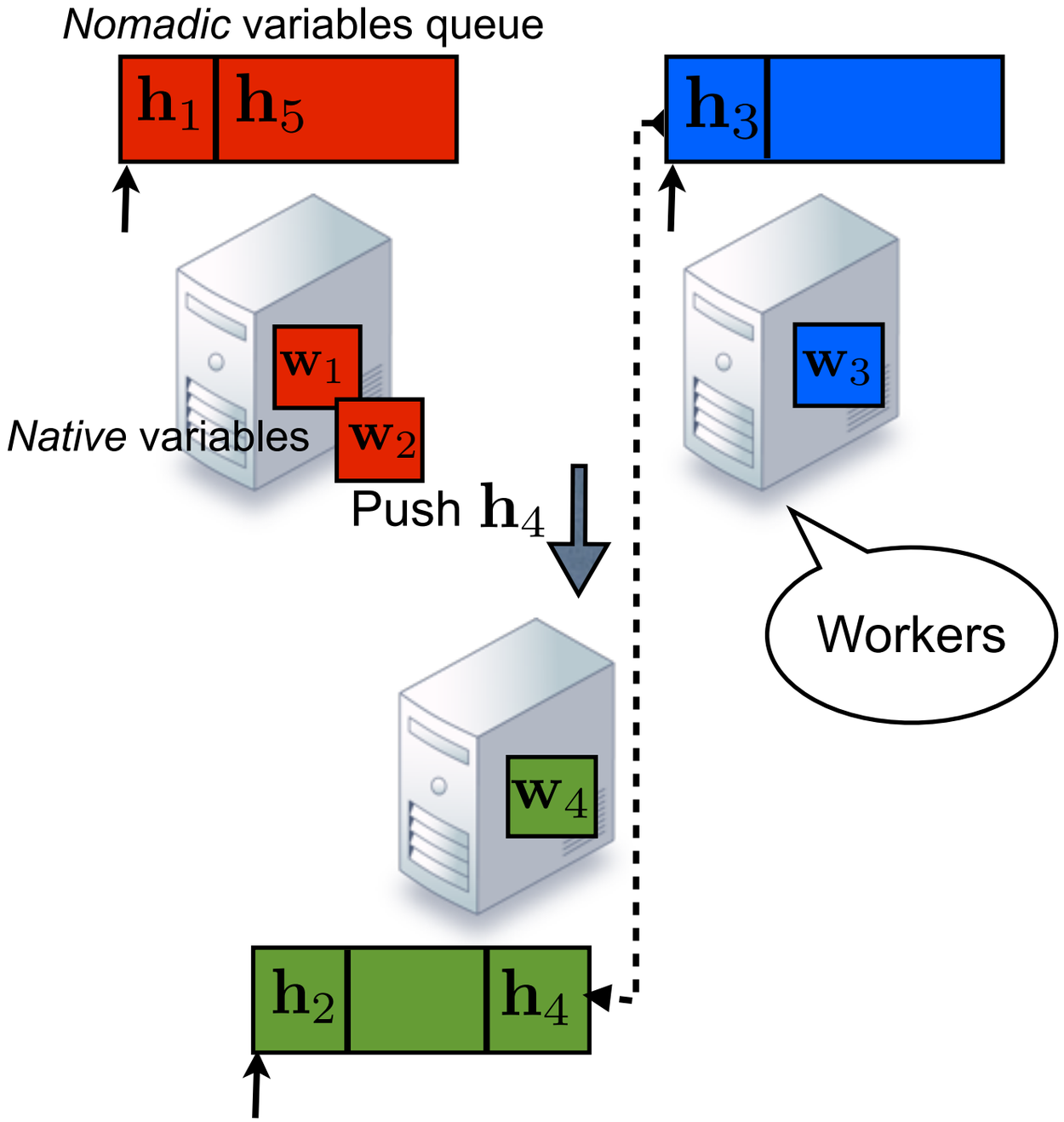




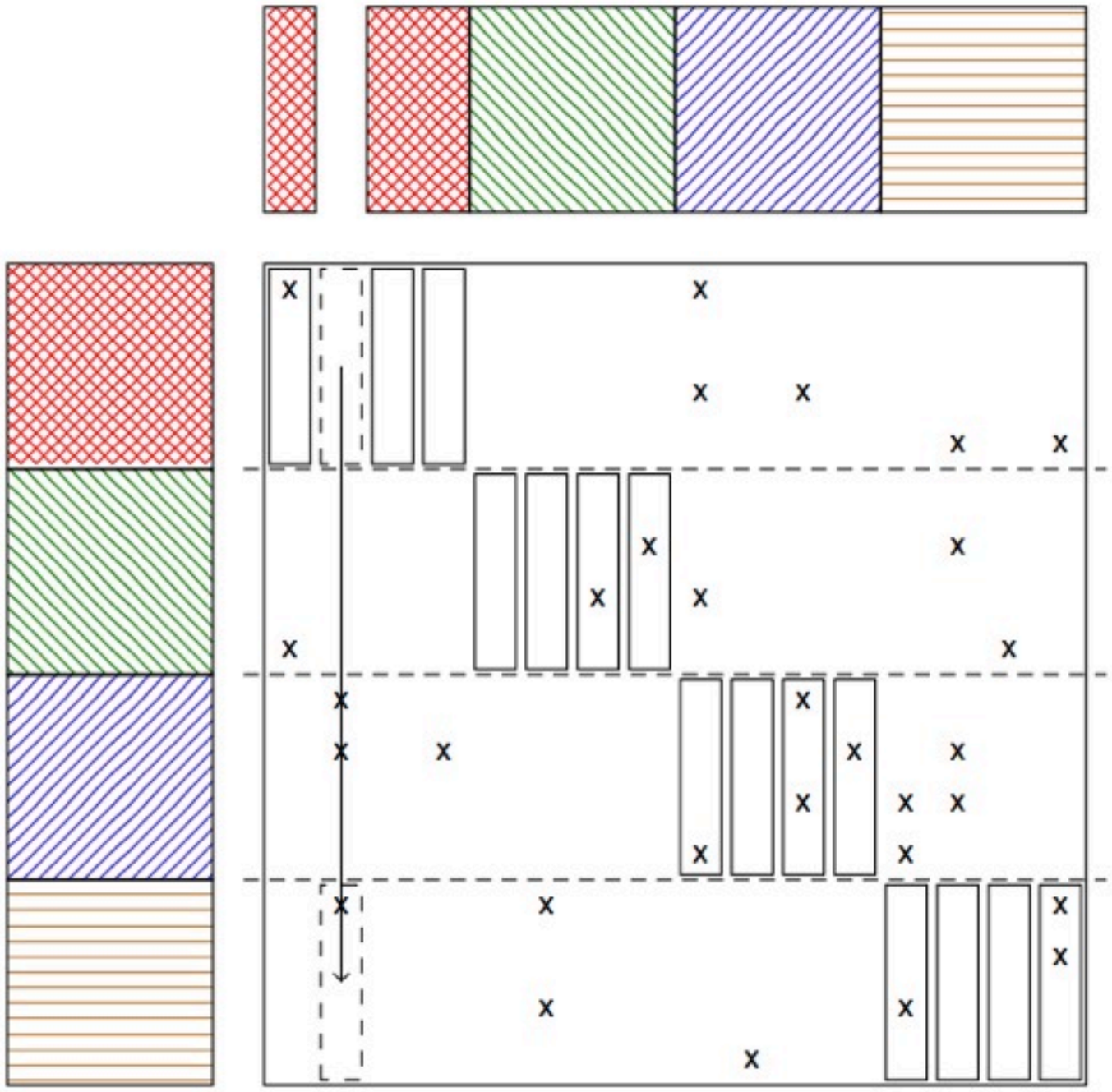


NOMAD

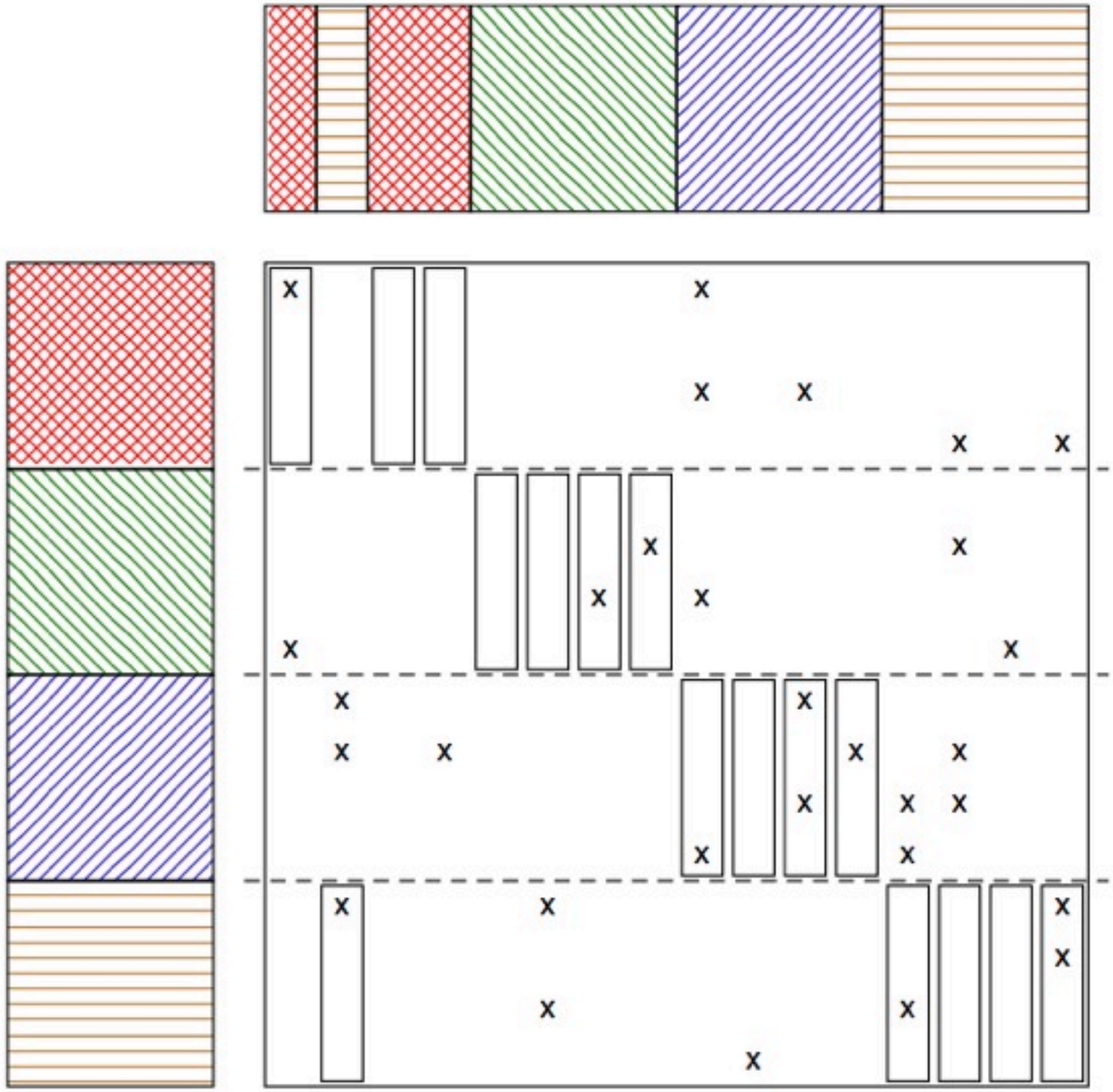



Nomadic variables queue
$\mathbf{h}_1$ $\mathbf{h}_5$
$\mathbf{h}_3$
$\mathbf{w}_1$
$\mathbf{w}_3$
Native variables
$\mathbf{w}_2$
Push $\mathbf{h}_4$
Workers
$\mathbf{w}_4$
$\mathbf{h}_2$
$\mathbf{h}_4$

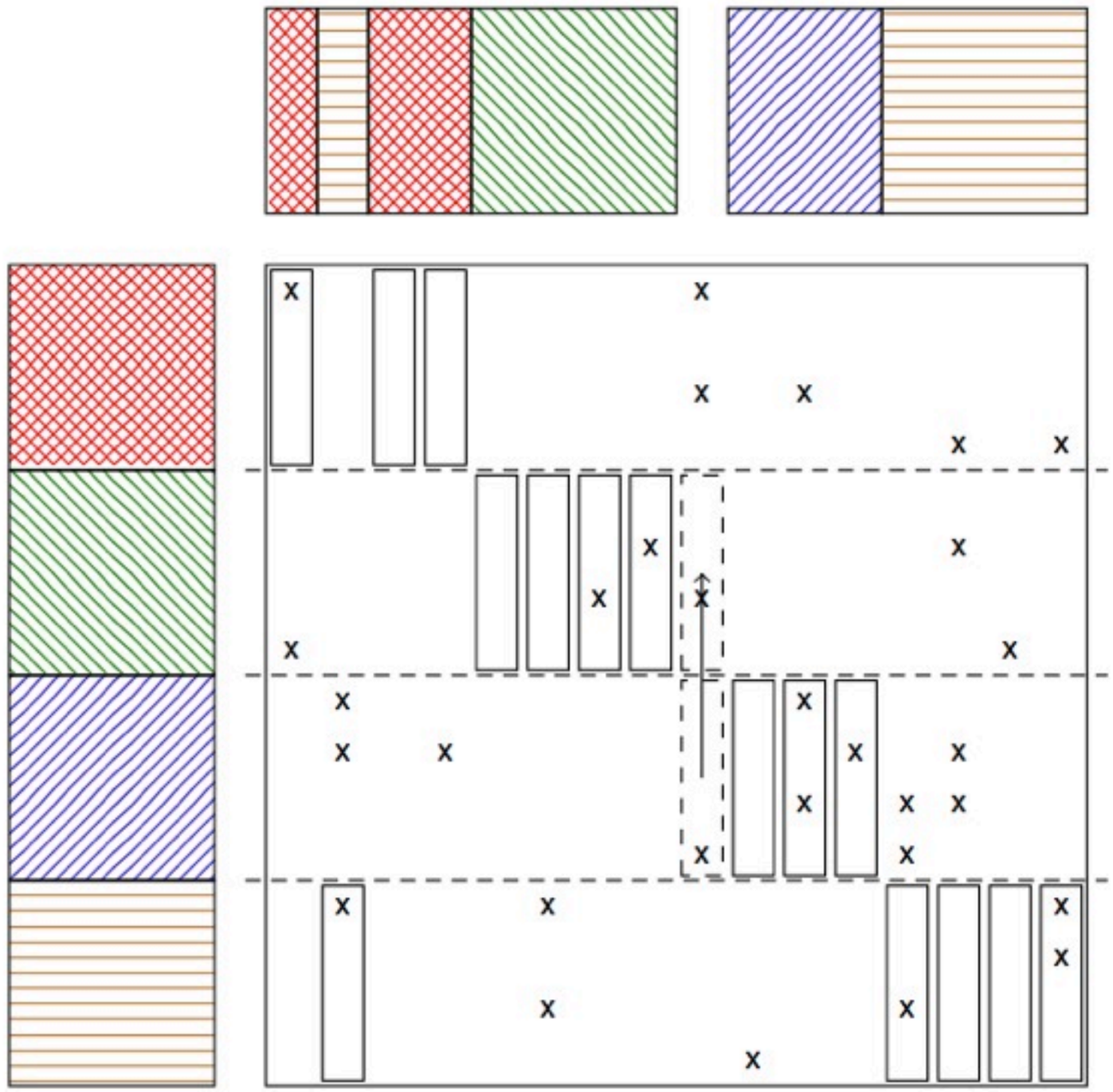
NOMAD



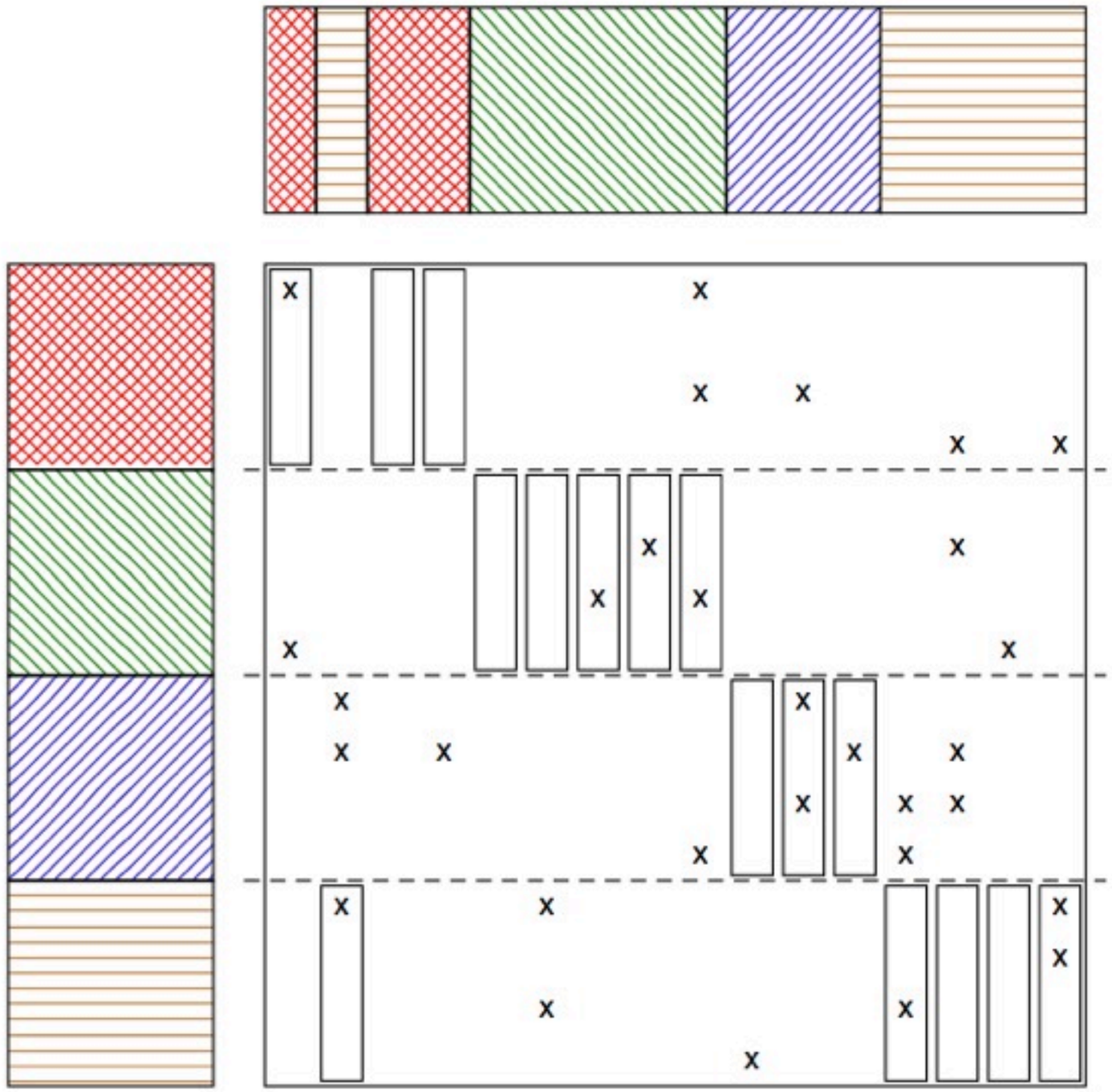
NOMAD



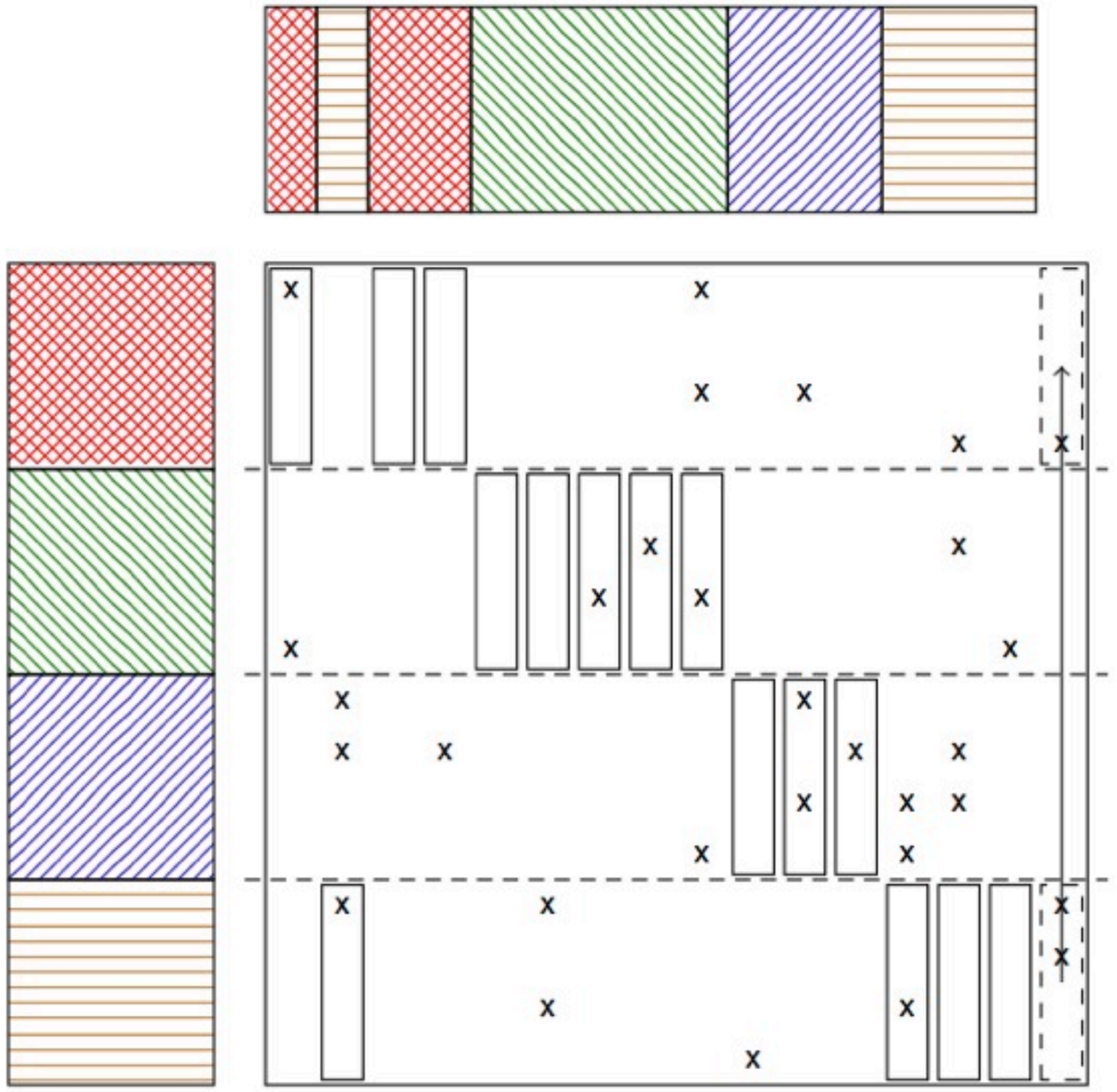
NOMAD



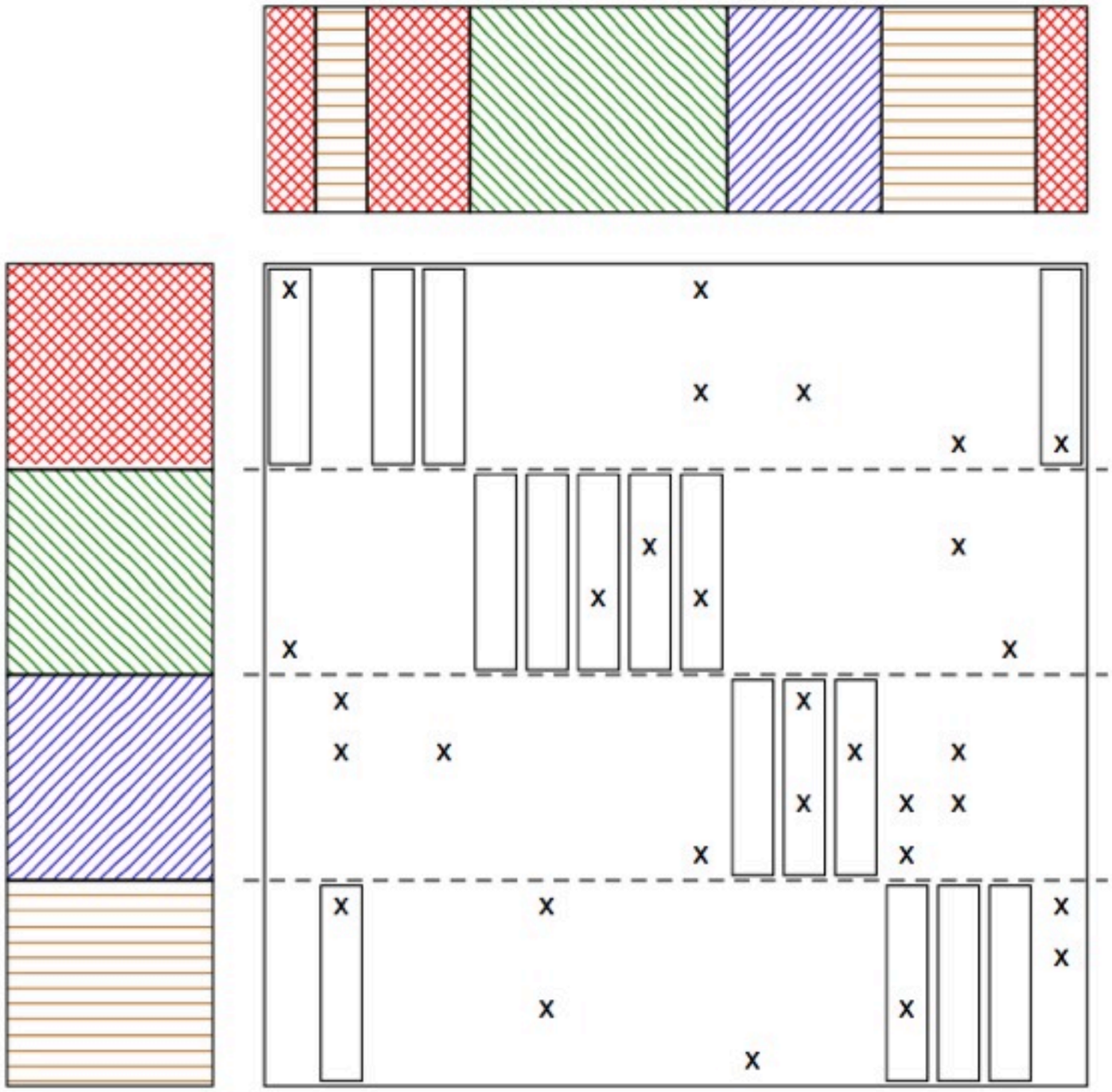
NOMAD



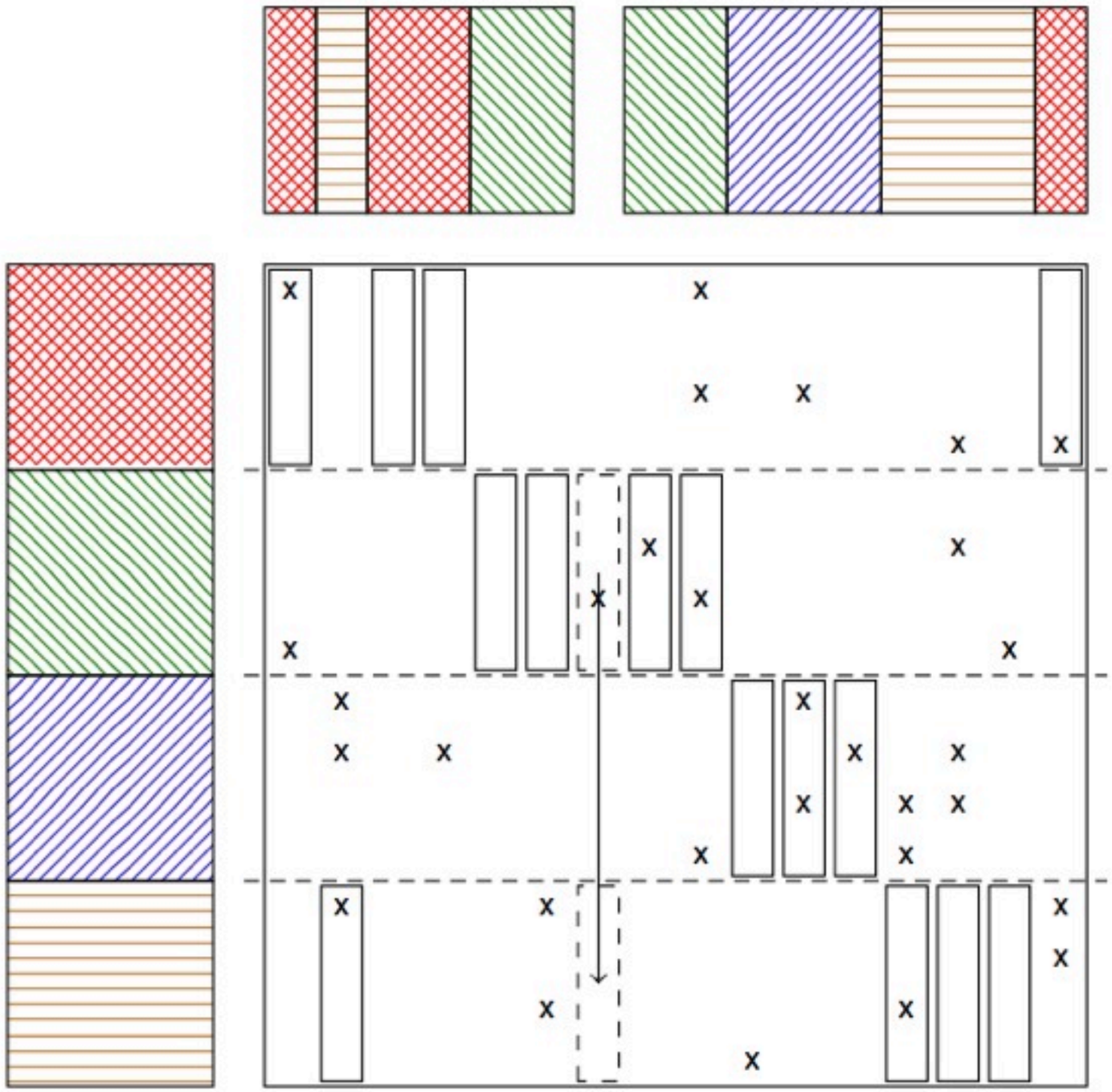
NOMAD



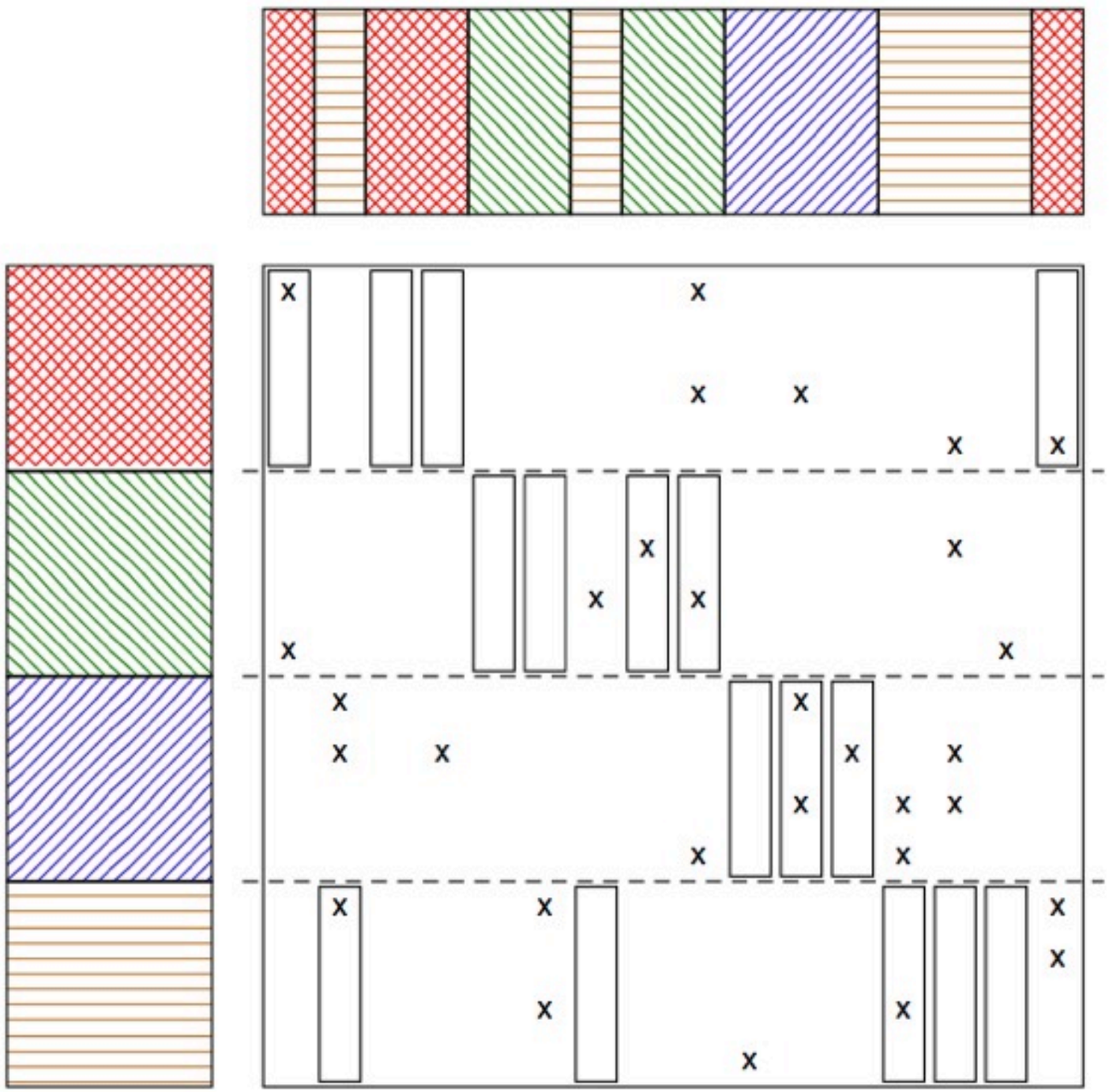
NOMAD



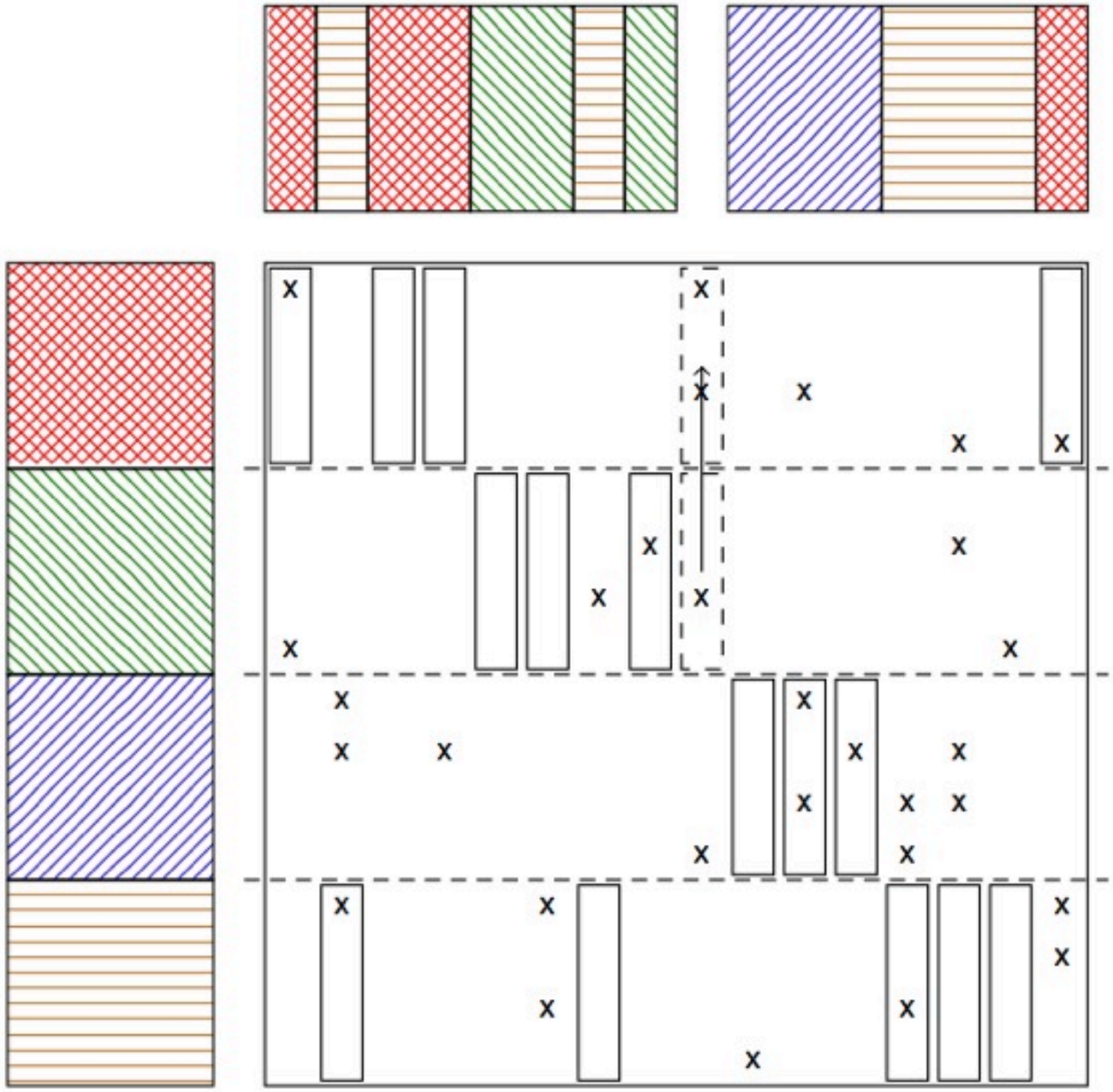
NOMAD



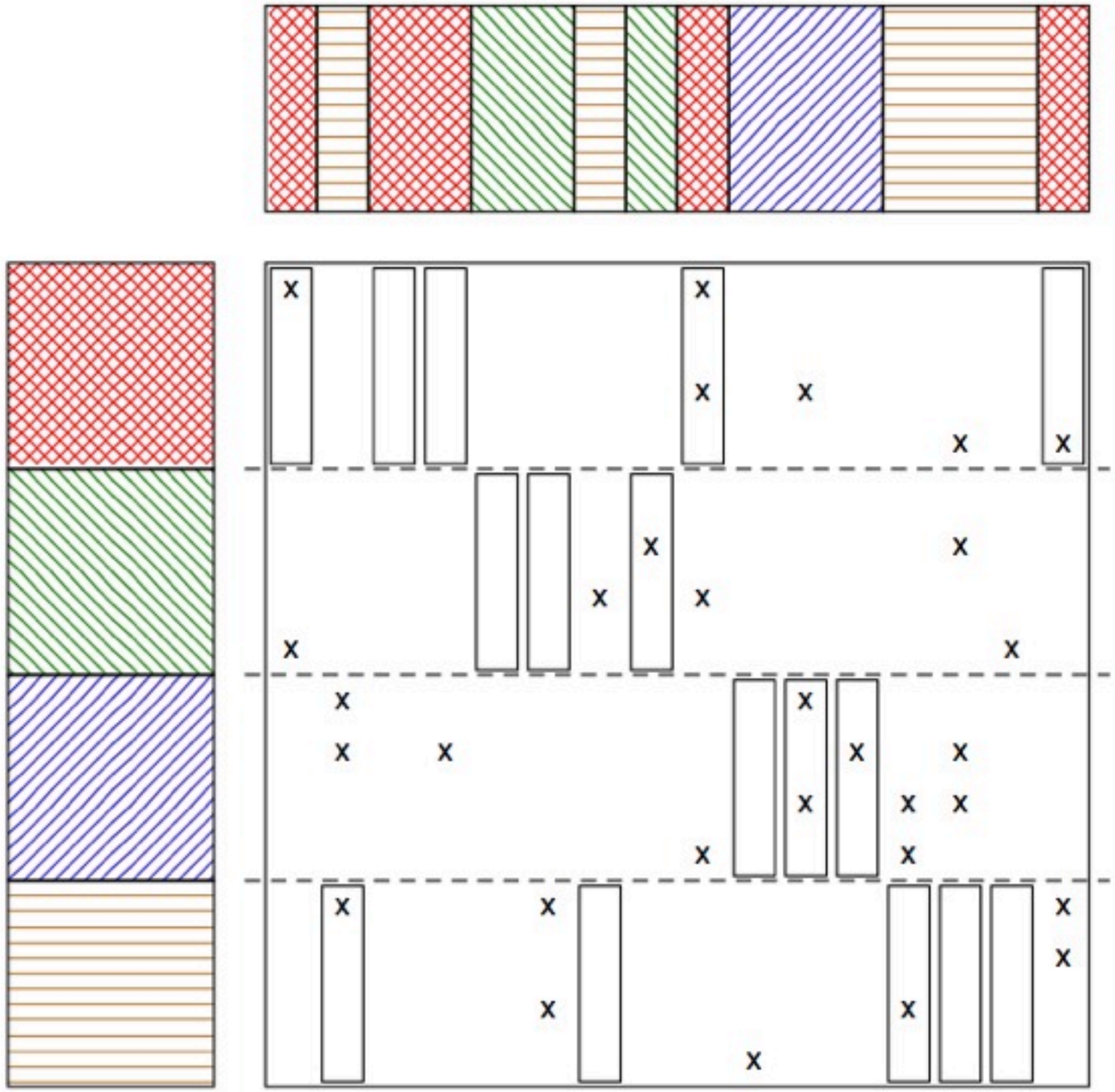
NOMAD



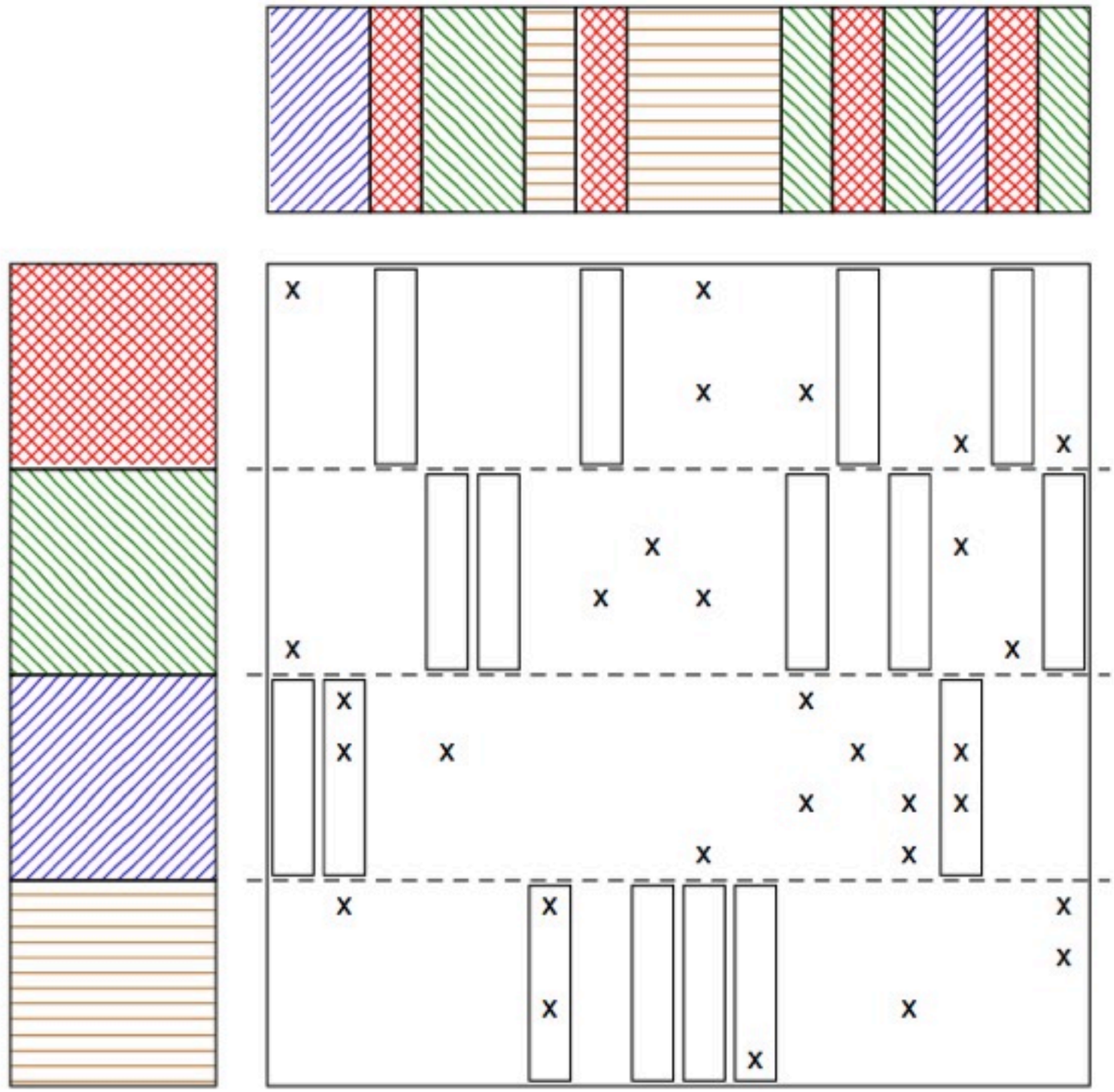
NOMAD



NOMAD



NOMAD



NOMAD Algorithm

1. **Initialize**: Randomly assign $\mathbf{h}_j$ columns to worker queues
2. **Parallel Foreach** *q* **in** {1,2,...,*p*}
3. **If** queue[*q*] not empty **then**
4. $(j, \mathbf{h}_j) \longleftarrow$ queue[*q*].pop()
5. **for** (i, j) **in** $\Omega_j^{(q)}$ do
6. Do SGD updates
7. **end for**
8. Sample *q*' uniformly from {1,2,...,*p*}
9. queue[*q*'].push($(j, \mathbf{h}_j)$)
10. **end if**
11. **Parallel End**

NOMAD Algorithm

1. **Initialize**: Randomly assign $\mathbf{h}_j$ columns to worker queues
2. **Parallel Foreach** *q* **in** {1,2,...,*p*}
3. **If** queue[*q*] not empty **then**
4. $(j, \mathbf{h}_j) \longleftarrow$ queue[*q*].pop()
5. **for** (i, j) **in** $\Omega_j^{(q)}$ do
6. Do SGD u
7. **end for**
8. Sample *q*' uniformly from {1,2,...,*p*}
9. queue[*q*'].push($(j, \mathbf{h}_j)$)
10. **end if**
11. **Parallel End**

Distributed setting: Write over the network

Concurrent object

Algorithm Complexity

- Average space required per worker:

$$O(mk/p + nk/p + |\Omega|/p)$$

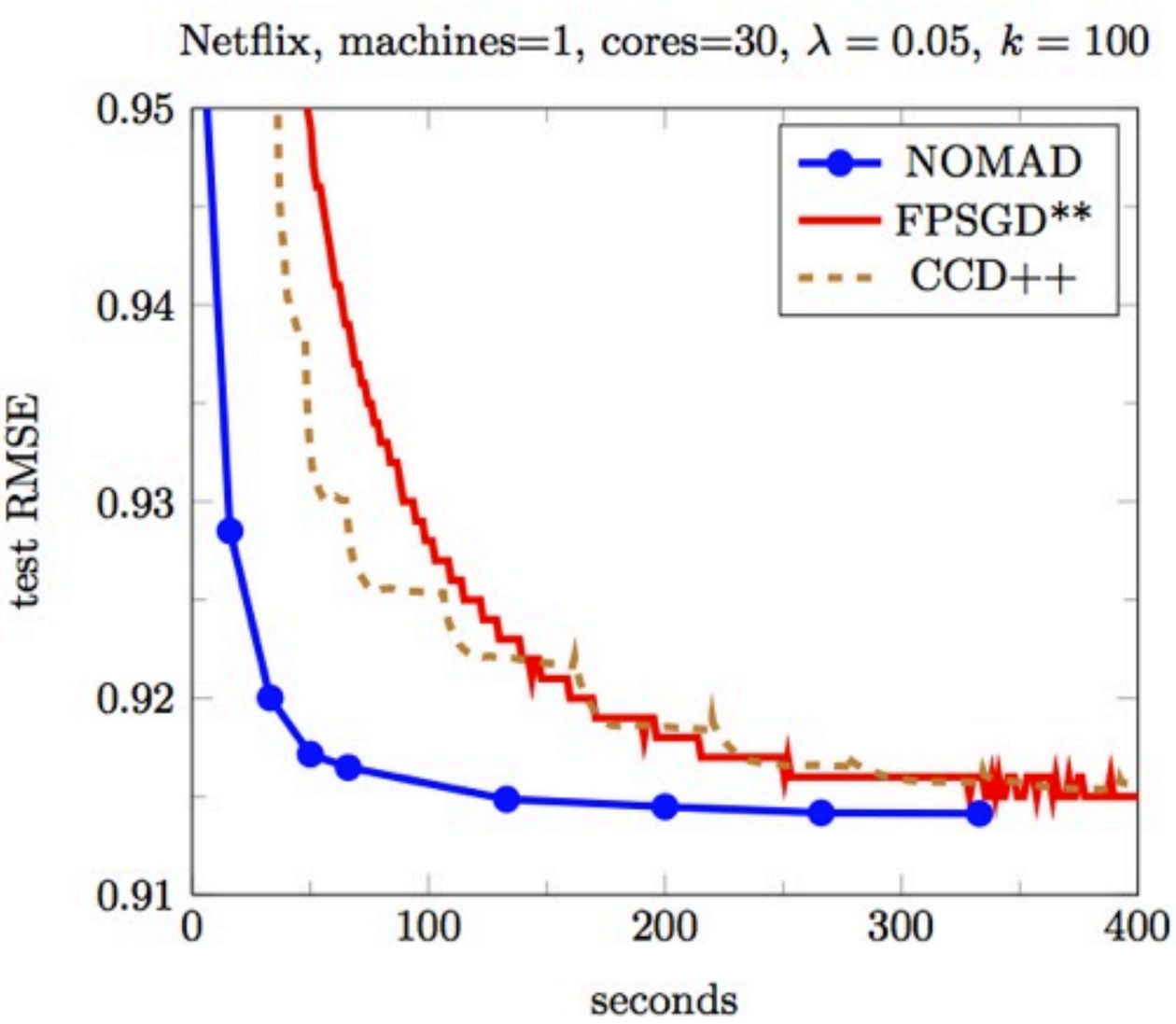
- Average time for one sweep:

$$O(|\Omega|k/p)$$

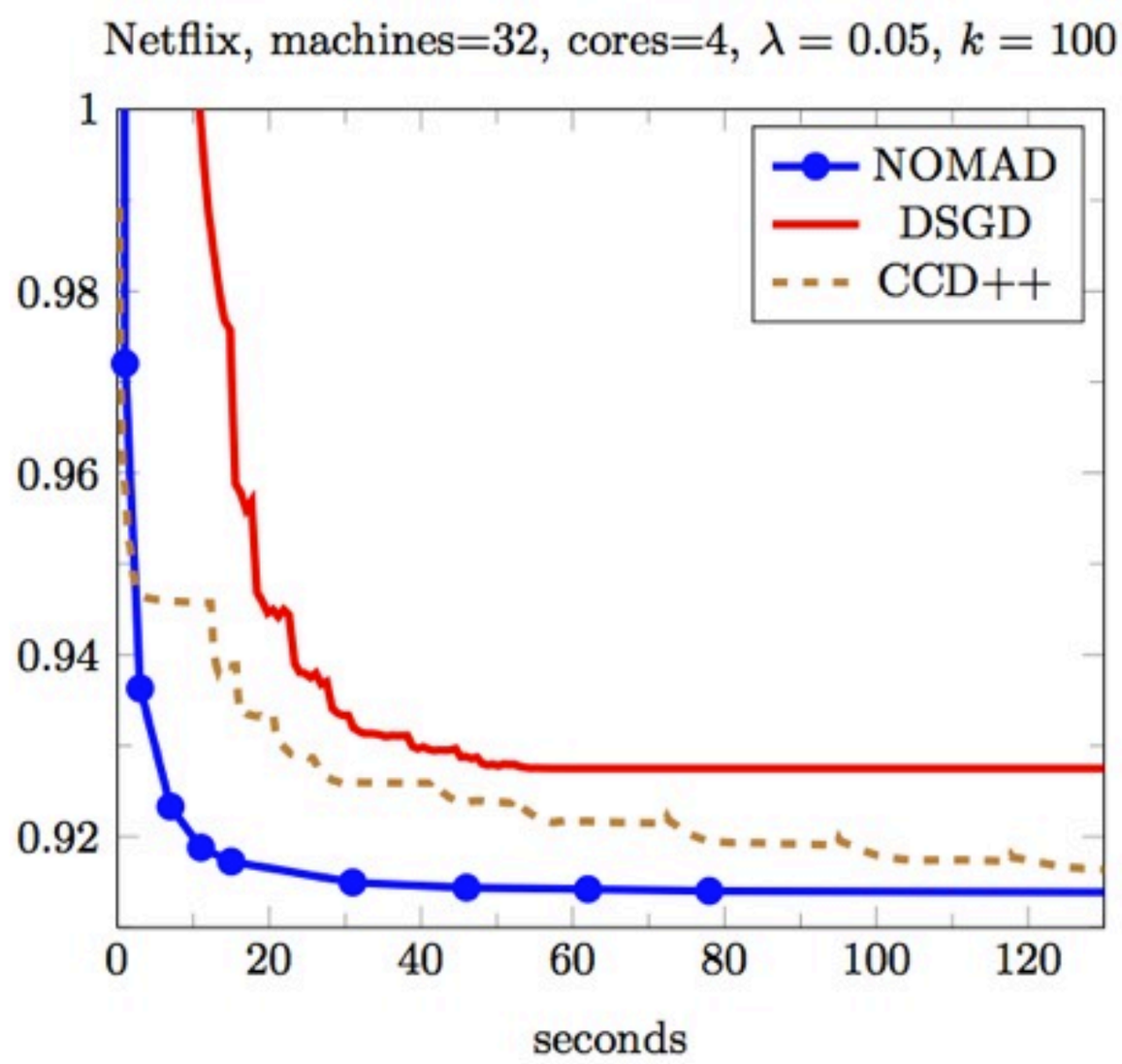
Results on Applications

Recommender Systems

Netflix dataset: 2,649,429 users, 17,770 movies, ~100M ratings



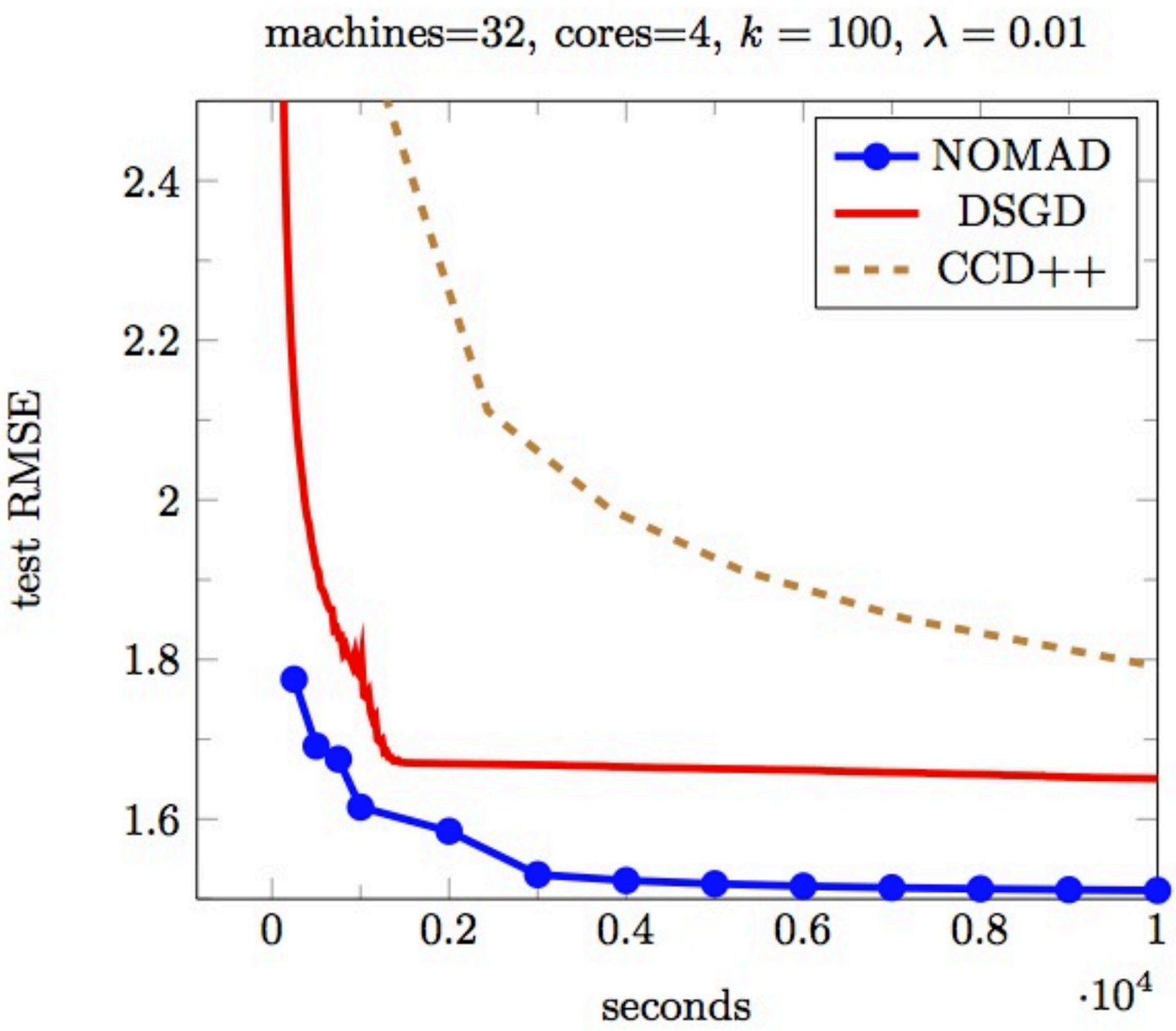
Multicore



Distributed

Recommender Systems

Synthetic dataset: ~85M users, 17,770 items, ~8.5B observations

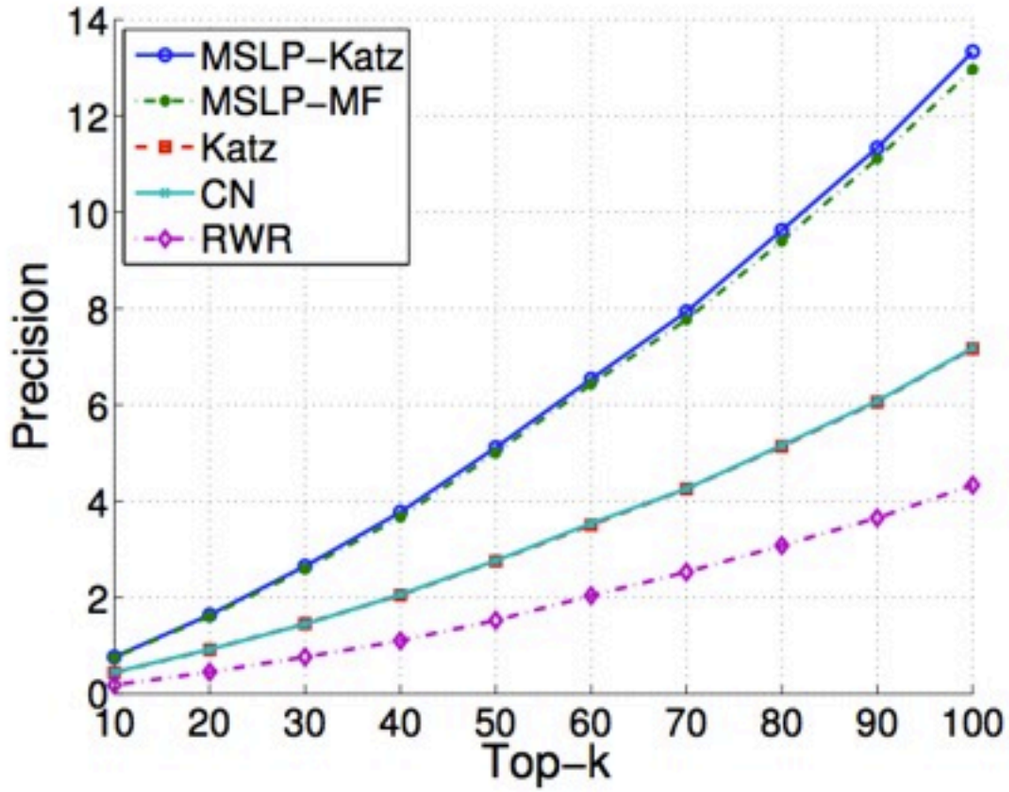


Link Prediction

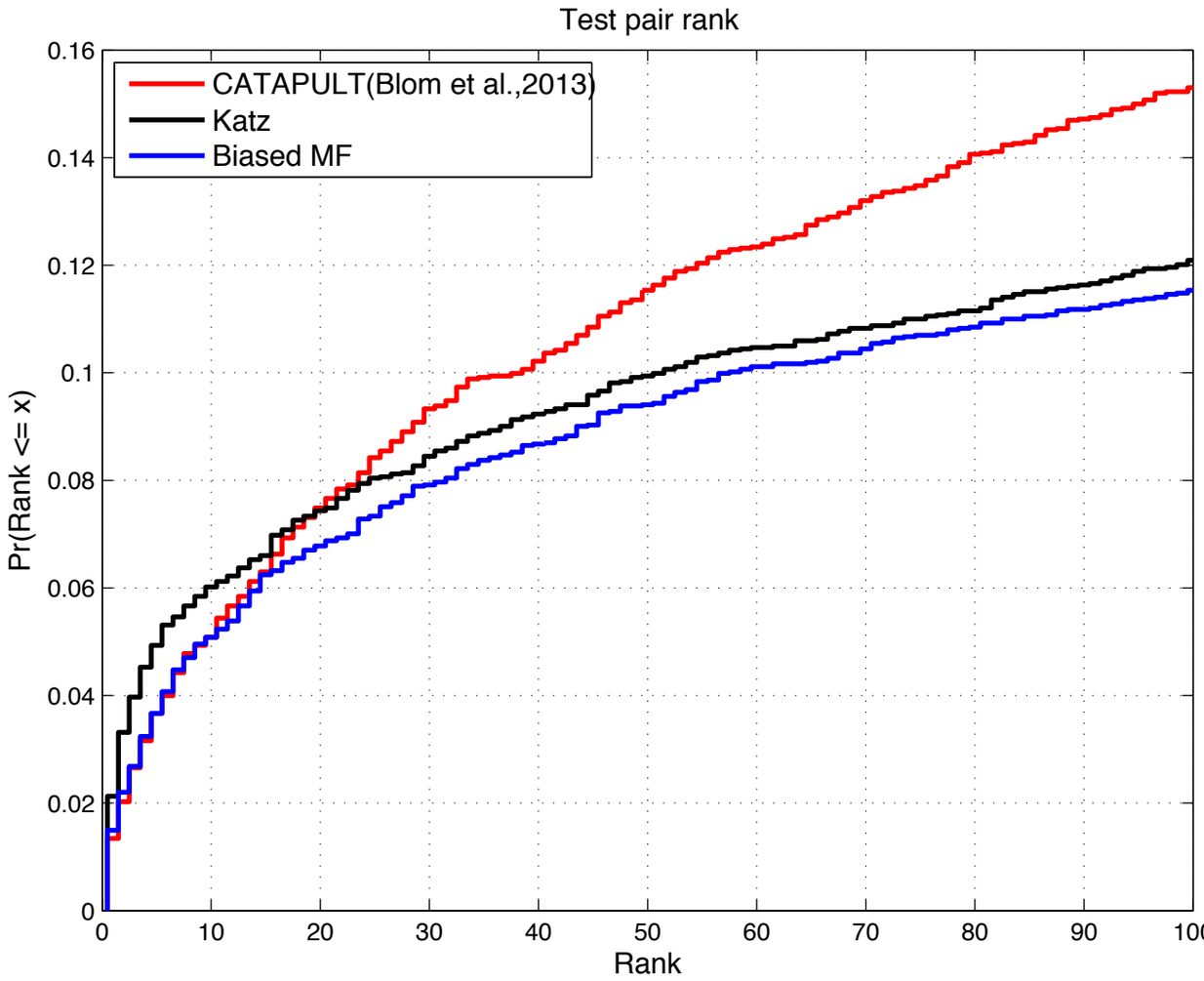
- Flickr dataset: 1.9M users & 42M links.
- Test set: sampled 5K users.

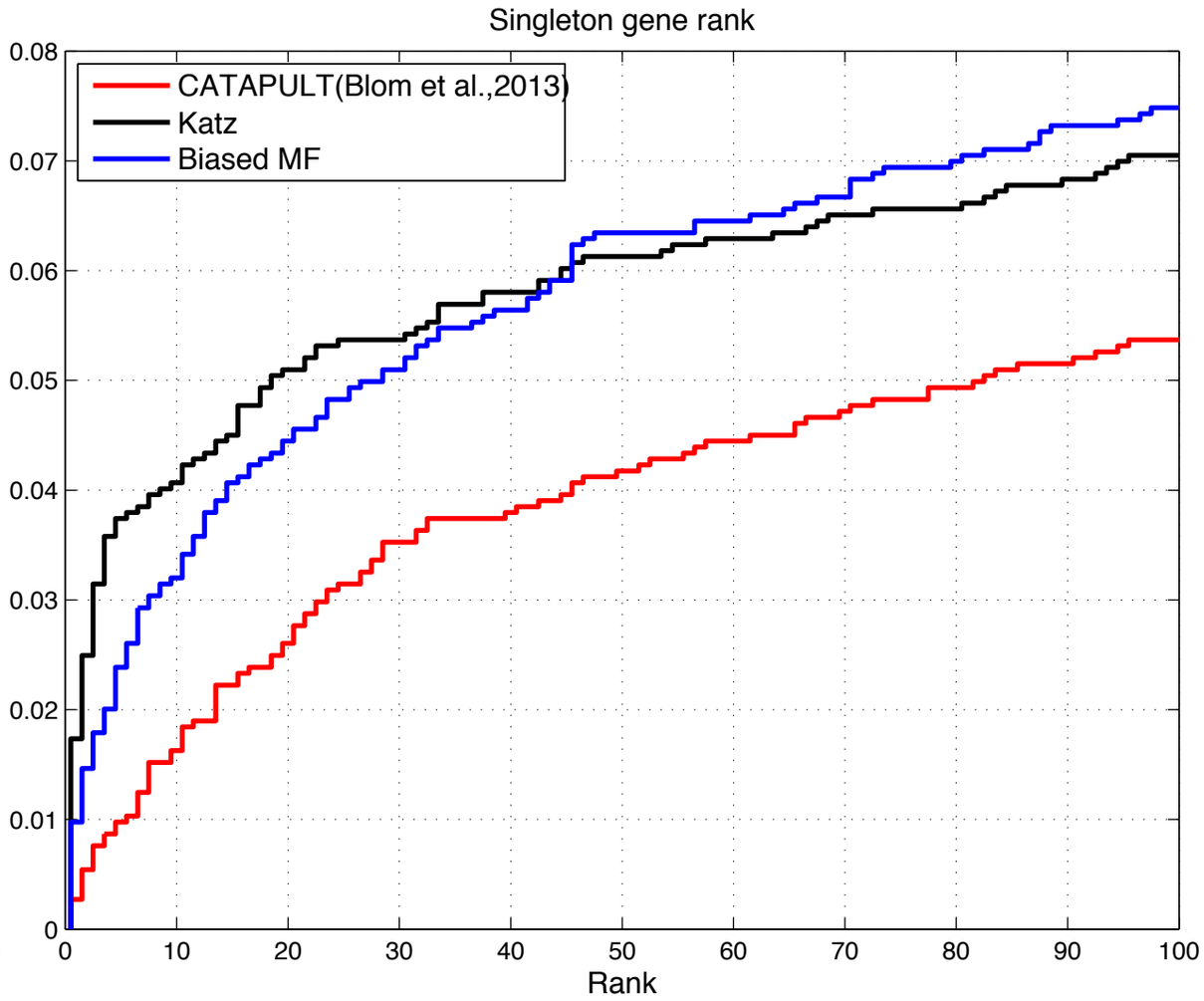
Method	Prec	AUC
Preferential Attachment	1.02	0.6981
Common Neighbor	7.08	0.8649
Adamic-Adar	7.29	0.8758
Random Walk w/ Restarts	5.49	0.7872
Logistic Regression*	2.54	0.7115
Katz	7.17	0.8429
MSLP-Katz	**13.34**	**0.8924**
MF	12.05	0.9078
MSLP-MF	**13.07**	**0.9145**

*using network-based features.



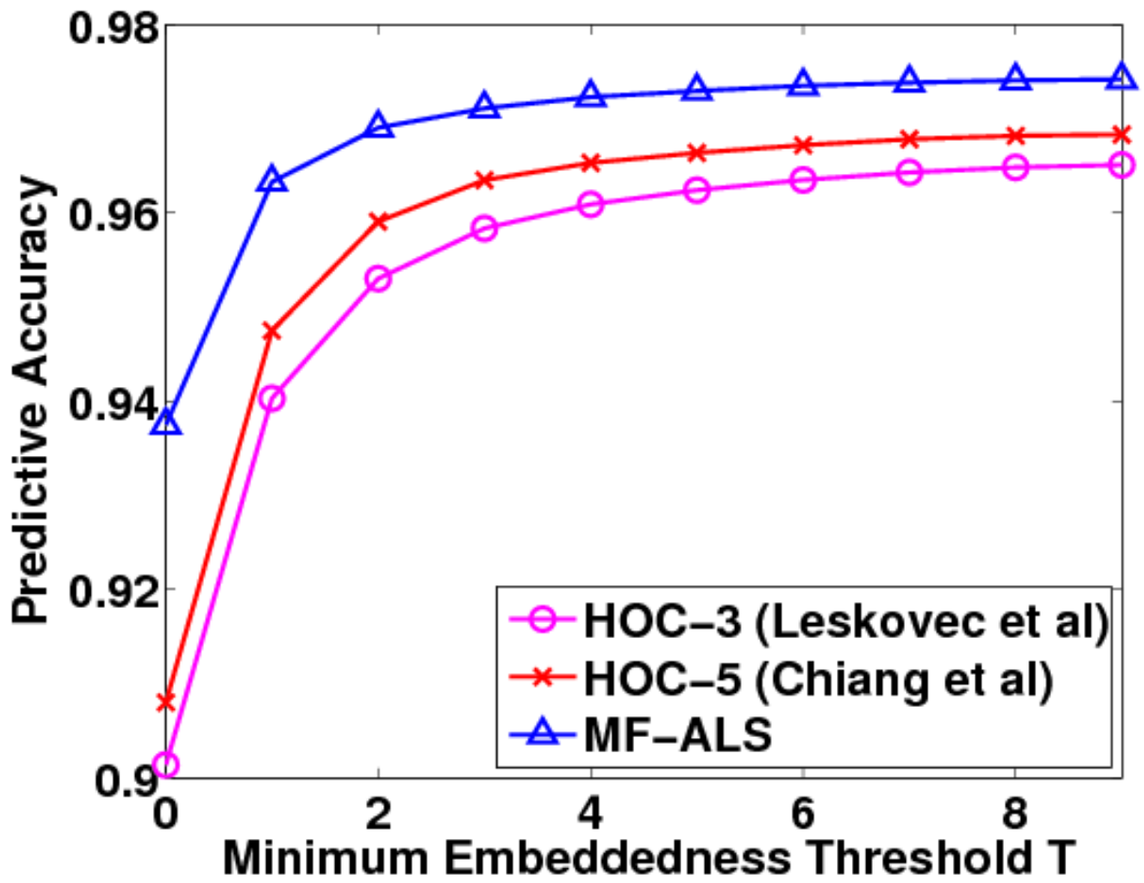
Predicting Gene-Disease Links





Sign Prediction

- Epinions dataset (+ve & -ve reviews):
  - 131K nodes, 840K edges, 15% edges negative.
  - MF-ALS is faster and achieves higher accuracy.



	Time
HOC-3	560.64 s.
HOC-4	6156.8 s.
HOC-5	$> 10,000$ s.
MF-ALS	**28.67 s.**

MF-ALS takes 455 secs
on network with
1.1M nodes & 120M edges

Conclusions

- Rapid growth of unstructured data demands scalable machine learning solutions for analysis
- Machine learning problems arising in network analysis can be cast in the matrix completion framework
- Our proposed asynchronous distributed algorithm NOMAD outperforms state-of-the-art matrix completion solvers
- Beyond Matrix Completion: Asynchronous distributed framework for solving machine learning problems