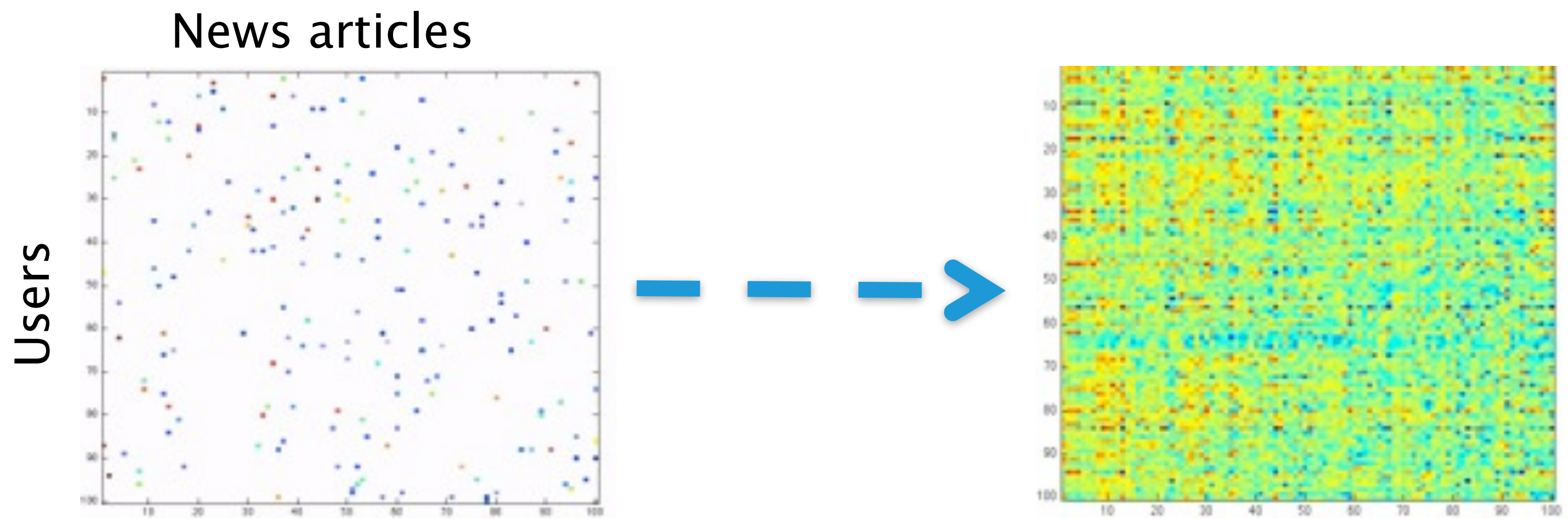
Matrix Completion

- References
  - R. Meka, P. Jain, I. Dhilon: Matrix Completion from Power-law distributed samples, NIPS 2009
  - N. Ruchansky, M. Crovella, E. Terzi: Matrix Completion with Queries, KDD 2015

The recommender-system challenge



Want to predict **all** preferences, but
we know only 1% of the entries!

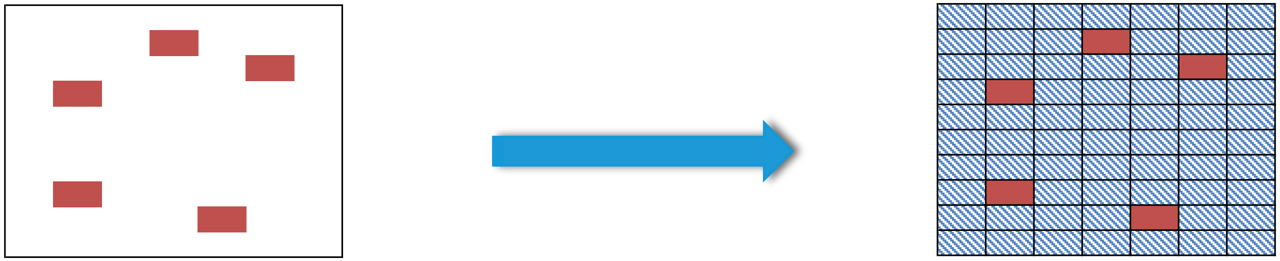
Matrix completion

Input:

Partially-observed matrix

Goal:

Find a **low-error completion** of the remaining entries



Not any input can be recovered well, and there exist some conditions **the true matrix** must meet.

How can we reconstruct a matrix with only 1% of the entries?

Statistical methods for matrix completion

Assumptions:

1. Underlying **true matrix** T is low rank

Find an estimate $\widehat{T}$ by solving the optimization problem:

$$minimize \quad rank(\widehat{T})$$

$$subject\ to\ T_{ij} = \widehat{T}_{ij} \text{ for } (i,j) \in \Omega$$

Observed set of entries

This problem is **NP-hard**.

Statistical methods for matrix completion

- Click to edit Master text styles

Assumptions:

1. Underlying **true matrix** T is low rank

Find an estimate $\widehat{T}$ by solving the optimization problem:

$$minimize \quad rank(\widehat{T})$$

$$subject\ to\ T_{ij} = \widehat{T}_{ij} \text{ for } (i,j) \in \Omega$$

Observed set of entries

This problem is **NP-hard**.

Alternating least squares

Wen et.al. 2012, Jain et al. 2013...

Assumptions:

1. Underlying T is low rank
2. Known rank r
3. The matrix can be written as a product $T = XY^T$
   1. X is of size nxr
   2. Y is of size mxr

Problem:

$$\min_{X,Y,\widehat{T}} \frac{1}{2}\|XY - \widehat{T}\|_F^2$$

$$subject\ to\ T_{ij} = \widehat{T}_{ij}\ (i,j) \in \Omega$$

Characteristics

Such optimization approaches **output an estimate of the whole matrix** on any input – any size Ω.

Compute error on observed entries:

$$\widehat{T}_\Omega \text{ vs. } T_\Omega$$

What if we want **very small error?**
How may entries Ω do we need to have?

Information-theoretic lower bound

When Ω is **sampled at random** the number of entries needs to be at least:

$$rn \log n$$

Coupon Collector argument: given **n** coupons, how many times do you need to draw (with replacement) to collect at least **r** of each?

Candes, Plan 2009

In practice

Most matrices have **less** entries than the lower bound.

Netflix: 480189 x 17770, winning solution was rank=40

$|\Omega|$ = 9.91e+07

$rn \log n$ = 2.51e+08

Not enough entries.

Since the lower bound relies on random sampling, we can **sample unknown entries randomly**. (e.g. Netflix survey)

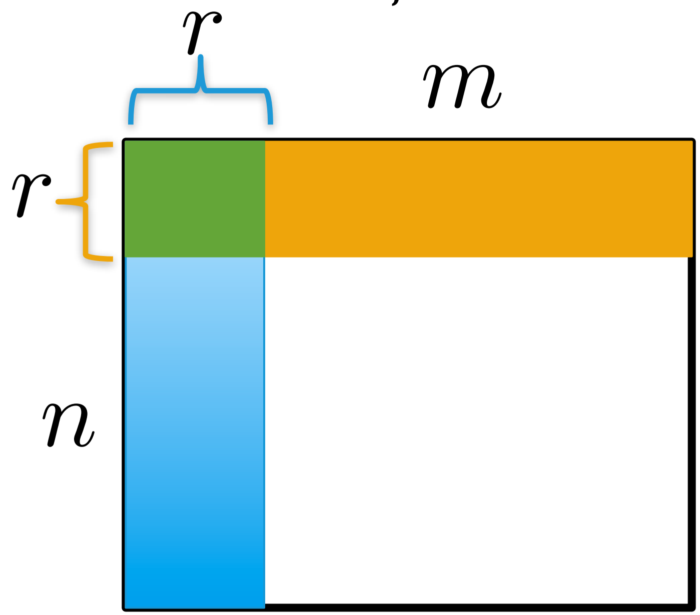
This would require about **151,000,000** samples!

Algebraic lower bound

In an $n \times m$ matrix of rank r, there are

$$r(n+m-r)$$

degrees of freedom. Once this many carefully-chosen entries are fixed, there should be a unique recovery.

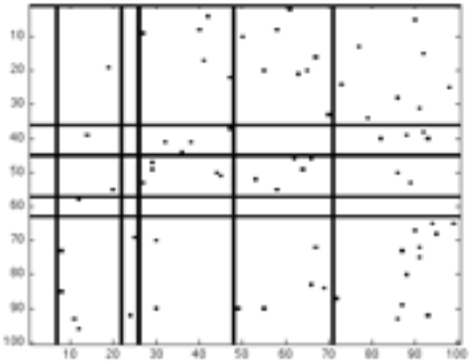


$$nr + mr - r^2 = r(n+m-r)$$

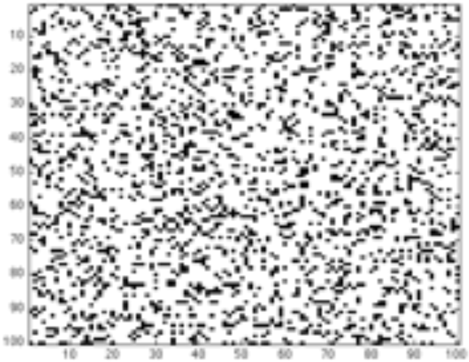
Targeted than random?

A **targeted** addition of entries (versus random), would only require:

$$r(n+m-r) \ll rn\log(n)$$



Carefully sampled



Randomly sampled

Given a non-completable partially-observed matrix, what is the minimum set of unobserved entries that need to be queried so that the error is low?

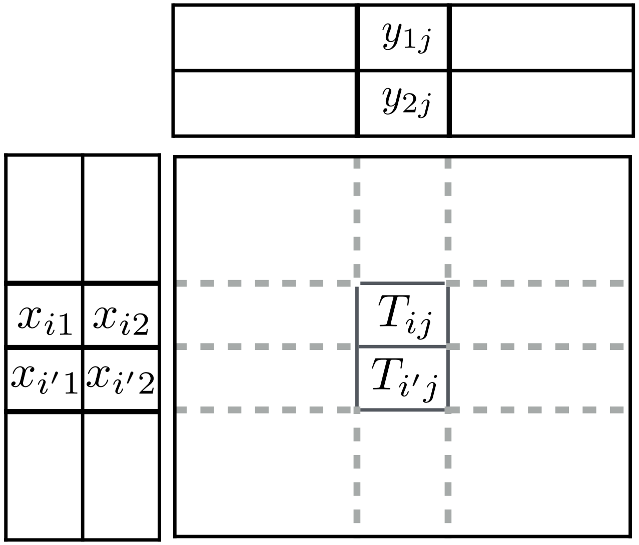
Can we do that by querying entries as close as possible to

$$r(n+m-r) \ll rn\log(n)\,?$$

YES!

Methodology overview

- $T = XY^T$
- Matrix completion as a sequence of linear systems that compute rows of X and columns of Y



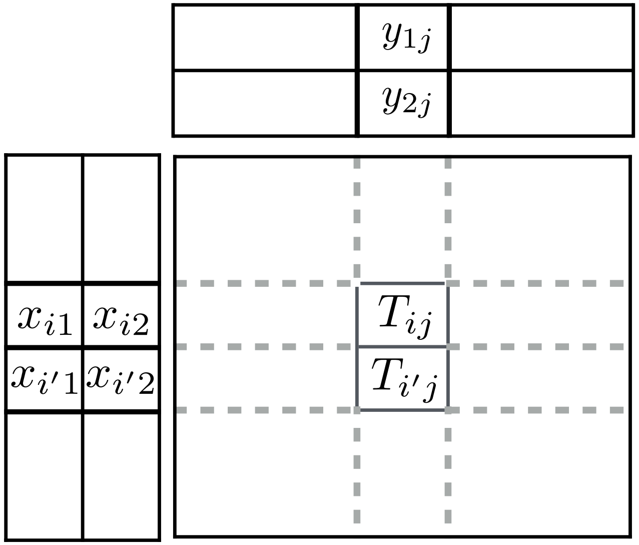
r=2

$$\begin{aligned} T_{ij} &= x_{i1}y_{1j} + x_{i2}y_{2j} \\ T_{i'j} &= x_{i'1}y_{1j} + x_{i'2}y_{2j} \end{aligned}$$

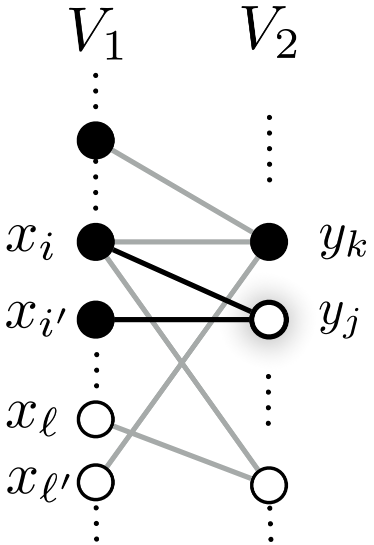
Meka et.al. 2009

Methodology overview

$$
\begin{aligned}
T_{ij} &= x_{i1}y_{1j} + x_{i2}y_{2j} \\
T_{i'j} &= x_{i'1}y_{1j} + x_{i'2}y_{2j}
\end{aligned}
$$



r=2

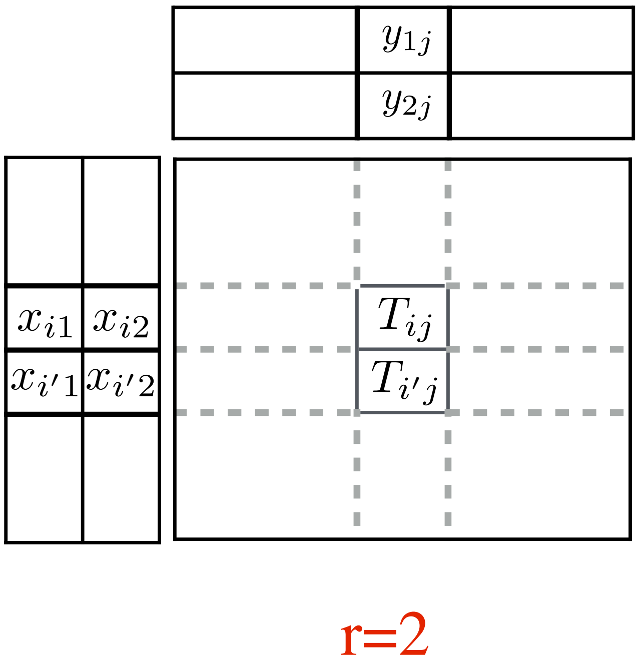


Information-propagation view

Methodology overview

$$\begin{aligned} T_{ij} &= x_{i1}y_{1j} + x_{i2}y_{2j} \\ T_{i'j} &= x_{i'1}y_{1j} + x_{i'2}y_{2j} \end{aligned}$$

$$A_x y = t.$$



- Solvable ☺
- Incomplete ☹
- Unstable ☹

- Incomplete and unstable systems are resolved through **queries**

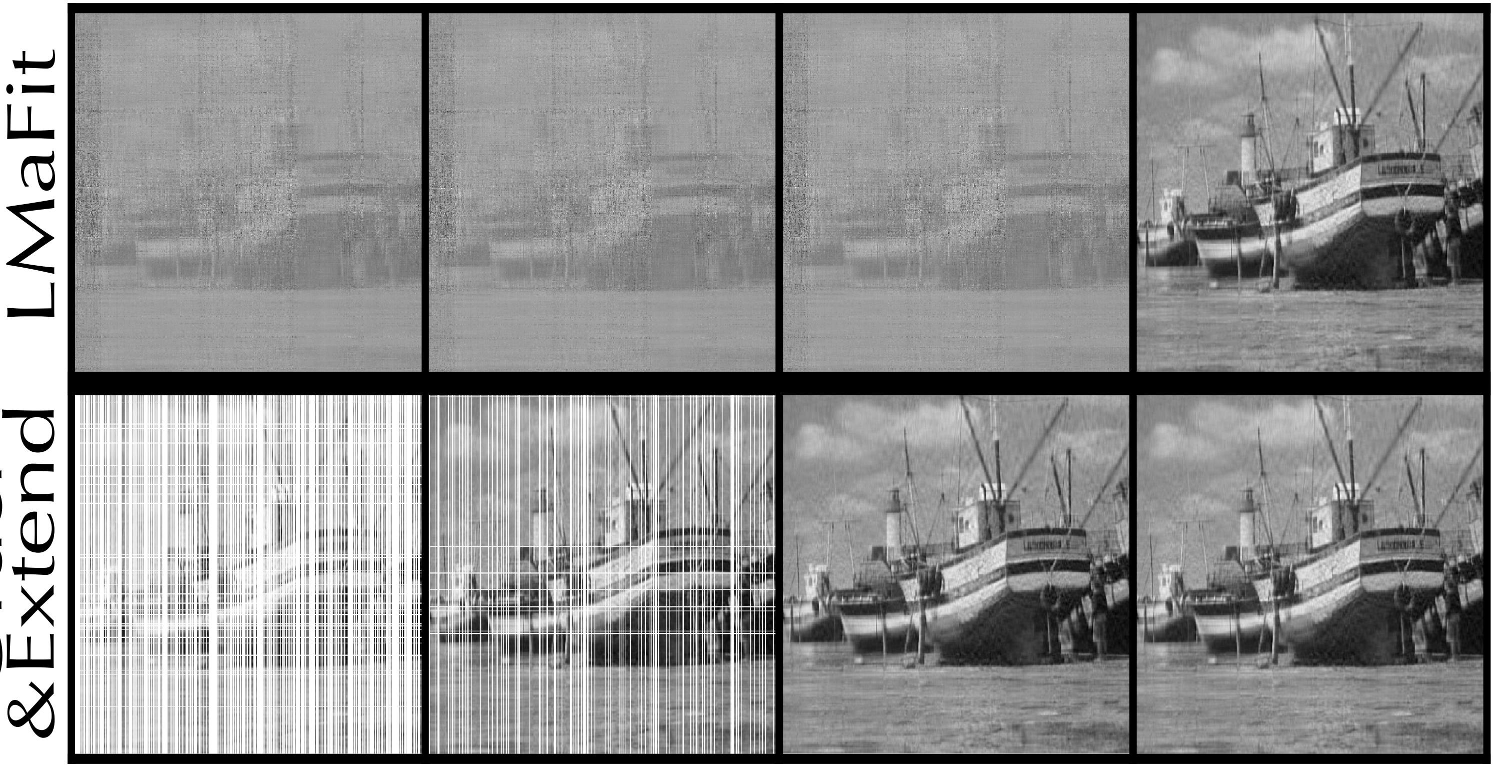
The **Order&Extend** algorithm

- **Order**: Find an ordering π that minimizes the number of incomplete systems
- **Extend:** Solve the linear systems imposed by π upon encountering
  - an *incomplete* system: ask directly the required entries from **T**
  - an *unstable* system: judiciously pick the entries from **T**

- Running time: $O(n+m)$

[Ruchansky, Crovella, T. 2015]

Partial completion by Order (without extend)



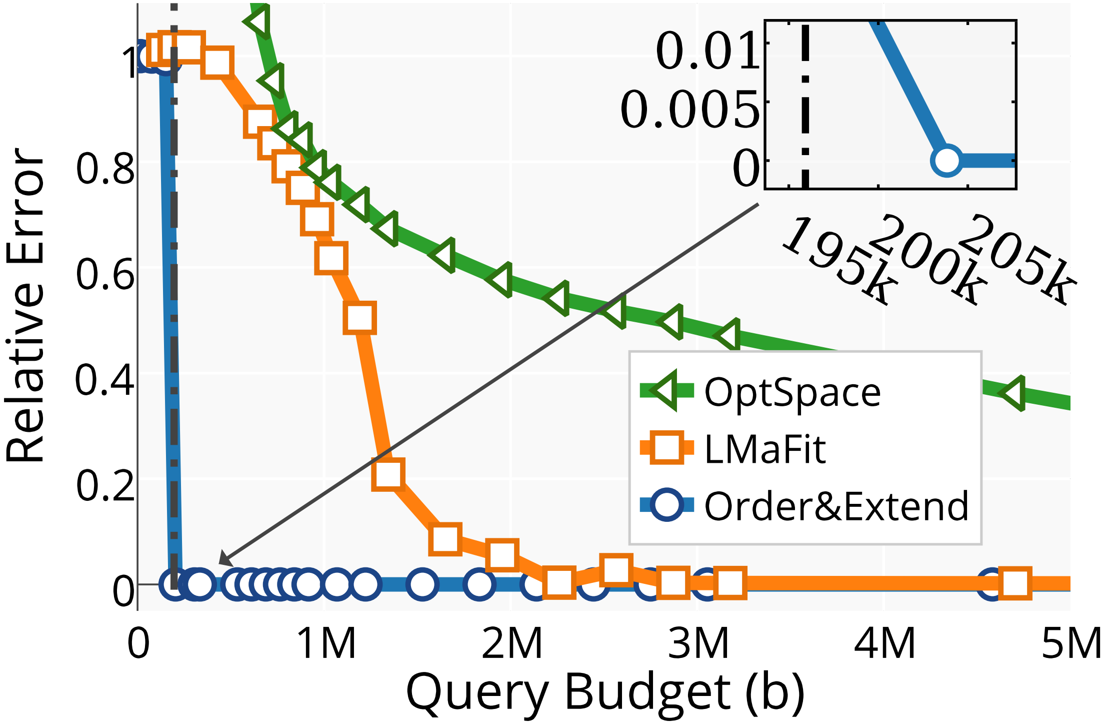
Experiments with matrices of rank r

MovieLens

User – Movie ratings

~6K x 4K with 5% known entries

Dense submatrix: ~5K x 3.5K



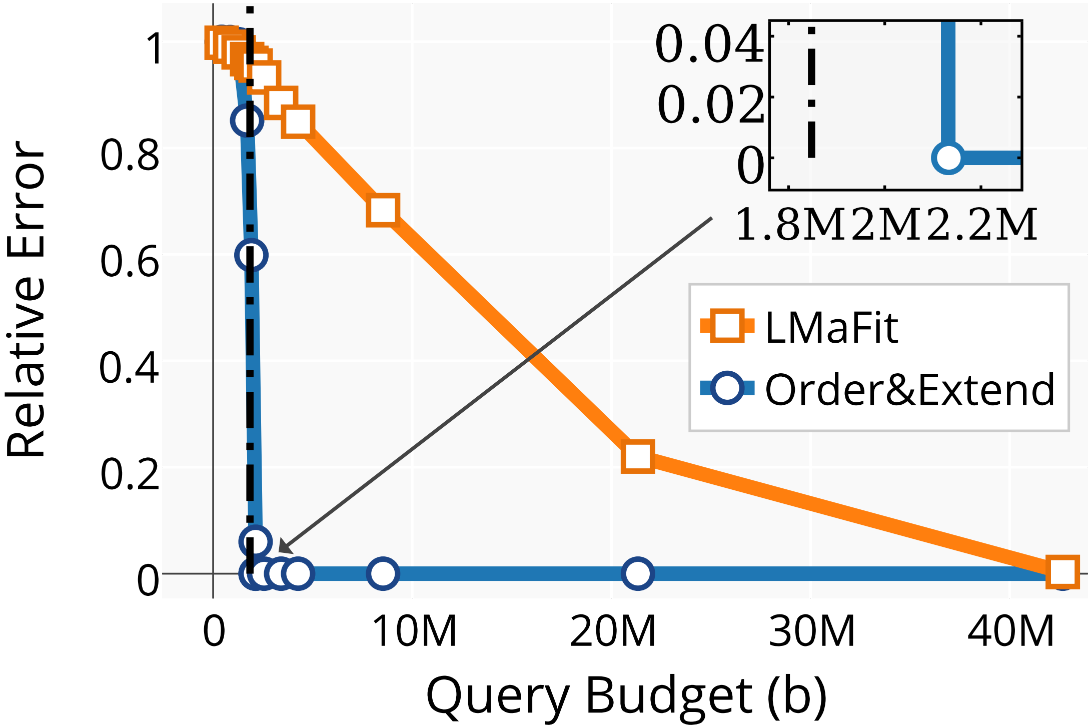
Experiments with matrices of rank r

Netflix

User – Movie ratings

~480K x 18K with 1% known entries

Dense submatrix: ~48K x 9K

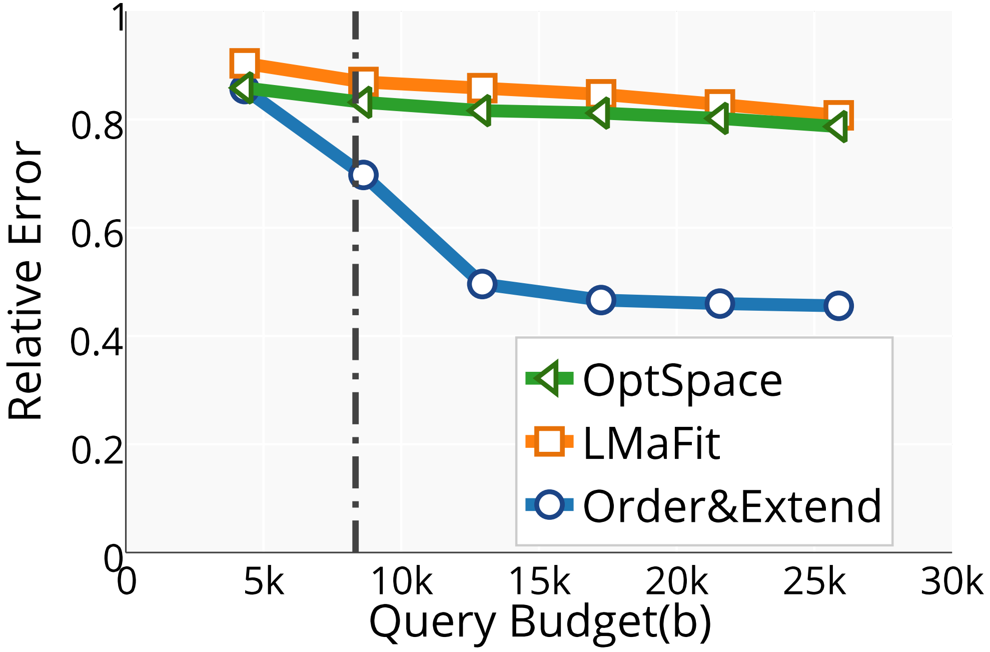


Experiments with noisy matrices

Traffic

Volume of traffic for source-destination pairs

~2Kx150, fully observed

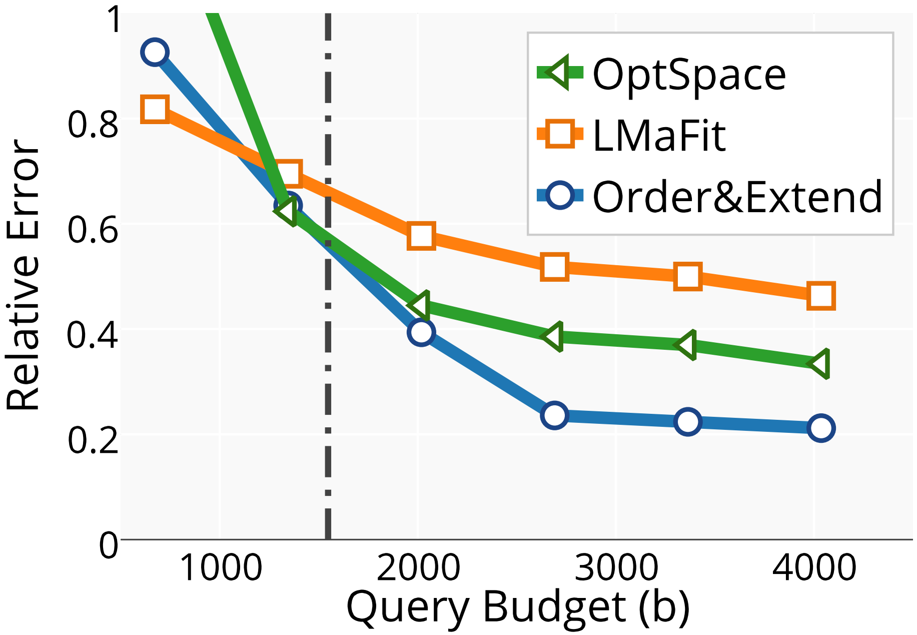


Experiments with noisy matrices

Lattency

Ping delay between hosts

~1Kx20


OptSpace
LMaFit
Order&Extend
Relative Error
Query Budget (b)