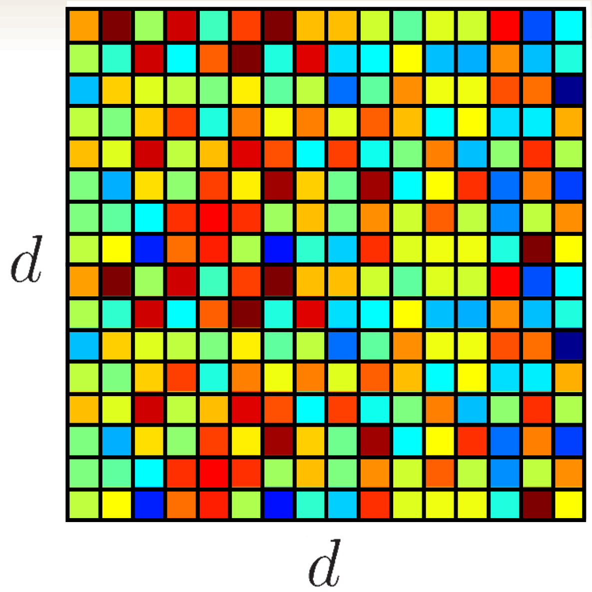
Recovery of low-rank matrices from incomplete observations

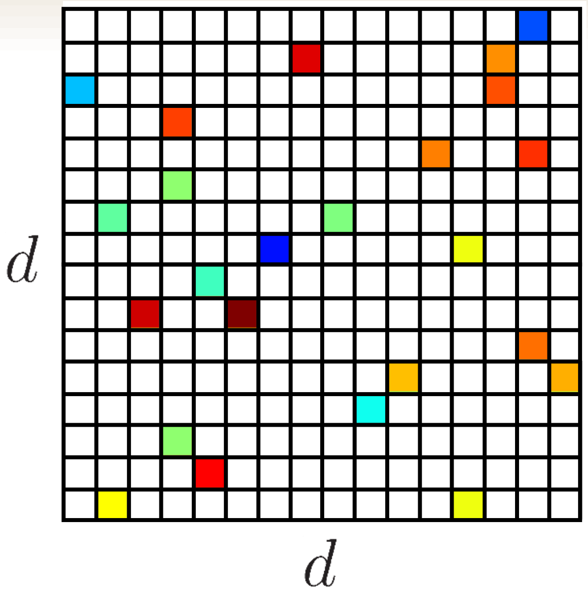
Mark A. Davenport

Georgia Institute of Technology
School of Electrical and Computer Engineering

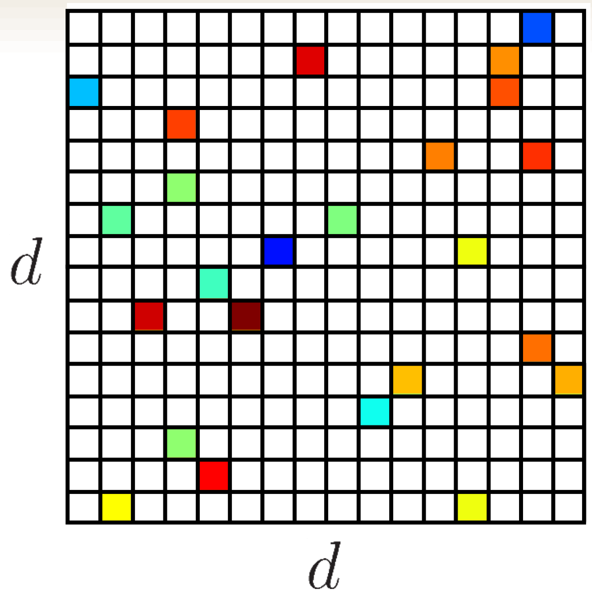
Matrix completion



Matrix completion

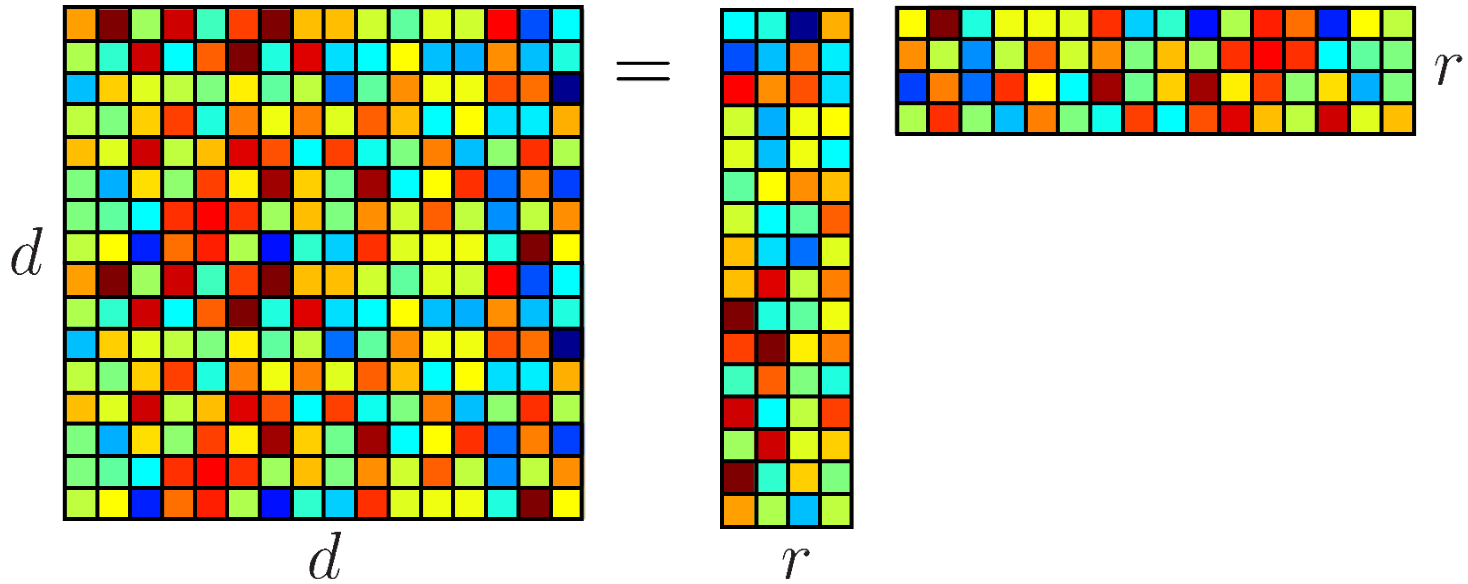


Matrix completion



- When is it possible to recover the original matrix?
- How can we do this efficiently?
- How many samples will we need?

Low-rank matrices



Singular value decomposition:

$$M = U\Sigma V^* \quad \Rightarrow \quad \approx dr \ll d^2$$

degrees of freedom

Applications

- Recommendation systems
- Recovery of incomplete survey data
- Analysis of voting data
- Analysis of student response data
- Localization/multidimensional scaling
- Blind deconvolution
- Phase recovery
- Quantum state tomography
- …

Low-rank matrix recovery

Given:

- a $d \times d$ matrix M of rank r
- samples of M on the set Ω : $Y = M_\Omega$

How can we recover M ?

Low-rank matrix recovery

Given:

- a $d \times d$ matrix M of rank r
- samples of M on the set Ω : $Y = M_\Omega$

How can we recover M ?

$$\widehat{M} = \arg\inf_{X : X_\Omega = Y} \operatorname{rank}(X)$$

Low-rank matrix recovery

Given:

- a $d \times d$ matrix M of rank r
- samples of M on the set Ω : $Y = M_\Omega$

How can we recover M ?

$$\widehat{M} = \arg\inf_{X : X_\Omega = Y} \operatorname{rank}(X)$$

Can we replace this with something computationally feasible?

Nuclear norm minimization

Convex relaxation!

Replace $\operatorname{rank}(X)$ with $\|X\|_* = \sum_{j=1}^{d} \sigma_j$

$$\boxed{\widehat{M} = \underset{X : X_\Omega = Y}{\arg\inf} \|X\|_*}$$

If $|\Omega| = O(rd \log d)$, under certain natural assumptions, this procedure can recover M exactly!

[Candès, Recht, Tao, Plan, Gross, Keshavan, Montenari, Oh, ...]

Matrix completion in practice

- Noise

$$Y = (M + Z)_\Omega$$

- ***Quantization***
  - Netflix: Ratings are integers between 1 and 5
  - Survey responses: True/False, Yes/No, Agree/Disagree
  - Voting data: Yea/Nay
  - Quantum state tomography: Binary outcomes

Matrix completion in practice

- Noise

$$Y = (M + Z)_\Omega$$

- ***Quantization***
  - Netflix: Ratings are integers between 1 and 5
  - Survey responses: True/False, Yes/No, Agree/Disagree
  - Voting data: Yea/Nay
  - Quantum state tomography: Binary outcomes

Extreme quantization *destroys low-rank structure*

1-bit matrix completion

Extreme case

$$Y = \mathrm{sign}(M_\Omega)$$

Claim: Recovering M from Y is impossible!

1-bit matrix completion

Extreme case

$$Y = \mathrm{sign}(M_\Omega)$$

Claim: Recovering M from Y is impossible!

$$M = \begin{bmatrix} \lambda & \lambda & \lambda & \lambda \\ \lambda & \lambda & \lambda & \lambda \\ \lambda & \lambda & \lambda & \lambda \\ \lambda & \lambda & \lambda & \lambda \end{bmatrix}$$

No matter how many samples we obtain, all we can learn is whether $\lambda > 0$ or $\lambda < 0$

Is there any hope?

If we consider a noisy version of the problem, recovery becomes feasible!

$$Y = \mathrm{sign}(M_\Omega + Z_\Omega)$$

$$M+Z = \begin{bmatrix} \lambda + Z_{1,1} & \lambda + Z_{1,2} & \lambda + Z_{1,3} & \lambda + Z_{1,4} \\ \lambda + Z_{2,1} & \lambda + Z_{2,2} & \lambda + Z_{2,3} & \lambda + Z_{2,4} \\ \lambda + Z_{3,1} & \lambda + Z_{3,2} & \lambda + Z_{3,3} & \lambda + Z_{3,4} \\ \lambda + Z_{4,1} & \lambda + Z_{4,2} & \lambda + Z_{4,3} & \lambda + Z_{4,4} \end{bmatrix}$$

Fraction of positive/negative observations tells us something about λ

Example of the power of *dithering*

Observation model

For $(i,j) \in \Omega$ we observe

$$Y_{i,j} = \begin{cases} +1 & \text{with probability } f(M_{i,j}) \\ -1 & \text{with probability } 1 - f(M_{i,j}) \end{cases}$$

If f behaves like a CDF, then this is equivalent to

$$Y_{i,j} = \operatorname{sign}(M_{i,j} + Z_{i,j})$$

where $Z_{i,j}$ is drawn according to a suitable distribution

We will assume that Ω is drawn uniformly at random

Examples

- Logistic regression / Logistic noise

$$f(x) = \frac{e^x}{1 + e^x}$$

$Z_{i,j} \sim$ logistic distribution

- Probit regression / Gaussian noise

$$f(x) = \Phi(x/\sigma)$$

$$Z_{i,j} \sim \mathcal{N}(0, \sigma^2)$$

Maximum likelihood estimation

Log-likelihood function:

$$F(X) = \sum_{(i,j)\in\Omega_+} \log(f(X_{i,j})) + \sum_{(i,j)\in\Omega_-} \log(1 - f(X_{i,j}))$$

$$\boxed{\begin{aligned} \widehat{M} = \arg\max_X \, & F(X) \\ \text{s.t.} \quad & \frac{1}{d\alpha}\|X\|_* \leq \sqrt{r} \\ & \|X\|_\infty \leq \alpha \end{aligned}}$$

Recovery of the matrix

***Theorem** (Upper bound achieved by convex ML estimator)*

Assume that $\frac{1}{d\alpha}\|M\|_* \leq \sqrt{r}$ and $\|M\|_\infty \leq \alpha$. If Ω is chosen at random with $\mathbb{E}|\Omega| = m > d\log d$, then with high probability

$$\frac{1}{d^2}\|\widehat{M} - M\|_F^2 \leq C\alpha L_\alpha \beta_\alpha \sqrt{\frac{rd}{m}}$$

where

$$L_\alpha := \sup_{|x|\leq\alpha} \frac{|f'(x)|}{f(x)(1 - f(x))} \qquad \beta_\alpha := \sup_{|x|\leq\alpha} \frac{f(x)(1 - f(x))}{(f'(x))^2}$$

Probit model

$$L_\alpha \approx \frac{\frac{\alpha}{\sigma}+1}{\sigma} \qquad \beta_\alpha \approx \sigma^2 e^{\alpha^2/2\sigma^2}$$

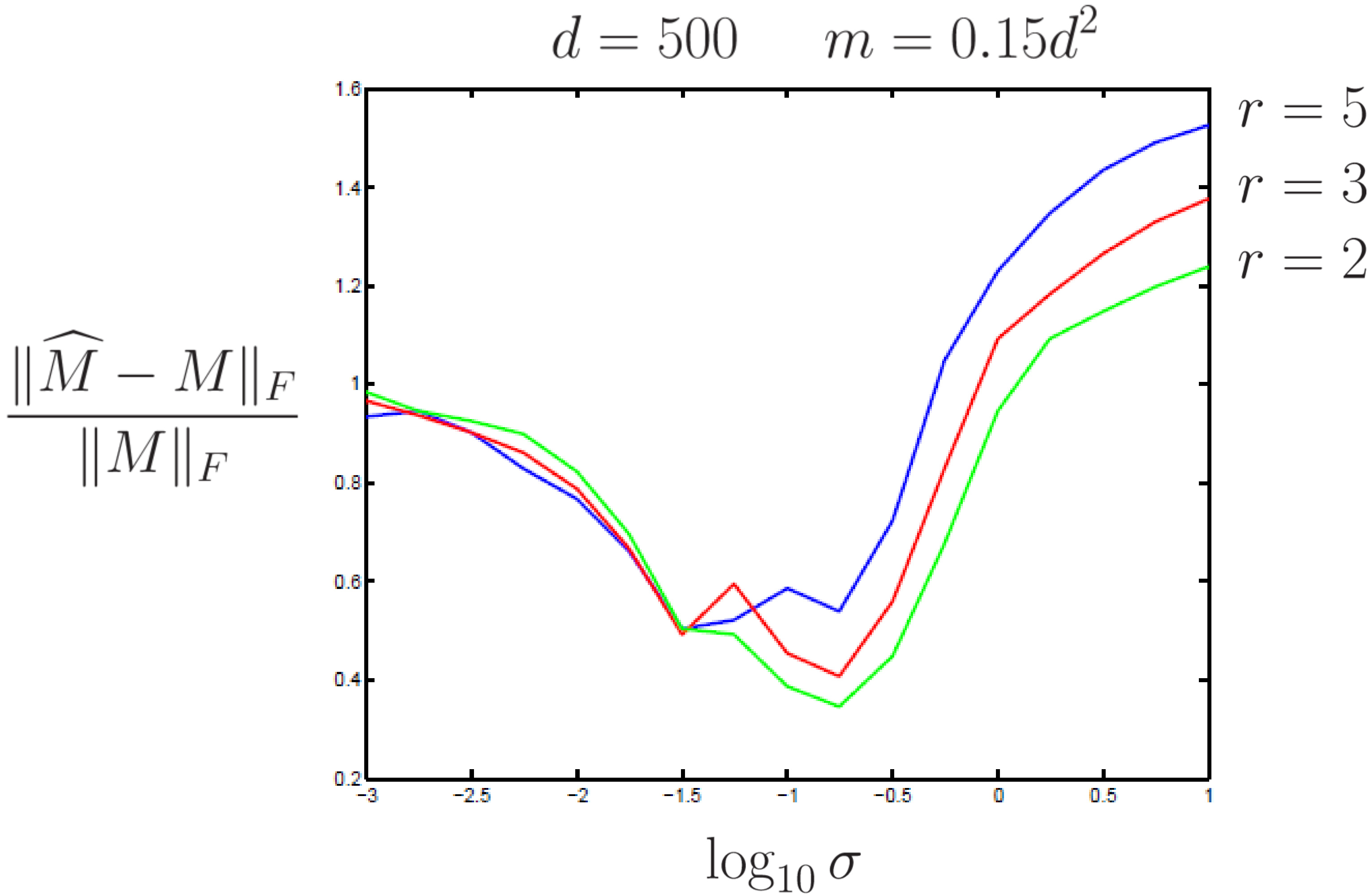
***Theorem** (Upper bound achieved by convex ML estimator)*

$$\frac{1}{d^2}\|\widehat{M}-M\|_F^2 \leq C\left(\frac{\alpha}{\sigma}+1\right)e^{\alpha^2/2\sigma^2}\sigma\alpha\sqrt{\frac{rd}{m}}$$

For any fixed α , optimal bound is achieved by $\sigma \approx 1.3\alpha$, in which case the bound reduces to

$$\frac{1}{d^2}\|\widehat{M}-M\|_F^2 \leq 3.1C\alpha^2\sqrt{\frac{rd}{m}}$$

Synthetic simulations



MovieLens data set

- 100,000 movie ratings on a scale from 1 to 5
- Convert to binary outcomes by comparing each rating to the average rating in the data set
- Evaluate by checking if we predict the correct sign
- Training on 95,000 ratings and testing on remainder
  - “standard” matrix completion: 60% accuracy

1: 64%	2: 56%	3: 44%	4: 65%	5: 74%

  - 1-bit matrix completion: 73% accuracy

1: 79%	2: 73%	3: 58%	4: 75%	5: 89%

Conclusions

- 1-bit matrix completion is hard!
- What did you really expect?
- Sometimes 1-bit is all we can get…
- We have algorithms that are near optimal
- Open questions
  - Simpler/better/faster/stronger algorithms?
  - More general likelihood models?
  - Incorporating dynamics?

Thank You!