One-bit matrix completion

Yaniv Plan

University of Michigan

Joint work with

(a) Mark Davenport

(b) Ewout van den Berg

(c) Mary Wootters

Low-rank matrices

Example: Netflix matrix

$$\begin{pmatrix} M_{1,1} & M_{1,2} & M_{1,3} & M_{1,4} \\ M_{2,1} & M_{2,2} & M_{2,3} & M_{2,4} \\ M_{3,1} & M_{3,2} & M_{3,3} & M_{3,4} \\ M_{4,1} & M_{4,2} & M_{4,3} & M_{4,4} \end{pmatrix}, \quad M_{i,j} = \text{How much user } i \text{ likes movie } j$$

Low-rank matrices

Example: Netflix matrix

$$\begin{pmatrix} M_{1,1} & M_{1,2} & M_{1,3} & M_{1,4} \\ M_{2,1} & M_{2,2} & M_{2,3} & M_{2,4} \\ M_{3,1} & M_{3,2} & M_{3,3} & M_{3,4} \\ M_{4,1} & M_{4,2} & M_{4,3} & M_{4,4} \end{pmatrix}, \quad M_{i,j} = \text{How much user } i \text{ likes movie } j$$

Rank-1 model:

- a_j = Amount of action in movie j
- x_i = How much user i likes action

$$M_{i,j} = x_i \cdot a_j$$

Low-rank matrices

Example: Netflix matrix

$$\begin{pmatrix} M_{1,1} & M_{1,2} & M_{1,3} & M_{1,4} \\ M_{2,1} & M_{2,2} & M_{2,3} & M_{2,4} \\ M_{3,1} & M_{3,2} & M_{3,3} & M_{3,4} \\ M_{4,1} & M_{4,2} & M_{4,3} & M_{4,4} \end{pmatrix}, \quad M_{i,j} = \text{How much user } i \text{ likes movie } j$$

Rank-1 model:

- a_j = Amount of action in movie j
- x_i = How much user i likes action

$$M_{i,j} = x_i \cdot a_j$$

Rank-2 model:

- b_j = Amount of comedy in movie j
- y_i = How much user i likes comedy

$$M_{i,j} = x_i \cdot a_j + y_i \cdot b_j$$

Low-rank assumption

- The number of characteristics that determine user preferences should be smaller than the number of movies or users.

- M should depend *linearly* on the characteristics:

$$M_{i,j} \neq exp(x_i \cdot a_j).$$

- These characteristics *need not be known*.

Matrix completion

Matrix completion: Completion of $\mathbf{M}$ from a subset of the entries.

$$\begin{pmatrix} ? & M_{1,2} & ? \\ ? & ? & M_{2,3} \\ M_{3,1} & ? & M_{3,3} \\ M_{4,1} & M_{4,2} & ? \end{pmatrix} \xrightarrow{\text{Matrix Completion}} \begin{pmatrix} M_{1,1} & M_{1,2} & M_{1,3} \\ M_{2,1} & M_{2,2} & M_{2,3} \\ M_{3,1} & M_{3,2} & M_{3,3} \\ M_{4,1} & M_{4,2} & M_{4,3} \end{pmatrix}$$

[Incomplete set of researchers: Srebro, Fazel, Candès, Recht, Rennie, Jaakkola, Montanari, Soo, Wainwright, Negahban, Yu, Koltchinskii, Lounici, Tsybakov, Klopp, Cai, Zhou, P.,... 2004-present]

Matrix completion

Matrix completion: Completion of $\mathbf{M}$ from a subset of the entries.

$$\begin{pmatrix} ? & M_{1,2} & ? \\ ? & ? & M_{2,3} \\ M_{3,1} & ? & M_{3,3} \\ M_{4,1} & M_{4,2} & ? \end{pmatrix} \xrightarrow{\text{Matrix Completion}} \begin{pmatrix} M_{1,1} & M_{1,2} & M_{1,3} \\ M_{2,1} & M_{2,2} & M_{2,3} \\ M_{3,1} & M_{3,2} & M_{3,3} \\ M_{4,1} & M_{4,2} & M_{4,3} \end{pmatrix}$$

[Incomplete set of researchers: Srebro, Fazel, Candès, Recht, Rennie, Jaakkola, Montanari, Soo, Wainwright, Negahban, Yu, Koltchinskii, Lounici, Tsybakov, Klopp, Cai, Zhou, P.,... 2004-present]

Imputation: Dealing with incomplete statistical data [Rubin, Little 1987; Daniels, Hogan 2009].

- Case deletion.
- Mean imputation
- Regression mean imputation.
- Multiple Imputation.
- Bayesian factor models.

One-bit matrix completion: Motivation

Binary data with missing entries

Senate Voting

$$\begin{array}{cc} & \text{Senate bills} \\ \text{Senators} & \begin{pmatrix} \frownie & ? & \smiley & ? & \smiley \\ ? & \frownie & ? & ? & ? \\ ? & ? & ? & \smiley & \frownie \\ ? & \frownie & \smiley & ? & ? \\ \smiley & ? & ? & ? & \smiley \end{pmatrix} \end{array}$$

<u>Mathoverflow</u>

$$\begin{array}{cc} & \text{Math questions} \\ \text{Math enthusiasts} & \begin{pmatrix} ? & ? & ? & \text{☺} & \text{☺} \\ \text{☹} & ? & \text{☺} & ? & \text{☺} \\ ? & \text{☹} & \text{☺} & ? & ? \\ ? & \text{☹} & ? & ? & ? \\ \text{☺} & ? & ? & ? & \text{☺} \end{pmatrix} \end{array}$$

Pandora

$$
\begin{array}{cc}
 & \text{Songs} \\
\text{People} & \begin{pmatrix}
\text{☹} & ? & \text{☺} & ? & \text{☺} \\
? & ? & ? & \text{☺} & \text{☹} \\
\text{☹} & ? & ? & ? & \text{☺} \\
? & \text{☹} & \text{☺} & ? & ? \\
? & \text{☹} & ? & ? & ?
\end{pmatrix}
\end{array}
$$

Research literature

$$
\begin{array}{cc}
 & \text{Papers} \\
\text{Researchers} & \begin{pmatrix}
\frownie & ? & \smiley & ? & \smiley \\
? & \frownie & ? & ? & ? \\
? & ? & ? & \smiley & \frownie \\
? & \frownie & \smiley & ? & ? \\
\smiley & ? & ? & ? & \smiley
\end{pmatrix}
\end{array}
$$

Netflix

$$\begin{array}{cc} & \text{Movies} \\ \text{People} & \begin{pmatrix} ? & ? & ? & \smiley & \smiley \\ \frownie & ? & \smiley & ? & \smiley \\ ? & \frownie & \smiley & ? & ? \\ ? & \frownie & ? & ? & ? \\ \smiley & ? & ? & ? & \smiley \end{pmatrix} \end{array}$$

Reddit

$$
\begin{array}{cc}
 & \text{Articles} \\
\text{People} & \begin{pmatrix}
\text{☹} & ? & \text{☺} & ? & \text{☺} \\
? & ? & ? & \text{☺} & \text{☹} \\
\text{☹} & ? & ? & ? & \text{☺} \\
? & \text{☹} & \text{☺} & ? & ? \\
? & \text{☹} & ? & ? & ?
\end{pmatrix}
\end{array}
$$

Incomplete binary survey

$$\begin{array}{cc} & \text{Survey questions} \\ \text{People} & \begin{pmatrix} \frownie & ? & \smiley & ? & \smiley \\ ? & \frownie & ? & ? & ? \\ ? & ? & ? & \smiley & \frownie \\ ? & \frownie & \smiley & ? & ? \\ \smiley & ? & ? & ? & \smiley \end{pmatrix} \end{array}$$

Sensor triangulation

	Sensors				
	☺	☺	☹	☺	☹
	☺	☺	☺	☹	☹
Sensors	☹	☺	☺	☹	☺
	☺	☹	☹	☺	☺
	☹	☹	☺	☺	☺

Senate Voting

$$\mathbf{Y} = \begin{array}{cc} & \text{Senate bills} \\ \text{Senators} & \begin{pmatrix} ? & ? & ? & \text{☺} & \text{☺} \\ \text{☹} & ? & \text{☺} & ? & \text{☺} \\ ? & \text{☹} & \text{☺} & ? & ? \\ ? & \text{☹} & ? & ? & ? \\ \text{☺} & ? & ? & ? & \text{☺} \end{pmatrix} \end{array}$$

Q: Low-rank model?

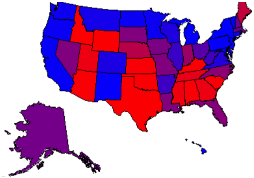
A: A classical numerical experiment with voting data.

Senate voting

Y: Voting history of US senators on 299 bills from 2008-2010.

Senate voting

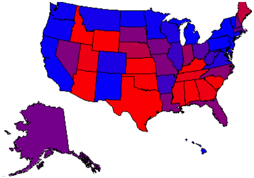
Y: Voting history of US senators on 299 bills from 2008-2010.



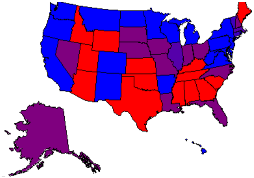
(f) First singular vector of Y

Senate voting

Y: Voting history of US senators on 299 bills from 2008-2010.

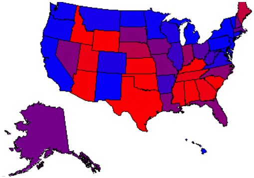


(h) First singular vector of Y

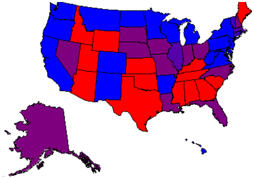


(i) Senate party affiliations

Y: Voting history of US senators on 299 bills from 2008-2010.



(j) First singular vector of Y



(k) Senate party affiliations

$\Rightarrow$ A low-rank model?

- What matrix has low rank?

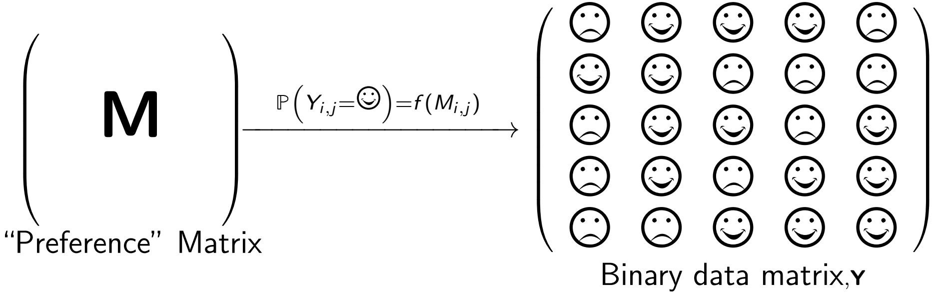
Low-rank assumption

Consider the **senate voting** example.

- Can the voting preferences of a certain senator be predicted given only a few characteristics of this senator?
- Does $\mathbf{Y}$ depend linearly on these characteristics?

Generalized linear model

$$\underbrace{\begin{pmatrix} \quad \mathbf{M} \quad \end{pmatrix}}_{\text{“Preference” Matrix}} \xrightarrow{\mathbb{P}\left(Y_{i,j}=\text{☺}\right)=f(M_{i,j})} \underbrace{\begin{pmatrix} \text{☹} & \text{☺} & \text{☺} & \text{☺} & \text{☹} \\ \text{☺} & \text{☺} & \text{☹} & \text{☹} & \text{☹} \\ \text{☹} & \text{☺} & \text{☺} & \text{☹} & \text{☺} \\ \text{☹} & \text{☺} & \text{☹} & \text{☺} & \text{☺} \\ \text{☹} & \text{☹} & \text{☺} & \text{☺} & \text{☺} \end{pmatrix}}_{\text{Binary data matrix,}\mathbf{Y}}$$



- $\mathbf{M}$ is unknown. $\mathbf{M}$ has (approximately) low rank.
- $f : \mathbb{R} \to [0, 1]$ is a known function (e.g., the logistic curve).
- $\mathbf{M} \in \mathbb{R}^{d \times d}$, $\mathbf{Y} \in \{☺, ☹\}^{d \times d}$.

Generalized linear model

$$\begin{pmatrix} & & \\ & \mathbf{M} & \\ & & \end{pmatrix} \xrightarrow{\mathbb{P}\left(Y_{i,j}=\text{☺}\right)=f(M_{i,j})} \begin{pmatrix} \text{☹} & ? & ? & ? & \text{☹} \\ ? & \text{☺} & ? & \text{☹} & ? \\ ? & ? & \text{☺} & ? & ? \\ ? & ? & \text{☹} & ? & ? \\ ? & ? & \text{☺} & ? & ? \end{pmatrix}$$

"Preference" Matrix

Incomplete binary matrix, $\mathbf{Y}$

- $\mathbf{M}$ is unknown. $\mathbf{M}$ has (approximately) low rank.
- $f : \mathbb{R} \to [0,1]$ is a known function (e.g., the logistic curve).
- $\mathbf{M} \in \mathbb{R}^{d \times d}, \mathbf{Y} \in \{\text{☺}, \text{☹}\}^{d \times d}$.
- $\Omega \subset \{1, 2, \ldots, d\} \times \{1, 2, \ldots, d\}$. You see $\mathbf{Y}_\Omega$.

Latent variable formulation

$$Y_{i,j} = \mathrm{sign}(M_{i,j} + Z_{i,j}) = \begin{cases} \smiley & \text{if } M_{i,j} + Z_{i,j} \geq 0 \\ \frownie & \text{if } M_{i,j} + Z_{i,j} < 0 \end{cases}$$

- Z is an iid noise matrix.
- $f(x) := \mathbb{P}\left(Z_{1,1} \geq -x\right)$.
- You see $\mathbf{Y}_\Omega$.

Goal: Efficiently approximate $\mathbf{M}$ and/or $f(\mathbf{M})$.

Data:
$$Y_\Omega = \begin{pmatrix} \frownie & ? & \smiley & ? & \smiley \\ ? & \frownie & ? & ? & ? \\ ? & ? & ? & \smiley & \frownie \\ ? & \frownie & \smiley & ? & ? \\ \smiley & ? & ? & ? & \smiley \end{pmatrix}.$$

Assumption: $\mathbf{M}$ has (approximately) low rank.

Approximately low-rank

Assumption:

$$\mathbf{M} \in \mathit{conv}(\textit{rank-r matrices with Frobenius norm } d)$$
$$\in (\text{Nuclear-norm ball}) \cdot d\sqrt{r}$$

$\Rightarrow \|\mathbf{M}\|_* \leq d\sqrt{r}$.

- $\|\mathbf{M}\|_* = \sum_i \sigma_i(\mathbf{M}) = \|(\sigma_1, \sigma_2, \ldots, \sigma_d)\|_1$.
- Robust extension of the rank [Chatterjee 2013].
- Facilitates convex programming reconstruction.

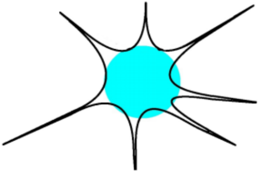


Figure: Nuclear-norm ball in high dimensions

Maximum likelihood estimation

Take our estimate $\hat{\mathbf{M}}$ be the solution to the following convex program:

$$\max_{\mathbf{X}} F_{\Omega,\mathbf{Y}}(\mathbf{X}) \quad \text{such that} \quad \frac{1}{d}\|\mathbf{X}\|_* \le \sqrt{r}$$

- $F_{\Omega,\mathbf{Y}}$: log-likelihood function.

Estimation of the distribution, $f(\mathbf{M})$

Theorem (Upper bound achieved by convex programming)

Let f be the logistic function. Assume that $\frac{1}{d}\left\|\mathbf{M}\right\|_* \leq \sqrt{r}$. Suppose the sampling set is chosen at random with $\mathbb{E}\left|\Omega\right| = m \geq d\log(d)$. Then with high probability,

$$\frac{1}{d^2}\sum_{i,j} d_H^2(f(\hat{M}_{i,j}), f(M_{i,j}))^2 \leq C \min\left(\sqrt{\frac{rd}{m}}, 1\right).$$

- $d_H(p,q)^2 := (\sqrt{p} - \sqrt{q})^2 + (\sqrt{1-p} - \sqrt{1-q})^2 =$ squared Hellinger distance.

Is this bound tight?

Estimation of the distribution, $f(\mathbf{M})$

Theorem (Upper bound achieved by convex programming)

Let f be the logistic function. Assume that $\frac{1}{d}\|\mathbf{M}\|_* \leq \sqrt{r}$. Suppose the sampling set is chosen at random with $\mathbb{E}|\Omega| = m \geq d\log(d)$. Then with high probability,

$$\frac{1}{d^2}\sum_{i,j} d_H^2(f(\hat{M}_{i,j}), f(M_{i,j}))^2 \leq C \min\left(\sqrt{\frac{rd}{m}}, 1\right).$$

Theorem (Lower bound achievable by any estimator)

In the setup of the above theorem,

$$\inf_{\hat{\mathbf{M}}(Y)} \sup_{\mathbf{M}} \mathbb{E}\, \frac{1}{d^2}\sum_{i,j} d_H^2(f(\hat{M}_{i,j}), f(M_{i,j}))^2 \geq c \min\left(\sqrt{\frac{rd}{m}}, 1\right).$$

Estimation of $\mathbf{M}$

Assumption: $\|\mathbf{M}\|_\infty \leq \alpha$.

$$\mathbf{M} \neq \begin{pmatrix} d & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\max_{\mathbf{X}} F_{\Omega,\mathbf{Y}}(\mathbf{X}) \quad \text{such that} \quad \frac{1}{d\alpha}\|\mathbf{X}\|_* \leq \sqrt{r} \quad \text{and} \quad \|\mathbf{X}\|_\infty \leq \alpha$$

1-bit matrix completion vs noisy matrix completion: error bounds

- Let $\mathbf{Y}^0 := \mathbf{M} + \mathbf{Z}$.
- $\mathbf{Z}$ is a matrix with iid Gaussian noise with variance σ^2.
- Let
$$Y_{i,j} := \mathrm{sign}(Y^0_{i,j}).$$
- $\Rightarrow f$ follows the *probit* model.

Question: How much harder is it to estimate $\mathbf{M}$ from $\mathbf{Y}$ in comparison to estimating $\mathbf{M}$ from $\mathbf{Y}^0$?

Two regimes: High SNR and low SNR.

Case 1: $\sigma \leq \alpha$ (high signal-to-noise ratio).

Theorem (Upper bound, convex programming, quantized input, $\mathbf{Y}$)

Let f be the probit function. Assume that $\frac{1}{d\alpha}\|\mathbf{M}\|_* \leq \sqrt{r}$ and $\|\mathbf{M}\|_\infty \leq \alpha$. Suppose the sampling set is chosen at random with $\mathbb{E}|\Omega| = m \geq d\log(d)$. Then with high probability,

$$\frac{1}{d^2}\left\|\hat{\mathbf{M}} - \mathbf{M}\right\|_F^2 \leq C\alpha^2 \exp\left(\frac{\alpha^2}{2\sigma^2}\right)\sqrt{\frac{rd}{m}}.$$

Theorem (Lower bound, achievable by any estimator, unquantized input)

In the setup of the above theorem, and under mild technical conditions,

$$\inf_{\hat{\mathbf{M}}(Y^0)} \sup_{\mathbf{M}} \mathbb{E}\frac{1}{d^2}\left\|\hat{\mathbf{M}} - \mathbf{M}\right\|_F^2 \geq c\alpha\sigma\sqrt{\frac{rd}{m}}.$$

Case 2: $\sigma \geq \alpha$ (low signal-to-noise ratio).

Theorem (Upper bound, convex programming, quantized input, $\mathbf{Y}$)

Let f be the probit function. Assume that $\frac{1}{d\alpha} \|\mathbf{M}\|_* \leq \sqrt{r}$ and $\|\mathbf{M}\|_\infty \leq \alpha$. Suppose the sampling set is chosen at random with $\mathbb{E}\,|\Omega| = m \geq d\log(d)$. Then with high probability,

$$\frac{1}{d^2} \left\|\hat{\mathbf{M}} - \mathbf{M}\right\|_F^2 \leq C\alpha\sigma\sqrt{\frac{rd}{m}}.$$

Theorem (Lower bound, achievable by any estimator, unquantized input)

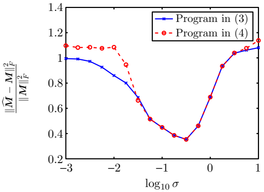
In the setup of the above theorem, and under mild technical conditions,

$$\inf_{\hat{\mathbf{M}}(Y^0)} \sup_{\mathbf{M}} \mathbb{E}\,\frac{1}{d^2} \left\|\hat{\mathbf{M}} - \mathbf{M}\right\|_F^2 \geq c\alpha\sigma\sqrt{\frac{rd}{m}}.$$

Dithering: noise helps!

Conclusion:

- When the noise is larger than the signal, quantizing to a single bit loses almost no information!
- When the noise is (significantly) smaller than the signal, increasing the noise improves recovery from quantized measurements!



Why we need noise

Take

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

Now remove the noise!

Why we need noise

Take

$$Y_{i,j} = \mathrm{sign}(M_{i,j})$$

Claim: Accurate reconstruction of $\mathbf{M}$ is impossible!

Why we need noise

Take

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

Suppose that

$$\mathbf{M} = \begin{pmatrix} \lambda & \lambda & \lambda & \lambda \\ \lambda & \lambda & \lambda & \lambda \\ \lambda & \lambda & \lambda & \lambda \\ \lambda & \lambda & \lambda & \lambda \end{pmatrix}.$$

Why we need noise

Take

$$Y_{i,j} = \text{sign}(M_{i,j})$$

Suppose that

$$\mathbf{M} = \begin{pmatrix} \lambda & \lambda & \lambda & \lambda \\ \lambda & \lambda & \lambda & \lambda \\ \lambda & \lambda & \lambda & \lambda \\ \lambda & \lambda & \lambda & \lambda \end{pmatrix}.$$

$\Rightarrow Y_{i,j} = \text{sign}(\lambda)$
$\Rightarrow$ Approximation of **M** is impossible even if every entry of **Y** is seen.

Why we need noise

Take

$$Y_{i,j} = \text{sign}(M_{i,j})$$

- Suppose that we know that $\mathbf{M}$ has rank 1 so that $\mathbf{M} = \mathbf{u}\mathbf{v}^T$ for some two vectors $\mathbf{u}, \mathbf{v} \in \mathbb{R}^d$.
- $\tilde{\mathbf{M}} = \tilde{\mathbf{u}}\tilde{\mathbf{v}}^T$ leads to exactly the same binary data as $\mathbf{M}$ if the signs of $\tilde{u}$ match the signs of $\mathbf{u}$ and similarly for $\tilde{\mathbf{v}}$ and $\mathbf{v}$.
- $\Rightarrow$ Approximation of $\mathbf{M}$ is impossible even if every entry of $\mathbf{Y}$ is seen.

Related literature

- [Srebro - Rennie - Jaakkola et al. 2004] Model free: If an estimate has low nuclear norm and matches the signs of the observed entries by a significant margin, then the error on unobserved entries is small.
- Our results:
  - If the model is correct, then the overall error is nearly minimax.
  - Noise helps!
- [Cai-Zhou 2013]: Extension to non-uniform sampling by using *max norm*.

general f

Let

$$L_\alpha := \sup_{|x| \le \alpha} \frac{|f'(x)|}{f(x)(1 - f(x))} \qquad .$$

Theorem (Upper bound achieved by convex programming)

$$d_H^2(f(\widehat{\mathbf{M}}), f(\mathbf{M})) \le C \alpha L_\alpha \sqrt{\frac{rd}{m}}.$$

Theorem (Lower bound achievable by any estimator)

$$d_H^2(f(\mathbf{M}), f(\widehat{\mathbf{M}})) \ge c \frac{\alpha}{L_1} \sqrt{\frac{rd}{m}}.$$

general f

Let

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

Theorem (Upper bound achieved by convex programming)

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

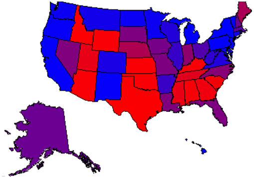
Theorem (Lower bound achievable by any estimator)

$$\frac{1}{d_1 d_2}\|\mathbf{M} - \widehat{\mathbf{M}}\|_F^2 \ge c\alpha\sqrt{\beta_{\frac{3}{4}\alpha}}\sqrt{\frac{rd}{m}}.$$

Experiments with real data.

Voting simulation

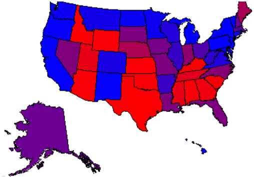
Binary data: Voting history of US senators on 299 bills from 2008-2010.



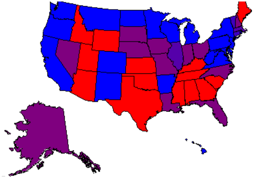
(a) First singular vector of $\hat{\mathbf{M}}$

Voting simulation

Binary data: Voting history of US senators on 299 bills from 2008-2010.



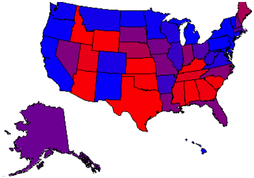
(b) First singular vector of $\hat{\mathbf{M}}$



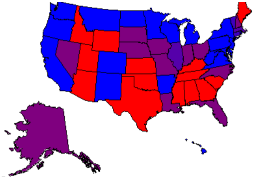
(c) Senate party affiliations

Voting simulation

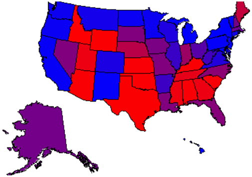
Binary data: Voting history of US senators on 299 bills from 2008-2010.



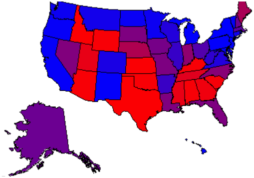
(d) First singular vector of $\hat{\mathbf{M}}$



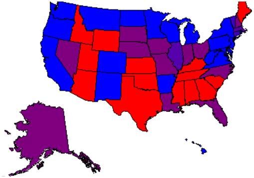
(e) Senate party affiliations



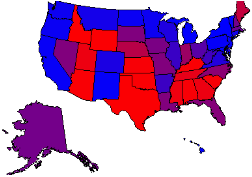
(f) First singular vector of $\mathbf{Y}_\Omega$.



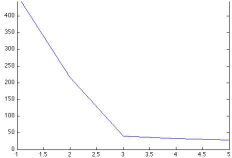
(g) First singular vector of $\hat{\mathbf{M}}$



(h) Senate party affiliations



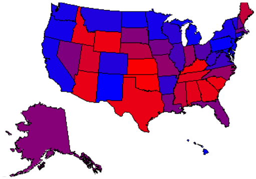
(i) First singular vector of $\mathbf{Y}_\Omega$.



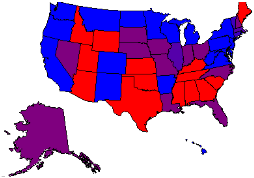
(j) The first five singular values of $\hat{\mathbf{M}}$: $463, 216, 40, 32, 29$

Voting simulation

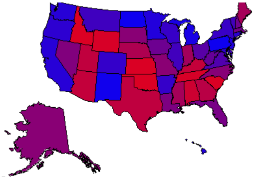
Randomly delete 90% of entries.



(k) First singular vector of $\hat{\mathbf{M}}$



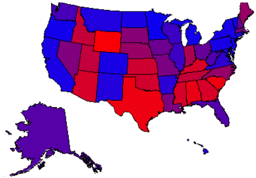
(l) Senate party affiliations



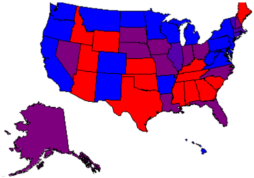
(m) First singular vector of $\mathbf{Y}_{\Omega}$.

Voting simulation

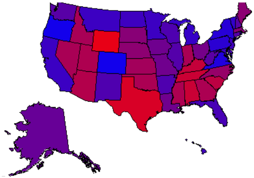
Randomly delete 95% of entries.



(n) First singular vector of $\hat{\mathbf{M}}$



(o) Senate party affiliations



(p) First singular vector of $\mathbf{Y}_\Omega$.

Voting simulation

With 95% of votes deleted:

86% of missing votes were correctly predicted. (Averaged over 20 experiments.)

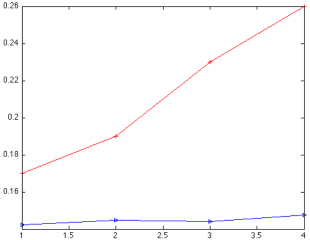


Figure: **Percent of missed predictions versus model rank** r

— Rank-r approximation of Y_Ω

— Nuclear-norm constrained maximum-likelihood estimation

MovieLens data set

- 100,000 movie ratings on a scale from 1 to 5 (sparsely sampled matrix).
- Convert to binary outcomes by comparing each rating to the mean.
- Training on 95,000 ratings and testing on remainder.
- **One-bit matrix completion:** Given +1s and -1s. Evaluate by checking if we predict the correct sign.
- **Standard matrix completion:** Given original values from 1 to 5. Evaluate by checking if the imputed value is above or below the mean.
  - "Standard" matrix completion: 60% accuracy
    1: 64%   2: 56%   3: 44%   4: 65%   5: 74%
  - Binary matrix completion: 73% accuracy
    1: 79%   2: 73%   3: 58%   4: 75%   5: 89%

Restaurant recommendations

[REU with Gao, Wootters, Vershynin]

- 100 restaurants, 107 users.
- 11 yes/no answers per user.
- $>$ 75% success rate in recommending 1 restaurant per user (estimated using cross validation).

Learning analytics

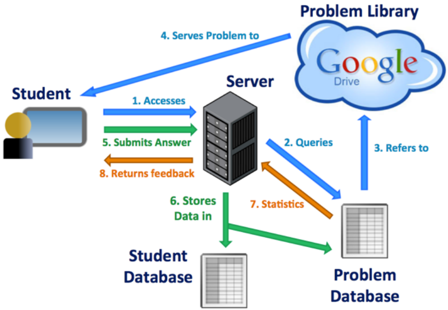


Figure: Problem Roulette [Evrard et al. 2013, Am. J. Phys.]

Goals:

- Recommend practice problems to students based on past performance.
- Learn which practice problems have the best teaching ability.

Learning analytics

Data from Phys 240:

- $\sim$ 450 students.
- $\sim$ 370 challenging multiple-choice problems.
- $\sim$ 20% of problems answered.

Last answer by each student used for cross validation.
68% of answers correctly predicted.

- How well do we predict individual probabilities?

Learning analytics

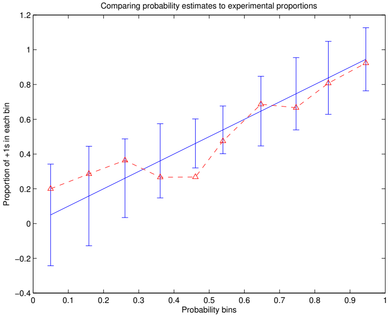
Data from Phys 240:

- ~ 450 students.
- ~ 370 challenging multiple-choice problems.
- $\sim 20\%$ of problems answered.

Last answer by each student used for cross validation.

68% of answers correctly predicted.

- How well do we predict individual probabilities?



Methods of proof

Upper bounds: Probability in Banach spaces/random matrix theory

Lower bounds: Information theoretic techniques: Fano's inequality

Bare-bones sketch of upper bound proof

Recall: $F_{\Omega,\mathbf{Y}}(\mathbf{X})$ is the log-likelihood of $\mathbf{X}$ (we maximize it).

1. For a fixed matrix, $\mathbf{X}$, $\mathbb{E}(F_{\Omega,\mathbf{Y}}(\mathbf{M}) - F_{\Omega,\mathbf{Y}}(\mathbf{X})) = c \cdot D(f(\mathbf{X})||f(\mathbf{M}))$.
2. **Lemma:** the following holds for all $\mathbf{X}$ satisfying $\frac{1}{d\alpha}\|\mathbf{X}\|_* \leq \sqrt{r}$:
$$|F_{\Omega,\mathbf{Y}}(\mathbf{X}) - \mathbb{E}\, F_{\Omega,\mathbf{Y}}(\mathbf{X})| \leq \delta.$$
3. The maximizer, $\hat{\mathbf{M}}$ satisfies $F_{\mathbf{Y},\Omega}(\hat{\mathbf{M}}) \geq F_{\mathbf{Y},\Omega}(\mathbf{M})$.
4. $$\begin{aligned} 0 &\geq F_{\Omega,\mathbf{Y}}(\mathbf{M}) - F_{\Omega,\mathbf{Y}}(\hat{\mathbf{M}}) \geq \mathbb{E}(F_{\Omega,\mathbf{Y}}(\mathbf{M}) - F_{\Omega,\mathbf{Y}}(\hat{\mathbf{M}})) - 2\delta \\ &= c \cdot D(f(\hat{\mathbf{M}})||f(\mathbf{M})) - 2\delta \end{aligned}$$

   Thus,
$$D(f(\hat{\mathbf{M}})||f(\mathbf{M})) \leq \frac{2}{c}\delta.$$

Key step: Proof of lemma.

Feedback on vague idea

- Problem 1: After deriving theory for 1-bit matrix completion, finding good data to test the method on.
  - Bias towards data on which my method works well.
- Problem 2: After getting the 1-bit matrix data for learning analytics, finding the best method to use to analyze the data.
  - Bias towards using my own method.

Feedback on vague idea

- Problem 1: After deriving theory for 1-bit matrix completion, finding good data to test the method on.
  - Bias towards data on which my method works well.
- Problem 2: After getting the 1-bit matrix data for learning analytics, finding the best method to use to analyze the data.
  - Bias towards using my own method.

Solution: Large online problem bank?

- Algorithms people submit code which should work out of the box.
- It is tested across a broad array of problems.
- A scientist analyzing a data set can find similar classes of data sets and has access to the code to try.

Thank you!

www.yanivplan.com