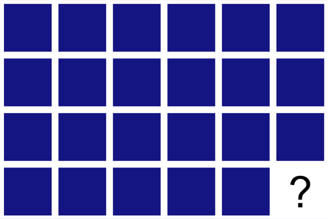
A Polynomial Model for Filling In Incomplete Data

Christopher Gadzinski
University of Coimbra

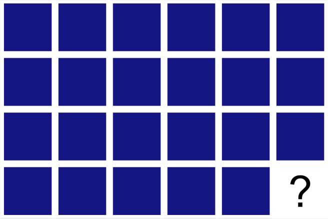
July 2021

A Regression Problem

- A process is producing vectors in $\mathbb{R}^m$.
- The process generates a dataset $(\mathbf{x}_1, \ldots, \mathbf{x}_n)$.
- A new vector $\mathbf{x}_{n+1}$ is generated, but we only observe some of its coordinates.

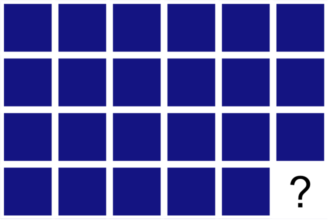


A Regression Problem



- Find the missing coordinates!
- This is **regression**.

Reminder: The Linear Regression Model



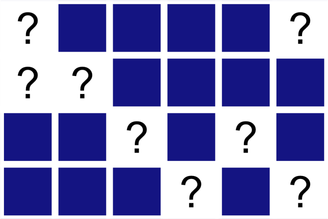
- **Hypothesis:** there is a linear function f taking

$$\text{independent variables} \xrightarrow{f} \text{dependent variables}$$

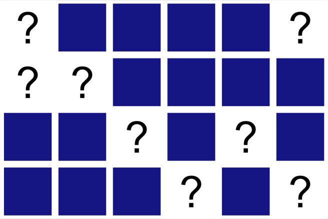
  in some approximate sense.
- **Strategy:** find a map f agreeing with the observed dataset $(\mathbf{x}_1, \ldots, \mathbf{x}_n)$ with least-squares optimization.

A Collaborative Filtering Problem

- A process is producing vectors in $\mathbb{R}^m$.
- The process generates a dataset $(\mathbf{x}_1, \ldots, \mathbf{x}_n)$...
- ...but coordinates are missing from every datapoint!



A Collaborative Filtering Problem

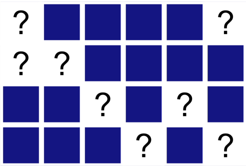


- We want to infer *all* the missing coordinates.
- This is a **collaborative filtering** problem.
- Application: recommender systems (for Netflix, Amazon . . .).

A Linear Model for Collaborative Filtering

- **Hypothesis:** our data is concentrated on a linear subspace.
- **Strategy:** solve "low rank matrix completion."

$$\begin{array}{rll} \textbf{minimize} & \operatorname{rank} M & \\ \textbf{subject to} & m_{i,j} = c_{i,j} & \text{for all } (i,j) \in \Omega \end{array}$$



LRMC in Practice

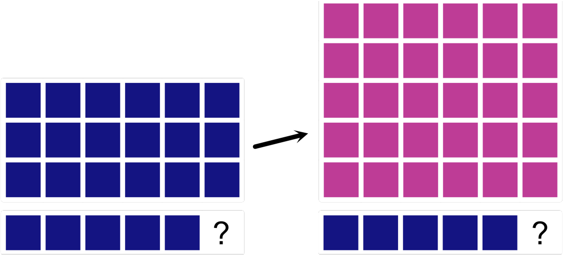
- Linear regression $\Leftrightarrow$ solving a linear system.
- Low rank matrix completion is “hard in general.”
- Some numerical methods work in practical problems.
- One popular strategy: minimize the sum of the singular values of M. This can be expressed as a *semidefinite program* and solved with iterative numerical algorithms.

Reminder: Linear Regression Over a Feature Space

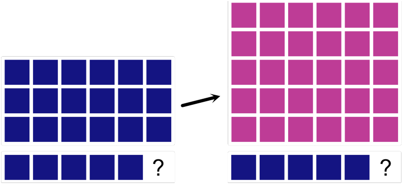
- What if we want to fit a model

$$\text{dependent variables} \xrightarrow{f} \text{independent variables}$$

  that lives in some linear space $\mathcal{M}$ of model functions?
- If $\mathcal{M}$ is finite-dimensional, this can be interpreted as linear regression over a transformed data set.



Reminder: Linear Regression Over a Feature Space



- Say $\{f_1, \ldots, f_r\}$ is a basis for $\mathcal{M}$. Then solving

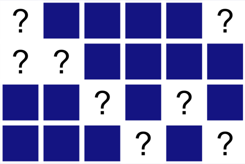
$$\textbf{find } f \in \mathcal{M} \textbf{ minimizing } \sum (f(x_i) - y_i)^2$$

is equivalent to solving

$$\textbf{find } w \in \mathbb{R}^r \textbf{ minimizing } \sum (\langle w, \phi(x_i) \rangle - y_i)^2$$

where $\phi(x) = (f_1(x), \ldots, f_r(x))$ is the **feature map**.

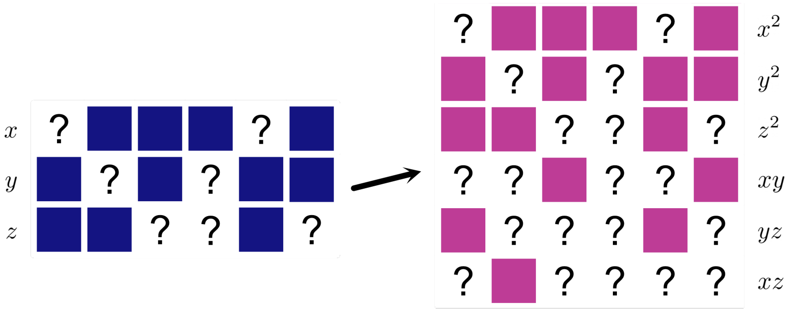
LRMC With a Feature Map?



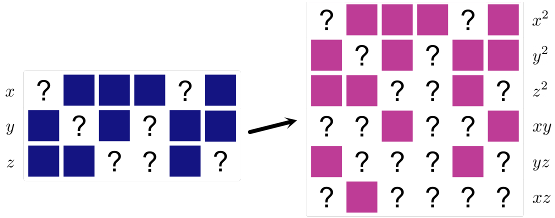
- Can we use the same trick with LRMC?
- This was proposed by Ongie et. al. in their paper, *Tensor Methods for Nonlinear Matrix Completion* (2020).
- They proposed the feature map for homogeneous quadratic polynomials:

$$\phi(x, y, z) = (x^2, y^2, z^2, xy, yz, xz).$$

LRMC With a Feature Map?



Research Questions

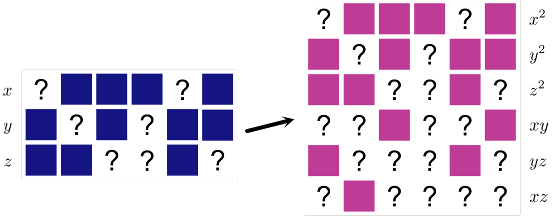


- Quadratic relationships on the **original data** turn into linear relationships on the **transformed data**.
- **Problem:** Can LRMC infer these relationships?

Research Questions

- **Problem:** Can LRMC infer these relationships?
- **Answer:** Suppose we're only observing k coordinates per datapoint. Then, LRMC can only infer our space of polynomial relationships if they are generated by "sparse polynomials," each involving at most k distinct coordinates.
- **This is a very restrictive property.**

Research Questions



- What's going on?
- Ideally, we'd optimize the unknown entries of the **original matrix** so that the rank of the **transformed matrix** is minimized.

Research Questions

- Ideally, we'd optimize the unknown entries of the **original matrix** so that the rank of the **transformed matrix** is minimized.

$$
\textbf{minimize} \quad \operatorname{rank} \begin{bmatrix} a_1 & \dots & a_n \\ b_1 & \dots & b_n \\ \vdots & & \vdots \\ f_1 & \dots & f_n \end{bmatrix}
$$

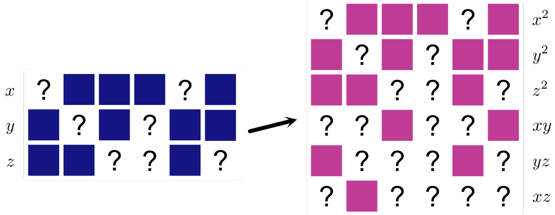
$$
\textbf{subject to} \begin{cases} \text{existence of some vectors } (x_i, y_i, z_i) \text{ so that} \\ \qquad (a_i, b_i, c_i, d_i, e_i, f_i) = (x_i^2, y_i^2, z_i^2, x_i y_i, y_i z_i, x_i z_i) \\ \text{constraints on the vectors } (x_i, y_i, z_i) \end{cases}
$$

- … But, by applying LRMC on the **transformed matrix**, we're solving a relaxation of this problem!

Research Questions

- Let (x, y, z) be a column of the **original matrix** with $x = 1$ and $y = 2$, and let
$$(a, b, c, d, e, f) = (x^2, y^2, z^2, xy, yz, xz)$$
be a column of the **transformed matrix**.
- We are telling the LRMC solver that $a = 1, b = 4, d = 2$.
- The true constraint on (a, b, c, d, e, f) is hard to use in our LRMC solver...
- But, there is another linear equation to use:
$$yz = 2z = 2xz \implies e = 2f.$$

Research Questions



- Let k **coordinates** be observed. Instead of $\binom{k+1}{2}$, we can actually enforce

$$\binom{k+1}{2} + (k-1)(m-k)$$

constraints on the **transformed column**.

- This is a big help on sparse data (where $k \ll m$).

What's Next?

- Does our new method suffer from the same severe limitations that the "naive" approach suffered from? (I don't think so!)
- Can we write an efficient LRMC-type solver that uses the new constraints and apply this to real-world collaborative filtering problems? **(Contact me if you want to do this!)**

me@cgad.ski