Parameterized Complexity and PCA

Fedor V. Fomin

Algorithmic Tractability via Sparsifiers
Leh 2019

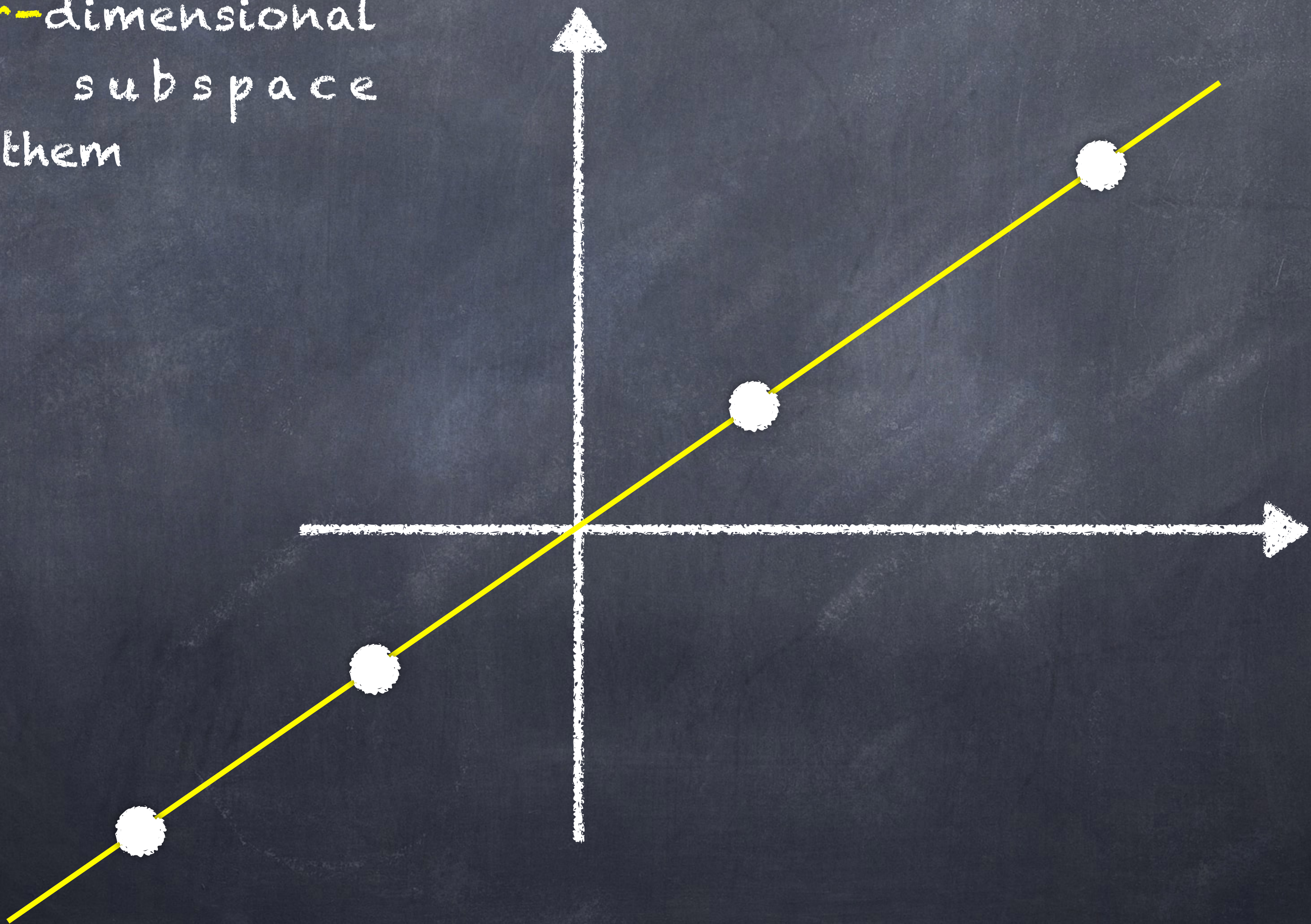
UNIVERSITY OF BERGEN

Parameterized Complexity

Is it all about graphs?

Toy problem

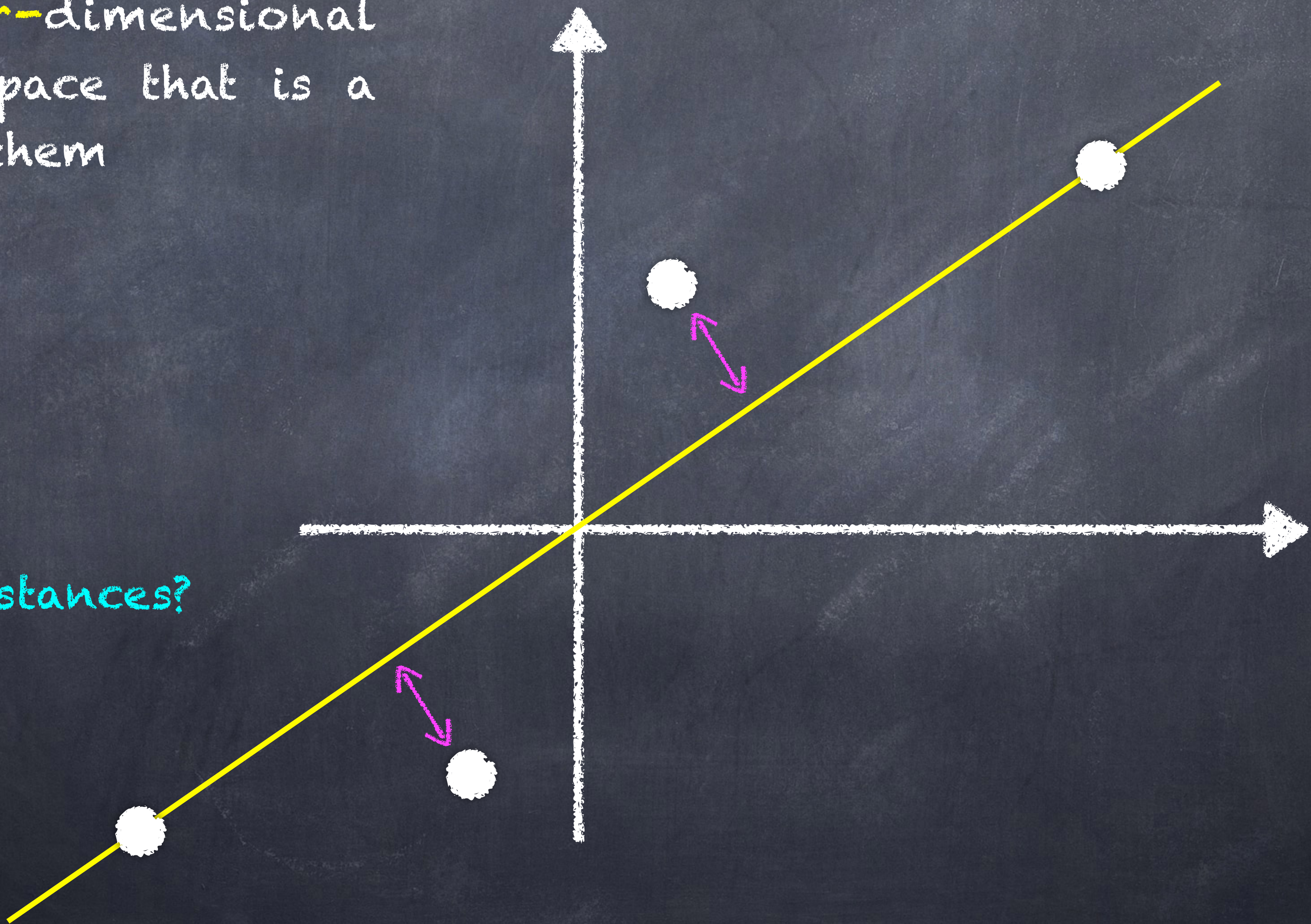
Given a set of points in R^d, find r-dimensional linear subspace containing them



Problem

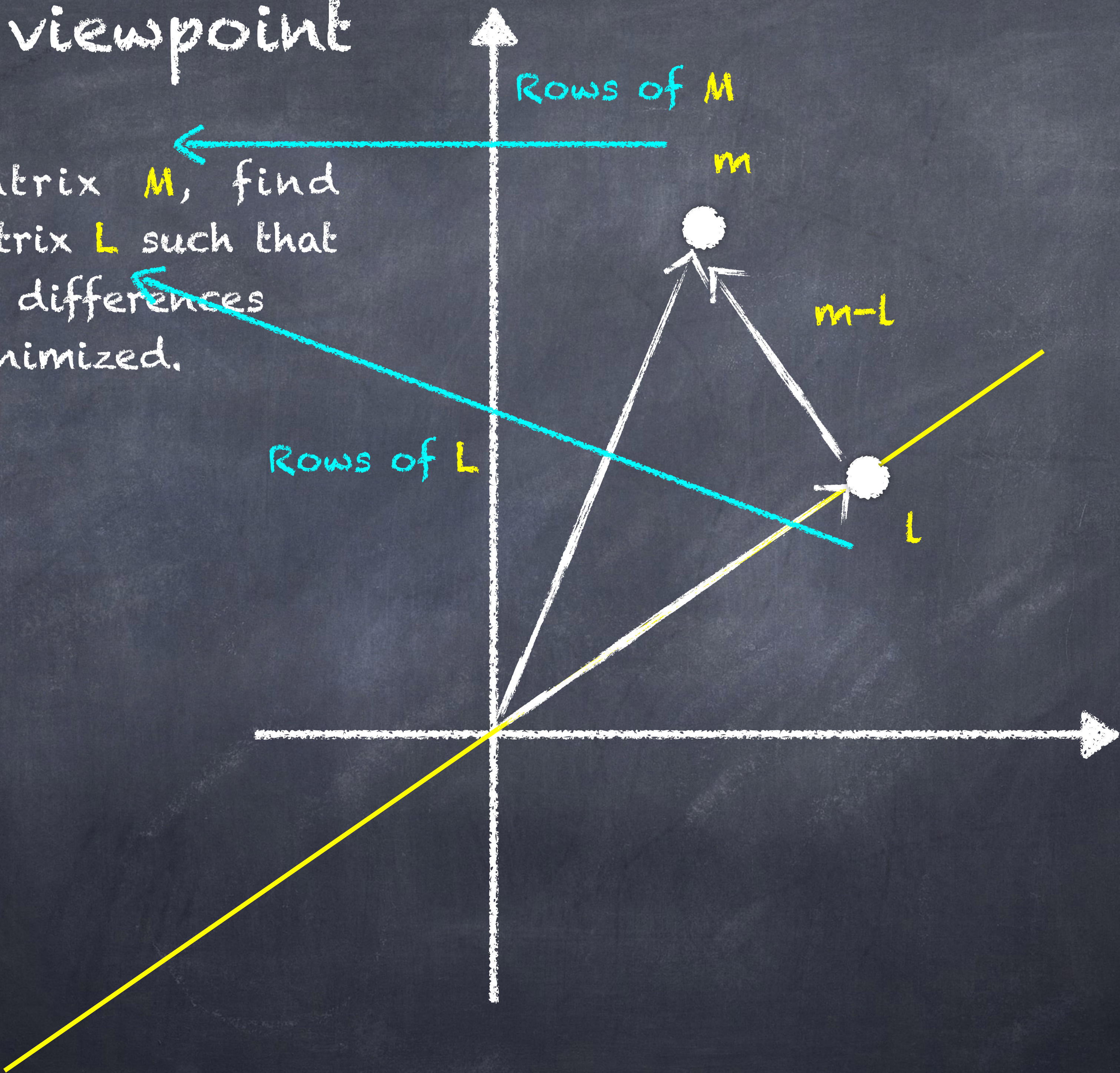
Given a set of points in R^d, find r-dimensional linear subspace that is a best fit to them

Sum of distances?



Matrix viewpoint

Input matrix M, find rank r matrix L such that the sum of differences m -l is minimized.

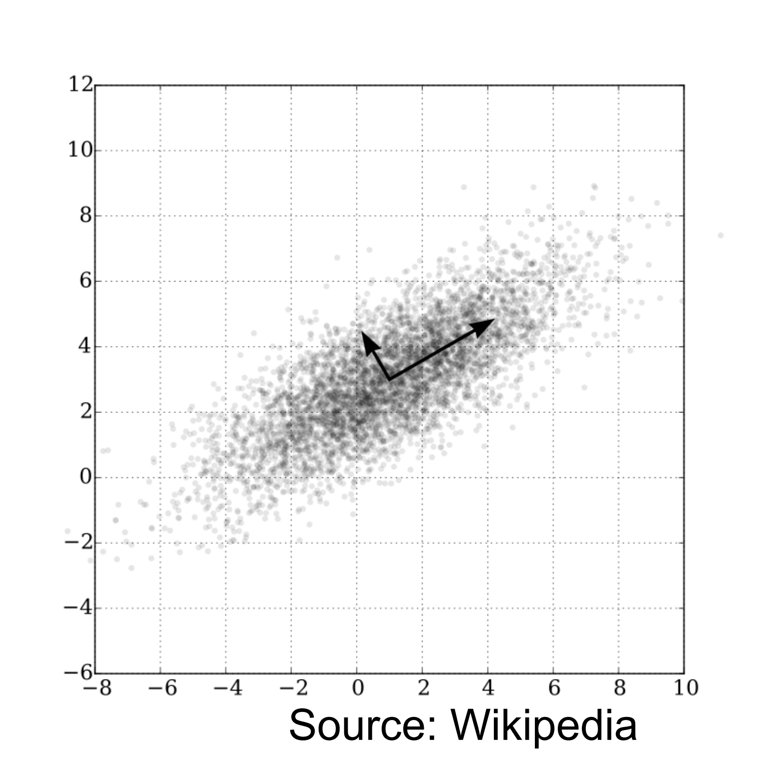


Principal component analysis (PCA)

$$\text{minimize } \|M - L\|_F^2$$
$$\text{subject to } \operatorname{rank}(L) \le r.$$

// $$\|A\|_F^2 = \sum_{i,j} a_{ij}^2$$

Frobenius norm



Sum of the squares of the Eucledian distances between rows (columns) of M-L

[Eckart & Young, 1936]

PCA

$$\text{minimize } \|M - L\|_F^2$$
$$\text{subject to } \operatorname{rank}(L) \leq r.$$

Singular Value Decomposition (SVD)

$$L_{OPT} = \sum_{i=1}^{r} \sigma_i u_i v_i^T$$

Singular vector

$$v_i = \arg\max |Mv|$$

$v_i \perp v_1, v_2, \ldots, v_{i-1}$
$|v_i| = 1$

Singular value

$$\sigma_i = |Mv_i|$$

Left-singular vector

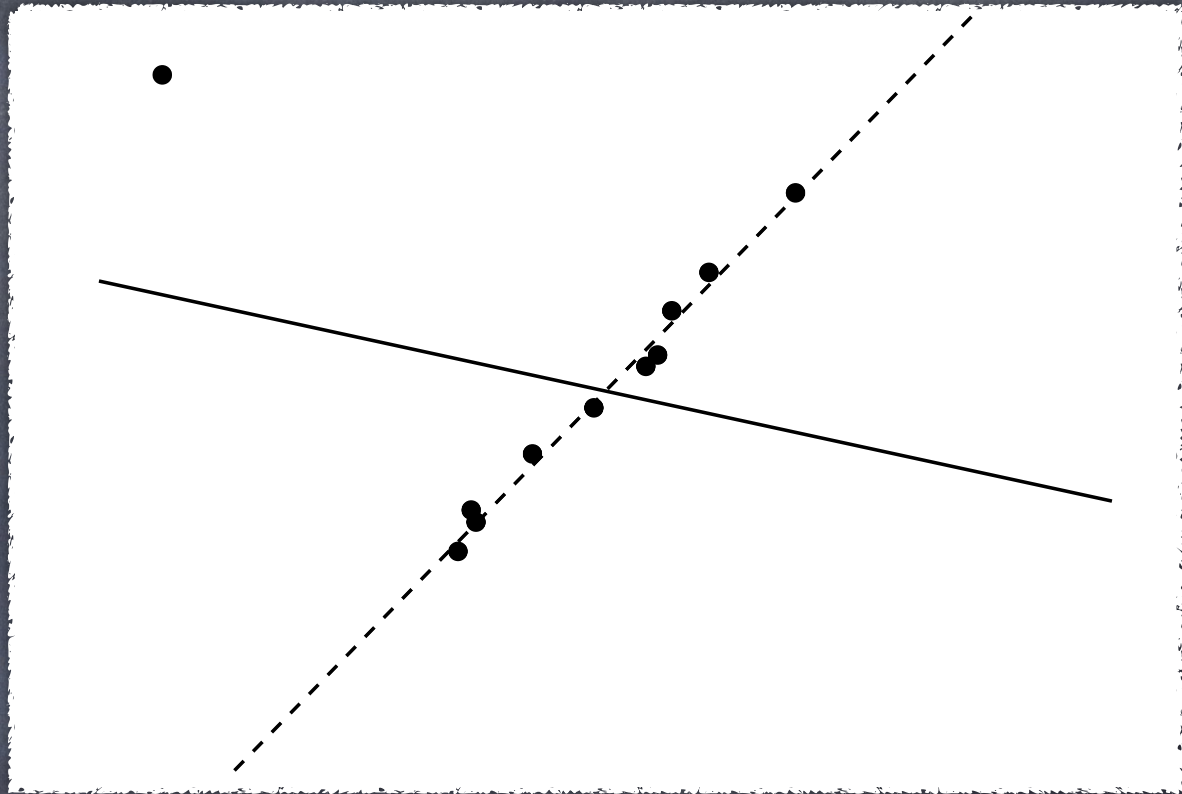
$$u_i = \frac{1}{\sigma_i} M v_i$$

[Eckart & Young, 1936]

PCA is not robust

Some observations are corrupted

How to reveal information about non-corrupted observations?



Robust PCA

number of non-zero elements

sparse component

$$M = L + S$$

data matrix

low rank

minimize $\text{rank}(L) + \|S\|_0$
subject to $M = L + S$

[Candès, Li, and Wright, 2011]

[Chandrasekaran et al., 2011]

This is an instance of a fundamental and largely unexplored question

"Can we reconcile computational efficiency and robustness in unsupervised learning?"

-Moritz Hardt and Ankur Moitra

Robust PCA

$$\text{minimize } \operatorname{rank}(L) + \|S\|_0$$
$$\text{subject to } M = L + S$$

Matrix Rigidity

Input: M, r, k

Task: Change at most k entries of M such that the resulting matrix is of rank at most r

Robust PCA

Input: M, r, k

Task: Change at most k entries of M such that the resulting matrix is of rank at most r

NP-complete for every $r>0$

W[1]-hard parameterized by k

We will see FPT parameterized by $r+k$!

Preprocessing

SIAM J. Discrete Math.
Vol. 32, No. 2, pp. 966–985

MATRIX RIGIDITY FROM THE VIEWPOINT OF PARAMETERIZED COMPLEXITY*

FEDOR V. FOMIN†, DANIEL LOKSHTANOV†, S. M. MEESUM‡, SAKET SAURABH§, AND MEIRAV ZEHAVI†

Abstract. For a target rank r, the *rigidity* of a matrix A over a field $\mathbb{F}$ is the minimum Hamming distance between A and a matrix of rank at most r. Rigidity is a classical concept in computational complexity theory: constructions of rigid matrices are known to imply lower bounds of significant importance relating to arithmetic circuits. Yet, from the viewpoint of parameterized complexity, the study of central properties of matrices in general, and of the rigidity of a matrix in particular, has been neglected. In this paper, we conduct a comprehensive study of different aspects of the computation of the rigidity of *general matrices* in the framework of parameterized complexity. Naturally, given parameters r and k, the Matrix Rigidity problem asks whether the rigidity of A for the target rank r is at most k. We show that in the case $\mathbb{F} = \mathbb{R}$ or $\mathbb{F}$ is any finite field, this problem is fixed-parameter tractable with respect to $k+r$. To this end, we present a dimension reduction procedure, which may be a valuable primitive in future studies of problems of this nature. We also employ central tools in real algebraic geometry, which are not well known in parameterized complexity, as a black box. In particular, we view the output of our dimension reduction procedure as an algebraic variety. Our

Robust PCA

Input: $n \times d$ matrix M, integers r, k

Task: Change at most k entries of M such that the resulting matrix is of rank at most r

Main idea: polytime procedure reducing (M,r,k) to equivalent instance (M',r,k), where matrix M' is $O(rk) \times O(rk)$ matrix

Step I. Preprocessing

If M has r+1 independent columns, at least one entry of these columns should be changed

$$\begin{pmatrix} \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \cdots \\ c_1 & c_2 & c_3 & c_4 & c_5 & c_6 & c_7 & c_8 & c_9 & \ddots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \cdots \end{pmatrix}$$

Choose a set of $r + 1$ linearly independent columns.

$$\left(\begin{array}{ccc|ccccccc} \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \cdots \\ c_1 & c_3 & c_6 & c_2 & c_4 & c_5 & c_7 & c_8 & c_9 & \ddots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \cdots \end{array}\right)$$

Step I. Preprocessing

If M has r+1 independent columns, at least one entry of these columns should be changed

$$\left(\begin{array}{ccc|cccccccc} \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \cdots \\ c_1 & c_3 & c_6 & c_2 & c_4 & c_5 & c_7 & c_8 & c_9 & \ddots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \cdots \end{array}\right)$$

Repeat at most $k+1$ times.

$$\left(\begin{array}{ccc|ccc|cccc} \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \cdots \\ c_1 & c_3 & c_6 & c_2 & c_4 & c_8 & c_6 & c_7 & c_9 & \ddots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \cdots \end{array}\right)$$

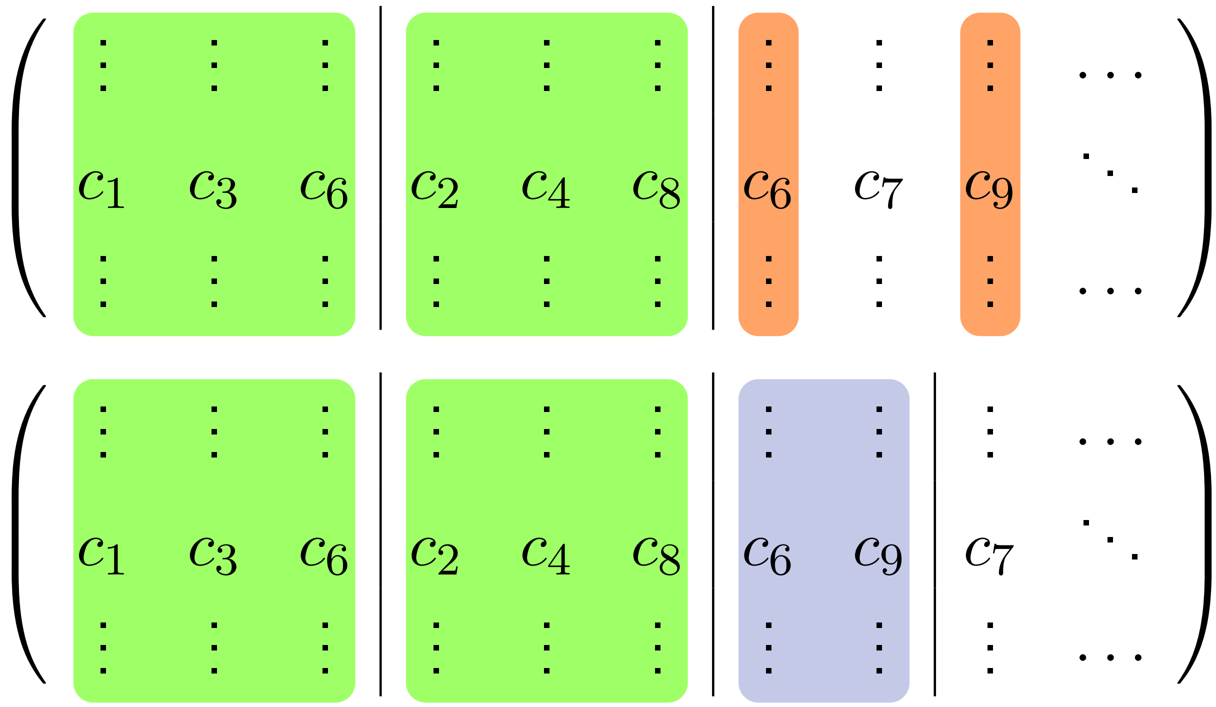
Step I. Preprocessing

If M has $r+1$ independent columns, at least one entry of these columns should be changed

If we obtained $k+1$ sets of $r+1$ independent columns, then the answer is NO

Step I. Preprocessing

If there are no r+1 independent columns, select a basis.



Preprocessing algorithm

Algorithm. USELESS-COLUMNS

INPUT : A, r, k.

1. *Let* $U \leftarrow A$.
2. *Repeat* at most $k+1$ times or till U is empty:
   - *If* $\text{rank}(U) \geq r+1$
     then delete $r+1$ linearly independent columns from U.
   - *If* $\text{rank}(U) \leq r$
     then delete a column basis from U.
3. *Output* U.

Step I. Preprocessing

- The resulting matrix U has at most $(k+1) \times (r+1)$ columns.
- U is a yes-instance if and only if M is.
- Proof: pigeonhole principle

Step I. Preprocessing

- Run the same preprocessing on rows, results in matrix with at most (k+1)x(r+1) columns and (k+1)x(r+1) rows.
- Is it a polynomial kernel?
- How to solve the problem?

Step II. Algorithm

Guess the span of matrix S from $M=L+S$ (this is $O(rk) \times O(rk)$ matrix with at most k non-zero elements). Total $rk^{O(r)}$ guesses. Each non-zero entry of S is a variable.

Matrix $M-S$ is of rank at most r if and only if its each $(r+1) \times (r+1)$ submatrix has zero determinant. (In total $rk^{O(r)}$ submatrices.)

↗

Polynomial equation with at most k variables and of degree at most k

Fact

Given a set $\mathcal{P}$ of ℓ polynomials of degree d in k variables with integer coefficients of bit length L, we can decide the feasibility of $\mathcal{P}$ with $L \log L \log\log L(\ell \cdot d)^{O(k)}$ bit operations.

Theorem. Robust PCA is solvable in time

$$\text{poly}(M)\, kr^{O(kr)}$$

We have to assume that matrix M is integer (or rational)

Exercise. Show that the following variant of Robust PCA is FPT parameterized by k and r.

Input: M, r, k

Task: Change at most k rows of M such that the resulting matrix is of rank at most r

PCA with outliers

ICML'19

Refined Complexity of PCA with Outliers

Fedor Fomin [*1] **Petr Golovach** [*1] **Fahad Panolan** [*1] **Kirill Simonov** [*1]

Abstract

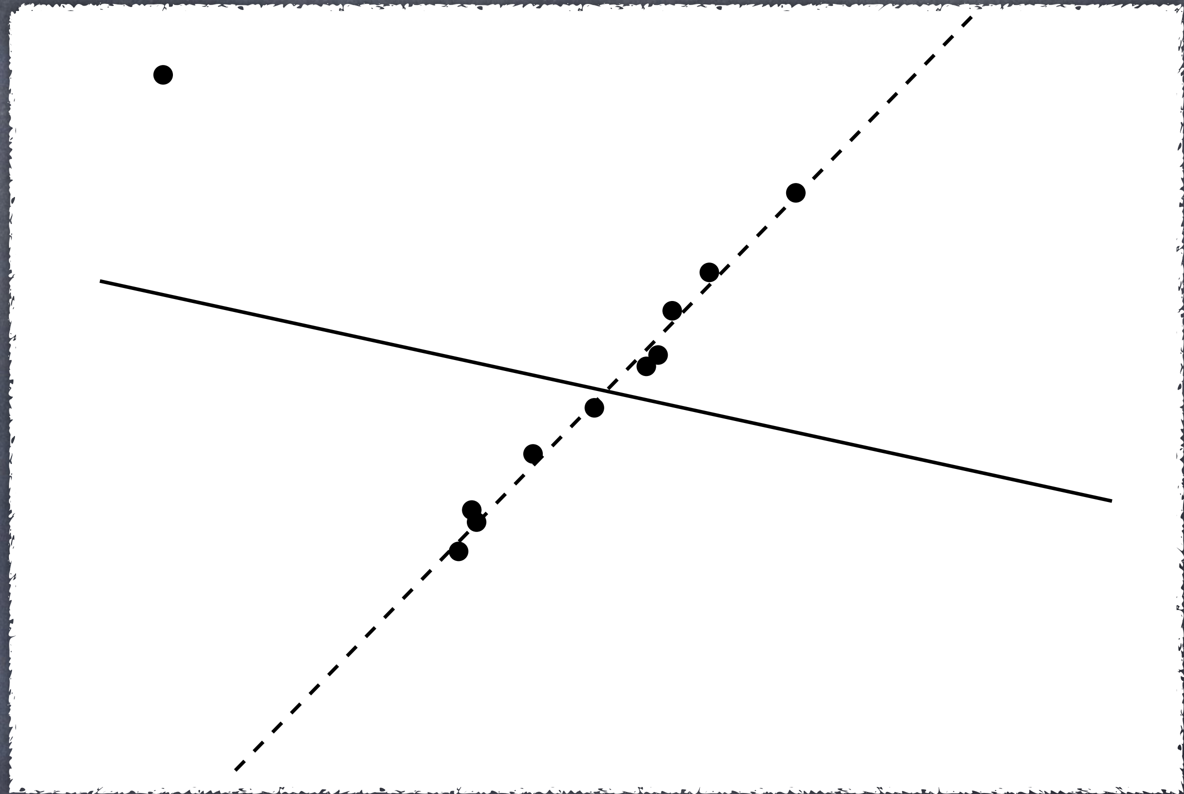
Principal component analysis (PCA) is one of the most fundamental procedures in exploratory data analysis and is the basic step in applications ranging from quantitative finance and bioinformatics to image analysis and neuroscience. However, it is well-documented that the applicability of PCA in many real scenarios could be constrained by an "immune deficiency" to outliers such as corrupted observations. We consider the following algorithmic question about the PCA with outliers. For a set of n points in $\mathbb{R}^d$, how to learn a subset of points, say 1% of the total number of points, such that the remaining part of the points is best

low-rank approximation of data matrix M by solving

$$\begin{aligned} &\text{minimize } \|M - L\|_F^2 \\ &\text{subject to } \operatorname{rank}(L) \le r. \end{aligned}$$

Here $||A||_F^2 = \sum_{i,j} a_{ij}^2$ is the square of the Frobenius norm of matrix A. By the Eckart-Young theorem (Eckart & Young, 1936), PCA is efficiently solvable via Singular Value Decomposition (SVD). PCA is used as a preprocess-

PCA with outliers



How to identify k points such that n-k remaining points fit well to an r-dimensional subspace?

Mathematical model

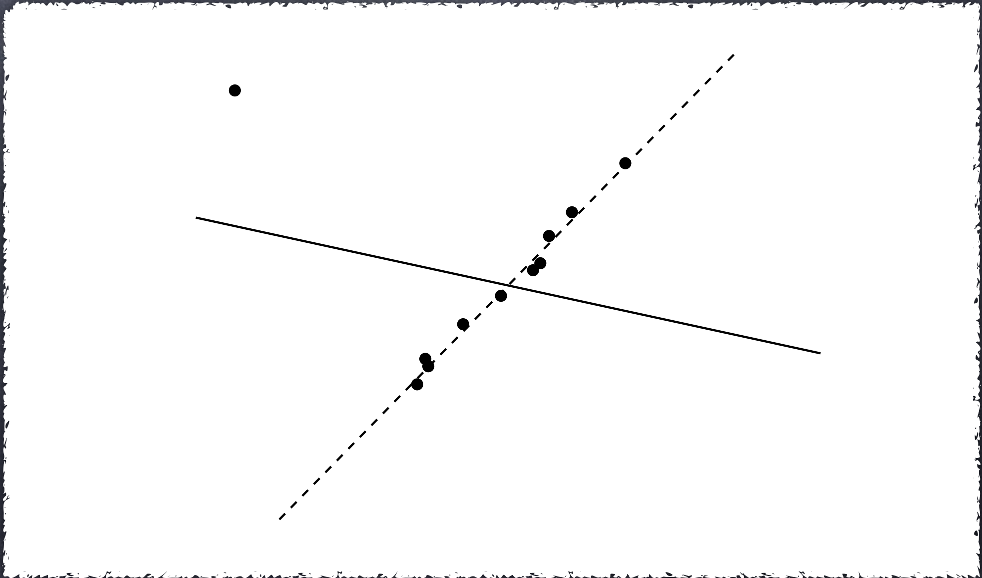
PCA with Outliers

Input: Data matrix $M \in \mathbb{R}^{n \times d}$, integer parameters r and k.

Task:

$$\begin{aligned} \text{minimize} \quad & \|M - L - S\|_F^2 \\ \text{subject to} \quad & L, S \in \mathbb{R}^{n \times d}, \\ & \operatorname{rank}(L) \leq r, \text{ and} \\ & S \text{ has at most } k \text{ non-zero rows.} \end{aligned}$$

How to identify k points such that n-k remaining points fit well to an r-dimensional subspace?



Related work

Robust PCA

sparse component

$$M = L + S$$

low rank

data matrix

[Candès, Li, and Wright, 2011]

[Chandrasekaran et al., 2011]

Noisy Robust PCA

sparse component

$$M = L + S + N$$

noise

low rank

data matrix

[Wright et al., 2009]

Theorem Robust PCA can be solved by solving $n^{O(d^2)}$ instances of PCA

PCA WITH OUTLIERS

Input: Data matrix $M \in \mathbb{R}^{n \times d}$, integer parameters r and k.

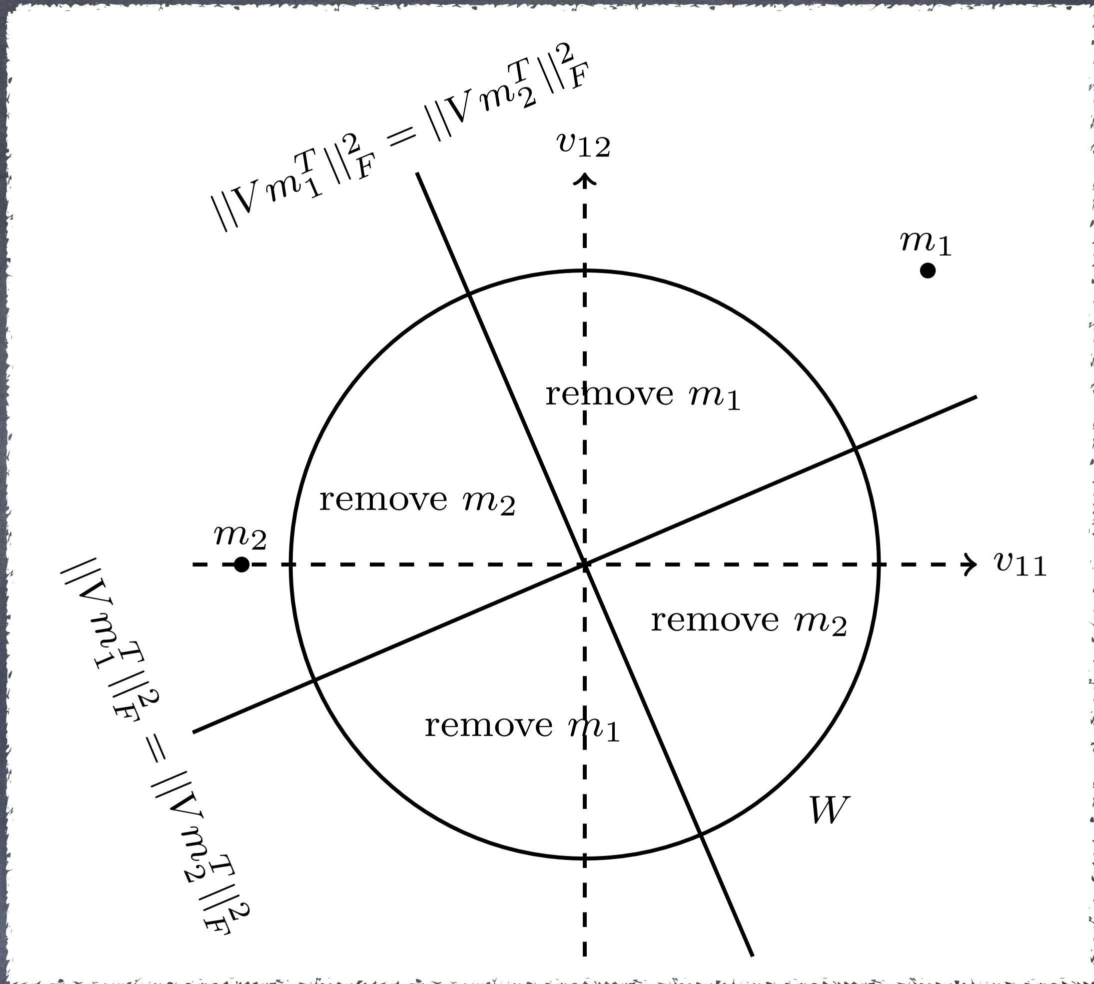
Task:

$$\begin{aligned} \text{minimize} \quad & \|M - L - S\|_F^2 \\ \text{subject to} \quad & L, S \in \mathbb{R}^{n \times d}, \\ & \text{rank}(L) \leq r, \text{ and} \\ & S \text{ has at most } k \text{ non-zero rows.} \end{aligned}$$

Sketch of the proof

Intuition:

d=2, k=1, r=1



Sketch of the proof

unit vector defining r-dimensional subspace

$$P_{i,j}(V) = ||Vm_i^T||_F^2 - ||Vm_j^T||_F^2$$

algebraic set W

polynomials

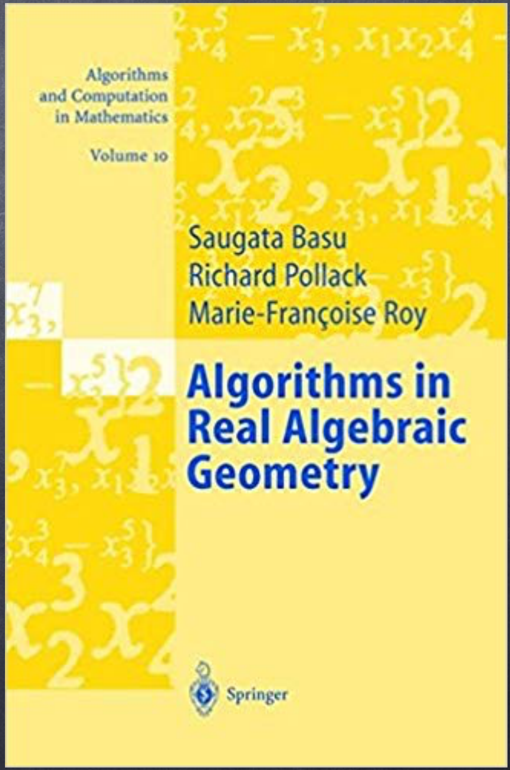
Cell $C \subset W$ s.t. for all V in C the signs of all polynomials do not change

For each cell C take some V from C, construct orthogonal subspace, remove the furthermost k points, run PCA on remaining n-k points

Sketch of the proof

The number of times we call PCA is proportional to the number of cells

The number of cells is $n^{O(d^2)}$ and the cells can be constructed within the same running time



Natural questions

Is the running time $n^{O(d^2)}$ the best we can hope for?

Could PCA with outliers be solved in polynomial time?

Could PCA with outliers be solved in time $f(d)n^{O(1)}$?

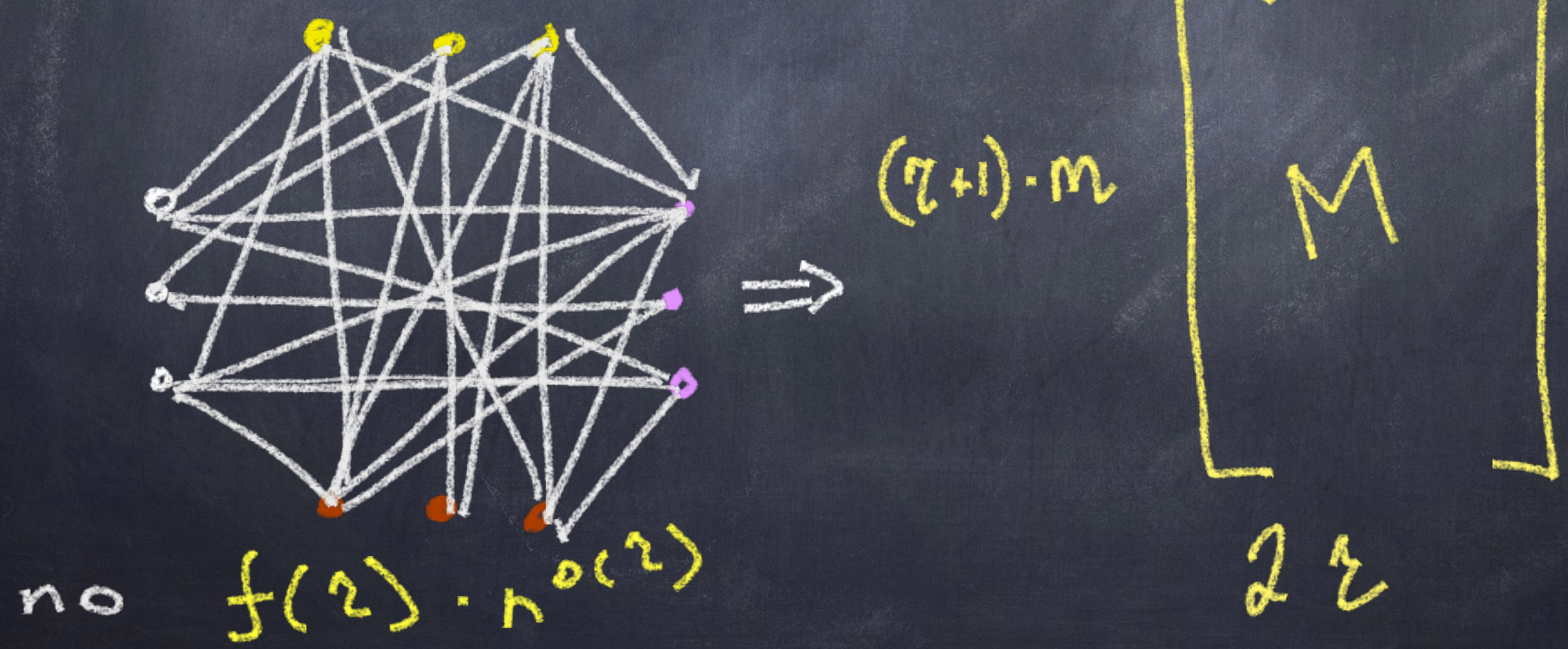
Exponential Time Hypothesis (ETH)

(ETH) 3-SAT with n variables and m clauses cannot be solved in time $2^{o(n+m)}$

[Impagliazzo, Paturi, and Zane, 2001]

Theorem. For any $c>1$, there is no c-approximation algorithm for **PCA With Outliers** with running time $f(d)\, N^{o(d)}$ for any computable function f unless ETH fails, where N is the bitsize of the input matrix M.

Proof: Reduction from **Multicolored Clique**



PCA with outliers

is solvable in time $|M|^{O(d^2)}$

cannot be solved in time $f(d)|M|^{o(d)}$ unless ETH fails

Conclusion

- Robust PCA FPT parametrized by d and k
- PCA with outliers: $n^{f(d)}$ algorithm

Open questions

- Robust PCA in time $n^{f(d)}$
- PCA with outliers $f(r)$-approximation in poly time