Learning Sparsely used Overcomplete Dictionaries

Prateek Jain

Microsoft Research, India

Joint work with Alekh Agarwal, Animashree Anandkumar, Praneeth Netrapalli, Rashish Tandon

Dictionary Learning

y

$\cong$

m

Data Point

A

$d \times r$

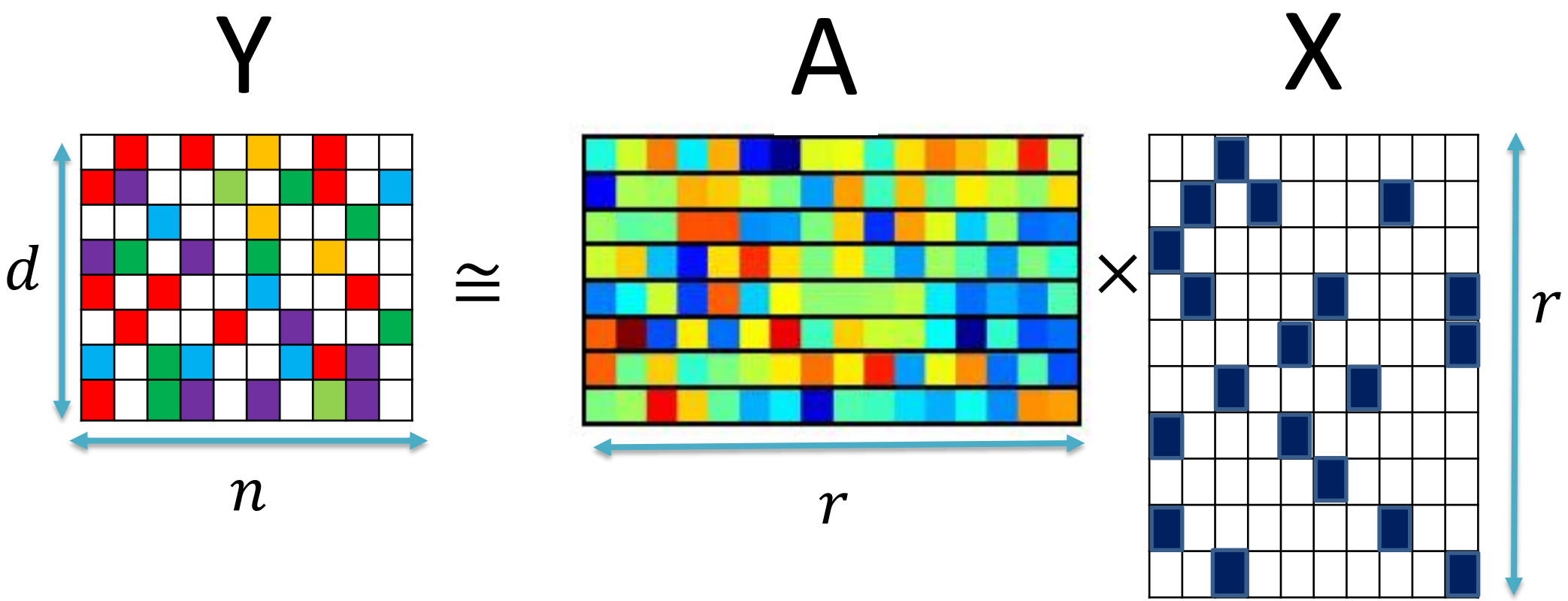
Dictionary

$\times$

x

r-dim, k-sparse vector

Dictionary Learning



- Overcomplete dictionaries: $r \gg d$
- Goal: Given Y, compute A, X
  - Using small number of samples n

Existing Results

- Generalization error bounds [VMB'11, MPR'12, MG'13, TRS'13]
  - But assumes that the optimal solution is reached
  - Do not cover exact recovery with finite many samples
- Identifiability of A, X [HS'11]
  - Require exponentially many samples
- Exact recovery [SWW'12]
  - Restricted to square dictionary $(d = r)$
  - In practice, overcomplete dictionary $(d \ll r)$ is more useful
    - Concurrent result by AGM13

Generating Model

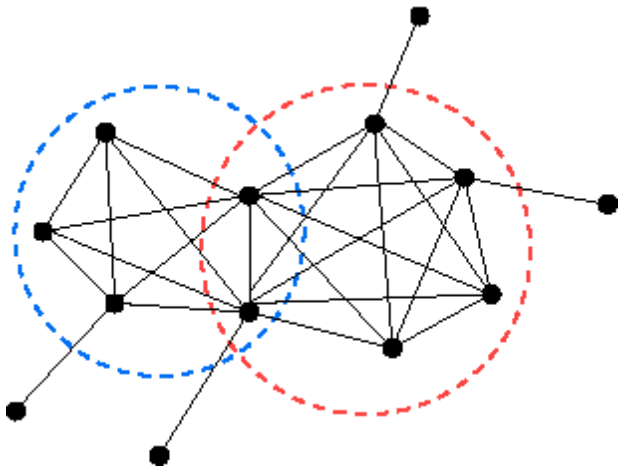
- Generate dictionary A
  - Assume A to be incoherent, i.e., $\langle A_i, A_j \rangle \leq \mu/\sqrt{d}$
  - $r \gg d$
- Generate random samples $X = [x_1, x_2, \ldots, x_n] \in R^{d \times n}$
  - Each x_i is k-sparse
- Generate observations: $Y = AX$

Algorithm

- Typically practical algorithm: alternating minimization
  - $X_{t+1} = argmin_X ||Y - A_t X||_F^2$
  - $A_{t+1} = argmin_A ||Y - A X_{t+1}||_F^2$
- Initialize A_0
  - Using clustering+SVD method of [AAN'13] or [AGM'13]

Initialization Method

- Key idea:
  - $\langle Y_i, Y_j \rangle = \langle AX_i, AX_j \rangle \geq \rho \rightarrow$ Share a common element
  - X_i: random k-sparse vector
  - Collect enough points sharing a common element A_1
    - Clustering
  - Use SVD to approximate A_1



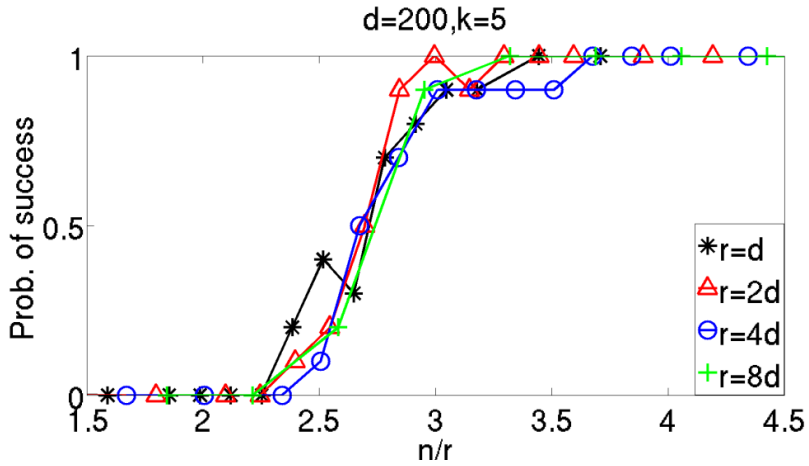
Results [AAJNT'13]

- Assumptions:
  - A is $\mu-$ incoherent ($\langle A_i, A_j \rangle \leq \mu/\sqrt{d}, ||A_i|| = 1$)
  - $1 \leq |X_{ij}| \leq 100$
  - Sparsity: $k \leq \frac{d^{\frac{1}{6}}}{\mu^{\frac{1}{3}}}$ (AGM13: $k \leq O(\sqrt{d})$)
  - $n \geq \tilde{O}(r^2)$ (AGM13: $n = \tilde{O}(r^2 \log 1/\epsilon)$)
    - For initialization $n = \tilde{O}(r)$ (AGM13: $n = \tilde{O}(r^2)$)
- After $\log(\frac{1}{\epsilon})$-steps of AltMin:

$$||A_T^i - A^i||_2 \leq \epsilon$$

Summary

- Dictionary Learning
  - Novel initialization method
  - Exact recovery for alternating minimization
- Future Work:
  - Sample complexity: $n = O(r^2 \log r) \Rightarrow n = O(r \log r)$?



Please visit the poster for more details!

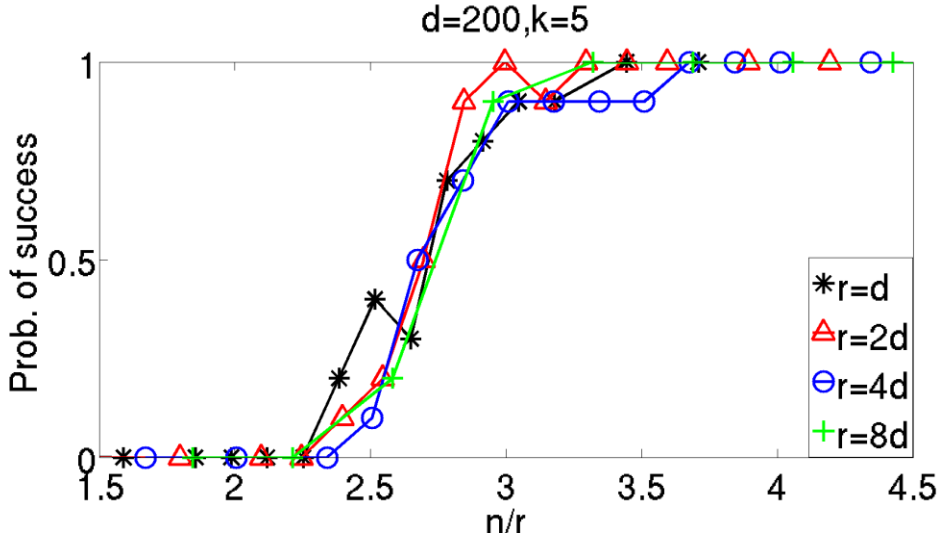
Proof Sketch

- Initialization step ensures that:

$$||A^i - A_0^i|| \leq \frac{1}{k^2}$$

- Lower bound on each element of X_{ij} + above bound:
  - $supp(x_i)$ is recovered exactly
  - Robustness of compressive sensing!
- A_{t+1} can be expressed exactly as:
  - $A_{t+1} = A + Error_{(A_t, X_t)}$
  - Use randomness in $supp(X_t)$

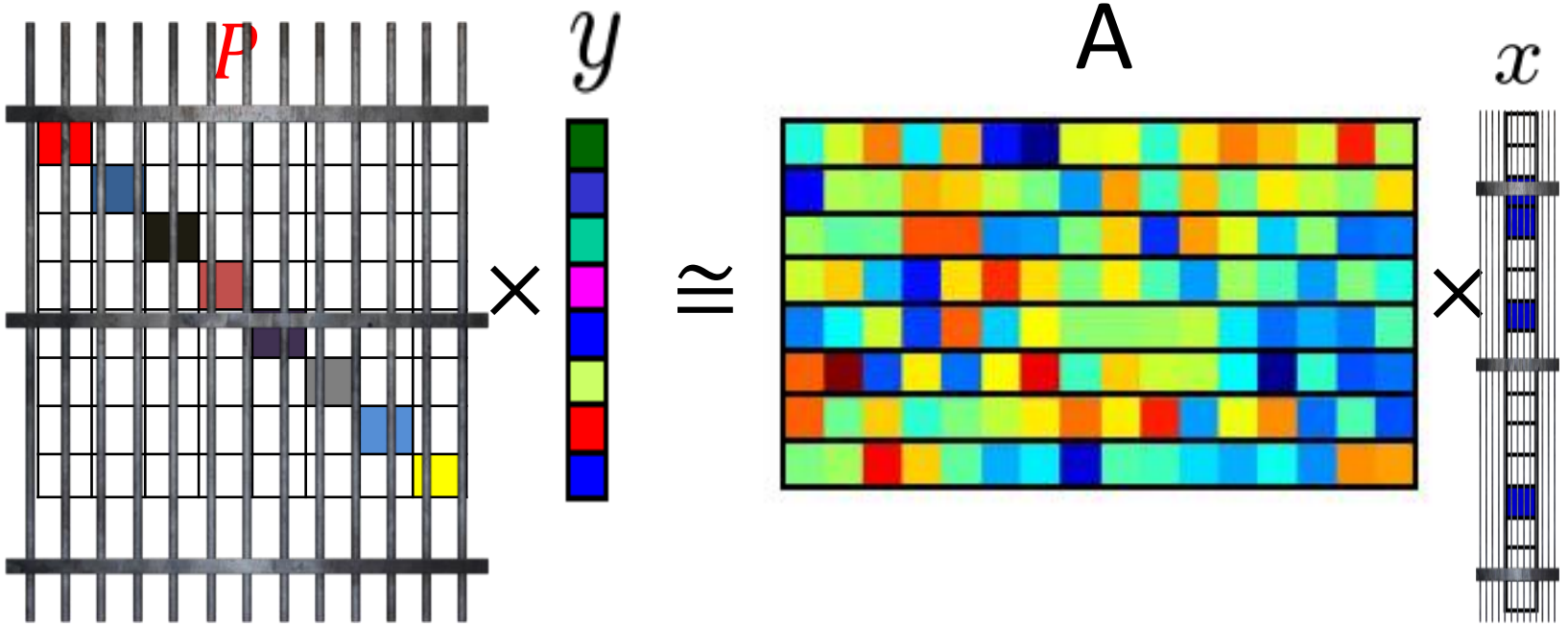
Simulations



Emirically: $n = O(r)$

Known result: $n = O(r^2 \log r)$

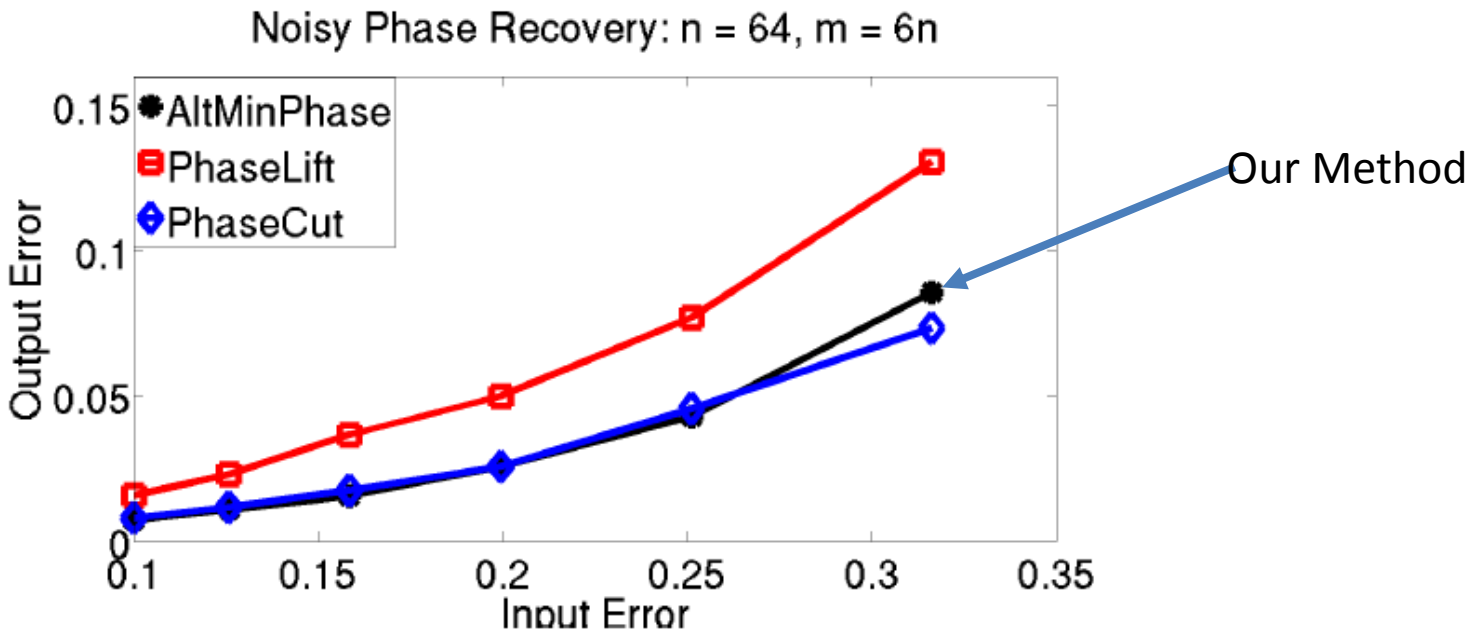
Alternating Minimization



$$x = \arg\min_x \|Py - Ax\|^2$$

$$P_{ii} = \text{Phase}(\langle a_i, x \rangle)$$

Empirical Results



- Smaller is better

Existing method: Trace-norm minimization

$$\min_{X} \sum_{(i,j)\in\Omega} (X_{ij} - M_{ij})^2$$

$$\text{s.t.} \quad \text{rank}(X) \leq k$$

- $||X||_*$: sum of singular values
- Candes and Recht prove that above problem solves matrix completion (under assumptions on Ω and M)
- However, convex optimization methods for this problem don't scale well

Our Results

- Crucial observation: Alternating minimization is similar to power method but with an "error term"
  - Power-method: a basic method to compute singular value decomposition
- Assumptions: Ω: set of known entries
  - Ω is sampled uniformly s.t. $|\Omega| = O(k^5 n \log n \ \beta^2)$
    - $\beta = \sigma_1/\sigma_k$
  - M: rank-k "incoherent" matrix
    - Most of the entries are similar in magnitude