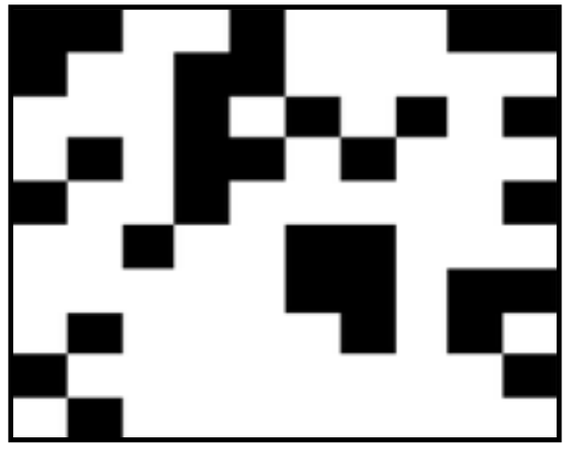
SGD for large-scale SDP

Benjamin Recht
University of California, Berkeley

Matrix Completion

M =



M_{ij} known for black cells

M_{ij} unknown for white cells

Rows index questions

Columns index users

- How do you fill in the missing data?

M = **L** **R***

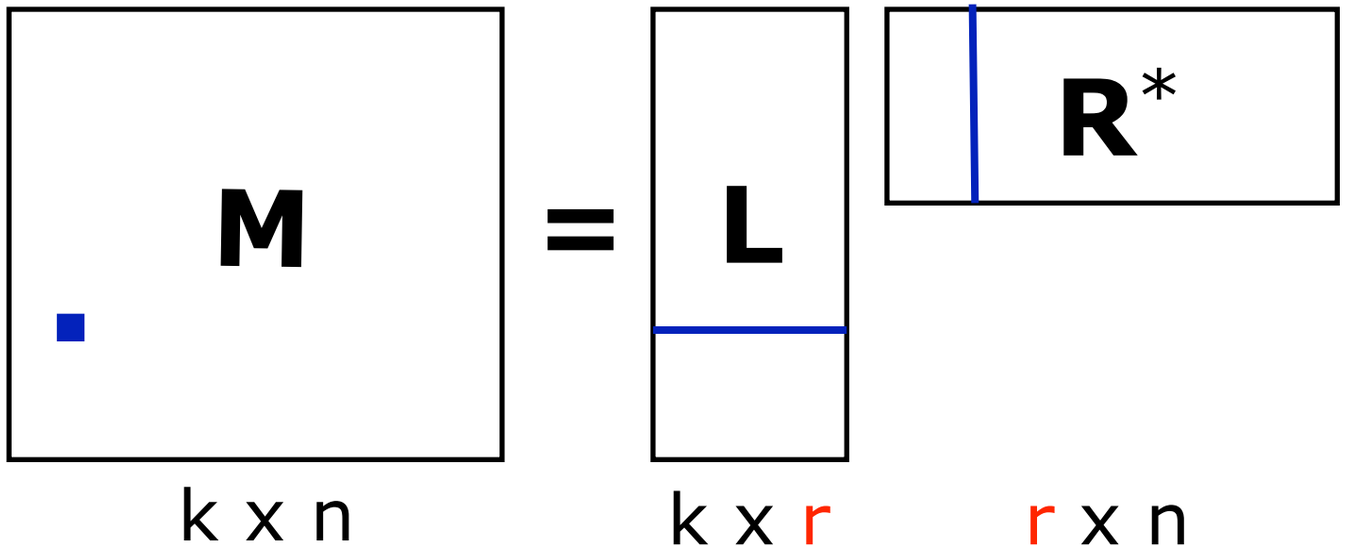
$p_1 \times p_2$ $p_1 \times r$ $r \times p_2$

$p_1 p_2$ entries $r(p_1+p_2)$ entries

Heuristic for Matrix Completion

IDEA: Replace *rank* with *nuclear norm*:

$$\begin{aligned} \text{minimize} \quad & \|\mathbf{X}\|_* \\ \text{subject to} \quad & \Phi(\mathbf{X}) = \mathbf{b} \end{aligned}$$



- Step 1: Pick (i,j) and compute residual:

$$e = (\mathbf{L}_i \mathbf{R}_j^T - M_{ij})$$

- Step 2: Take a mixture of current model and corrected model (α,β>0):

$$\begin{bmatrix} \mathbf{L}_i \\ \mathbf{R}_j \end{bmatrix} \leftarrow \begin{bmatrix} \alpha\mathbf{L}_i - \beta e \mathbf{R}_j \\ \alpha\mathbf{R}_j - \beta e \mathbf{L}_i \end{bmatrix}$$

Succeeds when number of samples is Õ(r(k+n))

Some guy on livejournal, 2006
Fazel, Parillo, Recht, 2007
Candes and Recht, 2008

Equivalent Formulations

$$\begin{array}{ll} \text{minimize} & \|X\|_* \\ \text{subject to} & \mathcal{A}(X) = b \end{array} \quad \Leftrightarrow \quad \begin{array}{ll} \text{minimize} & \sum_{i=1}^{k} \sigma_i(X) \\ \text{subject to} & \mathcal{A}(X) = b \end{array}$$

- Semidefinite embedding:

$$X = U \Sigma V^*$$

$$\begin{bmatrix} W_1 & X \\ X^* & W_2 \end{bmatrix} = \begin{bmatrix} U \\ V \end{bmatrix} \Sigma \begin{bmatrix} U \\ V \end{bmatrix}^*$$

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}(\mathrm{Tr}(W_1) + \mathrm{Tr}(W_2)) \\ \text{subject to} & \begin{bmatrix} W_1 & X \\ X^* & W_2 \end{bmatrix} \succeq 0 \\ & \mathcal{A}(X) = b \end{array}$$

- Low rank parametrization:

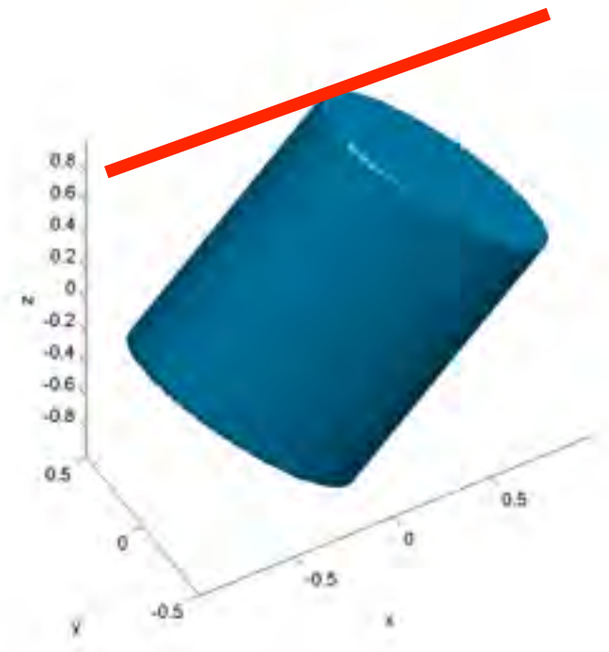
$$L = U \Sigma^{1/2}$$

$$R = V \Sigma^{1/2}$$

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}(\|L\|_F^2 + \|R\|_F^2) \\ \text{subject to} & \mathcal{A}(LR^*) = b \end{array}$$

Nuclear Norm minimization

$$\begin{aligned} \text{minimize} \quad & \|X\|_* = \sum_{i=1}^{k} \sigma_i(X) \\ \text{subject to} \quad & \Phi(X) = y \end{aligned}$$



Low-rank parameterization

$$\begin{aligned} \text{minimize} \quad & \tfrac{1}{2}(\|L\|_F^2 + \|R\|_F^2) \\ \text{subject to} \quad & \Phi(LR^*) = y \end{aligned}$$

$$X = U\Sigma V^*$$

$$L = U\Sigma^{1/2}$$

$$R = V\Sigma^{1/2}$$

Method of Multipliers

$$\text{minimize} \sum_{i=1}^{k}\sum_{a=1}^{r} L_{ia}^2 + \sum_{j=1}^{n}\sum_{a=1}^{r} R_{ja}^2 + \lambda\|\Phi(LR^*) - y\|_2^2$$

Theoretical Foundation

$$\text{minimize} \sum_{i=1}^{k}\sum_{a=1}^{r} L_{ia}^2 + \sum_{j=1}^{n}\sum_{a=1}^{r} R_{ja}^2 + \lambda\|\Phi(LR^*) - y\|_2^2$$

- Any local minimum of this nonlinear program is a global optimum of the nuclear norm minimization problem provided the nuclear norm has a rank smaller than r.

Burer and Monteiro, 2005

JELLYFISH

- SGD for Matrix Factorizations.

Example: $\text{minimize} \quad \sum_{(i,j)\in\Omega}(X_{ij} - M_{ij})^2 + \mu\|\mathbf{X}\|_*$

- Idea: approximate $\mathbf{X} \approx \mathbf{L}\mathbf{R}^T$

$$\text{minimize}_{(\mathbf{L},\mathbf{R})} \sum_{(i,j)\in\Omega} \left\{(\mathbf{L}_i\mathbf{R}_j^T - M_{ij})^2 + \mu_i\|\mathbf{L}_i\|_F^2 + \mu_j\|\mathbf{R}_j\|_F^2\right\}$$

- Step 1: Pick (i,j) and compute residual:

$$e = (\mathbf{L}_i\mathbf{R}_j^T - M_{ij})$$

- Step 2: Take a stochastic gradient step:

$$\begin{bmatrix} \mathbf{L}_i \\ \mathbf{R}_j \end{bmatrix} \leftarrow \begin{bmatrix} \alpha\mathbf{L}_i - \beta e\mathbf{R}_j \\ \alpha\mathbf{R}_j - \beta e\mathbf{L}_i \end{bmatrix}$$

JELLYFISH

Observation: With replacement sample=poor locality

Idea: Bias sample to improve locality.

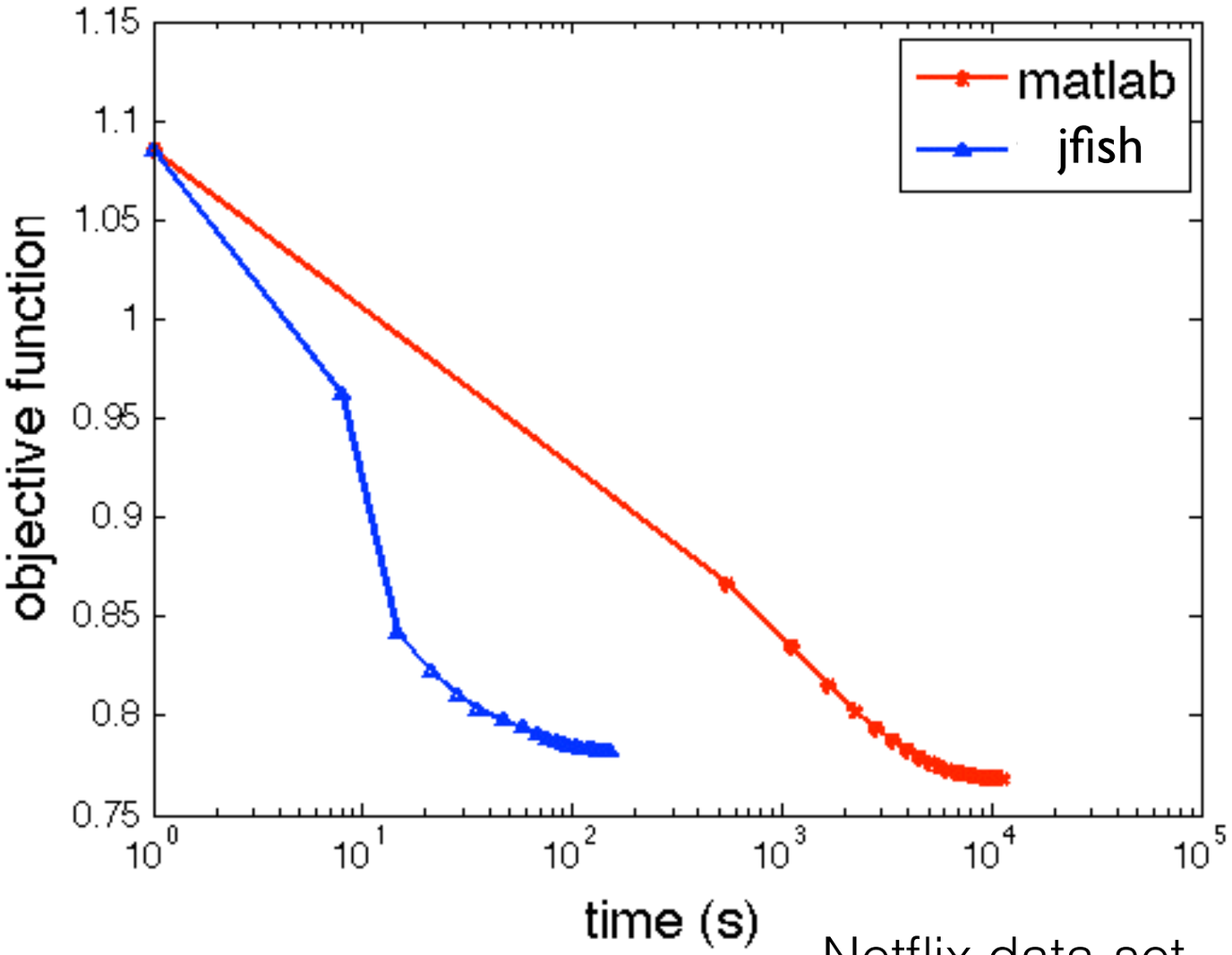
$$\begin{bmatrix} L_1 \\ L_2 \\ L_3 \end{bmatrix} \begin{bmatrix} R_1 & R_2 & R_3 \end{bmatrix} = \begin{bmatrix} L_1R_1 & L_1R_2 & L_1R_3 \\ L_2R_1 & L_2R_2 & L_2R_3 \\ L_3R_1 & L_3R_2 & L_3R_3 \end{bmatrix}$$

Algorithm: Shuffle the data.

1. Process $\{L_1R_1, L_2R_2, L_3R_3\}$ in parallel
2. Process $\{L_1R_2, L_2R_3, L_3R_1\}$ in parallel
3. Process $\{L_1R_3, L_2R_1, L_3R_2\}$ in parallel

Big win: No locks!
(model access)

Recht and Ré, 2010



Netflix data-set
100M examples
17770 rows
480189 columns

- 100x faster than standard solvers
- 25% speedup over HOGWILD! on 12 core machine
- 3x speedup on 40 core machine

Extensions

- Other structured matrix problems
  - max-norm
  - NMF (with non-negativity or simplex constraint)
    - LDA?

- Decoupling works for any algorithm where we can project the rows independently
- However, those not arising from SDPs have anything resembling guarantees.

Quadratic Inverse Problems

- Blind deconvolution
- Phase retrieval
- Dictionary learning
- Nonnegative matrix decomposition

Quadratic Inverse Problems

- Blind deconvolution
- Phase retrieval
- Dictionary learning
- Nonnegative matrix decomposition

$$I_j = \sum_k s_k h_{j-k}$$

Quadratic Inverse Problems

- Blind deconvolution
- Phase retrieval
- Dictionary learning
- Nonnegative matrix decomposition

$$I_j = |f_j^* s|^2 = s^* f_j f_j^* s$$

Quadratic Inverse Problems

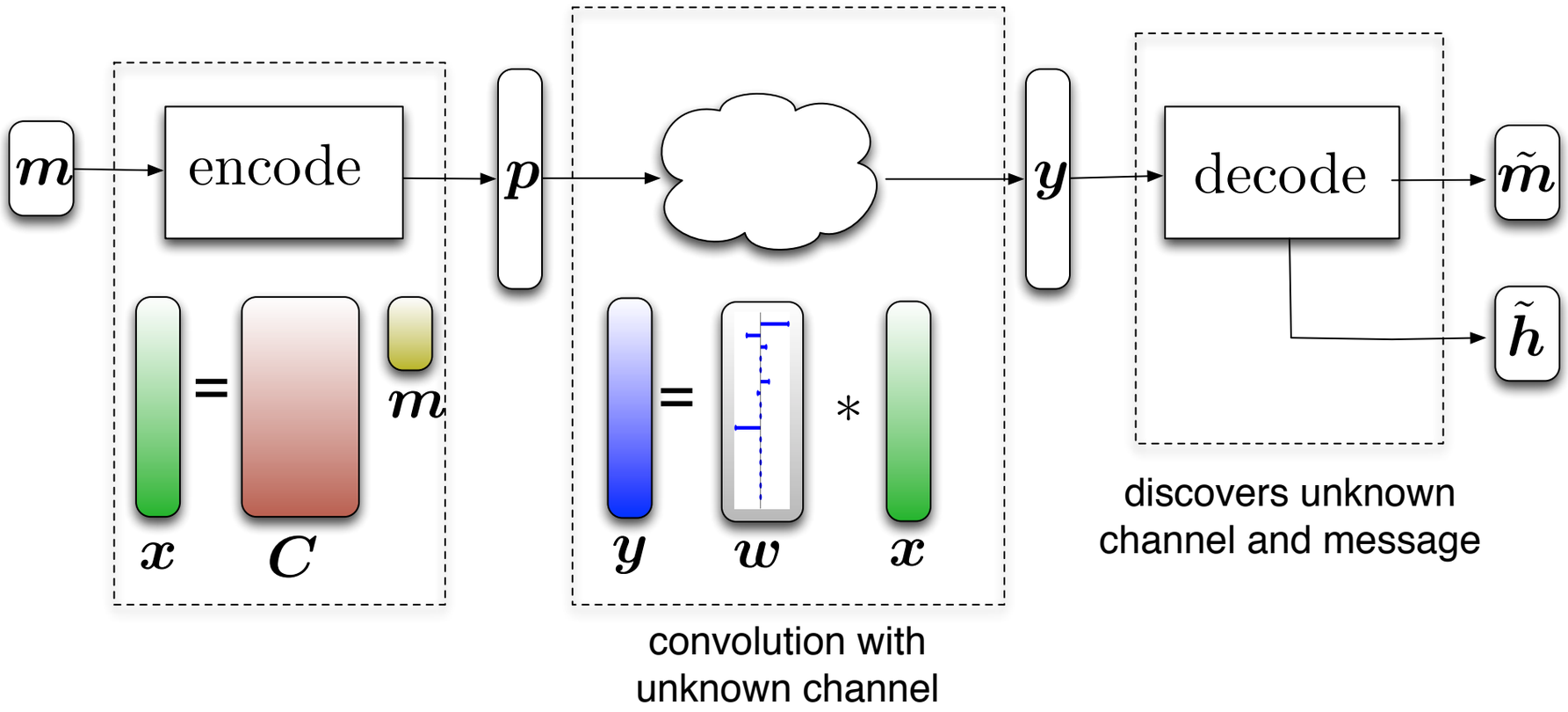
- Blind deconvolution
- Phase retrieval
- Dictionary learning
- Nonnegative matrix decomposition

$$X_{ij} = \sum_k D_{ik} C_{kj}$$

Challenges and Ideas

- Extremely ill-posed
  - ✦ Impose additional structures
  - ✦ Diversify measurements
- Nonlinear, non-convex
  - ✦ Lift to a high-dimensional space
  - ✦ Enforce low-rankness

Blind Deconvolution with Subspace Knowledge

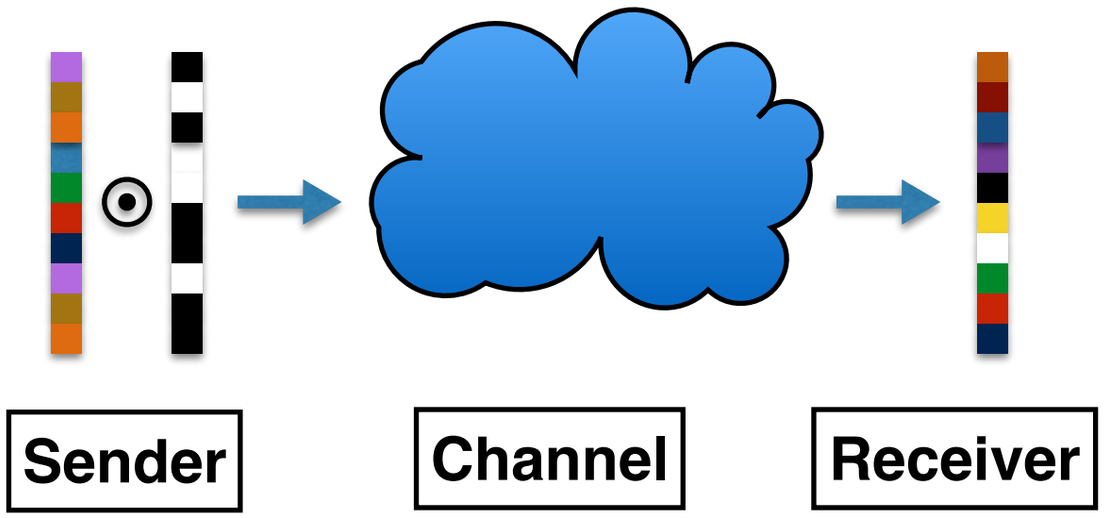


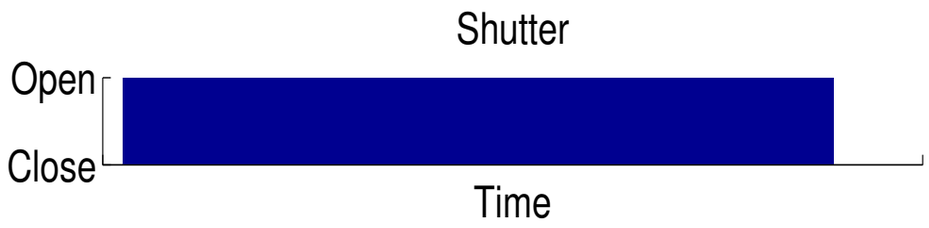
Ahmed, R, and Romberg, 2012

Blind Deconvolution with Random Masks

$$I^i = s * (m^i \odot h)$$

$$\text{or } (m^i \odot s) * h$$





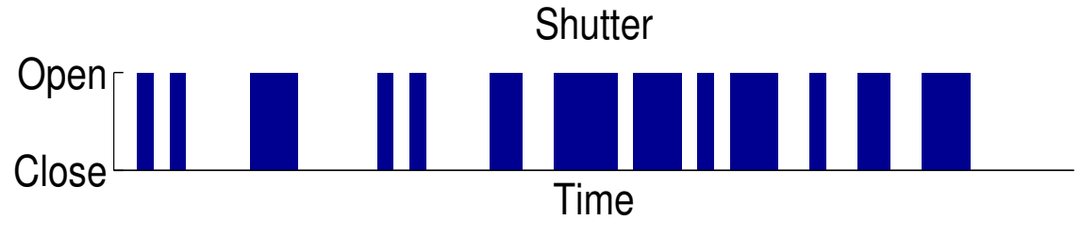


Image Courtesy: Raskar, Agrawal, Tumblin

Lifting + Low-Rank

$$I_j = \sum_k m_{j-k} s_k h_{j-k}$$

$$X = hs^T$$

$$\begin{bmatrix} I_1 \\ I_2 \\ I_3 \\ I_4 \\ \vdots \\ I_n \end{bmatrix} = \begin{bmatrix} m_1 & 0 & 0 & \cdots & 0 \\ 0 & m_2 & 0 & \cdots & 0 \\ 0 & 0 & m_3 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & m_n \end{bmatrix} \begin{bmatrix} s_1h_1 & s_2h_1 & s_3h_1 & \cdots & s_nh_1 \\ s_1h_2 & s_2h_2 & s_3h_2 & \cdots & s_nh_2 \\ s_1h_3 & s_2h_3 & s_3h_3 & \cdots & s_nh_3 \\ s_1h_4 & s_2h_4 & s_3h_4 & \cdots & s_nh_4 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ s_1h_n & s_2h_n & s_3h_n & \cdots & s_nh_n \end{bmatrix}$$

find X

subject to

$$I_j^i = \text{trace}(H^{jT} \,\text{diag}(m^i) X)$$

$$\text{rank}(X) = 1$$

minimize $\|X\|_*$

subject to

$$I_j^i = \text{trace}(H^{jT} \,\text{diag}(m^i) X)$$

Theoretical Foundation

$$\begin{aligned} \text{minimize} \quad & \|h\|_F^2 + \|s\|_F^2 \\ \text{subject to} \quad & I_j^i = \text{trace}\left(H^{j^T} \text{diag}(m^i) h s^T\right) \end{aligned}$$

- Any local minimum of this nonlinear program is a global optimum of the nuclear norm minimization problem provided the nuclear norm has a rank 1 solution.

Burer and Monteiro, 2005

Super-resolution

- Image size: 256x256
- filter size: 16x16
- channels: 20

SGD for large SDP

- Burer and Monteiro lifts
  - Make large SDPs practically solvable
  - Still gaps in our theoretical understanding: stationary points, initializations, stochastic convergence
- SGD with biased orderings
  - Bias for locality and speed
  - Little understanding of the theory of nonuniform orderings