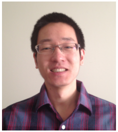
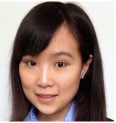
Fast algorithm for low-rank matrix recovery in Poisson noise

Yao Xie
Joint work with **Yang Cao**

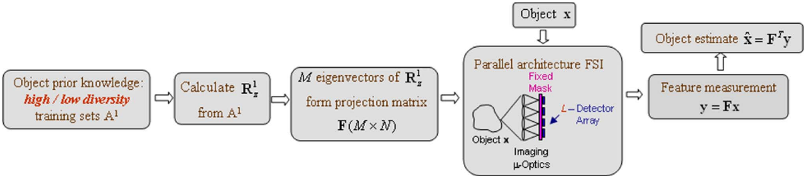
 

H. Milton Stewart School of Industrial and Systems Engineering

Georgia Institute of Technology

Presented at GlobalSIP 2014

Static feature-specific imaging (SFSI)



Formulation

- Goal: recover a matrix $M^* \in \mathbb{R}_+^{m_1 \times m_2}$
- Observation: N linear measurements with Poisson noise

$$y_i \sim \text{Poisson}([\mathcal{A}M^*]_i), \quad i = 1, \ldots, N,$$

- linear operator $\mathcal{A} : \mathbb{R}_+^{m_1 \times m_2} \to \mathbb{R}^N$ models the measuring process of physical devices

$$[\mathcal{A}M]_i = \langle A_i, M \rangle \triangleq \text{tr}(A_i^\top M),$$

$A_i \in \mathbb{R}^{m_1 \times m_2}$

Connection with earlier work

- matrix completion:
  - noiseless $y_i = [\mathcal{A}M]_i$ [Candes, Recht 2009],
  - with Gaussian measurement noise $y_i = [\mathcal{A}M]_i + w_i$ [Candes, Yaniv 2009] [Cai, Candes, Shen 2010]
- compressed sensing with Poisson noise [Raginsky et.al. 2010]

$$y_i \sim \mathrm{Poisson}([Ax]_i)$$

- feasibility study and performance bound for matrix recovery with Poisson noise [Xie, Chi, Calderbank 2013]
- low-rank signal recovery with Poisson noise (directly observe entries) [Soni, Haupt 2014]

$$Y_{ij} \sim \mathrm{Poisson}([M]_{ij}), \quad \{M \text{ is low-rank}\}$$

- this work: efficient algorithm for matrix recovery with Poisson noise

Regularized maximum-likelihood estimator

$$\begin{aligned}\widehat{M} &\triangleq \arg\min_{M\in\Gamma}[-\log p(y|\mathcal{A}M) + \lambda\rho(M)] \\ &= \arg\min_{M\in\Gamma}[\underbrace{-\sum_{j=1}^{N} y_j \log[\mathcal{A}M]_j - [\mathcal{A}M]_j}_{f(M)} + \lambda\rho(M)]\end{aligned}$$

- $\rho(M) > 0$: regularization function
- $\lambda > 0$: regularization parameter
- Γ: set of feasible estimators

Assumptions

- Total intensity of M^* is known a priori

$$I \triangleq \|M^*\|_{1,1},$$

where $\|X\|_{1,1} = \sum_i \sum_j [X]_{ij}$

- Positivity-preserving of $\mathcal{A}$: $[M]_{ij} \geq 0$ for all i, $j \Rightarrow [\mathcal{A}M]_i \geq 0$, for all i

- Flux-preserving of $\mathcal{A}$: $\sum_{i=1}^{N} [\mathcal{A}M]_i \leq \|M\|_{1,1}$

Sensing operator

- Linear sensing operator $\mathcal{A}$,

$$[A_i]_{jk} = \begin{cases} 0, & \text{with probability } p; \\ 1/N, & \text{with probability } 1-p. \end{cases}$$

- satisfies earlier assumptions, and the restrictive isometry property (RIP)

[Raginsky, Willett, Harmany, Marcia 2010], [Xie, Chi, Calderbank 2013]

Optimization problem

- $\rho(M) = \|M\|_*$
- Optimization problem

$$\min_{M \in \Gamma_0} f(M) + \lambda \|M\|_*,$$

where $f(M) = -\log p(y|\mathcal{A}M)$

$$\Gamma_0 \triangleq \{M \in \mathbb{R}_+^{m_1 \times m_2} : \|M\|_{1,1} = I\}.$$

- Convex optimization: Semidefinite program (SDP)
- More efficient algorithm: **eigenvalue thresholding**

Taylor expansion and approximation

- Taylor expansion of log-likelihood function at $(k-1)$th solution

$$
\begin{aligned}
Q_{t_k}(M, M_{k-1}) &\triangleq f(M_{k-1}) + \langle M - M_{k-1}, \nabla f(M_{k-1}) \rangle \\
&+ \frac{t_k}{2} \| M - M_{k-1} \|_F^2, \\
&\propto \frac{t_k}{2} \left\| M - \left(M_{k-1} - \frac{1}{t_k} \nabla f(M_{k-1}) \right) \right\|_F^2
\end{aligned}
$$

where t_k is the step size at kth iteration

Solution for the next iteration

$$M_k = \arg\min_M \left[\frac{1}{2} \left\| M - \left(M_{k-1} - \frac{1}{t_k} \nabla f(M_{k-1}) \right) \right\|_F^2 + \frac{\lambda}{t_k} \|M\|_* \right].$$

Theorem (Cai, Candes, Shen 2010)

For each $\tau \geq 0$, *and* $X \in \mathbb{R}^{n_1 \times n_2}$*:*

$$D_\tau(X) = \arg \min_{Y \in \mathbb{R}^{n_1 \times n_2}} \left\{ \frac{1}{2} \|Y - X\|_F^2 + \tau \|Y\|_* \right\}. \qquad (1)$$

$$D_\tau(X) \triangleq U D_\tau(\Sigma) V^T,$$

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

Solution of the approximate optimization problem

- Solution given by **Singular Value Thresholding (SVT)**

$$M_k = D_{\lambda/t_k}\left(M_{k-1} - \frac{1}{t_k}\nabla f(M_{k-1})\right).$$

Poisson noise Maximal Likelihood Singular Value thresholding (PMLSV) algorithm

Algorithm 1 PMLSV

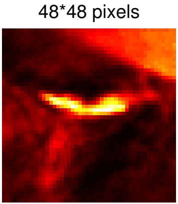
1: Initialize: $M_0 = \mathcal{P}(\sum_{i=1}^{n} y_i A_i)$, parameter γ, step size L
2: **for** $k = 1, 2, \ldots NOI$ **do**
3: $\quad \mathcal{G}(M_{k-1}) := \nabla[-\log p(y|\mathcal{A}M_{k-1})]$
4: $\quad C := M_{k-1} - \frac{1}{L}\mathcal{G}(M_{k-1})$
5: $\quad$ singular value decomposition: $C := UDV^T$
6: $\quad D_{\text{new}} := \text{diag}((\text{diag}(D) - \frac{\lambda}{L})_+)$
7: $\quad W_k := \mathcal{P}(UD_{\text{new}}V^T)$.
8: $\quad$ If $F(M_k) < F(M_{k-1})$, then $k = k+1$; else $L = \gamma L$, go to 6.
9: $\quad$ If $|F(M_k) - F(M_{k-1})| < 0.5/NOI$, then $k = k-1$, exit;
10: **end for**

Numerical examples

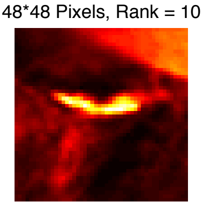
- normalized risk:

$$R(M^*, M) \triangleq \frac{1}{I^2} \|M^* - M\|_F^2.$$

- Parameters are as follows : $I = 2.37 \times 10^7$, $L = 10^{-5}$, $\gamma = 1.1$, and $NOI = 2500$.

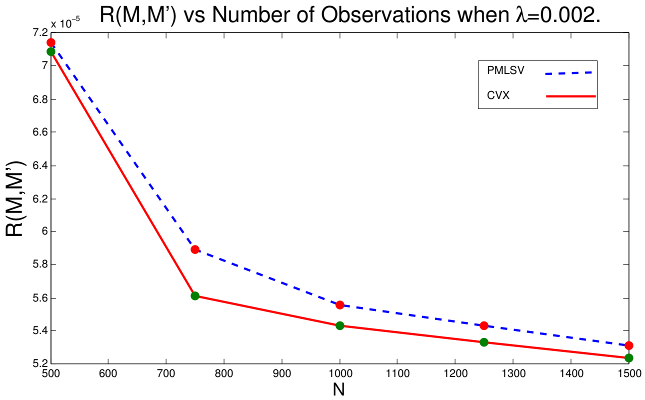


(a) original solar flare image



(b) solar flare image with rank 10

Solution quality compared with SDP



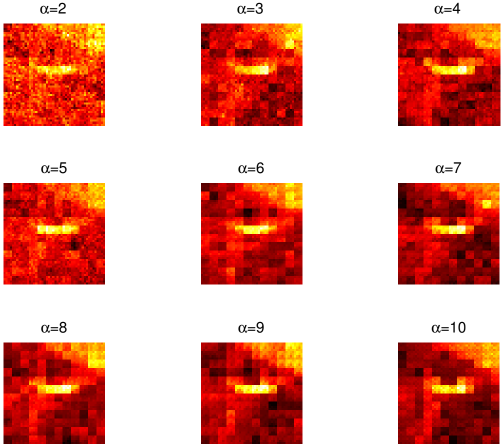
Computation time

Table 1: CPU time (in seconds) of solving SDP by using CVX and our PMLSV algorithm when fixing $\alpha = 4$ and $\lambda = 0.002$ with 500, 750, 1000, 1250 and 1500 measurements, respectively.

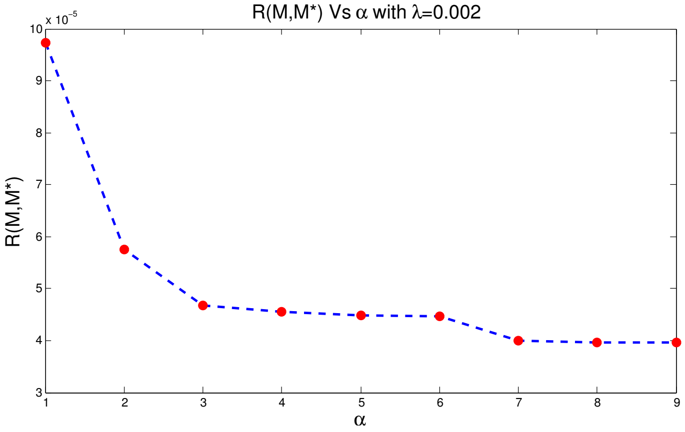
N	500	750	1000	1250	1500
SDP	725s	1146s	1510s	2059s	2769s
PMLSV	172s	232s	378s	490s	642s

Changing SNR

- to change SNR of the image, we scale the image intensity by $\alpha \geq 1$.
- recovery results when $N = 1000$ and $\lambda = 0.002$

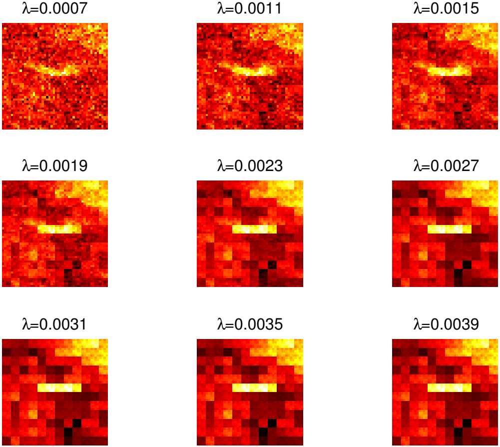


Changing SNR

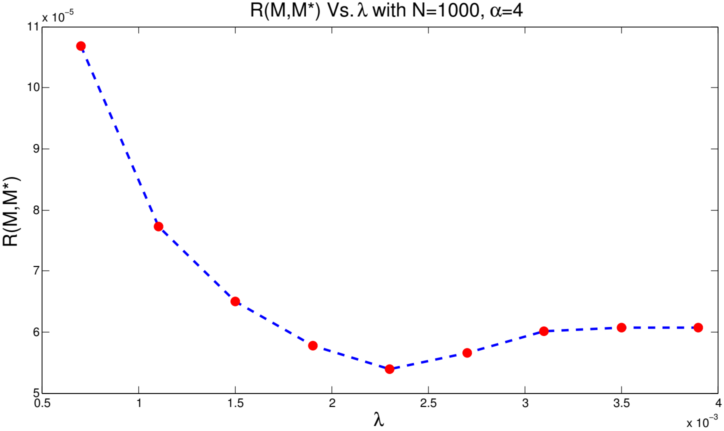


Choice of penalty parameter λ

Recovery results when $N = 1000$ and $\alpha = 4$,



Choice of penalty parameter λ



Extension to matrix completion

- $F_{\Omega,Y}(X) = \sum_{(i,j)\in\Omega} Y_{i,j} \log X_{i,j} - X_{i,j}$,
- matrix recovery by the following method

$$\begin{aligned} \widehat{M} &= \arg \max_{X \in \mathbb{R}^{d_1 \times d_2}} F_{\Omega,Y}(X), \\ s.t. \quad & \|X\|_* \le \alpha\sqrt{d_1 d_2 r}, \quad \gamma_1 \le \|X\|_\infty \le \gamma_2 \end{aligned} \tag{2}$$

Performance guarantee

Theorem

Assume that $\|M\|_* \leq \alpha\sqrt{d_1 d_2 r}$ *and* $\gamma_1 \leq \|M\|_\infty \leq \gamma_2$. Ω *is chosen at random binomial with* $\mathbb{E}|\Omega| = m$. $\widehat{M}$ *is is defined in (2). Then with probability at least* $(1 - C/(d_1 + d_2))$, *we have*

$$\frac{1}{d_1 d_2}\|M - \widehat{M}\|_F^2 \leq C^* \left(\frac{8\gamma_2 T}{1 - e^{-T}}\right)(\alpha\sqrt{r} + \frac{1}{\sqrt{d_1 d_2}})\sqrt{\frac{d_1 + d_2}{m}} \cdot \sqrt{1 + \frac{(d_1 + d_2)\log(d_1 d_2)}{m}}, \tag{3}$$

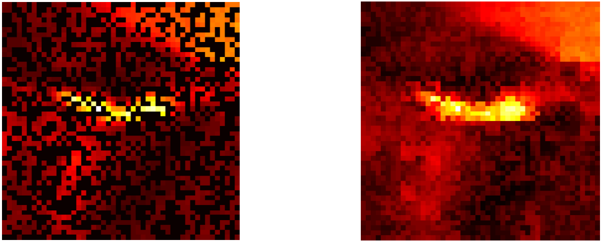
If $m \geq (d_1 + d_2)\log(d_1 d_2)$ *then this simplifies to*

$$\frac{1}{d_1 d_2}\|M - \widehat{M}\|_F^2 \leq \sqrt{2}C^* \left(\frac{8\gamma_2 T}{1 - e^{-T}}\right)(\alpha\sqrt{r} + \frac{1}{\sqrt{d_1 d_2}})\sqrt{\frac{d_1 + d_2}{m}}.$$

T, C^*, C: *constants.*

Fast algorithm for matrix completion

- algorithm similar to PMLSV can be used except that we modify the likelihood function, which only affect the gradient
- solution again based on singular value thresholding



(a) $p = 0.5$ (b) $\alpha = 1000, \gamma_1 = 1, \gamma_2 = 12000$

Figure : Image with missing data and recovery result

Summary

- fast algorithm for low-rank matrix recovery and matrix completion
- key idea: approximating the log likelihood function by sequential Taylor expansion
- known exact solution to the approximated cost function via **Singular Value Thresholding**