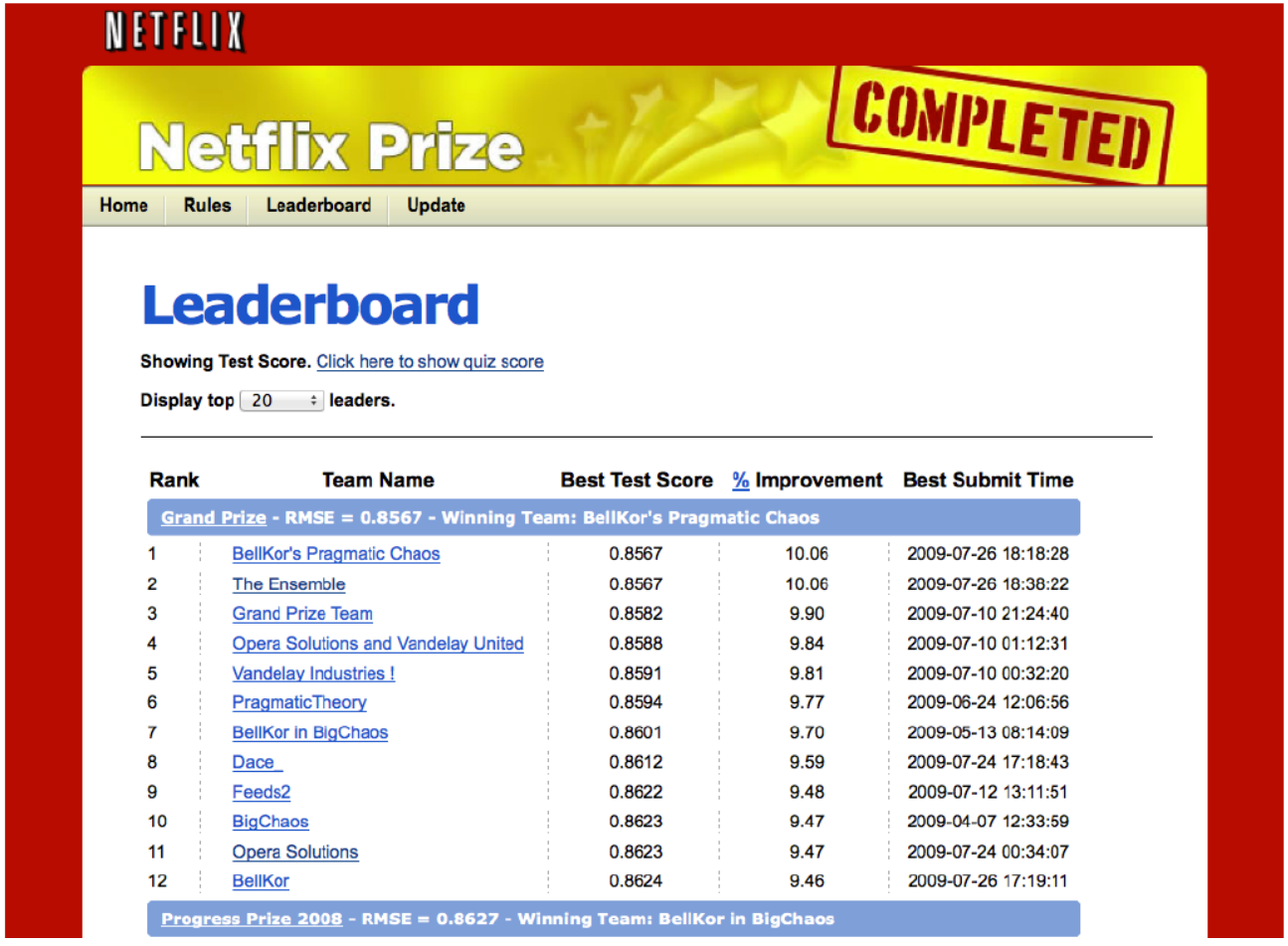
Matrix Completion

Stanley Chan

Purdue University

ECE/STAT 695 Spring 2018

The Netflix Challenge



NETFLIX

Netflix Prize

COMPLETED

Home | Rules | Leaderboard | Update

Leaderboard

Showing Test Score. Click here to show quiz score

Display top 20 **leaders.**

Rank	Team Name	Best Test Score	% Improvement	Best Submit Time
Grand Prize - RMSE = 0.8567 - Winning Team: BellKor's Pragmatic Chaos				
1	BellKor's Pragmatic Chaos	0.8567	10.06	2009-07-26 18:18:28
2	The Ensemble	0.8567	10.06	2009-07-26 18:38:22
3	Grand Prize Team	0.8582	9.90	2009-07-10 21:24:40
4	Opera Solutions and Vandelay United	0.8588	9.84	2009-07-10 01:12:31
5	Vandelay Industries !	0.8591	9.81	2009-07-10 00:32:20
6	PragmaticTheory	0.8594	9.77	2009-06-24 12:06:56
7	BellKor in BigChaos	0.8601	9.70	2009-05-13 08:14:09
8	Dace	0.8612	9.59	2009-07-24 17:18:43
9	Feeds2	0.8622	9.48	2009-07-12 13:11:51
10	BigChaos	0.8623	9.47	2009-04-07 12:33:59
11	Opera Solutions	0.8623	9.47	2009-07-24 00:34:07
12	BellKor	0.8624	9.46	2009-07-26 17:19:11
Progress Prize 2008 - RMSE = 0.8627 - Winning Team: BellKor in BigChaos				

- Predict user rating
- Missing data

Missing Data Problem

	Dirty Dancing	Meet the Parents	Top Gun	The Sixth Sense	Catch Me If You Can	The Royal Tenenbaums	Con Air	Big Fish	The Matrix	A Few Good Men
Customer 1					4					
Customer 2			3				3			3
Customer 3		2		4					2	
Customer 4	3									
Customer 5	5	5			4					
Customer 6						2	4			
Customer 7			5					3		
Customer 8						2				3
Customer 9	3				5			5		
Customer 10										

Optimization

The general problem of matrix completion is

$$\widehat{\boldsymbol{Z}} = \arg\min_{\boldsymbol{Z}} \|\boldsymbol{Z} - \boldsymbol{M}\|_F^2 \quad \text{subject to} \quad \Phi(\boldsymbol{Z}) \leq c. \tag{1}$$

Examples of $\Phi(\boldsymbol{Z})$:

- $\|\boldsymbol{Z}\|_1$: Sparse matrix
- $\mathrm{rank}(\boldsymbol{Z})$: Low rank
- $\|\boldsymbol{Z}\|_*$: Nuclear norm
- $\boldsymbol{Z} = \boldsymbol{L} + \boldsymbol{S}$, $\boldsymbol{S}$ sparse: Decomposition

Rank Minimization

Candes and Recht:

$$\begin{aligned} \underset{\boldsymbol{X}}{\text{minimize}} \quad & \operatorname{rank}(\boldsymbol{X}) \\ \text{subject to} \quad & X_{ij} = M_{ij}, \quad (i,j) \in \Omega. \end{aligned} \tag{2}$$

- $\Omega = \{(i,j) \mid M_{ij} \text{ is available.}\}$
- Finds the matrix with the minimum rank
- NP-hard

Nuclear Norm Minimization

The nuclear norm minimization problem is defined as

$$\begin{aligned} \underset{\boldsymbol{X}}{\text{minimize}} \quad & \|\boldsymbol{X}\|_* \\ \text{subject to} \quad & X_{ij} = M_{ij}, \quad (i,j) \in \Omega. \end{aligned} \tag{3}$$

Comparison between Nuclear Norm and Frobenius Norm:

- $\|\boldsymbol{X}\|_* = \mathrm{Tr}\left(\sqrt{\boldsymbol{X}^T\boldsymbol{X}}\right) = \sum_{i=1}^{n} \sigma_i(\boldsymbol{X})$
- $\|\boldsymbol{X}\|_F = \sqrt{\mathrm{Tr}(\boldsymbol{X}^T\boldsymbol{X})} = \sqrt{\sum_{i=1}^{n} \sigma_i^2(\boldsymbol{X})}$
- $\|\boldsymbol{X}\|_*$ promotes sparsity
- Nuclear norm minimization is the tightest convex relaxation of the rank minimization.
- $\{\boldsymbol{X} \mid \|\boldsymbol{X}\|_* \leq 1\}$ is the convex hull of set of rank-one matrices with spectral norm bounded by 1.

Random Orthogonal Model

Candes and Recht 2008:

$$\boldsymbol{M} = \sum_{k=1}^{r} \sigma_k \boldsymbol{u}_k \boldsymbol{v}_k^T, \tag{4}$$

- $\{\boldsymbol{u}_k\}_{k=1}^r$ is selected *uniformly* at random among all families of orthonormal vectors.
- So is $\{\boldsymbol{v}_k\}$
- That is: $\boldsymbol{M}$ has an intrinsic low rank

Main Theoretical Result

Candes and Recht 2008:

Theorem

Let $\boldsymbol{M} \in \mathbb{R}^{n_1 \times n_2}$ be a rank r matrix sampled from the random orthogonal model. Let $n = \max(n_1, n_2)$. Suppose we observe m entries of $\boldsymbol{M}$ uniformly at random. Then, there are constants c and C such that if

$$m \geq C n^{5/4} r \log n,$$

then the minimizer to the Nuclear norm minimization is unique and equal to $\boldsymbol{M}$ with probability $1 - cn^{-3}$.

Implications:

- Under the hypothesis of the Theorem, there is a unique low-rank matrix which is consistent with the observed entries
- This matrix can be recovered by a convex optimization algorithm.
- So the nuclear norm relaxation is formally equivalent to the NP hard problem.

Solving the Nuclear Norm Problem

- Can be translated into the semi-definite programming
- Standard packages exist, e.g., CVX, SDPT3
- Typically solves the problem using interior point method
- Polynomial runtime

Projection Operator

Define a projection operator $\mathcal{P}_\Omega$:

$$[\mathcal{P}_\Omega(\boldsymbol{X})]_{ij} = \begin{cases} X_{ij}, & (i,j) \in \Omega \\ 0, & (i,j) \notin \Omega \end{cases} \tag{5}$$

Then, the nuclear norm minimization can be written as

$$\begin{aligned} \underset{\boldsymbol{X}}{\text{minimize}} \quad & \|\boldsymbol{X}\|_* \\ \text{subject to} \quad & \mathcal{P}_\Omega(\boldsymbol{X}) = \mathcal{P}_\Omega(\boldsymbol{M}). \end{aligned} \tag{6}$$

We can relax the problem by

$$\begin{aligned} \underset{\boldsymbol{X}}{\text{minimize}} \quad & \tau\|\boldsymbol{X}\|_* + \tfrac{1}{2}\|\boldsymbol{X}\|_F^2 \\ \text{subject to} \quad & \mathcal{P}_\Omega(\boldsymbol{X}) = \mathcal{P}_\Omega(\boldsymbol{M}). \end{aligned} \tag{7}$$

As $\tau \to \infty$, the second problem becomes the first problem.

Solving the Problem

Recall:

$$\begin{aligned} \underset{\boldsymbol{X}}{\text{minimize}} \quad & \tau\|\boldsymbol{X}\|_* + \tfrac{1}{2}\|\boldsymbol{X}\|_F^2 \\ \text{subject to} \quad & \mathcal{P}_\Omega(\boldsymbol{X}) = \mathcal{P}_\Omega(\boldsymbol{M}). \end{aligned} \tag{8}$$

Define the Lagrangian:

$$\mathcal{L}(\boldsymbol{X}, \boldsymbol{Y}) = \tau\|\boldsymbol{X}\|_* + \frac{1}{2}\|\boldsymbol{X}\|_F^2 + \langle \boldsymbol{Y}, \mathcal{P}_\Omega(\boldsymbol{M} - \boldsymbol{X})\rangle \tag{9}$$

Uzawa's Algorithm:

$$\begin{aligned} \boldsymbol{X}^k &= \arg\min_{\boldsymbol{X}} \; \mathcal{L}(\boldsymbol{X}, \boldsymbol{Y}^{k-1}) \\ \boldsymbol{Y}^k &= \boldsymbol{Y}^{k-1} + \delta_k \mathcal{P}_\Omega(\boldsymbol{M} - \boldsymbol{X}^k) \end{aligned} \tag{10}$$

Subproblem Solution

$$\begin{aligned}\boldsymbol{X}^k &= \arg\min_{\boldsymbol{X}} \ \mathcal{L}(\boldsymbol{X}, \boldsymbol{Y}^{k-1}) \\ &= \arg\min_{\boldsymbol{X}} \ \tau\|\boldsymbol{X}\|_* + \frac{1}{2}\|\boldsymbol{X}\|_F^2 + \langle \boldsymbol{Y}^{k-1}, \mathcal{P}_\Omega(\boldsymbol{M} - \boldsymbol{X})\rangle \\ &= \arg\min_{\boldsymbol{X}} \ \tau\|\boldsymbol{X}\|_* + \frac{1}{2}\|\boldsymbol{X} - \mathcal{P}_\Omega \boldsymbol{Y}^{k-1}\|_F^2.\end{aligned}$$

Theorem (Theorem 2.1, Cai-Candes-Shen 2008)

For every $\tau \geq 0$ and $\boldsymbol{Y} \in \mathbb{R}^{n_1 \times n_2}$, it holds that

$$\arg\min_{\boldsymbol{X}} \ \tau\|\boldsymbol{X}\|_* + \frac{1}{2}\|\boldsymbol{X} - \mathcal{P}_\Omega \boldsymbol{Y}\|_F^2 = \mathcal{D}_\tau(\boldsymbol{Y}), \tag{11}$$

where $\mathcal{D}_\tau(\boldsymbol{Y})$ is the singular value shrinkage operator:

$$\mathcal{D}_\tau(\boldsymbol{Y}) = \boldsymbol{U}\mathcal{D}_\tau(\boldsymbol{\Sigma})\boldsymbol{V}^T, \quad \mathcal{D}_\tau(\boldsymbol{\Sigma}) = diag((\sigma_i - \tau)_+). \tag{12}$$

The SVT Algorithm

We now have the Singular Value Thresholding algorithm

$$\begin{aligned} \boldsymbol{X}^k &= \mathcal{D}_\tau(\boldsymbol{Y}^{k-1}) \\ \boldsymbol{Y}^k &= \boldsymbol{Y}^{k-1} + \delta_k \mathcal{P}_\Omega(\boldsymbol{M} - \boldsymbol{X}^k) \end{aligned} \tag{13}$$

- $\boldsymbol{X}^k = \mathcal{D}_\tau(\boldsymbol{Y}^{k-1})$ is similar to a shrinkage step
- τ is the shrinkage threshold
- $\boldsymbol{Y}^k = \boldsymbol{Y}^{k-1} + \delta_k \mathcal{P}_\Omega(\boldsymbol{M} - \boldsymbol{X}^k)$ is the update

Choice of δ_k

"Heuristic Choice" by Cai-Candes-Shen:

$$\delta = 1.2\frac{n_1 n_2}{m} \tag{14}$$

- $\boldsymbol{X} \in \mathbb{R}^{n_1 \times n_2}$, $m = |\Omega|$.
- Incoherence assumption: If $\mathrm{rank}(\boldsymbol{A})$ is not too large, then with high probability

$$(1-\epsilon)p\|\boldsymbol{A}\|_F^2 \leq \|\mathcal{P}_\Omega(\boldsymbol{A})\|_F^2 \leq (1+\epsilon)p\|\boldsymbol{A}\|_F^2, \quad p = m/(n_1 n_2)$$

- As part of the convergence proof, the algorithm will converge when

$$0 < \delta < 2\frac{\|\boldsymbol{X}^* - \boldsymbol{X}^k\|_F^2}{\|\mathcal{P}_\Omega(\boldsymbol{X}^* - \boldsymbol{X}^k)\|_F^2}.$$

- Put $\boldsymbol{A} = \boldsymbol{X}^* - \boldsymbol{X}^k$, then if $\epsilon = 1/4$, $\delta \leq 1.6p^{-1}$. They take $1.2p^{-1}$.

Stopping Criteria

Another "heuristic choice":

$$\frac{\|\mathcal{P}_\Omega(\boldsymbol{X}^k - \boldsymbol{M})\|_F}{\|\mathcal{P}_\Omega(\boldsymbol{M})\|_F} \leq \epsilon \tag{15}$$

- At optimal point, we must have $\mathcal{P}_\Omega(\boldsymbol{X} - \boldsymbol{M}) = 0$.
- Under suitable assumptions: $\|\mathcal{P}_\Omega(\boldsymbol{M})\|_F^2 \approx p\|\mathcal{P}_\Omega(\boldsymbol{M})\|_F^2$
- So roughly speaking: $\|\mathcal{P}_\Omega(\boldsymbol{X}^k - \boldsymbol{M})\|_F^2 \approx p\|\mathcal{P}_\Omega(\boldsymbol{X}^k - \boldsymbol{M})\|_F^2$
- Hence,

$$\frac{\|\mathcal{P}_\Omega(\boldsymbol{X}^k - \boldsymbol{M})\|_F}{\|\mathcal{P}_\Omega(\boldsymbol{M})\|_F} \approx \frac{\|\boldsymbol{X}^k - \boldsymbol{M}\|_F}{\|\boldsymbol{M}\|_F}.$$

Algorithm

Algorithm 1: Singular Value Thresholding (SVT) Algorithm

Input: sampled set Ω and sampled entries $\mathcal{P}_\Omega(\boldsymbol{M})$, step size δ, tolerance ϵ, parameter τ, increment ℓ, and maximum iteration count $k_{\max}$

Output: $\boldsymbol{X}^{\text{opt}}$

Description: Recover a low-rank matrix $\boldsymbol{M}$ from a subset of sampled entries

Set $\boldsymbol{Y}^0 = k_0 \delta \, \mathcal{P}_\Omega(\boldsymbol{M})$ (k_0 is defined in (5.3))
Set $r_0 = 0$
for $k = 1$ **to** $k_{\max}$
    Set $s_k = r_{k-1} + 1$
    repeat
        Compute $[\boldsymbol{U}^{k-1}, \boldsymbol{\Sigma}^{k-1}, \boldsymbol{V}^{k-1}]_{s_k}$
        Set $s_k = s_k + \ell$
    until $\sigma^{k-1}_{s_k - \ell} \le \tau$
    Set $r_k = \max\{j : \sigma^{k-1}_j > \tau\}$
    Set $\boldsymbol{X}^k = \sum_{j=1}^{r_k} (\sigma^{k-1}_j - \tau) \boldsymbol{u}^{k-1}_j \boldsymbol{v}^{k-1}_j$

    if $\|\mathcal{P}_\Omega(\boldsymbol{X}^k - \boldsymbol{M})\|_F / \|\mathcal{P}_\Omega \boldsymbol{M}\|_F \le \epsilon$ **then break**

    Set $Y^k_{ij} = \begin{cases} 0 & \text{if } (i,j) \notin \Omega, \\ Y^{k-1}_{ij} + \delta(M_{ij} - X^k_{ij}) & \text{if } (i,j) \in \Omega \end{cases}$
end *for* k
Set $\boldsymbol{X}^{\text{opt}} = \boldsymbol{X}^k$

Results

Unknown $\boldsymbol{M}$				Computational results		
size $(n \times n)$	rank (r)	m/d_r	m/n^2	time(s)	# iters	relative error
$1,000 \times 1,000$	10	6	0.12	23	117	1.64×10^{-4}
	50	4	0.39	196	114	1.59×10^{-4}
	100	3	0.57	501	129	1.68×10^{-4}
$5,000 \times 5,000$	10	6	0.024	147	123	1.73×10^{-4}
	50	5	0.10	950	108	1.61×10^{-4}
	100	4	0.158	3,339	123	1.72×10^{-4}
$10,000 \times 10,000$	10	6	0.012	281	123	1.73×10^{-4}
	50	5	0.050	2,096	110	1.65×10^{-4}
	100	4	0.080	7,059	127	1.79×10^{-4}
$20,000 \times 20,000$	10	6	0.006	588	124	1.73×10^{-4}
	50	5	0.025	4,581	111	1.66×10^{-4}
$30,000 \times 30,000$	10	6	0.004	1,030	125	1.73×10^{-4}

Table 1: Experimental results for matrix completion. The rank r is the rank of the unknown matrix $\boldsymbol{M}$, m/d_r is the ratio between the number of sampled entries and the number of degrees of freedom in an $n \times n$ matrix of rank r (oversampling ratio), and m/n^2 is the fraction of observed entries. All the computational results on the right are averaged over five runs.

Extension

Robust PCA:

$$(\widehat{\boldsymbol{A}}, \widehat{\boldsymbol{E}}) = \arg\min_{\boldsymbol{A},\boldsymbol{E}} \quad \|\boldsymbol{A}\|_* + \lambda\|\boldsymbol{E}\|_1, \quad \boldsymbol{A} + \boldsymbol{E} = \boldsymbol{D} \tag{16}$$

- $\boldsymbol{D}$ is data matrix
- $\boldsymbol{A}$ is low rank: background video frames
- $\boldsymbol{E}$ is sparse: foreground objects

To solve the problem, translate into

$$(\widehat{\boldsymbol{A}}, \widehat{\boldsymbol{E}}) = \arg\min_{\boldsymbol{A},\boldsymbol{E}} \quad \mu\|\boldsymbol{A}\|_* + \lambda\|\boldsymbol{E}\|_1 + \frac{1}{2}\|\boldsymbol{D} - \boldsymbol{A} - \boldsymbol{E}\|_F^2 \tag{17}$$

Algorithm

Algorithm 1: Robust PCA via Proximal Gradient with Continuation

Input: Observation matrix $D \in \mathbb{R}^{m \times n}$, weight λ.
$A_0, A_{-1} \leftarrow 0$, $E_0, E_{-1} \leftarrow 0$, $t_0, t_{-1} \leftarrow 1$, $\mu_0 \leftarrow .99\|D\|_{2,2}$, $\bar{\mu} \leftarrow 10^{-5}\mu_0$.
while not converged
  $\tilde{A}_k \leftarrow A_k + \frac{t_{k-1}-1}{t_k}(A_k - A_{k-1})$, $\tilde{E}_k \leftarrow E_k + \frac{t_{k-1}-1}{t_k}(E_k - E_{k-1})$.
  $Y_k^A \leftarrow \tilde{A}_k - \frac{1}{2}\left(\tilde{A}_k + \tilde{E}_k - D\right)$.
  $(U, S, V) \leftarrow \mathrm{svd}(Y_k^A)$, $A_{k+1} \leftarrow U\left[S - \frac{\mu}{2}\mathtt{I}\right]_+ V^*$.
  $Y_k^E \leftarrow \tilde{E}_k - \frac{1}{2}\left(\tilde{A}_k + \tilde{E}_k - D\right)$.
  $E_{k+1} \leftarrow \mathrm{sign}[Y_k^E] \circ \left[|Y_k^E| - \frac{\lambda\mu}{2}11^*\right]_+$.
  $t_{k+1} \leftarrow \frac{1+\sqrt{1+4t_k^2}}{2}$, $\mu \leftarrow \max(.9\mu, \bar{\mu})$.
end while
Output: A, E.

Example

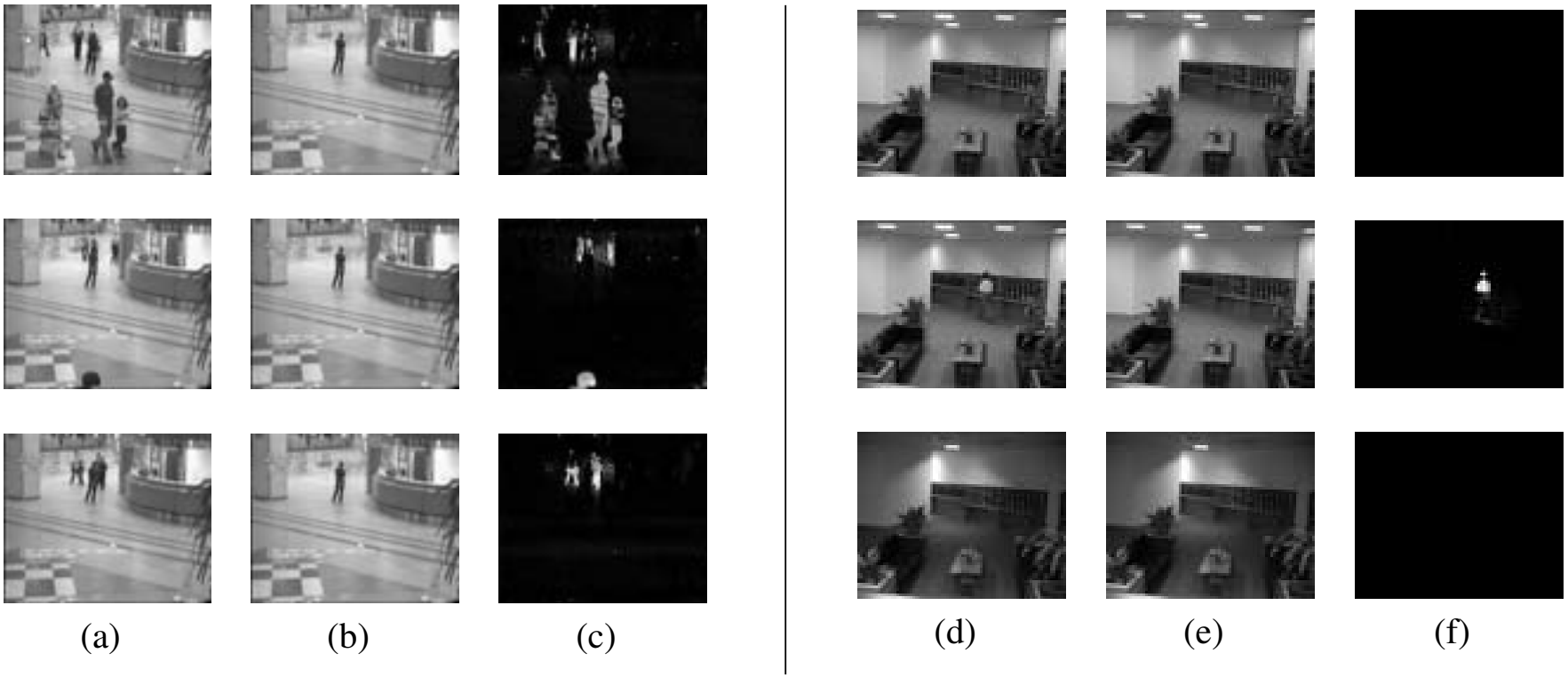


Figure 2: **Background modeling.** (a) Video sequence of a scene in an airport. The size of each frame is 72×88 pixels, and a total of 200 frames were used. (b) Static background recovered by our algorithm. (c) Sparse error recovered by our algorithm represents activity in the frame. (d) Video sequence of a lobby scene with changing illumination. The size of each frame is 64×80 pixels, and a total of 550 frames were used. (e) Static background recovered by our algorithm. (f) Sparse error. The background is correctly recovered even when the illumination in the room changes drastically in the frame on the last row.

Reference

- Jian-Feng Cai, Emmanuel J. Candes, Zuowei Shen, A Singular Value Thresholding Algorithm for Matrix Completion, arXiv:0810.3286
- Emmanuel J. Candes, Benjamin Recht, Exact Matrix Completion via Convex Optimization, arXiv:0805.4471
- John Wright, Yigang Peng, Yi Ma, Arvind Ganesh, and Shankar Rao, Robust Principal Component Analysis: Exact Recovery of Corrupted Low-Rank Matrices by Convex Optimization, arXiv:0905.0233