Online Tensor Decomposition and Imputation for Count Data

Chang Ye

Dept. of Electrical and Computer Engineering
University of Rochester
cye7@ur.rochester.edu
http://www.ece.rochester.edu/~cye7/

Co-author: Gonzalo Mateos
Acknowledgment: NSF Awards CCF-1750428 and ECCS-1809356

DSW 2019, Minneapolis, June 3, 2019

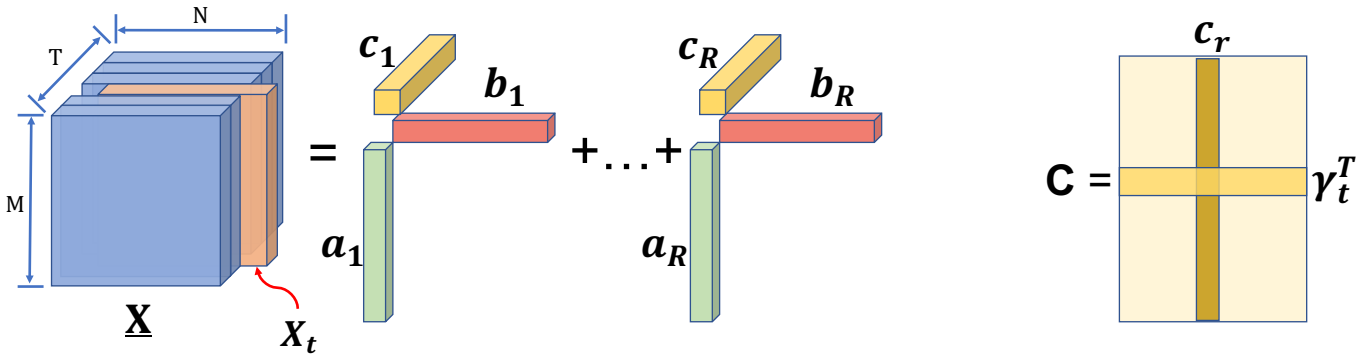
Tensors and PARAFAC decomposition

- Three-way tensor $\underline{\mathbf{X}} \in \mathbb{R}^{M \times N \times T}$. PARAFAC decomposition

$$\underline{\mathbf{X}} = \sum_{r=1}^{R} \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r$$

$\Rightarrow$ Rank R is the number of (rank-one) outer-products

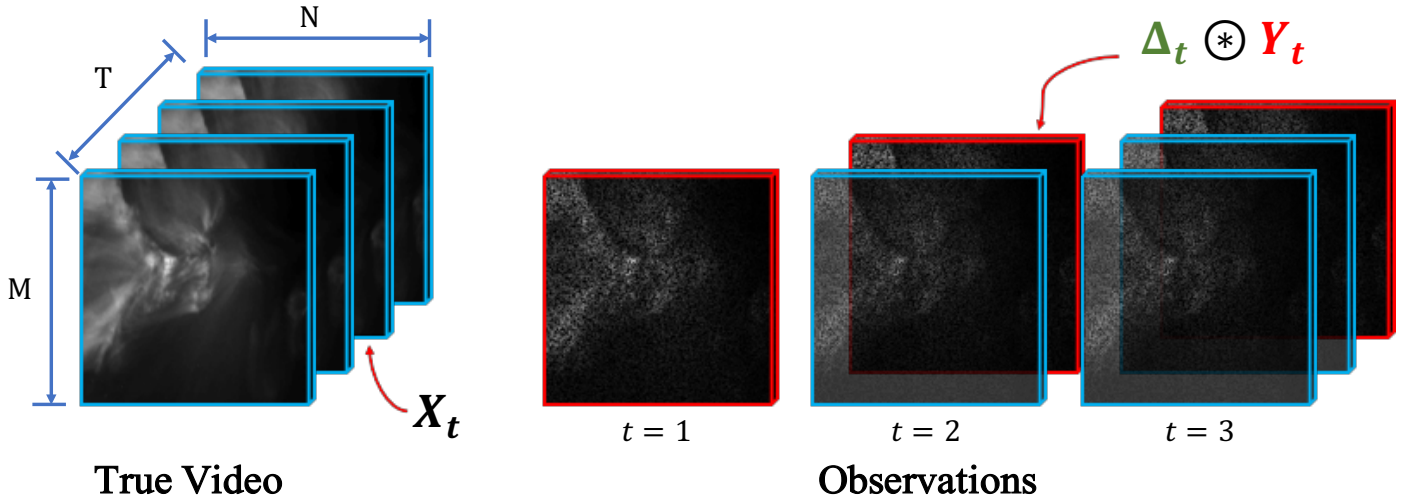
$\Rightarrow$ Factors $\mathbf{A} = [\mathbf{a}_1, ..., \mathbf{a}_R]$, $\mathbf{B} = [\mathbf{b}_1, ..., \mathbf{b}_R]$, $\mathbf{C} = [\mathbf{c}_1, ..., \mathbf{c}_R]$



- Write tensor slice w.r.t. subscript t as $\mathbf{X}_t = \mathbf{A}\text{diag}(\gamma_t)\mathbf{B}^T$

Streaming count data with misses

- **Ex:** solar flare video, NASA SDO program [Xie et al'13]

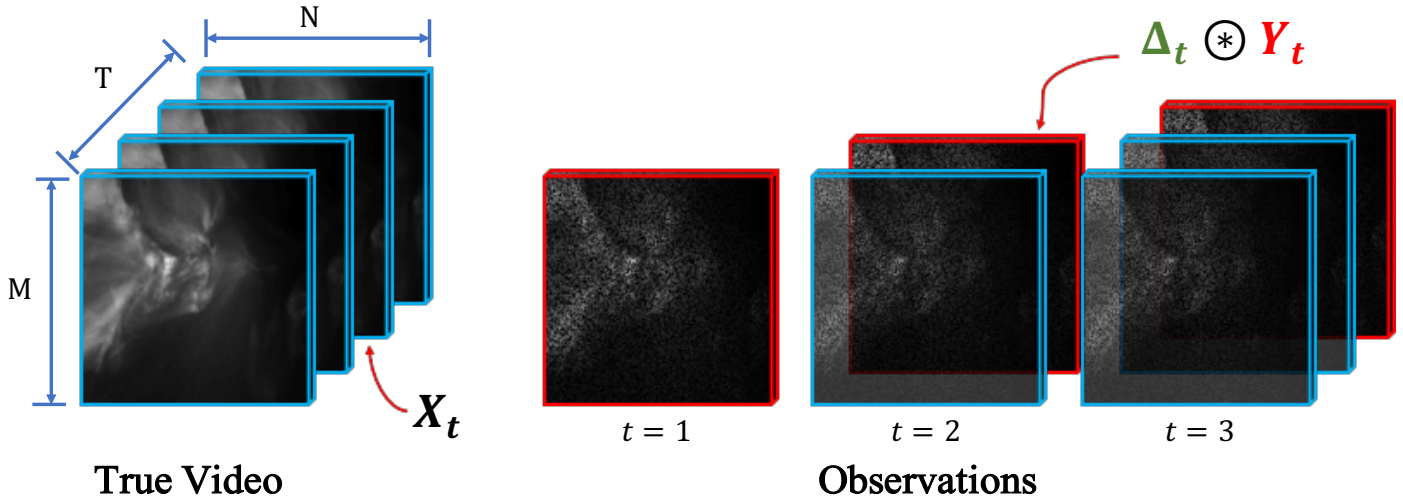


- Observations $\underline{\mathbf{Y}} \sim \text{Poisson}(\underline{\mathbf{X}})$, i.i.d. $\rightarrow \mathrm{P}\left(y_{mnt} = k\right) = \frac{x_{mnt}^k e^{-x_{mnt}}}{k!}$
- Tensor data may be incomplete. Model as $\{\mathcal{\Delta}_\tau \circledast \mathcal{Y}_\tau\}_{\tau=1}^t$

  $\Rightarrow$ Binary mask $\delta_{mnt} = 1$ if y_{mnt} is observed, else $\delta_{mnt} = 0$

Streaming count data with misses

- **Ex:** solar flare video, NASA SDO program [Xie et al'13]



- Observations $\underline{\mathbf{Y}} \sim \text{Poisson}(\underline{\mathbf{X}})$, i.i.d. $\rightarrow \mathrm{P}\left(y_{mnt} = k\right) = \frac{x_{mnt}^k e^{-x_{mnt}}}{k!}$
- Tensor data may be incomplete. Model as $\{\mathbf{\Delta}_\tau \circledast \mathbf{Y}_\tau\}_{\tau=1}^t$

  $\Rightarrow$ Binary mask $\delta_{mnt} = 1$ if y_{mnt} is observed, else $\delta_{mnt} = 0$

Goal: given streaming data $\{\mathbf{\Delta}_\tau \circledast \mathbf{Y}_\tau\}_{\tau=1}^t$, with $\underline{\mathbf{X}}$ assumed low-rank
(1) Adaptively learn factor matrices $\mathbf{A}, \mathbf{B}$ and coefficient γ_t at time t
(2) Impute the missing entries of $\mathbf{Y}_t$ as a byproduct

Our work in context

- Online matrix factorization for Gaussian data [Mairal el al'09]
  - Subspace tracking [Yang'95], [Yang-Kaveh'98]
  - Online robust PCA [He et al'11], [Chi et al'13], [Feng et al'13], ...
- Beyond Gaussian-distributed matrix data
  - Online categorical subspace learning [Shen et al'17]
  - Subspace tracking for Poisson-distributed data [Wang-Chi'18]

Our work in context

- Online matrix factorization for Gaussian data [Mairal el al'09]
  - Subspace tracking [Yang'95], [Yang-Kaveh'98]
  - Online robust PCA [He et al'11], [Chi et al'13], [Feng et al'13], ...
- Beyond Gaussian-distributed matrix data
  - Online categorical subspace learning [Shen et al'17]
  - Subspace tracking for Poisson-distributed data [Wang-Chi'18]
- Online imputation and decomposition for low-rank Gaussian tensors
  - Stochastic approximation [Nion-Sidiropoulos'09], [Mardani et al'15]
- Batch Poisson tensor completion [Chi-Kolda'12], [Bazerque et al'13]

Our work in context

- Online matrix factorization for Gaussian data [Mairal el al'09]
  - Subspace tracking [Yang'95], [Yang-Kaveh'98]
  - Online robust PCA [He et al'11], [Chi et al'13], [Feng et al'13], ...
- Beyond Gaussian-distributed matrix data
  - Online categorical subspace learning [Shen et al'17]
  - Subspace tracking for Poisson-distributed data [Wang-Chi'18]
- Online imputation and decomposition for low-rank Gaussian tensors
  - Stochastic approximation [Nion-Sidiropoulos'09], [Mardani et al'15]
- Batch Poisson tensor completion [Chi-Kolda'12], [Bazerque et al'13]
- **Contribution:** online tensor decomposition and imputation for count data

Problem formulation

- Negative Poisson log-likelihood function for data slice at time t

$$\ell_t(\mathbf{Y}_t, \mathbf{\Delta}_t; \mathbf{X}_t) = \sum_{mn} \delta_{mnt}(x_{mnt} - y_{mnt} \log x_{mnt})$$

Problem formulation

- Negative Poisson log-likelihood function for data slice at time t

$$\ell_t(\mathbf{Y}_t, \mathbf{\Delta}_t; \mathbf{X}_t) = \sum_{mn} \delta_{mnt}(x_{mnt} - y_{mnt} \log x_{mnt})$$

- Rank-regularized Poisson tensor completion formulation

$$\min_{\{\underline{\mathbf{X}},\mathbf{A},\mathbf{B},\mathbf{C}\}\in\mathcal{T}} \sum_{t=1}^{T} \underbrace{\left[\ell_t(\mathbf{Y}_t, \mathbf{\Delta}_t; \mathbf{X}_t) + \frac{\mu}{2T}\left(\|\mathbf{A}\|_F^2 + \|\mathbf{B}\|_F^2\right) + \frac{\mu}{2}\|\boldsymbol{\gamma}_t\|^2\right]}_{g_t(\mathbf{A},\mathbf{B},\boldsymbol{\gamma}_t)}$$

  - Feasible set $\mathcal{T} := \{\underline{\mathbf{X}}, \mathbf{A}, \mathbf{B}, \mathbf{C} : \mathbf{A} \geq \mathbf{0}, \mathbf{B} \geq \mathbf{0}, \mathbf{C} \geq \mathbf{0}, \underline{\mathbf{X}} = \sum_{r=1}^{R} \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r\}$
  - Tikhonov regularizer promotes low rank [Bazerque et al'13]

Problem formulation

- Negative Poisson log-likelihood function for data slice at time t

$$\ell_t(\mathbf{Y}_t, \mathbf{\Delta}_t; \mathbf{X}_t) = \sum_{mn} \delta_{mnt}(x_{mnt} - y_{mnt} \log x_{mnt})$$

- Rank-regularized Poisson tensor completion formulation

$$\min_{\{\underline{\mathbf{X}},\mathbf{A},\mathbf{B},\mathbf{C}\}\in\mathcal{T}} \sum_{t=1}^{T} \underbrace{\left[\ell_t(\mathbf{Y}_t, \mathbf{\Delta}_t; \mathbf{X}_t) + \frac{\mu}{2T}\left(\|\mathbf{A}\|_F^2 + \|\mathbf{B}\|_F^2\right) + \frac{\mu}{2}\|\boldsymbol{\gamma}_t\|^2\right]}_{g_t(\mathbf{A},\mathbf{B},\boldsymbol{\gamma}_t)}$$

  - Feasible set $\mathcal{T} := \{\underline{\mathbf{X}}, \mathbf{A}, \mathbf{B}, \mathbf{C} : \mathbf{A} \geq \mathbf{0}, \mathbf{B} \geq \mathbf{0}, \mathbf{C} \geq \mathbf{0}, \underline{\mathbf{X}} = \sum_{r=1}^{R} \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r\}$
  - Tikhonov regularizer promotes low rank [Bazerque et al'13]

- **Q:** Can we devise an adaptive algorithm to factorize $\underline{\mathbf{X}}$ 'on the fly'?

A stochastic approximation algorithm

- Alternating minimization with stochastic gradient iterations at time t
- **Step 1:** fiber-mode, non-negative coefficient updates

$$\hat{\boldsymbol{\gamma}}[t] = \underset{\boldsymbol{\gamma}_\tau \in \mathbb{R}_+^R}{\operatorname{argmin}}\, g_t(\hat{\mathbf{A}}[t-1], \hat{\mathbf{B}}[t-1], \boldsymbol{\gamma}_\tau)$$

$\Rightarrow$ Convex, solve via projected gradient descent

A stochastic approximation algorithm

- Alternating minimization with stochastic gradient iterations at time t
- **Step 1:** fiber-mode, non-negative coefficient updates

$$\hat{\boldsymbol{\gamma}}[t] = \underset{\boldsymbol{\gamma}_\tau \in \mathbb{R}_+^R}{\operatorname{argmin}}\, g_t(\hat{\mathbf{A}}[t-1], \hat{\mathbf{B}}[t-1], \boldsymbol{\gamma}_\tau)$$

  $\Rightarrow$ Convex, solve via projected gradient descent

- **Step 2:** first-order tensor subspace updates

$$\begin{bmatrix} \hat{\mathbf{A}}[t] \\ \hat{\mathbf{B}}[t] \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{A}}[t-1] - \alpha_t \nabla_{\mathbf{A}} g_t(\hat{\mathbf{A}}[t-1], \hat{\mathbf{B}}[t-1], \hat{\boldsymbol{\gamma}}[t]) \\ \hat{\mathbf{B}}[t-1] - \alpha_t \nabla_{\mathbf{B}} g_t(\hat{\mathbf{A}}[t-1], \hat{\mathbf{B}}[t-1], \hat{\boldsymbol{\gamma}}[t]) \end{bmatrix}_+$$

  $\Rightarrow$ Update at t only relies on $\boldsymbol{\Delta}_t \circledast \mathbf{Y}_t$ and $\hat{\boldsymbol{\gamma}}[t]$

  $\Rightarrow$ Interpret: minimize a surrogate quadratic upper-bound of g_t

- 'On the fly' imputation $\hat{\mathbf{X}}_t = \hat{\mathbf{A}}[t]\text{diag}(\hat{\boldsymbol{\gamma}}[t])\hat{\mathbf{B}}^T[t]$

Numerical test: Synthetic data

- Synthetic tensor data $\underline{\mathbf{X}}$, $M \times N \times T = 20 \times 15 \times 400$, $R = 10$
  - $\Rightarrow$ Entries of $\{\mathbf{A}, \mathbf{B}, \mathbf{C}\}$ are drawn from Uniform$[0, 1]$ distribution
  - $\Rightarrow$ Scaled such that $\mathbb{E}[x_{mnt}] = 100$
- Observations $\underline{\mathbf{Y}} \sim$ Poisson$(\underline{\mathbf{X}})$, i.i.d.
  - $\Rightarrow$ Missing data via $\underline{\mathbf{\Delta}}$ with i.i.d. Bernoulli(p) entries

Numerical test: Synthetic data

- Synthetic tensor data $\underline{\mathbf{X}}$, $M \times N \times T = 20 \times 15 \times 400$, $R = 10$
  - $\Rightarrow$ Entries of $\{\mathbf{A}, \mathbf{B}, \mathbf{C}\}$ are drawn from Uniform$[0, 1]$ distribution
  - $\Rightarrow$ Scaled such that $\mathbb{E}[x_{mnt}] = 100$
- Observations $\underline{\mathbf{Y}} \sim$ Poisson$(\underline{\mathbf{X}})$, i.i.d.
  - $\Rightarrow$ Missing data via $\underline{\mathbf{\Delta}}$ with i.i.d. Bernoulli(p) entries
- Figures of merit as in [Wang-Chi'18]

  (i) Relative tensor slice recovery error

$$e_t^{(re)} = \|\hat{\mathbf{X}}_t - \mathbf{X}_t\|_F / \|\mathbf{X}_t\|_F, \quad \hat{\mathbf{X}}_t = \hat{\mathbf{A}}[t]\text{diag}(\hat{\gamma}[\tau])\hat{\mathbf{B}}[t]^T$$

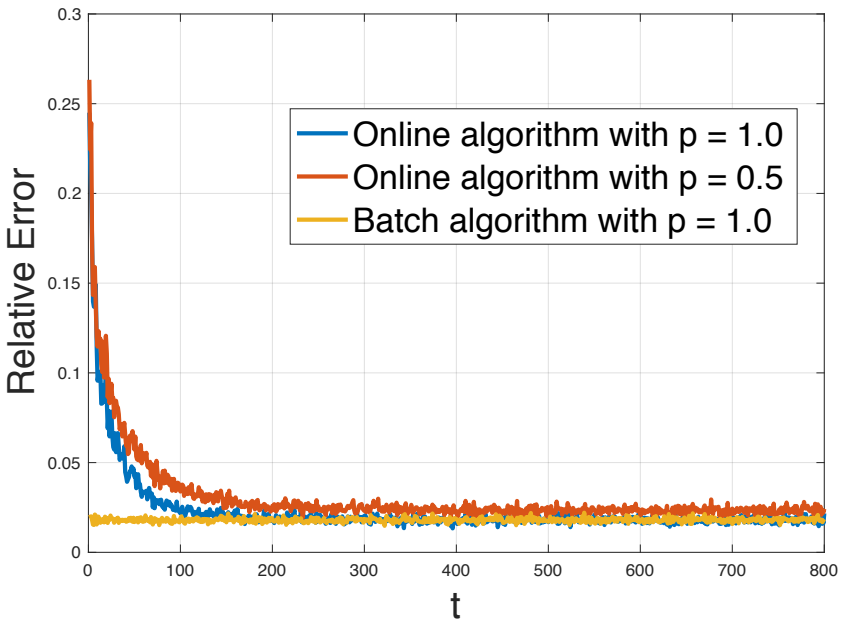
  (ii) Normalized subspace reconstruction error

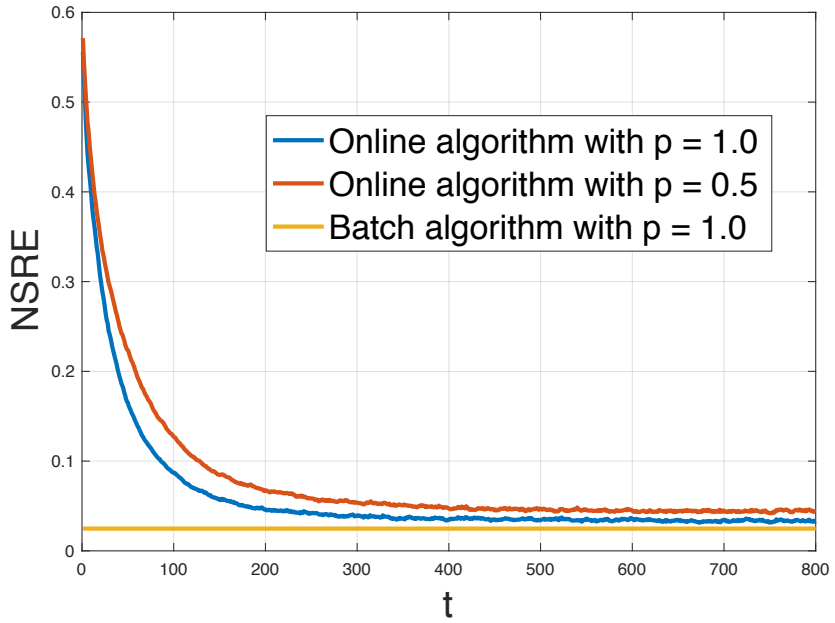
$$e_t^{(ss)} = \|P_{\hat{\mathbf{\Pi}}_{Ct}^{\perp}} \mathbf{\Pi}_C\|_F / \|\mathbf{\Pi}_C\|_F, \quad P_{\hat{\mathbf{\Pi}}_{Ct}^{\perp}} = (\mathbf{I} - \hat{\mathbf{\Pi}}_{Ct}\hat{\mathbf{\Pi}}_{Ct}^{\dagger})$$

  $\Rightarrow$ True subspace $\mathbf{\Pi}_C = (\mathbf{B} \odot \mathbf{A})$, and estimate $\hat{\mathbf{\Pi}}_{Ct} = (\hat{\mathbf{B}}[t] \odot \hat{\mathbf{A}}[t])$

Performance evaluation

- Set $\mu = 0.01$ for best performance, α_t chosen via line search
  - $\Rightarrow$ Baseline: batch Poisson tensor completion [Bazerque et al'13]

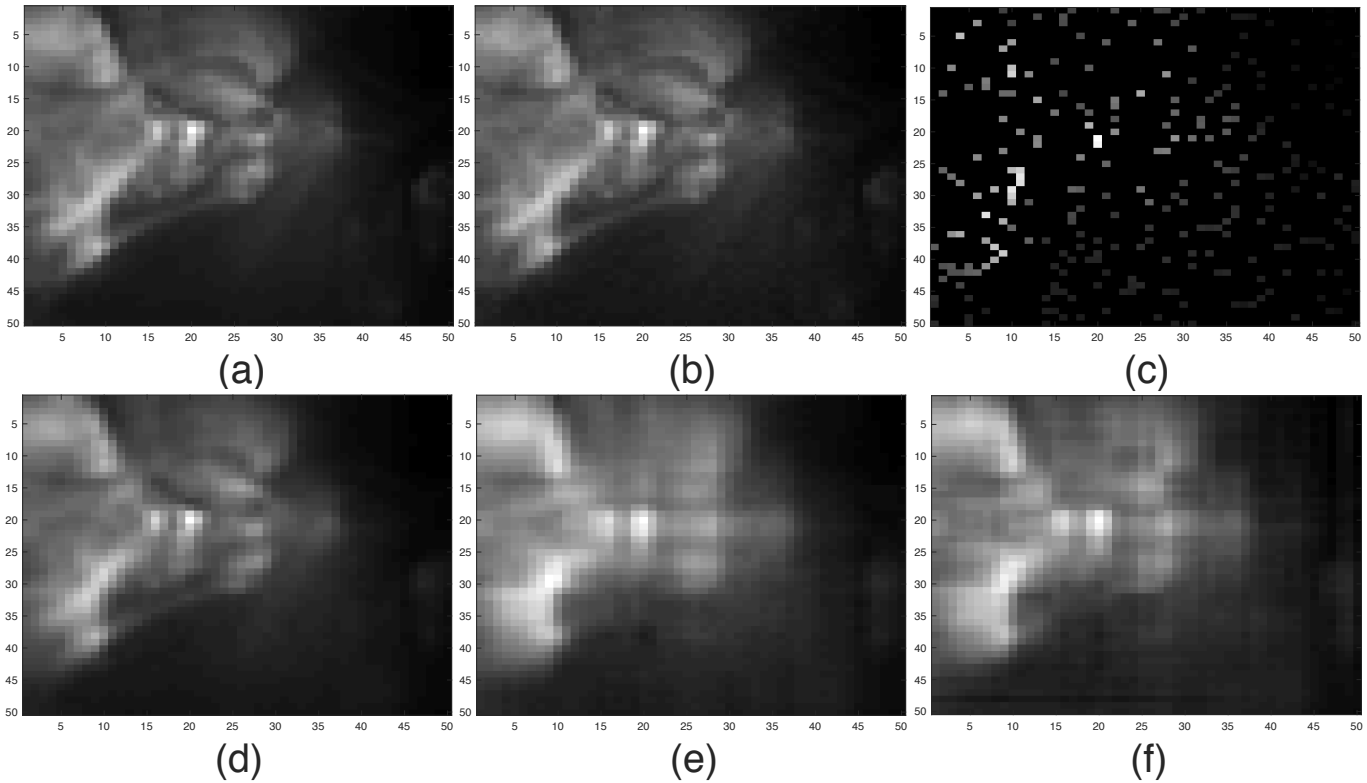




- Convergence apparent after $t \approx 150$ time slots
  - Performance close to batch benchmark, even when 50% data missing

Numerical test: Solar flare data

- Solar flare video of resolution 50×50, 300 frames [$t = 200$ in (a)]



- Input Poisson-corrupted data [(b) fully observed and (c) 90% missing]
  - $\Rightarrow$ Reconstruction via (d) batch algorithm and (e) proposed method
  - $\Rightarrow$ (f) Online imputation of $\mathbf{\Delta}_{200} \circledast \mathbf{Y}_{200}$

Concluding summary

- Online tensor decomposition and imputation for count data
  - $\Rightarrow$ From Gaussian to streaming Poisson-distributed data
  - $\Rightarrow$ Key: alternating minimization with stochastic gradient
- Ongoing work
  - $\Rightarrow$ Convergence and optimality analysis (non-Lipschitz gradient)
- Envisioned application domains
  (a) Real-time video denoising and imputation
  (b) Dynamic network traffic monitoring
  (c) Large-scale RNA sequencing analysis
  (d) Recommendation systems