# MEMORY-LIMITED STOCHASTIC APPROXIMATION FOR POISSON SUBSPACE TRACKING

*Liming Wang*[†] *and Yuejie Chi*[†,‡]

[†]Department of Electrical & Computer Engineering, The Ohio State University, OH 43210
[‡]Department of Biomedical Informatics, The Ohio State University, OH 43210

**ABSTRACT**

Poisson noise is ubiquitously encountered in applications including medical and photon-limited imaging. We consider the problem of recovering and tracking the underlying Poisson rate, where the rate vector is assumed to lie in an unknown low-dimensional subspace, with possibly missing entries. A stochastic approximation (SA) algorithm is proposed to solve the problem. This algorithm alternates between two steps. It sequentially identifies the underlying subspace, and recovers coefficients associated with the subspace. The SA algorithm is then modified to obtain a memory-efficient algorithm without storing all historic data. Two theoretical performance guarantees are establish regarding the convergence of SA algorithm. Numerical experiments are provided to demonstrate the proposed algorithms for Poisson video. The memory-limited SA algorithm is shown to empirically yield similar performances to the original SA algorithm.

***Index Terms*—** Poisson noise, subspace tracking, stochastic approximation

## 1. INTRODUCTION

For many applications such as X-ray imaging [1, 2], photonics [3] and astronomical imaging [4], a Gaussian noise model is not appropriate. The observed data in these application is typically characterized via the Poisson noise model, which significantly differs from the traditional Gaussian counterpart, and calls for new processing techniques.

Consider a high-dimensional data stream under Poisson noise, and the observed data vector $\mathbf{y}_n \in \mathbb{Z}_+^N$ at each time $n$ is often modeled as $\mathbf{y}_n \sim \text{Pois}(\mathbf{z}_n)$, where $\text{Pois}(\cdot)$ denotes the vector Poisson distribution, and $\mathbf{z}_n \in \mathbb{R}_+^N$ is the rate vector. Furthermore, the data vectors may not be fully observed due to packet loss, privacy considerations or missing data in many applications. Therefore, it is vital to propose online algorithms that can accurately learn and track the underlying structure of the Poisson model, e.g. changes in the rates, in both computational- and memory-efficient manners, as well as being robust to missing data.

In the Poisson compressed sensing (CS) framework, typically the rate vector is modeled as $\mathbf{z}_n = \mathbf{A}\mathbf{x}_n$, where the aim is to recover the sparse vector $\mathbf{x}_n$, whose dimension is much higher than that of $\mathbf{y}_n$, and the sensing matrix $\mathbf{A}$ is known *a priori*. Algorithms for recovering the rate have been studied in [5], and performance bounds for recovering algorithms have been developed in [6, 7]. The impact of designed sensing matrix $\mathbf{A}$ for Poisson model has been investigated in [8]. In [2], the Poisson CS framework is extended to the multiple measurement setting, where it proposes a batch algorithm to recover multiple sparse vectors $\{\mathbf{x}_n\}$. Similarly, [9, 10] developed batch algorithms for the Poisson matrix completion problem, which aims to recover the rate vectors $\{\mathbf{z}_n\}$, assuming it lies in a low-dimensional subspace. These batch algorithms become highly insufficient in terms of computational cost and storage complexity for large-scale data streams, and do not adapt to changes.

Dictionary model [11] is an effective dimensionality reduction technique for high-dimensional data such as images and video sequences, where the data is represented by a linear combination of the columns of a subspace matrix $\mathbf{D}$ and the dimension of this column subspace is significantly lower than the ambient dimension of the data. This model has been successfully applied to image and video processing under Gaussian noise. In [12, 13], efficient online algorithms based on Gaussian noise model for subspace tracking and reconstruction has been considered.

Inspired by the aforementioned work for Gaussian noise model, this paper pursues the overarching scheme for streaming data under Poisson noise, and proposes online algorithms for Poisson subspace identification as well as data reconstruction. We model the streaming data under the Poisson noise whose parameters possess a subspace representation. The underlying subspace is identified by minimizing the expectation of a suitable loss function. The stochastic approximation (SA) framework [14] is leveraged to develop the online algorithm, and two steps are derived to solve the optimization problem. Under a few mild assumptions, we also establish the convergence of proposed SA algorithm. However, distinct from the Gaussian model, the formulation of Poisson statistics prohibit an easy adaptation to a memory-efficient implementation. Alternatively, we derive a lower bound of the Poisson log-likelihood to mitigate this issue, yielding a *memory-efficient* modification of proposed SA algorithm which can also handle missing data. The memory-limited SA algorithm is shown to yield similar performance to the original SA algorithm. These algorithms have important practical applications such as Poisson video, which is demonstrated in the experiments.

The paper is organized as follows. In Section 2, we first introduce our model and the problem formulation. In Section 3, we propose the stochastic approximation algorithms, and develop memory-limited modifications to the SA algorithm, which also allow for the missing data case. We present the convergence guarantees of the proposed SA algorithm in Section 4. Numerical experiments for real Poisson video are presented in Section 5. We conclude the paper in Section 6.

## 2. SIGNAL MODEL AND PROBLEM STATEMENT

Consider the following signal model

$$\mathbf{y}_n \sim \text{Pois}(\mathbf{D}\mathbf{a}_n), \qquad n = 1, \ldots, M, \tag{1}$$

This work is supported in part by AFOSR under the grant FA9550-15-1-0205, by ONR under the grant N00014-15-1-2387.

where the $n$-th observation $\mathbf{y}_n = [y_{n,1}, \ldots, y_{n,N}]^T \in \mathbb{Z}_+^N$, $\mathbf{D} = [\mathbf{d}_1, \ldots, \mathbf{d}_N]^T \in \mathbb{R}_+^{N \times K}$ and $\mathbf{a}_n \in \mathbb{R}_+^K$. $\mathrm{Pois}(\cdot)$ denotes the vector-Poisson distribution, *i.e.,* for vector $\mathbf{x} \in \mathbb{R}_+^N$, $\mathrm{Pois}(\mathbf{x}) = \prod_{i=1}^N \mathrm{pois}((\mathbf{x})_i)$, where $\mathrm{pois}(\cdot)$ is the common scalar Poisson distribution with parameter $(\mathbf{x})_i$. Via this model, the underlying rate $\mathbf{z}_n = \mathbf{D}_n \mathbf{a}_n$ is assumed to reside in a low-dimensional subspace spanned by the columns of $\mathbf{D}$, and $\mathbf{a}_n$ specifies the linear combination of columns. Let $\mathbf{p}_n = [p_{n,1}, \ldots, p_{n,N}]^T \in \{0,1\}^N$ denote a binary mask at time $n$, where $p_{n,i} = 1$ if the $i$-th entry of $\mathbf{y}_n$ is observed, and $p_{n,i} = 0$ otherwise. Given the sequential full-observations $\{\mathbf{y}_n\}_{n=1}^M$ or partial-observations $\{\mathbf{p}_n \odot \mathbf{y}_n, \mathbf{p}_n\}_{n=1}^M$, the goal is to recover the span of the unknown subspace matrix $\mathbf{D}$ and corresponding rate vectors $\{\mathbf{z}_n\}_{n=1}^M$.

Given the assumed Poisson noise model, we manifest a loss function with respect to $\mathbf{y}_n$ and $\mathbf{D}$ as

$$\ell(\mathbf{y}_n, \mathbf{D}) := \min_{\mathbf{a}_n \in \mathbb{R}_+^K} \left[ -\log \mathrm{Pois}(\mathbf{y}_n; \mathbf{D}\mathbf{a}_n) + \lambda \|\mathbf{D}\|_F^2 \right], \quad (2)$$

where $\mathrm{Pois}(\mathbf{y}_n; (\mathbf{D}\mathbf{a}_n))$ denotes the vector Poisson likelihood function with rate $\mathbf{D}\mathbf{a}_n$, $\lambda > 0$ is a preset positive constant and the term $\|\mathbf{D}\|_F^2$ is a Tikhonov regularization term.

Motivated by the formulation in [14], the goal is to recover $\mathbf{D}$ by minimizing the expected loss $f(\mathbf{D}) := \mathbb{E}_{\mathbf{y}_n}[\ell(\mathbf{y}_n, \mathbf{D})]$ as

$$\hat{\mathbf{D}} = \operatorname*{argmin}_{\mathbf{D} \in \mathbb{R}_+^{N \times K}} [f(\mathbf{D})]. \quad (3)$$

Once $\hat{\mathbf{D}}$ is obtained, $\hat{\mathbf{a}}_n$ can be derived via

$$\hat{\mathbf{a}}_n = \operatorname*{argmin}_{\mathbf{a}_n \in \mathbb{R}_+^K} [-\log \mathrm{Pois}(\mathbf{y}_n; \hat{\mathbf{D}}\mathbf{a}_n)], \quad (4)$$

and the rate vector can be estimated as $\hat{\mathbf{z}}_n = \hat{\mathbf{D}}\hat{\mathbf{a}}_n$.

## 3. ALGORITHMS

### 3.1. A Stochastic Approximation Algorithm

The problem in (3) is a non-convex stochastic programming, and we seek its solution via leveraging the stochastic approximation (SA) framework [14]. We first define the empirical loss at time $t$ as

$$f_t(\mathbf{D}) := \frac{1}{t} \sum_{n=1}^{t} \ell(\mathbf{y}_n, \mathbf{D}). \quad (5)$$

By the strong law of large number, $f_t(\mathbf{D}) \to f(\mathbf{D})$ almost surely (a.s.) as $t \to \infty$.

At each time $t$, we aim to approximate the problem (3) via replacing the objective function $f(\mathbf{D})$ by the empirical loss $f_t(\mathbf{D})$. Hence, problem (3) can be approximated as

$$\hat{\mathbf{D}}_t = \operatorname*{argmin}_{\mathbf{D} \in \mathbb{R}_+^{N \times K}} \frac{1}{t} \sum_{n=1}^{t} \min_{\mathbf{a}_n} [-\log \mathrm{Pois}(\mathbf{y}_n; \mathbf{D}\mathbf{a}_n)] + \lambda \|\mathbf{D}\|_F^2. \quad (6)$$

Invoking the stochastic approximation framework, we aim to solve problem (6) by alternating between two steps, *nonnegative encoding* and *subspace update*. Specifically, at time $t$, we first learn the coefficient vector $\hat{\mathbf{a}}_t$, given the new data $\mathbf{y}_t$ and previously learned subspace $\hat{\mathbf{D}}_{t-1}$. Namely, the estimate $\hat{\mathbf{a}}_t$ is obtained by minimizing the loss function:

$$\hat{\mathbf{a}}_t = \operatorname*{argmin}_{\mathbf{a} \in \mathbb{R}_+^K} -\log \mathrm{Pois}(\mathbf{y}_t; \hat{\mathbf{D}}_{t-1}\mathbf{a}). \quad (7)$$

**Algorithm 1** Stochastic Approximation (SA)

**Input:** Data $\{\mathbf{y}_n\}_{n=1}^M$, $\lambda$, initialization $\mathbf{D}_0$

**Output:** Subspace estimates $\{\hat{\mathbf{D}}_t\}_{t=1}^M$ and $\{\hat{\mathbf{a}}_t\}_{t=1}^M$

1: **for** $t = 1$ to $M$ **do**

2: Estimate the coefficient $\hat{\mathbf{a}}_t$ by the following optimization via projected gradient descent:

$$\hat{\mathbf{a}}_t = \operatorname*{argmin}_{\mathbf{a} \in \mathbb{R}_+^K} -\log \mathrm{Pois}(\mathbf{y}_t; \hat{\mathbf{D}}_{t-1}\mathbf{a});$$

3: Update each row of the subspace $\hat{\mathbf{D}}_t$ by the following optimization via projected gradient descent:

$$\hat{\mathbf{d}}_{t,i} = \operatorname*{argmin}_{\mathbf{d}_i \in \mathbb{R}_+^K} -\frac{1}{t} \sum_{n=1}^{t} \log \mathrm{Pois}(y_{n,i}; \mathbf{d}_i^T \hat{\mathbf{a}}_n) + \lambda \|\mathbf{d}_i\|_2^2.$$

4: **end for**

Once we obtain $\hat{\mathbf{a}}_t$, the subspace $\hat{\mathbf{D}}_t$ is then updated by minimizing, based on previous estimates $\{\hat{\mathbf{a}}_n\}_{n=1}^t$ and observations $\{\mathbf{y}_n\}_{n=1}^t$:

$$\hat{\mathbf{D}}_t = \operatorname*{argmin}_{\mathbf{D} \in \mathbb{R}_+^{N \times K}} \left\{ -\frac{1}{t} \sum_{n=1}^{t} \log \mathrm{Pois}(\mathbf{y}_n; \mathbf{D}\hat{\mathbf{a}}_n) + \lambda \|\mathbf{D}\|_F^2 \right\}. \quad (8)$$

Moreover, (8) can be decomposed into a set of smaller problems for each row of the subspace matrix. Specifically, the $i$th row of $\mathbf{D}$ can be updated in parallel as

$$\hat{\mathbf{d}}_{t,i} = \operatorname*{argmin}_{\mathbf{d}_i \in \mathbb{R}_+^K} -\frac{1}{t} \sum_{n=1}^{t} \log \mathrm{Pois}(y_{n,i}; \mathbf{d}_i^T \hat{\mathbf{a}}_n) + \lambda \|\mathbf{d}_i\|_2^2. \quad (9)$$

We summarize the proposed stochastic approximate (SA) algorithm in Algorithm 1. Both (7) and (9) can be solved efficiently via projected gradient descent. Below we discuss the details for solving (7), and (9) can be solved similarly. Specifically, we find $\hat{\mathbf{a}}_t$ iteratively and at the $(k+1)$-th iteration, the update is calculated as

$$\hat{\mathbf{a}}_t^{(k+1)} = \mathrm{Proj}\left( \hat{\mathbf{a}}_t^{(k)} - \alpha_k \nabla g(\hat{\mathbf{a}}_t^{(k)}) \right),$$

where $g(\mathbf{a}) = -\log \mathrm{Pois}(\mathbf{y}_t; \hat{\mathbf{D}}_{t-1}\mathbf{a})$ and $\mathrm{Proj}(\mathbf{a}) := \max\{\mathbf{a}, 0\}$ is the projection operator, where $\max$ operator denotes the entry-wise maximization. Moreover, $\alpha_k$ is the step size and can be set as:

$$\alpha_k = \frac{\left( \hat{\mathbf{a}}_t^{(k)} - \hat{\mathbf{a}}_t^{(k-1)} \right)^T \left[ \nabla g(\hat{\mathbf{a}}_t^{(k)}) - \nabla g(\hat{\mathbf{a}}_t^{(k-1)}) \right]}{\left\| \nabla g(\hat{\mathbf{a}}_t^{(k)}) - \nabla g(\hat{\mathbf{a}}_t^{(k-1)}) \right\|_2^2}$$

following [15]. The regularization parameter $\lambda$ can be empirically determined via cross-validation and a random initialization $\mathbf{D}_0$ can be utilized.

### 3.2. Memory-limited SA Algorithm

When the data vectors are partially observed, we can replace the log-likelihood in (6) by $\sum_{i=1}^N p_{n,i} \log \mathrm{Pois}(y_{n,i}; \mathbf{d}_i^T \mathbf{a}_n)$ and modify accordingly. However, the optimization in (9) requires to store all previous $\{\hat{\mathbf{a}}_n\}$ and $\{\mathbf{y}_n\}$, yielding a significant storage cost, particularly when $t$ is large. In order to mitigate this issue, we modify the SA algorithm to a memory-limited algorithm that only demands a storage of sufficient statistics of previous data, which is more appealing when the data index is large.

In order to facilitate the derivation, we make two assumptions.
A1) $\mathbf{D} \in \mathcal{C}_2$ where $\mathcal{C}_2 \subset \mathbb{R}_+^{N \times K}$ is a compact set.
A2) There exist positive constants $a$ and $b$ such that $0 < a \leq (\mathbf{D}\mathbf{a}_n)_i \leq b$, for all $n$ and $i$.

In other words, we assume that entries of $\mathbf{D}\mathbf{a}_n$ are bounded. The first one essentially assumes that $\mathbf{D}$ has bounded entries and this is a very mild assumption for real applications. The second assumption is used to exclude the singular case, where some Poisson rates asymptotically approach zero. Similar assumption has also been utilized in [2, 6].

Our goal is to derive a memory-limited SA algorithm for Poisson data that only demands storing sufficient statistics of previous data, which is more appealing for streaming applications. Unfortunately, the Poisson log-likelihood function prohibits such an easy adaptation. Rather than dealing with the original Poisson log-likelihood function, we will establish an upper bound which is more amenable for memory-limited implementations and the upper bound the Poisson log-likelihood function is derived in the following proposition.

**Proposition 1.** *With previous assumptions, we have the following bound*

$$\begin{aligned} &-\sum_{n=1}^{t} p_{n,i} \log \mathrm{Pois}(y_{n,i}; \mathbf{d}_i^T \mathbf{a}_n) \leq \mathbf{d}_i^T \left( \sum_{n=1}^{t} p_{n,i} \mathbf{a}_n \right) \\ &- \left( \sum_{n=1}^{t} p_{n,i} y_{n,i} \right) \log \left[ \mathbf{d}_i^T \left( \sum_{n=1}^{t} p_{n,i} y_{n,i} \mathbf{a}_n \right) \right] \\ &+ \sum_{n=1}^{t} p_{n,i} \log(y_{n,i}!) \\ &+ \left( \sum_{n=1}^{t} p_{n,i} y_{n,i} \right) \left[ \log \left( \sum_{n=1}^{t} p_{n,i} y_{n,i} \right) + T \right], \end{aligned} \tag{10}$$

*where $T$ is a constant only depending on $a$ and $b$, as in assumption A2.*

Replacing the log-likelihood function by the above upper bound, then the rows of the subspace $\mathbf{D}$ can be similarly updated in parallel as

$$\hat{\mathbf{d}}_{t,i} = \underset{\mathbf{d}_i \in \mathbb{R}_+^K}{\operatorname{argmin}} \, \mathbf{d}_i^T \mathbf{s}_{t,i} - \beta_{t,i} \log(\mathbf{d}_i^T \widetilde{\mathbf{r}}_{t,i}) + \lambda \|\mathbf{d}_i\|_2^2. \tag{11}$$

where $\mathbf{s}_{t,i} = \frac{1}{t} \sum_{n=1}^{t} p_{n,i} \hat{\mathbf{a}}_n$, $\beta_{t,i} = \frac{1}{t} \sum_{n=1}^{t} p_{n,i} y_{n,i}$, and $\mathbf{r}_{t,i} = \sum_{n=1}^{t} \hat{\mathbf{a}}_n p_{n,i} y_{n,i}$. Hence, we can formulate a memory-limited SA algorithm with missing data that alternates between estimating $\hat{\mathbf{a}}_t$ and updating $\hat{\mathbf{D}}_t$, which is referred as the memory-limited SA algorithm. It is easy to see that the memory-limited SA algorithm only requires a storage independent of time index $t$. The algorithm is summarized in Algorithm 2.

## 4. CONVERGENCE ANALYSIS

In this section, we provide a convergence analysis for the proposed SA algorithm. We first define

$$\ell'(\mathbf{y}_n, \mathbf{D}, \mathbf{a}_n) := -\log \mathrm{Pois}(\mathbf{y}_n; \mathbf{D}\mathbf{a}_n) + \lambda \|\mathbf{D}\|_F^2, \tag{15}$$

**Algorithm 2** Memory-Limited Stochastic Approximation for Poisson Streaming Data with Missing Data

**Input:** Data $\{\mathbf{y}_n\}_{n=1}^M$, $\lambda$, initialization $\mathbf{D}_0$, $\widetilde{\mathbf{s}}_{0,i} = 0$, $\widetilde{\beta}_{0,i} = 0$ and $\widetilde{\mathbf{r}}_{0,i} = 0$ for all $1 \leq i \leq N$.

**Output:** Subspace estimates $\{\hat{\mathbf{D}}_t\}_{t=1}^M$ and $\{\hat{\mathbf{a}}_t\}_{t=1}^M$

**for** $t = 1$ to $M$ **do**

Estimate the coefficient $\hat{\mathbf{a}}_t$ by the following optimization via projected gradient descent

$$\hat{\mathbf{a}}_t = \underset{\mathbf{a} \in \mathbb{R}_+^K}{\operatorname{argmin}} - \sum_{i=1}^{N} p_{t,i} \log \mathrm{Pois}(y_{t,i}; \mathbf{d}_i^T \mathbf{a});$$

Update the sufficient statistics, for $1 \leq i \leq N$, as

$$\widetilde{\mathbf{s}}_{t,i} = \frac{t-1}{t} \widetilde{\mathbf{s}}_{t-1,i} + \frac{1}{t} p_{t,i} \hat{\mathbf{a}}_t, \tag{12}$$

$$\widetilde{\beta}_{t,i} = \frac{t-1}{t} \beta_{t,i} + \frac{1}{t} p_{t,i} y_{t,i}, \tag{13}$$

$$\widetilde{\mathbf{r}}_{t,i} = \widetilde{\mathbf{r}}_{t-1,i} + \hat{\mathbf{a}}_t p_{t,i} y_{t,i}; \tag{14}$$

Update each row of the subspace $\hat{\mathbf{D}}_t$ by the following optimization via projected gradient descent,

$$\hat{\mathbf{d}}_{t,i} = \underset{\mathbf{d}_i \in \mathbb{R}_+^K}{\operatorname{argmin}} \, \mathbf{d}_i^T \widetilde{\mathbf{s}}_{t,i} - \widetilde{\beta}_{t,i} \log(\mathbf{d}_i^T \widetilde{\mathbf{r}}_{t,i}) + \lambda \|\mathbf{d}_i\|_2^2.$$

**end for**

and

$$f_t'(\mathbf{D}) := \frac{1}{t} \sum_{i=1}^{t} \ell'(\mathbf{y}_i, \mathbf{D}, \hat{\mathbf{a}}_i). \tag{16}$$

where $\hat{\mathbf{a}}_i$ and $\hat{\mathbf{D}}_t$ are the output of the SA algorithm 1. Note that $f_t'(\mathbf{D})$ captures the empirical loss under the online algorithm.

In order to facilitate the convergence analysis, in addition to previous assumptions A1) and A2), we make two further assumptions.
A3) The observations $\{\mathbf{y}_n\}$ are supported on a compact set $\mathcal{C}_1$.
A4) There exists a unique minimizer $\mathbf{a}_n$ of $\ell(\mathbf{y}_n, \mathbf{D})$, for any $(\mathbf{y}_n, \mathbf{D}) \in \mathcal{C}_1 \times \mathcal{C}_2$.

A3) essentially assumes that the observed data is bounded and such an assumption is naturally satisfied for real data. A4) is valid if $-\log \mathrm{Pois}(\mathbf{y}_n; \mathbf{D}\mathbf{a}_n)$ is a strictly convex function. Therefore, such an assumption can be guaranteed if a strictly convex regularizer such as $\|\mathbf{a}_n\|_2^2$ is added to the loss function. However, we omit such regularizers for simplicity.

Our first theorem states the almost sure convergence of the SA algorithm that the empirical loss $f_t'(\hat{\mathbf{D}}_t)$ and the original loss $f(\hat{\mathbf{D}}_t)$ converge to the same limit under the output of Algorithm 1.

**Theorem 1.** *With assumptions A1-A4, the stochastic processes $\{f_t(\hat{\mathbf{D}}_t)\}$,$\{f_t'(\hat{\mathbf{D}}_t)\}$ and $\{f(\hat{\mathbf{D}}_t)\}$ converge a.s. to the same limit.*

In addition, our second theorem states that the estimated subspace $\hat{\mathbf{D}}_t$ produced by Algorithm 1 also almost surely converge to a local minimum of $f$.

**Theorem 2.** *With assumptions A1-A4, consider a sequence $\{\hat{\mathbf{D}}_t\}$ such that Theorem 1 holds. Then with probability 1, $\hat{\mathbf{D}}_t$ converges to a local minimum of the expected loss $f$.*

Unfortunately, both the proof techniques for Theorem 1 and 2 can only be applied to SA algorithm 1, and cannot be easily adapted to proposed memory-limited SA algorithm. However, Theorem 1 and 2 still serve as a convergence implication for the online algorithm, provided that the online algorithm yields similar performances to the SA algorithm. Moreover, we can show that the gap of the bound in Proposition 1 does not grow with the increase of time index. As we present in Sec. 5, it is found that the bound is empirically tight for numerical experiments.

## 5. EXPERIMENTS

In this section, we showcase the proposed algorithms, *i.e.,* the SA and memory-limited SA algorithms on real video sequences under Poisson noise. The gray-scale video is of a resolution $50 \times 50$ with total 250 frames and the $n$th frame is regarded as a 2500-dimensional vector $\mathbf{z}_n$ of its gray scale. In order to determine the rank of the data $[\mathbf{z}_1, \ldots, \mathbf{z}_{250}]$, we use SVD to calculate the approximate rank. Hence, we set $N = 2500$, rank $K = 40$, $M = 250$ and $\lambda = 0.2$. The observations are the Poisson counts $\mathbf{y}_n \sim \text{Pois}(\mathbf{z}_n)$, where each entry of $\mathbf{y}_n$ is observed independently with probability $p$. We compute the relative video reconstruction error at the $n$th frame as $\|\hat{\mathbf{D}}_n \hat{\mathbf{a}}_n - \mathbf{z}_n\|_2 / \|\mathbf{z}_n\|_2$.

Illustration of original video frame and recovery are showcased in Fig. 1. Relative errors of the recovered parameters via the SA and the memory-limited SA algorithms when $p = 1$ and $p = 0.5$ are shown in Fig. 2 and 3, respectively. It is demonstrated that the performance improves with the increase of the data stream index. We also compare our results to the state-of-the-art subspace tracking algorithm in [16] that assumes the Gaussian model for incoming data, and treat the Poisson observations as the input of the Gaussian algorithm in [16]. However, we find that its performance is so poor that the relative error is significantly greater than 1 and does not improve at all with the increase of the time index. Hence, we omit showing the performance of the Gaussian algorithm in these figures.

**Fig. 1**: Illustration of recovered video frame. Top left is the original video frame. Top right is the Poisson observation. Bottom left is the recovered video when $p = 1$. Bottom right is the recovered video frame when $p = 0.5$.

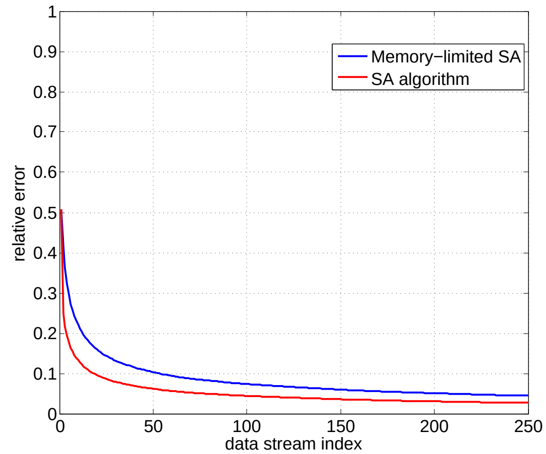


**Fig. 2**: The relative reconstruction errors for SA and memory-limited SA algorithms when the data is fully observed with $p = 1$.

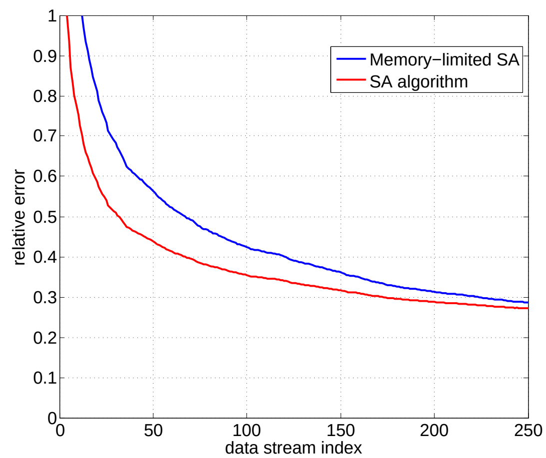


**Fig. 3**: The relative reconstruction errors for SA and memory-limited SA algorithms when the data is partially observed with $p = 0.5$.

## 6. CONCLUSION

We have considered the problem of recovering and tracking the underlying Poisson rate for streaming data under Poisson noise, where the rate has been posed to lie in a low-dimensional subspace structure, possibly with missing data entries. A stochastic programming has been proposed to recover the underlying subspace. A stochastic approximation algorithm has first been derived. The SA algorithm has been decomposed into two steps where the subspace and its coefficients are sequentially updated with new data. Theoretical convergence guarantees have been established for the SA algorithm. The SA algorithm has been proved to converge to the same point as the original stochastic programming. In addition, the estimated subspace has been shown to converge to a local minimum of original objective. In order to mitigate the storage requirement, the SA algorithm has been modified to a memory-limited SA algorithm. We have demonstrated that the memory-limited SA algorithm yields similar performance to the SA algorithm. Both algorithms have been showcased to achieve promising performances.

## 7. REFERENCES

[1] M. Seeger, H. Nickisch, R. Pohmann, and B. Schölkopf, "Optimization of k-space trajectories for compressed sensing by bayesian experimental design," *Magnetic resonance in medicine*, vol. 63, no. 1, pp. 116–126, 2010.

[2] L. Wang, J. Huang, X. Yuan, K. Krishnamurthy, J. Greenberg, V. Cevher, M. R. Rodrigues, D. Brady, R. Calderbank, and L. Carin, "Signal recovery and system calibration from multiple compressive poisson measurements," *SIAM Journal on Imaging Sciences*, vol. 8, no. 3, pp. 1923–1954, 2015.

[3] D. J. Brady, *Optical imaging and spectroscopy*. John Wiley & Sons, 2009.

[4] R. Berry and J. Burnell, "Astronomical image processing," *Willman-Bell, Inc*, 2000.

[5] Z. T. Harmany, R. F. Marcia, and R. M. Willett, "This is SPIRAL-TAP: Sparse poisson intensity reconstruction algorithms-theory and practice," *IEEE Transactions on Image Processing*, vol. 21, no. 3, pp. 1084–1096, 2012.

[6] M. Raginsky, S. Jafarpour, Z. T. Harmany, R. F. Marcia, R. Willett, and R. Calderbank, "Performance bounds for expander-based compressed sensing in Poisson noise," *IEEE Transactions on Signal Processing*, vol. 59, no. 9, pp. 4139–4153, 2011.

[7] M. Raginsky, R. Willett, Z. Harmany, and R. Marcia, "Compressed sensing performance bounds under Poisson noise," *IEEE Transactions on Signal Processing*, vol. 58, no. 8, pp. 3990–4002, 2010.

[8] L. Wang, D. E. Carlson, M. Rodrigues, D. Wilcox, R. Calderbank, and L. Carin, "Designed measurements for vector count data," in *Advances in neural information processing systems*, 2013, pp. 1142–1150.

[9] Y. Xie, Y. Chi, and R. Calderbank, "Low-rank matrix recovery with poisson noise," in *Global Conference on Signal and Information Processing (GlobalSIP), 2013 IEEE*. IEEE, 2013, pp. 622–622.

[10] Y. Cao and Y. Xie, "Poisson matrix completion," in *Information Theory (ISIT), 2015 IEEE International Symposium on*. IEEE, 2015, pp. 1841–1845.

[11] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM review*, vol. 43, no. 1, pp. 129–159, 2001.

[12] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online dictionary learning for sparse coding," in *Proceedings of the 26th annual international conference on machine learning*. ACM, 2009, pp. 689–696.

[13] Y. Chi, Y. C. Eldar, and R. Calderbank, "Petrels: Parallel estimation and tracking of subspace by recursive least squares from partial observations," *IEEE Trans. on Signal Processing*, 2013.

[14] L. Bottou, "Online learning and stochastic approximations," *On-line learning in neural networks*, vol. 17, no. 9, p. 142, 1998.

[15] J. Barzilai and J. M. Borwein, "Two-point step size gradient methods," *IMA journal of numerical analysis*, vol. 8, no. 1, pp. 141–148, 1988.

[16] H. Guo, C. Qiu, and N. Vaswani, "An online algorithm for separating sparse and low-dimensional signal sequences from their sum," *IEEE Transactions on Signal Processing*, vol. 62, no. 16, pp. 4284–4297, 2014.