# Online Denoising of Discrete Noisy Data

Pejman Khadivi Ravi Tandon Naren Ramakrishnan
Discovery Analytics Center and Department of Computer Science
Virginia Tech, Blacksburg, Virginia 24060
E-mail: {pejman, tandonr, naren}@cs.vt.edu

***Abstract*—Real-time data-driven systems often utilize discrete valued time series data and their functionality is highly dependent on the accuracy of such data. In order to improve the performance of these systems, an important pre-processing step is the denoising of data before performing any action (e.g. forecasting or control activities). Existing algorithms have primarily focused on the offline denoising problem, which requires the entire data to be collected before the denoising process. In this paper, the problem of online discrete denoising is considered. The online denoising problem is motivated by real-time applications, where the data must be utilizable soon after it is collected. Three online denoising algorithms are proposed which can strike a tradeoff between delay and accuracy of denoising. It is also shown that the proposed online algorithms asymptotically converge to a class of optimal offline block denoisers.**

***Index Terms*—Online Denoising, Discrete Denoising**

## I. Introduction

With emerging data-driven applications, such as data-driven system design, control systems, and data mining, noise removal from data sources is becoming increasingly important [1], [2], [3]. In such applications, we typically encounter data in the form of discrete valued time series, which could either be generated automatically at the output of sensors, or reported manually (such as in collection of disease counts reported by health care personnel [1]). However, such data is often noisy in nature, where the noise could be an artifact of incorrect measurements, faulty sensors, or imperfect data collection mechanisms. In time-sensitive applications, (such as forecasting or control activities), immediate usability of such noisy data is of critical importance. This leads to the problem of *online denoising*, in which data must be denoised immediately after being collected.

Prior work in denoising literature (such as [4], [5], [6], [7]) has primarily focused on the problem of *offline denoising*, which assumes the availability of the entire data. Weissman et al. in [4] proposed the discrete universal denoiser (DUDE) algorithm for *offline denoising* of $x^n$ from its noisy version $z^n$, with the assumption of an i.i.d noise generating mechanism (modeled through a noisy memoryless channel $p(z|x)$). DUDE assumes the statistical knowledge of the noisy mechanism, i.e., $p(z|x)$, but makes no assumptions on the distribution of the underlying data $x^n$. It is shown in [4], that this algorithm converges asymptotically to the optimum Bayes offline denoiser.

Several other offline denoising algorithms have been subsequently developed that are inspired by DUDE, such as [5], [7], and [8]. In [5], original DUDE algorithm is extended for denoising of grayscale images. To this end, an extension for the case of large alphabets is presented. [8] proposes another version of this algorithm named as S-DUDE, which attempts to address the non-stationarity of data. To handle correlation in the noise generating mechanism, an extension for the case of noisy channels with memory is presented in [9] and [10]. Furthermore, [11] addresses the issue of knowledge uncertainty in the statistics of the noise generating mechanism. While the above extensions are indeed valuable, they are primarily targeted at making the denoising mechanisms more robust and do not address the timeliness aspect of denoising, which is of importance in many applications, when data must be denoised on the fly.

In this paper, we focus on online discrete denoising problem to address the issue of data correction in time-sensitive applications. For this purpose, we precisely formulate the online denoising problem and propose three algorithms for online universal denoising. These algorithms can strike a tradeoff between time-sensitivity and denoising accuracy.

- *Repetable Online Denoising (ROD)* algorithm, which upon the collection of a new data point, denoises the entire past and current data points. This algorithm sacrifices delay for accuracy and is suitable for scenarios in which the past data could also be useful.
- *One-time Online Denoising (OOD)* algorithm, which only denoises the currently observed data point. This algorithm sacrifices accuracy for delay and is suitable for time-sensitive applications, where the data must be used immediately for real-time forecasting or real-time control commands.
- *Hybrid Online Denoising (HOD)* algorithm, which combines the timeliness of OOD with the accuracy of ROD. In this algorithm, denoising starts by initially applying the ROD algorithm. After a specific period of time, when data statistics become sufficiently reliable, denoising is substituted with OOD to speed up the denoising process.

The proposed algorithms belong to the class of block denoisers and are applicable for denoising the data streams when characteristics of the noise generation mechanism are known and the data points are from a finite alphabet. Universal denoising of an online sequence of symbols from a finite alphabet, also known as universal filtering, has been addressed before in [12]. In [12], authors study the association between the filtering problem and the problem of prediction of individual sequences and provide a general formulation for the construction of filters. They also show that how their general universal filtering solution may result in an online

version of DUDE. However, the filtering solution that has been provided in [12] does not address the issue of tradeoff between time-sensitivity and denoising accuracy which is the focus of this paper. In this paper, we construct a universal online denoiser based on DUDE that supports unbalanced contexts. We provide three algorithms to address the tradeoff between time-sensitivity and denoising accuracy and prove that the proposed algorithms asymptotically converge to the optimum block denoiser. We also present numerical results which support the theoretical aspects of the paper.

## II. Problem Formulation

We consider an arbitrary discrete-valued $n$ length sequence $x^n = (x_1, \ldots, x_n)$, which is sequentially observed through a noisy mechanism. In particular, at time $t$, a noisy version of $x_t$, which is denoted by $z_t$ is observed, where $x_t, z_t \in \mathcal{A} = \{\alpha_1, \cdots, \alpha_M\}$. The noise generating mechanism is assumed to be i.i.d. over time and described through the transition matrix $\Pi_{M\times M}$, where $\Pi(a,b)$ is the probability of observing $b$ if $a$ is the underlying true data. The $i^{\text{th}}$ column of $\Pi$ is illustrated by $\pi_i$. We denote by $z^t = (z_1, \ldots, z_t)$ (resp., $x^t = (x_1, \ldots, x_t)$) as the noisy sub-sequence (resp., underlying data sub-sequence) available up and until time $t$. To measure denoising accuracy, we define a loss function through the matrix, $\Lambda_{M\times M}$, where $\Lambda(a,b)$ is the loss incurred by estimating $a$ by $b$. The $i^{\text{th}}$ column of $\Lambda$ is illustrated by $\lambda_i$. Also, the maximum possible loss is defined as $\Lambda_{max} = \max_{a,b}\Lambda(a,b)$. The goal of online denoising is to sequentially produce a denoised version $\hat{x}^t$ using $z^t$ for each time $t$.

**Definition 1** *Online discrete denoising is the process of reproducing a new sequence of symbols, $\hat{x}^t$, at each time $t$, based on a received noisy sequence, $z^t$, such that:*

1) *The total loss in $\hat{x}^t$ is less than the total loss in $z^t$, i.e.,*
$$\sum_{i=1}^{t}\Lambda(x_i,\hat{x}_i) \le \sum_{i=1}^{t}\Lambda(x_i,z_i) \qquad (1)$$
2) *Reproduced symbol, $\hat{x}_t$, is generated with an acceptable amount of delay, i.e., $\delta << n$, after receiving $z_t$, where $n$ is the length of the entire sequence. Note that under no delay constraint, this reduces to offline denoising.*

We first review the concept of offline denoising and describe DUDE as a universal block denoiser [4]. Let us assume that we have received the *entire noisy sequence* $z^n$ and we want to denoise the symbol at time $t$, i.e. $z_t$. The optimum Bayes denoiser is the one that minimizes the expected loss of estimating $x_t$. In other words, considering the posterior distribution of the *entire* noiseless data, $x^n$, we have

$$\hat{X}^{opt}(z^n)[t] = \arg\min_{\hat{x}\in\mathcal{A}}\sum_{\alpha\in\mathcal{A}}\Lambda(\alpha,\hat{x})\Pr(X_t=\alpha|z^n), \qquad (2)$$

where the summation is the expected loss of denoising $z_t$ to $\hat{x}$, knowing the posterior distribution of $x_t$, i.e., $\Pr(X_t=\alpha|z^n)$. In vector notation, (2) can be shown as follows

$$\hat{X}^{opt}(z^n)[t] = \arg\min_{x\in\mathcal{A}}\lambda_{\hat{x}}^T\mathbf{P}_{X_t|z^n} = \arg\min_{x\in\mathcal{A}}\lambda_{\hat{x}}^T\mathbf{P}_{X_t,z^n}. \qquad (3)$$

where $\mathbf{P}_{X_t|z^n} = [\Pr(X_t=\alpha_1|z^n)\ldots\Pr(X_t=\alpha_M|z^n)]^T$.

In [4], it has been shown that this optimum denoiser can also be formulated as follows:

$$\hat{X}^{opt}(z^n)[t] = \arg\min_{\hat{x}\in\mathcal{A}}\left[\mathbf{P}_{Z_t,z^{n\setminus t}}\right]^T\Pi^{-1}[\lambda_{\hat{x}}\odot\pi_{z_t}] \qquad (4)$$

where, $\mathbf{P}_{Z_t,z^{n\setminus t}} = [\Pr(Z_t=\alpha_1, Z^{n\setminus t}=z^{n\setminus t})\ldots\Pr(Z_t=\alpha_M, Z^{n\setminus t}=z^{n\setminus t})]^T$, and $\odot$ is a pair-wise vector multiplication for vectors $\mathbf{u}$ and $\mathbf{v}$ defined as follows:

$$(\mathbf{u}\odot\mathbf{v})[i] = u_iv_i. \qquad (5)$$

Based on the formulation in (4), the DUDE algorithm [4] develops an empirical estimation procedure for $\mathbf{P}_{Z_t,z^{n\setminus t}}$. In particular, it considers a window of size $2k+1$, symmetrically wrapped around time $t$, i.e. $(z_{t-k},\cdots,z_{t+k})$. Then, the algorithm takes a pass through the whole sequence and counts all the occurrences of $z_{t-k},\cdots,z_{t-1},\beta,z_{t+1},\cdots,z_{t+k}$ for all the possible values of $\beta$. This counting process results in an empirical distribution of symbols that are located at the center of the window. The resulting empirical distribution, which is shown to be an approximation of $\mathbf{P}_{Z_t,z^{n\setminus t}}$, is then used for denoising of $z_t$. In the next section, we propose online universal denoising algorithms for the case of known channel and discrete finite alphabets. We will prove later that the online algorithms converge to the optimum denoiser.

## III. Online Universal Denoiser

In this section we introduce the online universal denoiser. We provide the general denoising rule and then discuss various versions of the algorithm that may be considered as the online version of DUDE [4]. The optimum Bayes denoiser, defined by (4), can be appropriately modified for the online problem such that each new noisy symbol, is corrected with a small appropriate delay. Let us assume that we can tolerate the delay of $\delta$ symbols and we have received symbols up to time $t+\delta$. Then, the online version of (4) can be written as follows

$$\hat{X}^{opt}\left(z^{t+\delta}\right)[t] = \arg\min_{\hat{x}\in\mathcal{A}}\left[\mathbf{P}_{Z_t,z^{(t+\delta)\setminus t}}\right]^T\Pi^{-1}[\lambda_{\hat{x}}\odot\pi_{z_t}] \qquad (6)$$

Based on (6), with each new received symbol, a new symbol, which is $\delta$ steps back of the latest received one, can be optimally denoised. However, it should be noted that with any new received symbol, we have more information about the whole sequence, which in turn, may improve the accuracy of the denoising process. Due to the importance of delay in online denoising, it is desirable to start denoising of a newly received symbol as soon as possible. Therefore, instead of the symmetric double-sided context of [4], we use unbalanced context windows to estimate the conditional distribution $\mathbf{P}_{Z_t,z^{(t+\delta)}}$. Hence, we define the following vector:

$$\mathbf{C}(z^{(t+\delta)},b^k,c^\delta)[\beta] = \left|\left\{i : k+1\le i\le t, z_{i-k}^{i+\delta} = b^k\beta c^\delta\right\}\right| \qquad (7)$$

with the assumption that $\delta \le k$. In fact, $\mathbf{C}(z^{(t+\delta)},b^k,c^\delta)[\beta]$ is the number of times that we have observed $\beta$, wrapped in the context of $(b^k\cdot c^\delta)$ in sequence $z^{(t+\delta)}$. Following the terminology of [4], $b^k$ is the left context, $c^\delta$ is the right context, and $(b^k\cdot c^\delta)$ is the double-sided context. Obviously, when $\delta <$

$k$, the double-sided context is unbalanced and when $\delta = 0$, the vector $\mathbf{C}$ is defined only based on the left-context. Using (7), online denoising of symbol at time $t$ is performed as follows

$$\hat{X}^{k,\delta}(z^{t+\delta})[t] = \arg\min_{\hat{x}\in\mathcal{A}} \mathbf{C}^T(z^{t+\delta}, z_{t-k}^{t-1}, z_{t+1}^{t+\delta})\Pi^{-1}\left[\lambda_{\hat{x}} \odot \pi_{z_t}\right]. \tag{8}$$

Thus, the core aspect of the algorithm is related to counting the number of times that an *unbalanced context* appears inside the received sequence up and until time $t+\delta$. However, as $t$ grows, and more data is collected, it is a time consuming process to start counting the contexts from the beginning of $z^{t+\delta}$. For real-time denoising applications, a more appropriate implementation is to keep the counts up to time $t+\delta$ in memory, and update the counts with a newly received symbol (we denote this memory by $\mathcal{C}$ and the counts by $\mathbf{C}$). It should be noted that with an alphabet of size $M$ and a double-sided context of size $k+\delta$, we need to save $M^{k+\delta}$ vectors, each of size $M$. In other words, the total amount of memory which is required to keep the current counts in the memory is $O\left(M^{k+\delta+1}\right)$. Therefore, there is a trade-off between the time and memory that depends on the values of $M$, $k$, and $\delta$. Depending on the time-sensitivity of the desired denoising process, we next present three online denoising algorithms.

### A. One-time Online Denoiser

In time-sensitive applications, when there is a hard deadline to use the time series, deadline satisfaction is the most important constraint of the system. Therefore, it may not be possible to trade-off time-sensitivity with denoising accuracy. In the first version of the online denoising algorithm, which we name it as One-time Online Denoiser (OOD), the denoiser does not re-process a previously denoised symbol, because it has already been utilized. Hence, it is intuitive to expect higher values of loss for smaller values of $t$. However, as time grows, we will have more reliable information about the data statistics (in the form of $\mathbf{C}$), and hence, it is to be expected that denoising via OOD improves with time. The pseudo-code of OOD is illustrated in Algorithm 1. In this algorithm, $\mathcal{C}$ represents the memory that keeps the updated counts of $\mathbf{C}$ for all possible unbalanced contexts.

The OOD algorithm starts by retrieving the previous counts for the context vector $(z_{t-k}^{t-1} \cdot z_{t+1}^{t+\delta})$ around the new observed symbol $z_t$. We denote this count vector by $\mathbf{C}$ and increment one of its elements, i.e., $\mathbf{C}[z_t]$ by 1. Using the updated count vector $\mathbf{C}$, OOD denoises the $t$th symbol and returns $\hat{x}_t$. Finally, dictionary of all counts ($\mathcal{C}$) is updated by new count vector $\mathbf{C}$.

### B. Repeatable Online Denoiser

In some applications, when we are not facing with hard deadlines, it is possible to trade-off the denoising accuracy with time-sensitivity. In the second version of denoising algorithms, which we name it as Repeatable Online Denoising (ROD), the denoiser is able to go back and reprocess the whole sequence. Comparing with OOD, ROD is a slower denoising algorithm, while it has higher accuracy and converges faster to the optimum denoiser. With a new received symbol, when we move from $t+\delta$ to $t+\delta+1$, the ROD algorithm reconsiders all the similar contexts in the past and using (8), denoises the corresponding symbols again. The pseudo-code of ROD is illustrated in Algorithm 2.

**Algorithm 1** One-time Online Denoising (OOD) Algorithm

**function** OOD($t$,$z_{t-k}^{t+\delta}$,$\mathcal{C}$,$k$,$\delta$,$\Pi$,$\Lambda$)
  $\mathbf{C}(z_{t-k}^{t-1}, z_{t+1}^{t+\delta}) \leftarrow \mathcal{C}[z_{t-k}^{t-1} z_{t+1}^{t+\delta}]$ ▷ (retrieve context counts around $z_t$)
  $\mathbf{C}(z_{t-k}^{t-1}, z_{t+1}^{t+\delta})[z_t] \leftarrow \mathbf{C}(z_{t-k}^{t-1}, z_{t+1}^{t+\delta})[z_t] + 1$ ▷ (update context count)
  $\hat{x}_t = \arg\min_{\hat{x}\in\mathcal{A}} \mathbf{C}^T\Pi^{-1}\left[\lambda_{\hat{x}} \odot \pi_{z_t}\right]$ ▷ (denoise)
  Update $\mathcal{C}[z_{t-k}^{t-1} z_{t+1}^{t+\delta}]$ By $\mathbf{C}$ ▷ (update context dictionary)
  **return** $\hat{x}_t$

**Algorithm 2** Repeatable Online Denoising (ROD) Algorithm

**function** ROD($t$,$z^{t+\delta}$,$\hat{x}^{t+\delta}$,$\mathcal{C}$,$k$,$\delta$,$\Pi$,$\Lambda$)
  $\mathbf{C} \leftarrow \mathcal{C}[z_{t-k}^{t-1} z_{t+1}^{t+\delta}]$
  $\mathbf{C}[z_t] \leftarrow \mathbf{C}[z_t] + 1$
  **for** $i = k+1$ to $t$ **do**
    **if** $z_{i-k}^{i-1} = z_{t-k}^{t-1}$ and $z_{i+1}^{i+\delta} = z_{t+1}^{t+\delta}$ **then**
      $\hat{x}_i = \arg\min_{\hat{x}\in\mathcal{A}} \mathbf{C}^T\Pi^{-1}\left[\lambda_{\hat{x}} \odot \pi_{z_i}\right]$
  Update $\mathcal{C}[z_{t-k}^{t-1} z_{t+1}^{t+\delta}]$ By $\mathbf{C}$
  **return** $\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_t$

**Algorithm 3** Hybrid Online Denoising (HOD) Algorithm

**function** HOD($t$,$z_{t-k}^{t+\delta}$,$\mathcal{C}$,$k$,$\delta$,$\Pi$,$\Lambda$,$\eta$)
  **if** $t \le \eta$ **then**
    **return** ROD($t$,$z^{t+\delta}$,$\hat{x}^{t+\delta}$,$\mathcal{C}$,$k$,$\delta$,$\Pi$,$\Lambda$)
  **return** OOD($t$,$z_{t-k}^{t+\delta}$,$\mathcal{C}$,$k$,$\delta$,$\Pi$,$\Lambda$)

### C. Hybrid Online Denoiser

To overcome the loss in accuracy of OOD and complexity/delay of ROD, a third solution is to use a hybrid algorithm. In this algorithm, which we name it as Hybrid Online Denoising (HOD), ROD algorithm is used for denoising the symbols for the first $\eta$ symbols, where $\eta$ is large enough that lets the unbalanced context counts in $\mathbf{C}$ to be stabilized. After receiving the first $\eta$ symbols, OOD is subsequently used for denoising. Pseudo-code of HOD is presented in Algorithm 3.

## IV. Properties of Proposed Online Denoisers

In this section, we address the convergence properties of the proposed methods. Consider denoising $z^{t+\delta}$ to the noiseless $x^{t+\delta}$ using an online denoiser and for this purpose we use an unbalanced block denoiser to denoise $z_t$ to $x_t$. If we use a denoising function, such as $f_t(\cdot)$, that makes decision based on the unbalanced context $z_{t-k}^{t+\delta}$ around the received symbol $z_t$, then the instantaneous loss occured by denoising the $t^{\text{th}}$ symbol is $\Lambda\left(x_t, \hat{x}_{f_t}\left(z_{t-k}^{t+\delta}\right)\right)$. The total and normalized average loss incurred by denoising function $f_t(\cdot)$ are defined next.

**Definition 2** *Total cumulative loss (uptil time $t$) incurred by denoising function $f_t(\cdot)$ used to denoise $z^t$ (when the noiseless sequence is $x^t$) is $\mathcal{L}_{f_t}$ and is defined as*

$$\mathcal{L}_{f_t} = \sum_{t'=k+1}^{t} \Lambda\left(x_{t'}, \hat{x}_{f_t}\left(z_{t'-k}^{t'+\delta}\right)\right). \tag{9}$$

**Definition 3** *Relative Average Loss (RAL) incurred by denoising function $f_t(\cdot)$ is $\overline{\mathcal{L}}_{f_t}$ and is defined as follows*

$$\overline{\mathcal{L}}_{f_t} = \frac{\mathcal{L}_{f_t}}{(t-k)\Lambda_{max}}. \tag{10}$$

It is clear from the above definition that the relative average loss (RAL) for any denoising algorithm satisfies $\overline{\mathcal{L}}_{f_t} \leq 1$, for all $t$. The behavior and performance of the online denoiser depends on the particular choice of the denoising function $f_t$. For instance, the denoising function $f_{t'}(z_{t'-k}^{t'+\delta})$ works only based on the available data up to time $t'+\delta$. It should be mentioned that in OOD, at time $t'$, $f_{t'}(z_{t'-k}^{t'+\delta})$ is only used to denoise $z_{t'}$ while in ROD, $f_{t'}(z_{t'-k}^{t'+\delta})$ is also used to re-denoise *all* the previously denoised symbols from $t=k+1$ to $t=t'-1$. It is readily seen that ROD is actually an online version of an unbalanced offline block denoiser that behaves similar to DUDE [4]. In fact, when we want to denoise the symbol at time $t$ and data up to $t+\delta$ is available, ROD denoises the whole data $z^{t+\delta}$. Subsequently, when we receive a new symbol, $z_{t+\delta+1}$, ROD denoises the whole data $z^{t+\delta+1}$. This is similar to using an offline block denoiser repeatedly and hence, similar to [4], it can be shown that ROD asymptotically converge to the optimum Bayes denoiser.

We next focus on the properties of the OOD algorithm. In OOD algorithm, at each time step, we only denoise one symbol. In other words, when we receive $z_{t+\delta}$, OOD denoises $z_t$. The following lemma shows that the asymptotic behavior of OOD is close to unbalance offline block denoiser that uses (8) when all the data, $z^n$ is available.

**Lemma 1** *Let*

$$\varphi(p) = \begin{cases} \frac{1}{1-2p}\log\frac{1-p}{p} & 0 \leq p < \frac{1}{2} \\ \frac{1}{2p(1-p)} & \frac{1}{2} \leq p \leq 1 \end{cases}$$

*Also, define the following vectors at time $t$:*

$$\mathbf{C}_t^{OFF} = \mathbf{C}(z^n, z_{t-k}^{t-1}, z_{t+1}^{t+\delta}), \quad \mathbf{C}_t^{OOD} = \mathbf{C}(z^{t+\delta}, z_{t-k}^{t-1}, z_{t+1}^{t+\delta})$$

*Then, for all $z^n \in \mathcal{A}^n$ we have*

$$\Pr\left(\left\|\frac{\mathbf{C}_t^{OFF}}{n} - \frac{\mathbf{C}_t^{OOD}}{t+\delta}\right\|_1 \geq \epsilon\right) \leq (2^M-2)e^{-\frac{t\epsilon^2}{4}\min\limits_{A\subseteq\mathcal{A}}\varphi(P(A))} \tag{11}$$

*where $P(A) = \sum_{\alpha\in A}\frac{\mathbf{C}_t^{OFF}[\alpha]}{n}$.*

*Proof*: It is shown in [4] (Proposition 1) that for a probability distribution vector $\mathbf{P}$ with length of $M$ and its empirical estimation $\hat{\mathbf{P}}$, that has been estimated using $t$ observations, we have

$$\Pr\left(\left\|\mathbf{P}-\hat{\mathbf{P}}\right\|_1 \geq \epsilon\right) \leq (2^M-2)e^{-\frac{t\epsilon^2}{4}\min_{A\subseteq\mathcal{A}}\varphi(P(A))}$$

The proof of the Lemma follows directly by substituting $\mathbf{P}$ by $\mathbf{C}_t^{OFF}/n$ and $\hat{\mathbf{P}}$ by $\mathbf{C}_t^{OOD}/(t+\delta)$.

The above lemma (i.e., (11)) shows that the empirical pmf (obtained by the counts in the observed vector $z^{t+\delta}$ for a context) of OOD converges in probability to the empirical pmf of the unbalanced offline block denoiser. Similar to (8), let us define the offline unbalanced denoiser (denoted by OFF) as:

$$\hat{X}^{k,\delta}(z^n)[t] = \arg\min_{\hat{x}\in\mathcal{A}} \mathbf{C}^T(z^{t+\delta}, z_{t-k}^{t-1}, z_{t+1}^{t+\delta})\Pi^{-1}\left[\lambda_{\hat{x}}\odot\pi_{z_t}\right]. \tag{12}$$

Lemma 1 shows that when we estimate the empirical distribution of $z_t$ wrapped in the context of $(z_{t-k}^{t-1}\cdot z_{t+1}^{t+\delta})$ using OOD, this distribution asymptotically converges to the empirical distribution of $z_t$ wrapped in the context of $(z_{t-k}^{t-1}\cdot z_{t+1}^{t+\delta})$ using the offline block denoiser. Now we show that convergence property of Lemma 1 results in the convergence of the final estimation of $\hat{x}_t^{OOD}$ to $\hat{x}_t^{OFF}$ where $\hat{x}_t^{OFF}$ is the result of denoising by the offline block denoiser. It is easy to observe that denoising rules of (8) and (12) for OOD and offline denoisers can be written as follows:

$$\hat{x}^{OOD}[t] = \arg\min_{\hat{x}\in\mathcal{A}}\left(\frac{\mathbf{C}_t^{OOD}}{t+\delta}\right)^T\Pi^{-1}\left[\lambda_{\hat{x}}\odot\pi_{z_t}\right], \tag{13}$$

and

$$\hat{x}^{OFF}[t] = \arg\min_{\hat{x}\in\mathcal{A}}\left(\frac{\mathbf{C}_t^{OFF}}{n}\right)^T\Pi^{-1}\left[\lambda_{\hat{x}}\odot\pi_{z_t}\right]. \tag{14}$$

Then, we can write the OOD denoiser output at time $t$ as

$$\begin{aligned}
&\hat{x}^{OOD}[t] \\
&= \arg\min_{\hat{x}\in\mathcal{A}}\left(\frac{\mathbf{C}_t^{OOD}}{t+\delta} + \frac{\mathbf{C}_t^{OFF}}{n} - \frac{\mathbf{C}_t^{OFF}}{n}\right)^T\Pi^{-1}\left[\lambda_{\hat{x}}\odot\pi_{z_t}\right] \\
&= \arg\min_{\hat{x}\in\mathcal{A}}\left(\frac{\mathbf{C}_t^{OFF}}{n}\right)^T\Pi^{-1}\left[\lambda_{\hat{x}}\odot\pi_{z_t}\right] + \\
&\qquad\left(\underbrace{\frac{\mathbf{C}_t^{OOD}}{t+\delta} - \frac{\mathbf{C}_t^{OFF}}{n}}_{\to 0 \text{ in probability}}\right)^T\Pi^{-1}\left[\lambda_{\hat{x}}\odot\pi_{z_t}\right].
\end{aligned}$$

However, we proved in Lemma 1 that $\mathbf{C}_t^{OOD}/(t+\delta)$ asymptotically converges to $\mathbf{C}_t^{OFF}/n$. Hence, as $t$ grows and more data is collected, it is expected that $\hat{x}^{OOD}[t]$ converges to $\hat{x}^{OFF}[t]$. We have showed that OOD asymptotically converges to the offline block denoiser with unbalanced context.

Using an approach similar to [4], it can be shown that the offline block denoiser with unbalanced context that denoises based on (12) asymptotically converges to the optimum Bayes denoiser.

**Proposition 1** *The offline unbalanced block denoiser that denoises based on (12) asymptotically converges to the Bayes optimal denoiser as defined in (4).*

## V. Numerical Results

To investigate the performance of the proposed algorithms, we performed various experiments. In this paper, we present

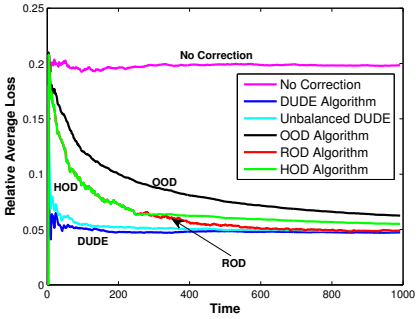


Fig. 1. Relative average loss of BSC example.

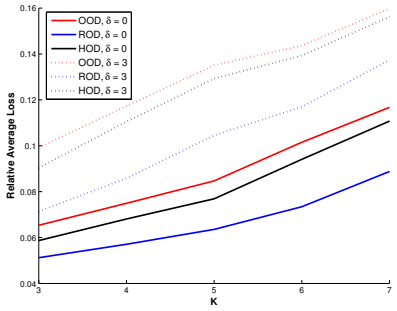


Fig. 2. Effect of $(k, \delta)$ on ratio of errors in BSC example.

some numerical results for the case of binary alphabet, i.e., $\mathcal{A} = \{0, 1\}$, and the noisy mechanism is modeled through a binary symmetric channel (BSC) with crossover probability $\mu$. We run the denoising algorithm for 100 randomly generated texts and report the average RAL over all cases. Each of the random texts contained binary data of length $n = 1000$. Before generating the $i^{\text{th}}$ random text, we first generate a pattern dictionary which contains 12 patterns, $\mathcal{P}_i^{\mathcal{S}} = \{\mathcal{P}_{i,1}, \cdots, \mathcal{P}_{i,12}\}$, where $\mathcal{P}_{i,j}$, for $j = 1$ to 10 are randomly generated patterns with length 10 bits, generated using uniform distribution of 0s and 1s. Also, $\mathcal{P}_{i,11}$ and $\mathcal{P}_{i,12}$ are 0 and 1, respectively. The $i^{\text{th}}$ text is the random sequence of patterns in $\mathcal{P}_i^{\mathcal{S}}$, where each pattern is selected with equal probability. We assumed that the crossover probability of the BSC is $\mu = 0.2$ and the loss matrix is $\Lambda(0, 0) = \Lambda(1, 1) = 0$, and $\Lambda(0, 1) = \Lambda(1, 0) = 1$. Furthermore, we assumed that $k$ is 3 and we show results for $\delta = 0$ and $\delta = 3$.

The RAL of various approaches for $\delta = 0$ are compared in Figure 1. In this figure, the proposed OOD, ROD, and HOD algorithms are compared with DUDE and Unbalanced DUDE, where by unbalanced, we mean that left context of size $k$ and right context of size $\delta$. Relative average loss of the noisy uncorrected sequence is also illustrated in this figure. It can be observed in Fig. 1 that as we proved in the previous section, online algorithms converge to the offline algorithm. It is also observable that ROD (red curve) shows lower RAL than OOD and HOD which shows that ROD convergence is faster than other ones. However, the faster convergence of ROD comes at a higher computational cost since ROD re-denoises all the past symbols upon receiving a new noisy symbol.

The effect of $\delta$ on the RAL of the proposed algorithms is shown in Fig. 2. It should be noted that when $k$ is 3, $\delta = 3$ means that online denoisers use balanced double-sided contexts as is used in DUDE [4]. From Fig. 2 it is obvious that for this specific test case, the average performance of online denoisers is better when $\delta$ is 0 which shows that *balanced denoisers are not necessarily better than unbalanced denoisers*. Results of Fig. 2 confirms that ROD results in better RAL compared to the other online denoisers.

## VI. Conclusions

In this paper, we studied the problem of online discrete denoising which is applicable in various real-time data driven applications. We presented three algorithms for different situations to strike the trade-off between the time-sensitivity and denoising accuracy. We proved that the proposed algorithms asymptotically converge to the optimum offline block denoisers. Furthermore, we provided numerical results for the case of binary data source and BSC channel, which support the theoretical justifications. Future directions include the extension of online algorithms to larger alphabet sizes.

## Acknowledgment

Supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior National Business Center (DoI/NBC) contract number D12PC000337, the US Government is authorized to reproduce and distribute reprints of this work for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoI/NBC, or the US Government.

## References

[1] P. Chakraborty, P. Khadivi et al., "Forecasting a moving target: Ensemble models for ILI case count predictions," in *Proceedings of the SIAM International Conference on Data Mining*, 2014, pp. 262–270.

[2] S. Buadhachain and G. Provan, "A model-based control method for decentralized calibration of wireless sensor networks," in *American Control Conference*, 2013, pp. 6571–6576.

[3] S. Sarkar, X. Jin, and A. Ray, "Data-driven fault detection in aircraft engines with noisy sensor measurements," *Journal of Engineering for Gas Turbines and Power*, vol. 13, August 2011.

[4] T. Weissman et al., "Universal discrete denoising: Known channel," *IEEE Transactions on Information Theory*, vol. 51, no. 1, pp. 5–28, 2005.

[5] G. Motta, E. Ordentlich, and et al., "The idude framework for grayscale image denoising," *IEEE Transactions on Image Processing*, vol. 20, no. 1, pp. 1–21, 2011.

[6] S. Pyatykh and J. Hesser, "Salt and pepper noise removal in binary images using image block prior probabilities," *Journal of Visual Communication and Image Representation*, vol. 25, no. 5, p. 748754, 2014.

[7] A. Buades, B. Coll, and J. M. Morel, "A review of image denoising algorithms, with a new one," *Multiscale Modeling & Simulation*, vol. 4, no. 2, pp. 490–530, 2005.

[8] T. Moon and T. Weissman, "Discrete denoising with shifts," *IEEE Trans. on Information Theory*, vol. 55, no. 11, pp. 5284–5301, 2009.

[9] R. Zhang and T. Weissman, "Discrete denoising for channels with memory," *Communications in Information and Systems*, vol. 5, no. 2, pp. 257–288, 2005.

[10] C. D. Giurcaneanu and B. Yu, "Efficient algorithms for discrete universal denoising for channels with memory," in *Proc. of the IEEE ISIT*, 2005.

[11] G. Gemelos, S. Sigurjonsson, and T. Weissman, "Algorithms for discrete denoising under channel uncertainty," *IEEE Transactions on Signal Processing*, vol. 54, no. 6, p. 22632276, 2006.

[12] T. Weissman et al., "Universal filtering via prediction," *IEEE Transactions on Information Theory*, vol. 53, no. 4, pp. 1253–1264, 2007.