Channel Input Distribution Estimation Using a Minimum I-divergence Algorithm

Kerkil Choi, *Student Member, IEEE*, Aaron D. Lanterman, *Member, IEEE*, and Majid Fozunbal, *Student Member, IEEE*

Abstract

Given a channel with a known transition probability, we consider the problem of finding the input distribution that most closely achieves a desired output distribution. We pose the problem as a linear inverse problem subject to nonnegativity constraints, and employ an iterative algorithm for minimizing Csiszar's *I*-divergence between the desired channel output and the channel output derived from an estimated channel input. We also show how to modify the algorithm to incorporate symmetry constraints on the input distribution. Particular examples involving Rician channels are shown.

Index Terms

Rician channel, linear inverse problem, nonnegativity constraint, edge artifact

I. Introduction

Fozunbal, McLaughlin, and Schafer [4] have recently presented results concerning the capacity of Rician channels. In their exposition, an integral equation is presented relating the input and the output distributions of the Rician channels. Researchers exploring the Rician channel may propose input distributions and find the resulting output distribution. The inverse problem of finding the input distribution that yields, as closely as possible, a desired output distribution is much more difficult. To provide an analytic tool for researchers, we formulate an iterative

Kerkil Choi and Aaron D. Lanterman are with School of Electrical and Computer Engineering, Georgia Institute of Technology, Mail Code 0250, Atlanta, GA 30332 (emails: kerkil@ece.gatech.edu, lanterma@ece.gatech.edu). Majid Fozunbal was with School of Electrical and Computer Engineering, Georgia Institute of Technology. He is currently with HP Research Labs, 1501 Page Mill Road, MS 1181 Palo Alto, CA 94041 (email: majid.fozunbal@hp.com).

algorithm for solving this inverse problem. Although this work was motivated by recent results on Rician channels, our algorithm could be applied to other channels as well.

A. Nonnegative Linear Inverse Problems

Problems involving the reconstruction of an input from a blurred output under a linear blurring function are omnipresent in engineering and science. In particular, inverse problems of linear systems with nonnegative parameters, subject to nonnegativity constraints on the solution, are often of interest. Vardi and Lee [2] showed that deterministic linear inverse problems with nonnegativity constraints can be thought of as statistical estimation problems from incomplete data based on an infinite number of observed samples, which allows us to use the weak law of large numbers. Hence, they showed that maximum likelihood estimation and the EM algorithm provide a direct method of addressing such problems. Snyder *et al.* [3] address the same issue, and conclude that solutions obtained by minimizing Csiszár's I-divergence measure asymptotically correspond to certain maximum likelihood estimators. They also showed that the sequence of estimators from their method has a nice set of properties such as guaranteed convergence to the global minimum, preserved nonnegativity of solutions, and monotonically decreasing I-divergence. Csiszár justified the use of his I-divergence measure by proving that, if all the functions involved are required to be nonnegative, minimizing his measure is the only choice consistent with the axioms he proposed. The algorithm proposed by Snyder *et al.* has been employed in various fields such as medical imaging, astronomical imaging, and image restoration [5], [6], [8].

Because all involved functions (the input, the output, and the kernel) in the problem of interest in this paper are nonnegative, we apply the minimum I-divergence method. For Rician channels, Fozunbal *et al.* [4] have shown that the input distribution should be symmetric; hence, we derive a new algorithm that preserves the symmetry of the solutions. We also show that if the transition kernel is symmetric with respect to the origin (as is the case with Rician channels), then the new symmetry-preserving algorithm and the original algorithm produce the same estimate at each iteration, assuming the algorithms are initialized with the same *symmetric* initial estimate. Hence, although the proposed algorithms do not improve the rate of convergence per iteration or the quality of the solutions in the Rician case, they noticeably improve computation time.

We illustrate the algorithm with two kinds of scenarios. In the first scenario, we test the

algorithm with a known input distribution to verify the accuracy of the algorithms. The second scenario matches how we expect researchers to use the algorithm, in that we give an output distribution to the algorithm and ask it to find the input distribution that gets as close as possible to the desired output, with the understanding that an input distribution that gives the *exact* desired output may not exist.

The estimated inputs may show some artifacts in some situations. These artifacts are also discussed.

B. The Channel Mapping

Let X and Y be random variables that represent the input and the output of a channel, respectively, defined over the entire real line. Consider a discrete-time channel specified by

$$Y = HX + N, \tag{1}$$

where H is a normal random variable with mean $\overline{h}$ and variance σ_h^2, and N is zero-mean additive white Gaussian noise with variance σ_n^2.

Let $F_X(x)$ and $F_Y(y)$ denote the distribution functions of X and Y, respectively. They are related as follows:

$$F_Y(y) = \int_{-\infty}^{y} \int_{-\infty}^{\infty} p(v|x) dF_X(x) dv, \tag{2}$$

where the kernel $p(v|x)$, the channel transition density [4], is specified by

$$p(v|x) = \frac{1}{\sqrt{2\pi(\sigma_h^2 x^2 + \sigma_n^2)}} e^{-\frac{1}{2(\sigma_h^2 x^2 + \sigma_n^2)}(v - \overline{h}x)^2}. \tag{3}$$

This paper considers the case where the kernel $p(v|x)$ and the output distribution $p_Y(y)$ are known. We are concerned with estimating the input distribution $p_X(x)$. To estimate $p_X(x)$, we suggest using the minimum *I*-divergence algorithm proposed in [3].

C. Organization

This paper is structured as follows. In Section II, our application of the minimum *I*-divergence method is described, and the symmetry-preserving algorithms are proposed. Additionally, the equivalence of the estimates from the original algorithm and the proposed algorithms is proved. Simulation results are presented and analyzed in Section III. Our discussion concludes in Section IV.

II. ALGORITHMS

A. Minimum I-divergence Algorithm

Csiszár's *I*-divergence is an information-theoretic discrepancy measure between two nonnegative functions. This measure is a generalization of the Kullback-Leibler distance designed to consider functions whose integrals may not be equal. In [1], Csiszár concludes that if both functions being compared are required to be nonnegative, his *I*-divergence measure is the only discrepancy measure consistent with the axioms he proposes.

The authors of [3] proposed an algorithm for nonnegative linear inverse problems that minimizes this discrepancy measure. The algorithm produces a sequence of estimates with nice properties, such as guaranteed nonnegativity of every estimate in the sequence, monotone convergence to a global minimum, and so on.

The relation between the input and the output distributions in (2) is equivalent to a Fredholm equation of the first kind relating densities $f_Y(y)$ and $f_X(x)$:

$$f_Y(y) = \int_{-\infty}^{\infty} p(y|x) f_X(x) dx. \tag{4}$$

Note that in (4), all functions involved are nonnegative. This motivates for applying the minimum *I*-divergence algorithm. For computer implementation, we assume the random variables X and Y are defined over finite dimensional sets $\mathcal{X} \subset \mathbb{R}$ and $\mathcal{Y} \subset \mathbb{R}$, respectively. Then (4) becomes

$$p_Y(y) = \sum_{x \in \mathcal{X}} p(y|x) p_X(x), \tag{5}$$

where $p_X(x)$ and $p_Y(y)$ are the probability mass functions of X and Y, respectively, and $p(y|x)$ has been similarly discretized. Since the discretization of densities of probability mass functions is just an artifice of computer implementation, and the underlying continuous densities are what we are really interested in, the remainder of this paper will use the shorter term "density" instead of "probability mass function."[1]

Our goal is to find $\hat{p}_X(x)$, an estimate of $p_X(x)$ that minimizes Csiszár's *I*-divergence,

$$\begin{aligned} &I[p_Y(y)||\hat{p}_Y(y)] \\ &= \sum_y p_Y(y) \ln \left[\frac{p_Y(y)}{\hat{p}_Y(y)} \right] - \sum_y [p_Y(y) - \hat{p}_Y(y)], \end{aligned} \tag{6}$$

[1]Mathematically inclined readers that might be bothered by this abuse of termonology may think of these densities as Radon-Nikodym derivatives with respect to a discrete counting measure instead of the usual Lebegue measure.

where $p_Y(y)$ is the given channel output, and the output $\hat{p}_Y(y)$ is generated by a particular estimate $\hat{p}_X(x)$ according to

$$\hat{p}_Y(y) = \sum_{x \in \mathcal{X}} p(y|x)\hat{p}_X(x). \tag{7}$$

Using the Kuhn-Tucker conditions, we can obtain an iterative algorithm [3] that minimizes (6), which is given by

$$\hat{p}_X^{(k+1)}(x) = \frac{\hat{p}_X^{(k)}(x)}{\sum_{y \in \mathcal{Y}} p(y|x)} \sum_{y \in \mathcal{Y}} \left[\frac{p(y|x)p_Y(y)}{\sum_{x' \in \mathcal{X}} p(y|x')\hat{p}_X^{(k)}(x')} \right]. \tag{8}$$

In the particular case of interest here, the summation of $p(y|x)$ in the denominator over all y given a fixed x is 1 because $p(y|x)$ is a probability mass function. However, the term is left in the expression because modifications to it will be made in a subsequent algorithm.

As mentioned before, a sequence produced by this algorithm has a nice set of properties (Section III of [3]). This kind of algorithm has found application in diverse areas. For example, Lucy [5] and Richardson [6] first derived it for image restoration problems in the 1970's using heuristic arguments.

The steps of the algorithm are described as follows:

1) Begin with an input estimate $\hat{p}_X^{(0)}(x)$ that is a valid probability mass function (nonnegative).
2) Divide the known output density by the output density $\hat{p}_Y^{(k)}(y)$ derived by plugging $\hat{p}_X^{(k)}(x)$ into (7). Call this function $U^{(k)}(y)$.
3) Compute the summation over y in the numerator.
   Call this $W^{(k)}(x) = \sum_y p(y|x)U^{(k)}(y)$.
4) Update the estimate of $\hat{p}_X(x)$ by

$$\hat{p}_X^{(k+1)}(x) = \hat{p}_X^{(k)}(x)W^{(k)}(x). \tag{9}$$

5) Repeat steps 2) through 4) until a convergence criterion is met.

B. Symmetry-Preserving Minimum I-divergence Algorithm

1) Symmetry of the input: Theorem 1 of [4] shows that the channels described in Section I have a bijection property: for a given symmetric output distribution, there exists a unique, symmetric input distribution that induces the given output distribution. Considering this property,

we propose a modification of the algorithm given in Section II-A that preserves the symmetry of the estimated input density. The modified algorithm also exploits symmetry for computational efficiency.

Assume that the input density is known to be symmetric with respect to the origin. Given this assumption, we note the following relations:

$$\begin{aligned}
& I\left[p_Y(y)||\hat{p}_Y(y)\right] \\
= \; & I\left[p_Y(y)\Big|\Big|\sum_{x\in\mathcal{X}} p(y|x)\hat{p}_X(x)\right] \\
= \; & I\Bigg[p_Y(y)\Big|\Big|p(y|0)\hat{p}_X(0) \\
& \quad + \sum_{x\in\mathcal{X}^+} p(y|x)\hat{p}_X(x) + \sum_{x\in\mathcal{X}^-} p(y|x)\hat{p}_X(x)\Bigg] \\
= \; & I\Bigg[p_Y(y)\Big|\Big|p(y|0)\hat{p}_X(0) \\
& \quad + \sum_{x\in\mathcal{X}^+} \left\{p(y|x)\hat{p}_X(x) + p(y|-x)\hat{p}_X(-x)\right\}\Bigg] \\
= \; & I\Bigg[p_Y(y)\Big|\Big|p(y|0)\hat{p}_X(0) \\
& \quad + \sum_{x\in\mathcal{X}^+} \left\{p(y|x)\hat{p}_X(x) + p(y|-x)\hat{p}_X(x)\right\}\Bigg] \\
= \; & I\Bigg[p_Y(y)\Big|\Big|p(y|0)\hat{p}_X(0) \\
& \quad + \sum_{x\in\mathcal{X}^+} \left\{p(y|x) + p(y|-x)\right\}\hat{p}_X(x)\Bigg],
\end{aligned} \tag{10}$$

where $\mathcal{X}^+$ and $\mathcal{X}^-$ are defined as

$$\begin{aligned}
\mathcal{X}^+ &= \{x \in \mathcal{X} : x > 0\}, \\
\mathcal{X}^- &= \{x \in \mathcal{X} : x < 0\}.
\end{aligned} \tag{11}$$

The second to last equality holds by the assumption of the symmetry of the input density. Inspired

by the structure of (10), we define a new kernel[2]

$$q(y|x) = \begin{cases} p(y|x) + p(y|-x), & x > 0 \\ p(y|0), & x = 0 \\ 0, & x < 0 \end{cases} . \tag{12}$$

Note that this kernel is still nonnegative, since $q(y|x)$ is defined by adding two elements in $p(y|x)$, which are nonnegative. Hence, we can still apply the (original) minimum *I*-divergence algorithm. With this new kernel, however, we only need to work on the estimate values for $x \in \mathcal{X}^+ \bigcup \{0\}$. Following the same procedure by which the original algorithm in Section II-A is derived, we find a new algorithm:

$$\hat{p}_X^{(k+1)}(x) = \frac{\hat{p}_X^{(k)}(x)}{Q(x)} \sum_{y \in \mathcal{Y}} \left[\frac{q(y|x) p_Y(y)}{\sum_{x' \in \mathcal{X}^+ \bigcup \{0\}} q(y|x') \hat{p}_X^{(k)}(x')} \right], \quad \forall x \in \mathcal{X}^+ \bigcup \{0\}, \tag{13}$$

where

$$\begin{aligned} Q(x) &= \sum_{y \in \mathcal{Y}} q(y|x) \\ &= \begin{cases} \sum_{y \in \mathcal{Y}} p(y|x) + p(y|-x), & x > 0 \\ \sum_{y \in \mathcal{Y}} p(y|0), & x = 0 \end{cases} \\ &= \begin{cases} 2, & x > 0 \\ 1, & x = 0 \end{cases} . \end{aligned} \tag{14}$$

Specifically, this algorithm minimizes Csiszár's *I*-divergence between the known output $p_Y(y)$ and the output $\tilde{p}_Y(y)$ produced by a new system defined as

$$\tilde{p}_Y(y) = \sum_{x \in \mathcal{X}^+ \bigcup \{0\}} q(y|x) \hat{p}(x), \tag{15}$$

[2]Here we deal with the case where the number of input samples is odd. Nevertheless, other researchers should be able to apply our method easily to the case of an even number of samples by following the procedure described in this section.

where the estimate for $x \in \mathcal{X}^-$ is tentatively assumed to be zero. Once the whole estimate for $x \in \mathcal{X}^+ \bigcup \{0\}$ is obtained, the estimate for $x \in \mathcal{X}^-$ is defined as

$$\hat{p}_X^{(k+1)}(x) = \hat{p}_X^{(k+1)}(-x) \qquad \forall x \in \mathcal{X}^-. \tag{16}$$

All the properties and relative theorems [3] still hold for the proposed algorithm associated with the newly defined system.

2) Symmetry of the output: The symmetry of the output allows an additional improvement of computational efficiency. Let

$$\tilde{p}_Y^{(k)}(y) = \sum_{x' \in \mathcal{X}^+ \bigcup \{0\}} q(y|x') \hat{p}_X^{(k)}(x'), \tag{17}$$

and

$$r_Y^{(k)}(y) = \frac{p_Y(y)}{\tilde{p}_Y^{(k)}(y)}. \tag{18}$$

Then, the algorithm (13) becomes

$$\hat{p}_X^{(k+1)}(x) = \frac{\hat{p}_X^{(k)}(x)}{Q(x)} \sum_{y \in \mathcal{Y}} q(y|x) r_Y^{(k)}(y), \qquad \forall x \in \mathcal{X}^+ \bigcup \{0\}. \tag{19}$$

Recall that since the input $p_X(x)$ is symmetric, so are $p_Y(y)$ and $\hat{p}_Y(y)$, and in turn so is $r_Y(y)$, namely $r_Y(y) = r_Y(-y)\ \forall y \in \mathcal{Y}$. Using this symmetry, (19) can be rewritten as

$$\hat{p}_X^{(k+1)}(x) = \frac{\hat{p}_X^{(k)}(x)}{Q(x)} \left[\sum_{y \in \mathcal{Y}^+} \{q(y|x) + q(-y|x)\} r^{(k)}(y) + q(0|x) r^{(k)}(0) \right], \qquad \forall x \in \mathcal{X}^+ \bigcup \{0\}, \tag{20}$$

where $\mathcal{Y}^+$ is defined as $\mathcal{Y}^+ = \{y \in \mathcal{Y} : y > 0\}$. Let

$$s(y|x) = \begin{cases} q(y|x) + q(-y|x), & y > 0 \\ q(0|x), & y = 0 \\ 0, & y < 0 \end{cases}. \tag{21}$$

Also, let $|\mathcal{Y}^+|$ denote the cardinality of $\mathcal{Y}^+$. Then (20) becomes

$$\hat{p}_X^{(k+1)}(x) = \frac{\hat{p}_X^{(k)}(x)}{S(x)} \sum_{y \in \mathcal{Y}^+ \bigcup \{0\}} s(y|x) r^{(k)}(y), \qquad \forall x \in \mathcal{X}^+ \bigcup \{0\}, \tag{22}$$

where

$$
\begin{aligned}
S(x) &= \sum_{y\in\mathcal{Y}^+\bigcup\{0\}} s(y|x) \\
&= \begin{cases} \sum_{y\in\mathcal{Y}^+} s(y|x) + s(0|x), & x>0 \\ \sum_{y\in\mathcal{Y}^+} s(y|0) + s(0|0), & x=0 \end{cases} \\
&= \begin{cases} \sum_{y\in\mathcal{Y}^+} \{q(y|x) + q(-y|x)\} + q(0|x), & x>0 \\ \sum_{y\in\mathcal{Y}^+} \{q(y|0) + q(-y|0)\} + q(0|0), & x=0 \end{cases} \\
&= \begin{cases} \sum_{y\in\mathcal{Y}^+} \{p(y|x) + p(y|-x) + p(-y|x) + p(-y|-x)\} + p(0|x) + p(0|-x), & x>0 \\ \sum_{y\in\mathcal{Y}^+} \{p(y|0) + p(-y|0)\} + p(0|0), & x=0 \end{cases} \\
&= \begin{cases} 2 - p(0|x) - p(0|-x) + p(0|x) + p(0|-x), & x>0 \\ 1 - p(0|0) + p(0|0), & x=0 \end{cases} \\
&= \begin{cases} 2, & x>0 \\ 1, & x=0 \end{cases} .
\end{aligned} \tag{23}
$$

In (22), the number of multiplications in the summation in the numerator decreases to about half the number in (20). Again, the remaining part of the estimate can be obtained by (16).

This new algorithm can be implemented as follows:

1) Begin with a feasible nonnegative, symmetric input estimate $\hat{p}_X^{(0)}(x)$, $\forall x \in \mathcal{X}^+\bigcup\{0\}$.
2) Divide the known output density by the estimated output $\tilde{p}_Y^{(k)}(y)$ to obtain $r_Y^{(k)}(y)$, $\forall y \in \mathcal{Y}^+\bigcup\{0\}$.
3) Compute the term in (22) where the summation over y is calculated.
   Call this $V^{(k)}(x) = \sum_{y\in\mathcal{Y}\bigcup\{0\}} s(y|x) r_Y^{(k)}(y)$.
4) Update the estimate of $\hat{p}_X(x)$ by
$$\hat{p}_X^{(k+1)}(x) = \frac{1}{S(x)} \hat{p}_X^{(k)}(x) V^{(k)}(x). \tag{24}$$
5) Repeat steps 2) through 4) until a convergence criterion is met.
6) Complete the estimate using (16).

C. Equivalence of the Algorithms

1) Symmetry of the Estimates: This section shows that if the transition kernel is centrosymmetric (as in the Rician case), then the the original minimum *I*-divergence algorithm and our symmetry-preserving minimum *I*-divergence algorithm are essentially the same, provided that both the algorithms are initialized with the same symmetric density. In other words, if the initial estimate is symmetric and the kernel has the desired symmetry, the original minimum *I*-divergence algorithm preserves the symmetry of the input density estimates, and the estimates from the two algorithms at each iteration are the same.

We first show that when $\hat{p}_X^{(k)}(x)$ is symmetric and the kernel has the desired properties, then the original minimum *I*-divergence algorithm induces the same symmetry on $\hat{p}_X^{(k+1)}(x)$. If the algorithm is initialized with a symmetric density $\hat{p}_X^{(0)}(x)$ (such as a uniform density), then mathematical induction implies that the original minimum *I*-divergence algorithm preserves the symmetry for all k. First, suppose that $\hat{p}_X^{(k)}(x)$ is symmetric, and suppose the kernel is $p(y|x)$ is symmetric with respect to the origin. This is clearly the case for the Rician kernel:

$$\begin{aligned} p(-y|-x) &= \frac{1}{\sqrt{2\pi(\sigma_h^2 x^2+\sigma_n^2)}} e^{-\frac{1}{2(\sigma_h^2 x^2+\sigma_n^2)}(-y+\bar{h}x)^2} \\ &= \frac{1}{\sqrt{2\pi(\sigma_h^2 x^2+\sigma_n^2)}} e^{-\frac{1}{2(\sigma_h^2 x^2+\sigma_n^2)}(y-\bar{h}x)^2} = p(y|x). \end{aligned} \tag{25}$$

Then, the estimate of the output $\hat{p}_Y^{(k)}(y)$ is symmetric because

$$\begin{aligned} \hat{p}_Y^{(k)}(-y) &= \sum_{x\in\mathcal{X}} p(-y|x)\hat{p}_X^{(k)}(x) \\ &= \sum_{x\in\mathcal{X}} p(y|-x)\hat{p}_X^{(k)}(x) \\ &= \sum_{x'\in\mathcal{X}} p(y|x')\hat{p}_X^{(k)}(-x') \\ &= \sum_{x'\in\mathcal{X}} p(y|x')\hat{p}_X^{(k)}(x') = \hat{p}_Y^{(k)}(y), \end{aligned} \tag{26}$$

where $x' = -x$. The second equality holds by the symmetry of the kernel in (25), and the fourth equality holds by the symmetry of the estimate of the input $\hat{p}_X^{(k)}(x)$. Furthermore, in (9), $W^{(k)}$

is symmetric since

$$\begin{aligned} W^{(k)}(-x) &= \sum_{y\in\mathcal{Y}} p(y|-x)\frac{p_Y(y)}{\hat{p}_Y^{(k)}(y)} \\ &= \sum_{y\in\mathcal{Y}} p(-y|x)\frac{p_Y(y)}{\hat{p}_Y^{(k)}(y)} \\ &= \sum_{y'\in\mathcal{Y}} p(y'|x)\frac{p_Y(-y')}{\hat{p}_Y^{(k)}(-y')} \\ &= \sum_{y'\in\mathcal{Y}} p(y'|x)\frac{p_Y(y')}{\hat{p}_Y^{(k)}(y')} = W^{(k)}(x), \end{aligned} \tag{27}$$

where $y' = -y$. The second equality holds by the symmetry of the kernel in (25), and the fourth equality holds by the symmetry, proven in (26), of the output and the derived output by $\hat{p}_X(x)$. Consequently, by the assumption that $\hat{p}_X^{(k)}(x)$ is symmetric, the $(k+1)^{st}$ estimate of the input given by (9) is symmetric:

$$\begin{aligned} \hat{p}_X^{(k+1)}(-x) &= \hat{p}_X^{(k)}(-x)W^{(k)}(-x) \\ &= \hat{p}_X^{(k)}(x)W^{(k)}(x) = \hat{p}_X^{(k+1)}(x). \end{aligned} \tag{28}$$

2) Equivalence of the iterations: Next, we show that the estimates from the two algorithms at each iteration are the same. Note that the algorithms in (19) and (22) are the same since the algorithm in (19) has been modified while the mathematical equivalence of each step in the modifications is maintained. Hence, it is sufficient to show the equivalence of the algorithms in (8) and (19). Since the kernel $p(y|x)$ is centrosymmetric (as seen in Fig. 4), the new kernel $q(y|x)$ possesses one-dimensional symmetry with respect to the x-axis:

$$\begin{aligned} q(y|x) &= p(y|x) + p(y|-x) \\ &= p(-y|-x) + p(-y|x) = q(-y|x). \end{aligned} \tag{29}$$

Then, the algorithm in (19) can be rewritten as follows:

$$\hat{p}^{(k+1)}(x) = \begin{cases} \dfrac{\hat{p}^{(k)}(x)}{2}\displaystyle\sum_{y\in\mathcal{Y}} q(y|x)r^{(k)}(y), & x > 0 \\ \hat{p}^{(k)}(0)\displaystyle\sum_{y\in\mathcal{Y}} q(y|0)r^{(k)}(y), & x = 0 \end{cases}$$

$$= \begin{cases} \dfrac{\hat{p}^{(k)}(x)}{2}\left[\displaystyle\sum_{y\in\mathcal{Y}^+} q(y|x)r^{(k)}(y) + \sum_{y\in\mathcal{Y}^-} q(y|x)r^{(k)}(y) + q(0|x)r^{(k)}(0)\right], & x>0 \\ \hat{p}^{(k)}(0)\displaystyle\sum_{y\in\mathcal{Y}} p(y|0)r^{(k)}(y), & x=0 \end{cases}. \tag{30}$$

In (30), when $x=0$, it is clear that if the algorithms (8) and (19) are initialized with the same symmetric density, they produce the same estimates at each iteration, because

$$\begin{aligned} \tilde{p}_Y^{(k)}(y) &= \sum_{x'\in\mathcal{X}^+\bigcup\{0\}} q(y|x')\hat{p}_X^{(k)}(x') \\ &= \sum_{x'\in\mathcal{X}^+} \{p(y|x') + p(y|-x')\}\hat{p}_X^{(k)}(x') + p(y|0)\hat{p}_X^{(k)}(0) \\ &= \sum_{x'\in\mathcal{X}^+} p(y|x')\hat{p}^{(k)}(x') + \sum_{x'\in\mathcal{X}^+} p(y|-x')\hat{p}^{(k)}(-x') + p(y|0)\hat{p}_X^{(k)}(0) \\ &= \sum_{x'\in\mathcal{X}^+} p(y|x')\hat{p}^{(k)}(x') + \sum_{x''\in\mathcal{X}^-} p(y|x'')\hat{p}^{(k)}(x'') + p(y|0)\hat{p}_X^{(k)}(0) \\ &= \sum_{x\in\mathcal{X}} p(y|x)\hat{p}_X^{(k)}(x) = \hat{p}_Y^{(k)}(y), \end{aligned} \tag{31}$$

where $x''=-x'$, and hence it follows that

$$r_Y^{(k)}(y) = \frac{p_Y(y)}{\tilde{p}_Y^{(k)}(y)} = \frac{p_Y(y)}{\hat{p}_Y^{(k)}(y)}. \tag{32}$$

The second equality holds by the definition of $q(y|x)$, and the third equality holds by the symmetry of $\hat{p}^{(k)}(x)$. When x is positive, the algorithm in (30) can be rewritten as follows:

$$\begin{aligned} \hat{p}^{(k+1)}(x) &= \frac{\hat{p}^{(k)}(x)}{2}\left[\sum_{y\in\mathcal{Y}^+} q(y|x)r^{(k)}(y) + \sum_{y\in\mathcal{Y}^-} q(y|x)r^{(k)}(y) + q(0|x)r^{(k)}(0)\right] \\ &= \frac{\hat{p}^{(k)}(x)}{2}\left[\sum_{y\in\mathcal{Y}^+} q(y|x)r^{(k)}(y) + \sum_{y'\in\mathcal{Y}^+} q(y'|x)r^{(k)}(y') + q(0|x)r^{(k)}(0)\right] \\ &= \frac{\hat{p}^{(k)}(x)}{2}\left[2\sum_{y\in\mathcal{Y}^+} \{p(y|x) + p(y|-x)\}r^{(k)}(y) + \{p(0|x) + p(0|-x)\}r^{(k)}(0)\right] \\ &= \hat{p}^{(k)}(x)\left[\sum_{y\in\mathcal{Y}^+} \{p(y|x) + p(-y|x)\}r^{(k)}(y) + p(0|x)r^{(k)}(0)\right] \end{aligned}$$

$$= \hat{p}^{(k)}(x) \left[\sum_{y \in \mathcal{Y}^+} p(y|x) r^{(k)}(y) + \sum_{y' \in \mathcal{Y}^-} p(y'|x) r^{(k)}(y') + p(0|x) r^{(k)}(0) \right]$$

$$= \hat{p}^{(k)}(x) \sum_{y \in \mathcal{Y}} p(y|x) r^{(k)}(y), \tag{33}$$

where $y' = -y$. The second equality holds by the symmetries of $q(y|x)$ and $r^{(k)}(y)$, the fourth equality holds because $p(0|x) = p(0|-x)$, and the second to last equality holds by the symmetry of $r^{(k)}(y)$. Note that, using the relation in (32), the last term in (33) is the exactly same as the algorithm in (8). Combining this finding with the preservation of the symmetry of the estimates from the original minimum *I*-divergence algorithm, we conclude that the original algorithm and the modified algorithms produce the same estimate at each iteration if they are initialized with the same symmetric function.

It should be noted that the equivalence of the original algorithm to the modified algorithms holds only under certain conditions and the special structure of the current application such as the symmetry of the kernel. Our new algorithms (19) and (22), however, can preserve the symmetry of the input even if the conditions for the equivalence (i.e. such as the symmetry of the kernel) do not hold. The estimates from the original algorithm and the proposed symmetry-preserving algorithms are not necessarily the same in general.

III. SIMULATIONS

This section presents numerical results. In our simulation study, we adhere to the estimation of symmetric input densities because the authors of [4] have concluded that the input density for a Rician channel should be symmetric. To help develop a feel for the nature of the Rician kernel, we plot it for various sets of parameters. Among the parameter sets, two interesting cases are selected to compare the behaviors of the sequence of the estimates. Since we have proven that the original minimum *I*-divergence algorithm and the symmetry-preserving minimum *I*-divergence algorithm for these kernels are equivalent on the initial conditions, we only show the results from the symmetry-preserving minimum *I*-divergence algorithm. For the comparison of the computational efficiency, statistics of the execution times of the two algorithms are given.

In the first set of studies presented in this section, a known input distribution is reconstructed to test the algorithm. Of course, in practice, a researcher will posit some output distribution and seek an associated input distribution. It would be interesting to see what input distribution

estimate our algorithm reports *when there is no input distribution that exactly gives the posited output distribution.* An interesting rectangular output is suggested, and an input density that generates a *closest* output is estimated and discussed.

Iterative reconstruction algorithms often suffer from various artifacts. As discussed in [9], these include *noise artifacts* and *edge artifacts,* which have been observed in emission tomography [8]. We are assuming that the "data" for the algorithm, namely the output distribution, are being proposed by a researcher, and hence noise is not a problem in the usual sense. "Edge artifacts" may arise when the kernel attenuates high frequency contents to the point that they cannot be reconstructed due to finite machine precision. Our Rician kernel certainly has this property; hence, we report how *edge artifacts* may be manifested by our algorithms.

A. Investigation of the Kernel

Figure 1 shows a symmetric input density. In this symmetric density, the $x \geq 0$ part is made by summing two differently weighted and shifted Weibull pdfs whose *characteristic life* and *shape* parameters are 2 and 5, respectively. Then, the $x < 0$ part is determined by symmetry. In implementation, the intermediate vector variables such as $W^{(k)}(x)$ in (9) and $V^{(k)}(x)$ in (24) are set to be longer than the input to avoid loss of information due to the support spreading of the data by the convolution-like operations in the algorithm. We must use the term "like," since the blurring is not technically a convolution since it is not space-invariant. In Fig. 1, only part of the x-axis is shown to best show details of the function.

Figure 2 shows a contour plot of the transition kernel parameterized with $\sigma_h = 0.5$, $\sigma_n = 0.6$, and $\overline{h} = 4$. Because the shape of the transition area of the kernel can be grasped better with the contour plot than with a three-dimensional mesh plot, just the contour plot is displayed. Figure 3 shows an output induced by the transition kernel in Fig. 2 given the input in Fig. 1. Note that the output is symmetric as mentioned in [4]. The output is somewhat blurred; it loses some important features of the input. To best show details, only a portion of the x-axis is shown (as in Fig. 3).

A broader transition kernel induces a blurrier output density. A blurrier output may cause "slower" convergence of the estimates than a less blurry output. Therefore, the behavior of convergence of the estimates is highly dependent on the shape of the kernel. The shape of the kernel is configured by the parameters, σ_h, σ_n, and $\overline{h}$. Hence, we show the effects of the

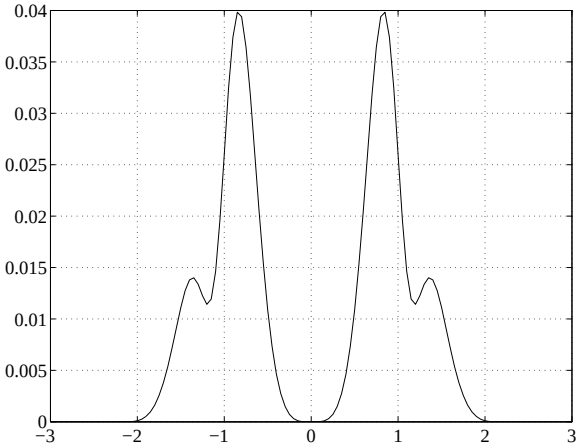


Fig. 1. Symmetric channel input density .

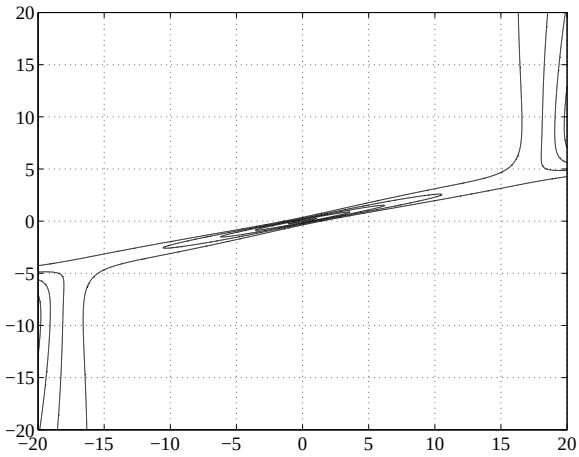


Fig. 2. Contour plot of the kernel parameterized with $\sigma_h = 0.5$, $\sigma_n = 0.6$, and $\overline{h} = 4$. The horizontal axis is associated with y (the column of a transition matrix), and the vertical axis is associated with x (the row of a transition matrix)

parameters on the shape of the kernel by showing changes of the shape induced by changes of the parameters. In our description, we regard the kernel in Fig. 2 as a standard. The effects of each parameter on the shape of the kernel are illustrated by setting the value of the parameter larger or smaller than the value of the same parameter used in Fig. 2 while leaving the other two parameters fixed. Figures 4(a) and 4(b), 4(c) and 4(d), and 4(e) and 4(f) show the kernels resulting from larger and smaller values of σ_h, σ_n, and $\overline{h}$, respectively. For instance, Figs. 4(a) and 4(b) show kernels parameterized with $\sigma_h = 0.1$ and $\sigma_h = 0.9$, respectively, while the other

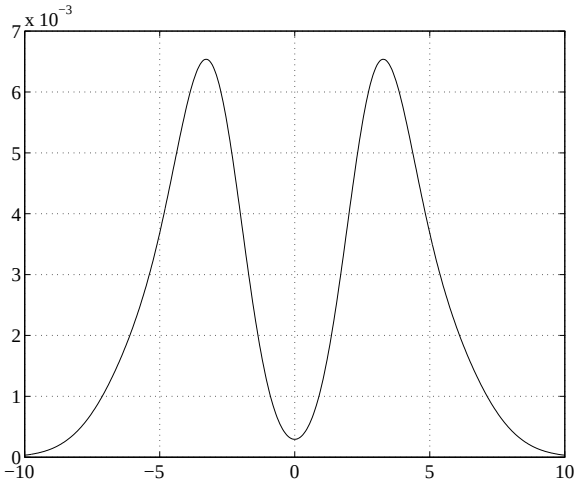


Fig. 3. Output induced by the transition kernel in Fig. 2 given the input in Fig. 1.

parameters $\sigma_n = 0.6$ and $\overline{h} = 4$ remain the same for both.

Compare Fig. 2 with Fig. 4, and Fig. 3 with Fig. 5. It may be reasonable to conclude that the effects of σ_h and σ_n on the shape of the kernel, hence the outputs, are not significant. More specifically, the outputs in Figs. 5(a) and 5(b) do not show significant differences to the output in Fig. 3, although the kernels in 4(a), and 4(b) look somewhat different. Note that the transition areas near the center have similar sharpness. On the other hand, the kernels in Figs. 4(e) and 4(f) and their associated outputs shown in 5(c) are considerably different. The output associated with the broadest kernel (the kernel in Fig. 4(e)) is distinctively blurred, and the features of the input is completely destroyed. On the contrary, the output associated with the sharpest kernel (the kernel in Fig. 4(f)) is much less blurred and retain most of the features of the input, although it seems to lose the finest details, such as the two small side lobes (See Case II in Fig. 5(c)). Since the behavior of the estimates associated with the kernels in 4(a) through 4(d) are similar to the behavior of the estimates associated with the kernel in 4(f), we only show the estimates associated with the kernels used to get the outputs in 5(c).

1) Estimation Results for Known Channel Inputs: Figure 6 shows some estimates of the input density provided by the symmetry-preserving minimum *I*-divergence algorithm in (19). Recall that the estimates from the original minimum *I*-divergence algorithm and the symmetry-preserving minimum *I*-divergence algorithms are essentially the same, and hence we only show the estimates from the symmetry-preserving minimum *I*-divergence algorithm. The algorithms

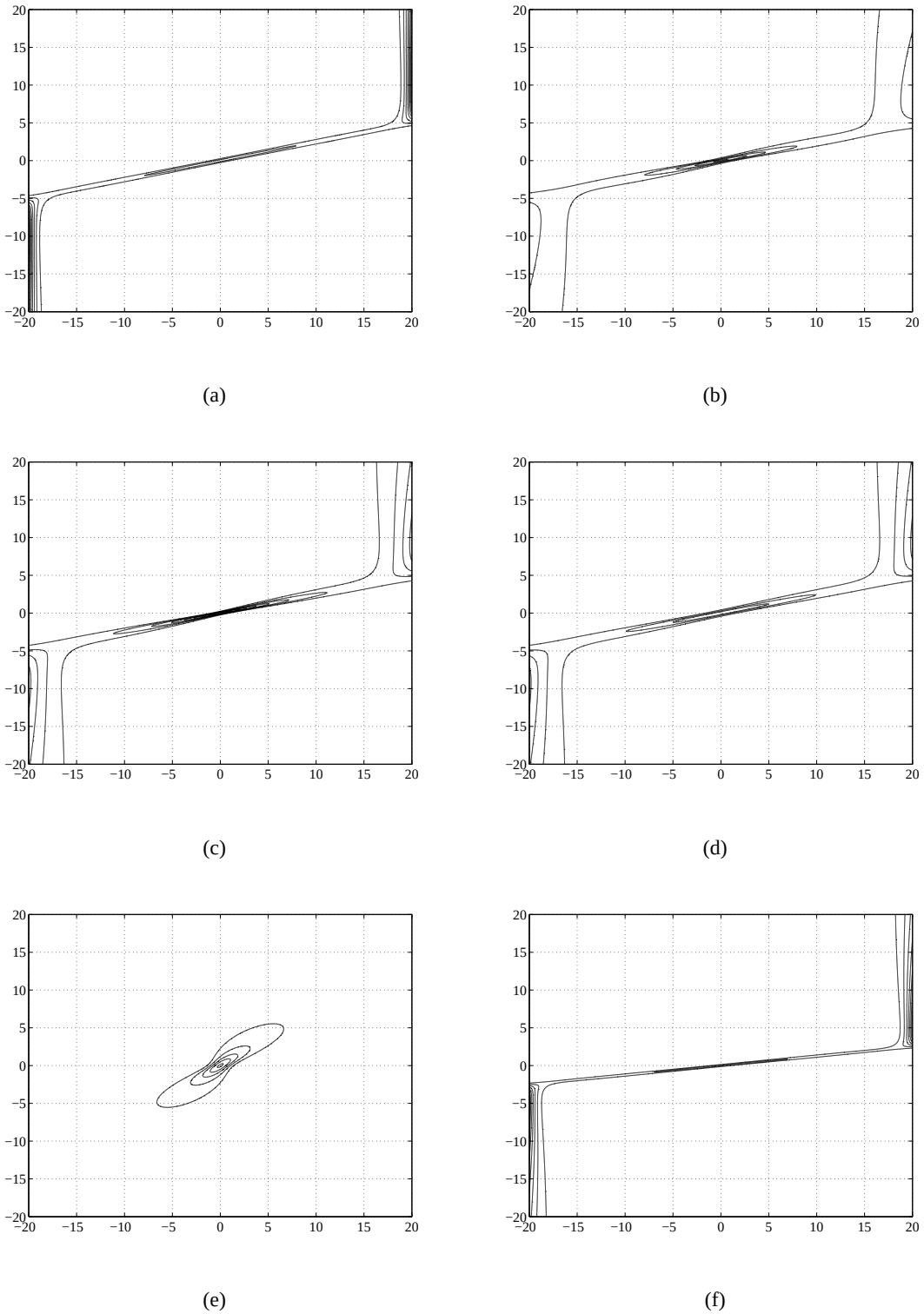
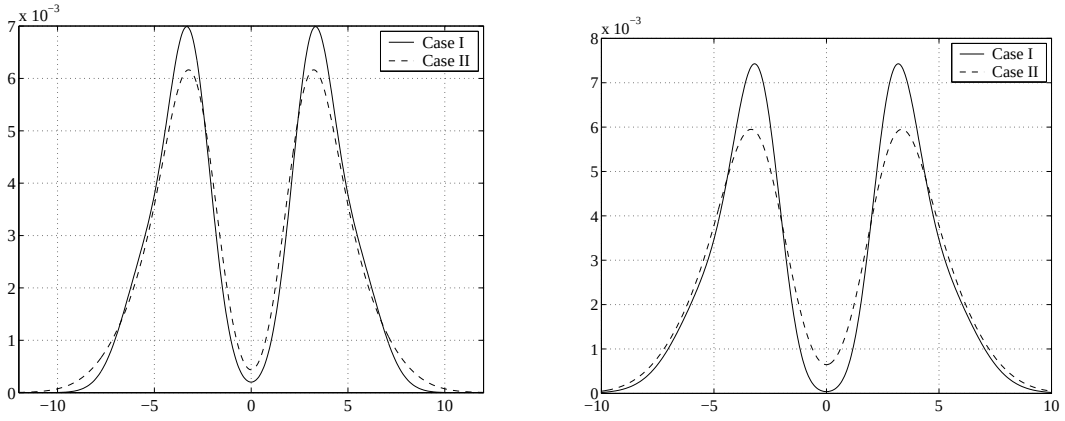
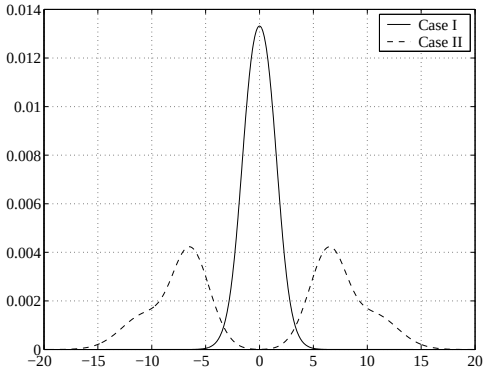


Fig. 4. Contour plots of transition kernels for various choices of parameters: (a) $\sigma_h = 0.1$, $\sigma_n = 0.6$, and $\overline{h} = 4$. (b) $\sigma_h = 0.9$, $\sigma_n = 0.6$, and $\overline{h} = 4$. (c) $\sigma_h = 0.5$, $\sigma_n = 0.2$, and $\overline{h} = 4$. (d) $\sigma_h = 0.5$, $\sigma_n = 1.0$, and $\overline{h} = 4$. (e) $\sigma_h = 0.5$, $\sigma_n = 0.6$, and $\overline{h} = 1$. (f) $\sigma_h = 0.5$, $\sigma_n = 0.6$, and $\overline{h} = 8$.



(a) (b)



(c)

Fig. 5. Channel outputs induced by the kernels given in Fig. 4 when the channel input in Fig. 1 is used: (a) Case I: $\sigma_h = 0.1$, $\sigma_n = 0.6$, and $\overline{h} = 4$; and Case II: $\sigma_h = 0.9$, $\sigma_n = 0.6$, and $\overline{h} = 4$. (b) Case I: $\sigma_h = 0.5$, $\sigma_n = 0.2$, and $\overline{h} = 4$; and Case II: $\sigma_h = 0.5$, $\sigma_n = 1.0$, and $\overline{h} = 4$. (c) Case I:$\sigma_h = 0.5$, $\sigma_n = 0.6$, and $\overline{h} = 1$; and Case II: $\sigma_h = 0.5$, $\sigma_n = 0.6$, and $\overline{h} = 8$.

are initialized with the same uniform density. To show the behavior of the estimates, the figures of the estimates are shown for some selected iterations at which big transitions in the forms of the estimates appear. For example, the estimate at the 5^{th} iteration starts showing the two main lobes, and the estimate at the 20000^{th} starts showing the two small side lobes. Convergence of the estimates associated with the sharpest kernel and convergence of the estimates associated with the broadest kernel are in striking contrast with each other, as shown in Figs. 6(a) and 6(b), and 6(c) and 6(d). The estimates resulting from the least blurry output converge after 100 iterations,

but the estimates from the blurriest output have not converged even after 20000 iterations. An important property of the original and the proposed algorithms is that of guaranteed convergence to the global optimum [3]. Based on this property and the observation that the algorithms take about 1800 iterations until the estimates show signs of the two small side lobes, it can be inferred that the estimates associated with the broadest kernel will converge after an enormous number of iterations.

All the experiments were performed with Matlab 6.5 by The Mathworks. The symmetry-preserving algorithm remarkably improves computation time. With the broadest kernel, the original minimum *I*-divergence algorithm takes 2808 seconds to reach the 20000^{th} iteration, but the symmetry-preserving minimum *I*-divergence algorithm given by (22) takes only 814 seconds.

B. Estimation Results for Arbitrary Specified Outputs

The previous section tested the algorithm by assuming a particular input, generating the outputs using the given kernel, and then watching the algorithm reconstruct the input. In practice, of course, a researcher will suggest an output, and an input that generates the exact output may not exist. In such cases, we aim at estimating an input that generates the *closest* output to the desired output in Csiszár's *I*-divergence sense. To demonstrate such an example, an ideal rectangle is proposed as the output distribution, and the resulting input estimates are shown and discussed.

Figure 7 shows a desired rectangular output. Note that there is no channel input density that could induce this channel output because of the sharpness of the edges. Figure 8 shows the input estimate resulting from 20000 iterations of the algorithm. The input estimate is extremely spiky; interestingly, this is consistent with the guess made in [4]. Figure 9 shows the output density induced by the estimated input density. This estimate is the *closest* achievable output, in the sense that the *I*-divergence measure between the resulting output and the desired output is the smallest possible given the nonnegativity constraint on the input. As expected, the desired sharp edges cannot be obtained. Interestingly, the overshoot and ringing observed in the resulting *output* density are reminicent of the edge artifacts seen in the *input* estimates discussed in the following section.

C. The Edge Artifact

1) Background on the Edge Artifact: The problem of input distribution estimation may be thought of as a classic linear inverse problem. For discussion, let $\{\hat{p}(x) : x \in \mathcal{X}\}$ denote the function to be estimated, and $\{\mu(y) : y \in \mathcal{Y}\}$ denote the function observed or measured. Then the linear inverse problem can be represented as

$$\mu(y) = \int_{\mathcal{X}} p(y|x)\hat{p}(x)dx, \tag{34}$$

where $p(y|x)$ is the transition kernel. These types of problems are notoriously unstable; small variations in $\mu(x)$ may cause relatively large variations in estimates of $\hat{p}(x)$. Such behavior may particularly affect the high frequency components in the estimates, and cause sharp transitions in the function being estimated to exhibit overshoot and ringing. These artifacts are called the *edge artifact* [9] since high frequency components are usually distributed along edges. The *edge artifact* is a manifestation of Gibbs' phenomenon.

We will demonstrate how such artifacts appear in estimates by reconstructing a uniform function, which is often used for demonstrating these artifacts.

2) Demonstration of Edge Artifacts: Figure 10 shows an image of a uniform channel input density, and Figs. 11(a) and 11(b) show the outputs induced by the transition kernels in Figs. 4(e) and 4(f), respectively. Note that the outputs are symmetric. The *edge artifacts* are demonstrated in Fig. 12. The algorithm is initialized with a uniform density.

Figures 12(a) and 12(b), and 12(c) and 12(d) show the estimates of the uniform inputs given the outputs induced by the broadest kernel and the sharpest kernel, respectively. Recall that the estimates for the broadest kernel converge much slower than the estimates for the sharpest kernel. The estimates in Figs. 12(a) and 12(b) show the ringing artifacts first, and then the overshoots, which are gradually increasing. In contrast, the estimates in Fig. 12(c) and 12(d) show the relatively large overshoots first, and then start showing the ringings at later iterations.

IV. Conclusion

We proposed a minimum *I*-divergence algorithm, along with new symmetry-preserving modifications, for estimating an input density given a kernel of a channel of interest and a target output density induced by the kernel. The rates of convergence depend on the shape of the kernel. The original algorithm and our proposed modifications are essentially equivalent at each iteration if

the kernel and the initial estimate possess certain symmetries. The proposed symmetry-preserving algorithms provide considerable improvement in computation time.

Our simulation study has shown that the algorithms, both the original and its modifications, produce desirable results. The appearance of *edge artifacts* in our experiments was discussed, and regularization methods were breifly mentioned; their exploration remains an avenue for future work. An experiment of practical interest was performed in which an ideal output was proposed for which there exists no corresponding input. As expected, the algorithms found the input that results in an output as close as possible to the desired output.

The minimum *I*-divergence method has found use in a large number of applications. We have introduced another application of the method. We hope this research will stimulate interest in finding other applications of the minimum *I*-divergence approach in the communications literature.

ACKNOWLEDGMENT

This work was supported by startup funds from the School of Electrical and Computer Engineering at the Georgia Institute of Technology and by the Demetrius T. Paris Junior Professorship.

REFERENCES

[1] I. Csiszár. “Why least squares and maximum entropy? - an axiomatic approach to inverse problems,” *Ann. Stat.*, vol. 19, pp. 2033-2066, Dec. 1991.

[2] Y. Vardi and D. Lee, “From Image Deblurring to Optimal Investments: Maximum Likelihood Solutions for Positive Linear Inverse Problems,” *J. R. Statist. Soc. B*, vol. 55, pp. 569–612, Nov. 1993.

[3] Donald L. Snyder, Timothy J. Schulz, and Joseph A. O’Sullivan, “Deblurring Subject to Nonnegativity Constraints,” *IEEE Trans. Signal Processing*, vol. 40, pp. 1143-1150, May 1992.

[4] M. Fozunbal, S. W. McLaughlin, R. W. Schafer, “A bijection property of flat-fading channels and its application to the capacity problem,” submitted to *IEEE Trans. Inform. Theory*, July 2003.

[5] L. B. Lucy, “An iterative technique for the rectification of observed distributions,” *Astronom. J.*, vol. 79, pp. 745-754, June 1974.

[6] W. H. Richardson, “Bayesian-based iterative method of image restoration,” *J. Opt. Soc. Amer.*, vol. 62, pp. 55-59, Jan. 1972.

[7] D. L. Snyder and D. G. Politte, “Image Reconstruction from list-mode data in an emission tomography system having time-of-flight measurements,” *IEEE Trans. Nuclear Sci.*, vol. NS-30, pp. 1843-1849, 1983.

[8] D. L. Snyder, M. I. Miller, L. J. Thomas, and D. G. Politte, “Noise and edge artifacts in maximum-likelihood reconstructions for emission tomography ,” *IEEE Trans. Med. Imaging*, vol. 6, no. 3 pp. 228-238, Sept. 1987.

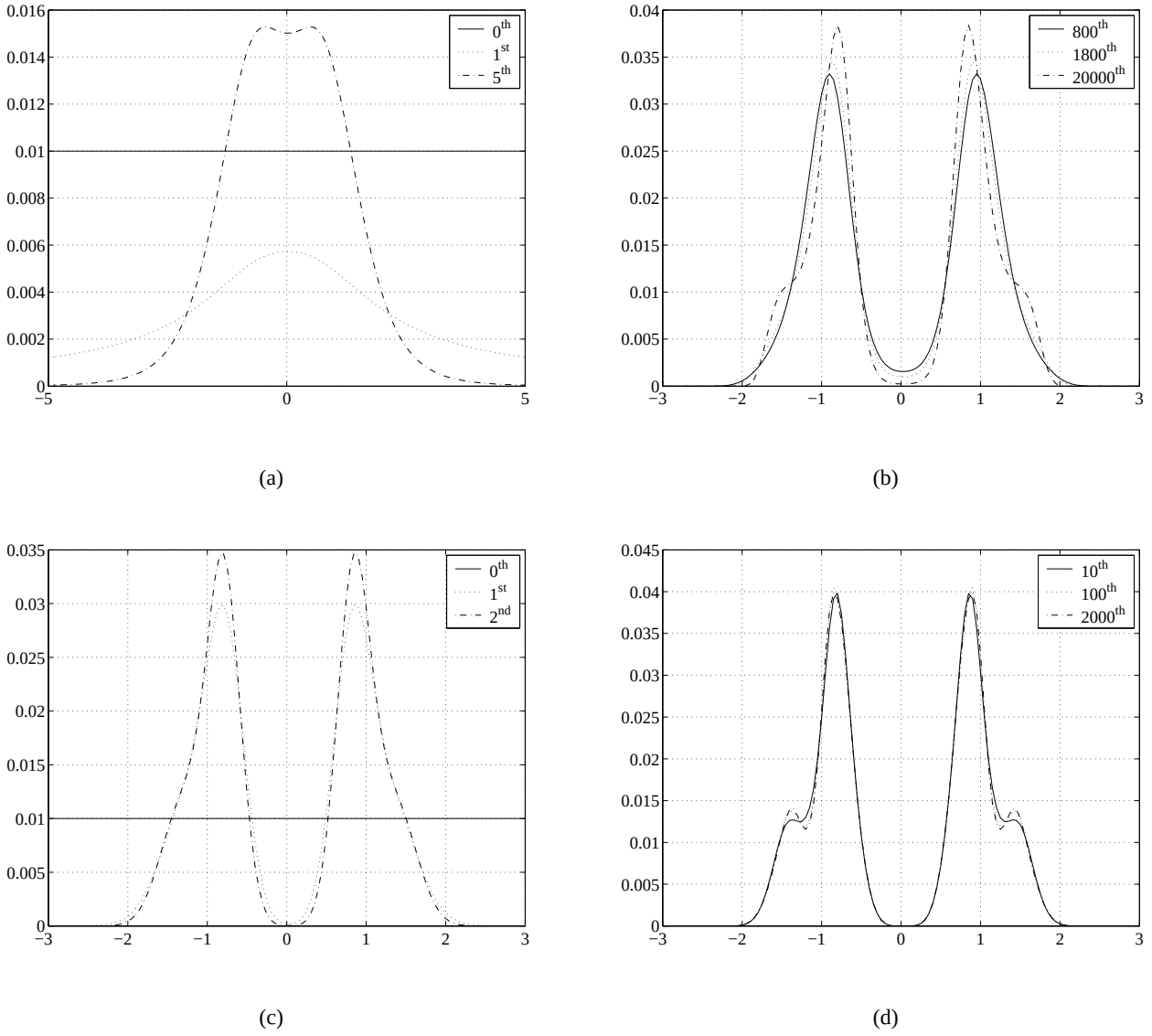
[9] D. L. Snyder and M. I. Miller, *Random Point Processes in Time and Space* (Springer-Verlag, New York, 1991).

[10] U. Grenander, *Abstract Inference* (Wiley, New York, 1981).

[11] I. J. Good and R. A. Gaskins, “Nonparametric Roughness Penalties for Probability Densities,” *Biometrica*, vol. 58, pp. 255-277, 1971.

[12] B. Roysam, J. A. Shrauner, and M. I. Miller, “Bayesian Imaging Using Good’s Roughness Measure - Implementation on a Massively Parallel Processer,” IEEE ICASSP-88, vol. M4.21, pp. 932-935, Mar. 1988.

[13] L. S. Joyce and W. L. Root, “Precision Bounds in Superresolution Processing,” *Journal of the Optical Society of America*, vol. 1, pp. 149-168, Feb. 1988.



(a) (b)

(c) (d)

Fig. 6. Input densities estimated by the symmetry-preserving minimum *I*-divergence algorithm: (a) Estimates at some selected early iterations when $\sigma_h = 0.5$, $\sigma_n = 0.6$, and $\overline{h} = 1$. (b) Estimate at some selected late iterations when $\sigma_h = 0.5$, $\sigma_n = 0.6$, and $\overline{h} = 1$. (c) Estimates at some selected early iterations when $\sigma_h = 0.5$, $\sigma_n = 0.6$, and $\overline{h} = 8$. (d) Estimates at some selected late iterations when $\sigma_h = 0.5$, $\sigma_n = 0.6$, and $\overline{h} = 8$.

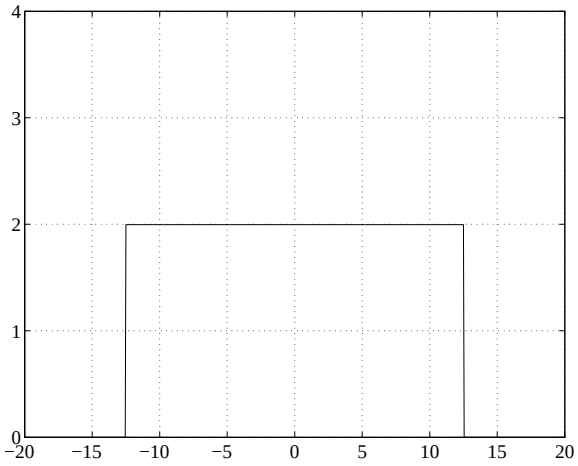
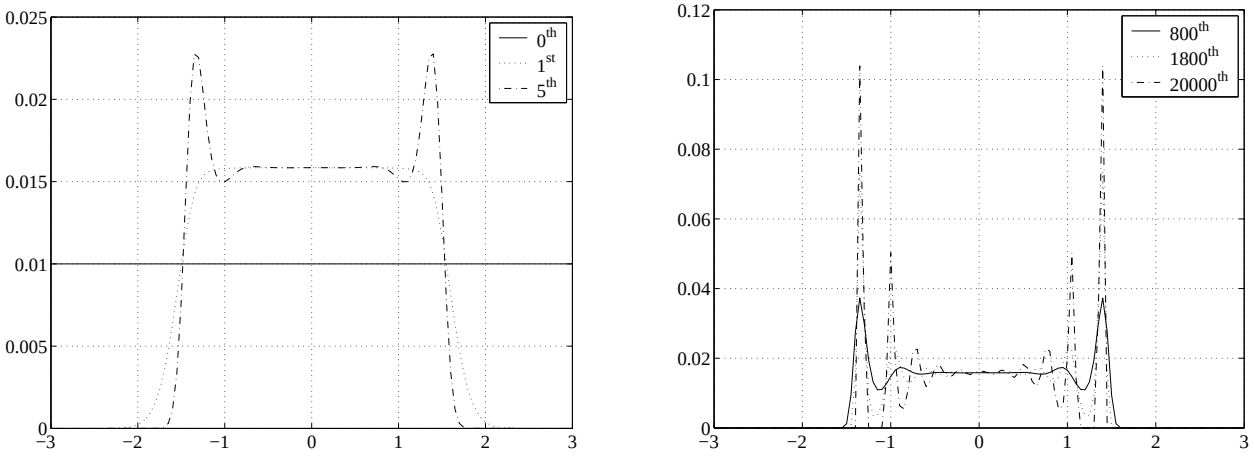


Fig. 7. Ideal rectangle output density.



(a) (b)

Fig. 8. Estimates of an input density generating the estimated output shown in Fig. 9. Early iterations are shown in (a), while later iterations are in (b).

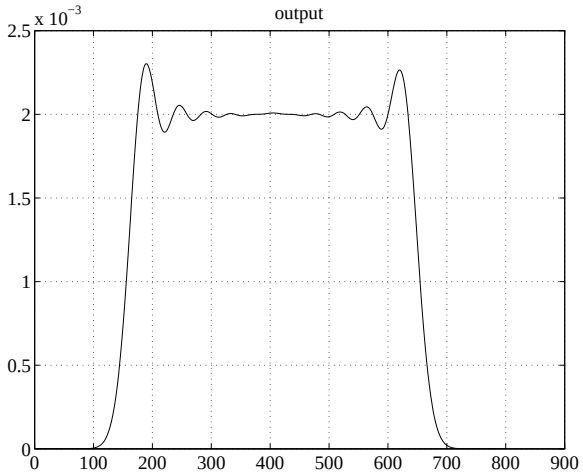


Fig. 9. (a) Induced output closest to the output in Fig. 7 given the kernel in Fig. 4(f) is known.

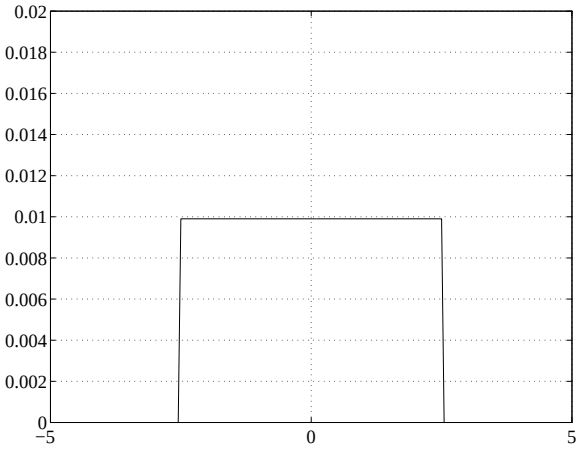
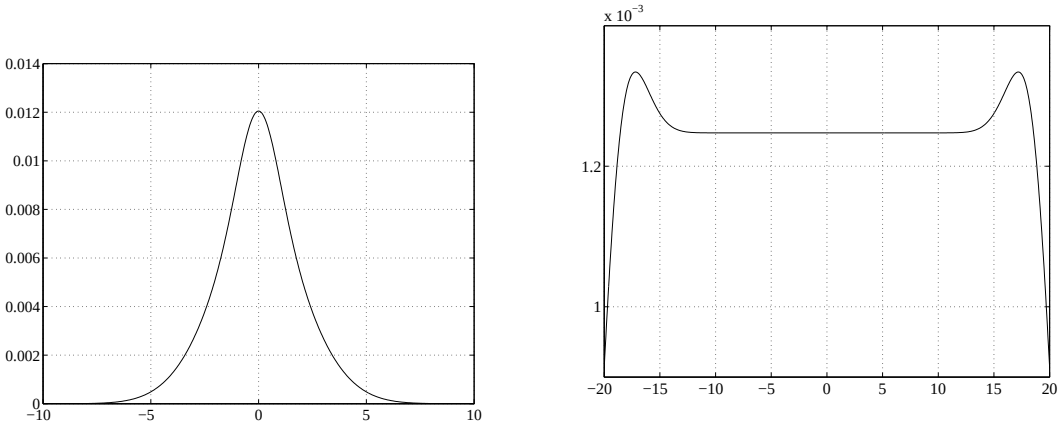


Fig. 10. Symmetric uniform input density for demonstration of the *edge artifact*.



(a) (b)

Fig. 11. (a) Output corresponding to the input density in Fig. 10 when $\sigma_h = 0.5$, $\sigma_n = 0.6$, and $\overline{h} = 1$ are used. (b) Output for the input density in Fig. 10 when $\sigma_h = 0.5$, $\sigma_n = 0.6$, and $\overline{h} = 8$ are used.

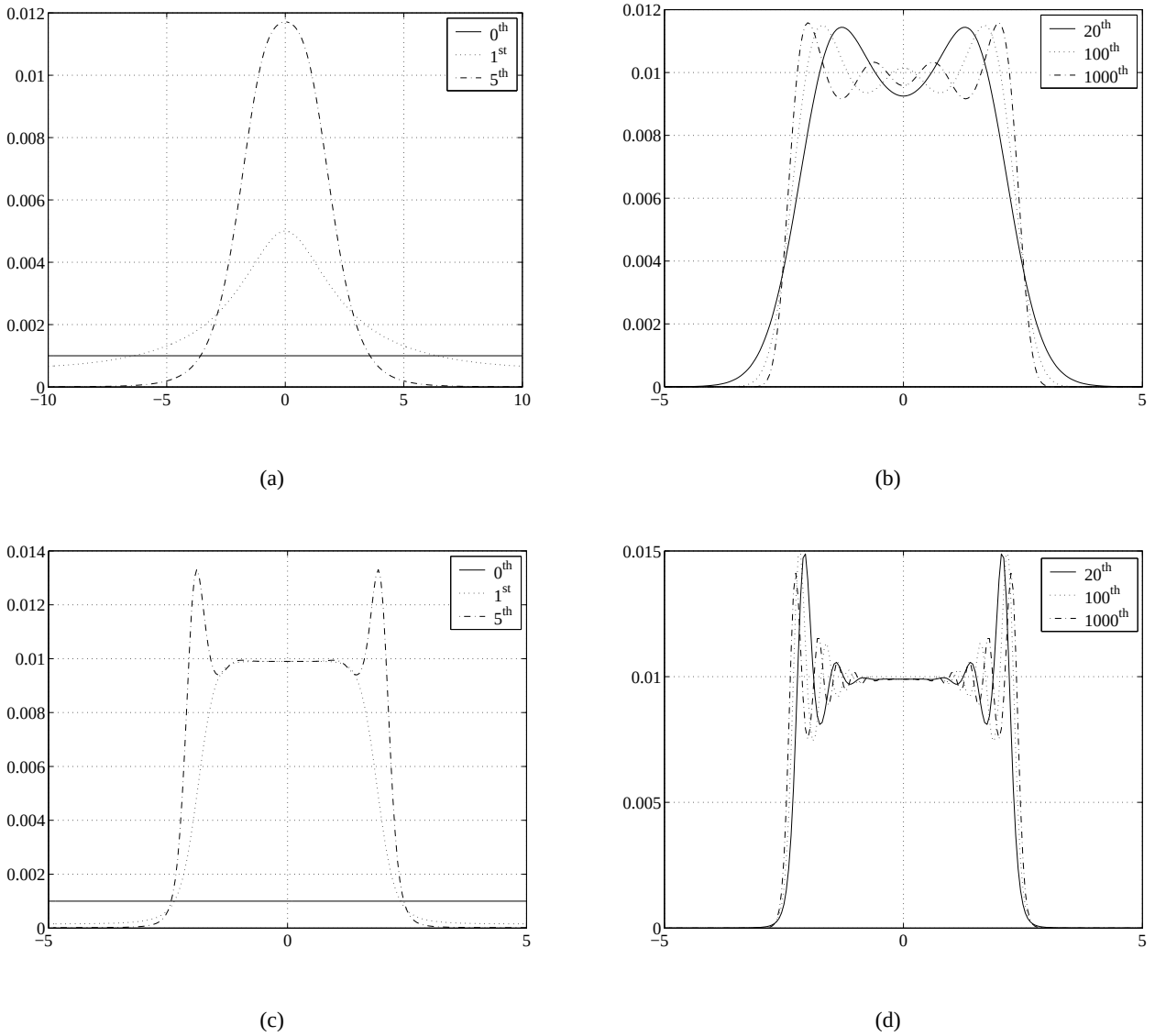


Fig. 12. Estimates for the input density given in Fig. 10 reconstructed by the original minimum *I*-divergence algorithm: (a) Estimates at some selected early iterations when $\sigma_h = 0.5$, $\sigma_n = 0.6$, and $\overline{h} = 1$. (b) Estimates at some selected late iterations when $\sigma_h = 0.5$, $\sigma_n = 0.6$, and $\overline{h} = 1$. (c) Estimates at some selected early iterations when $\sigma_h = 0.5$, $\sigma_n = 0.6$, and $\overline{h} = 8$. (d) Estimates at some selected late iterations when $\sigma_h = 0.5$, $\sigma_n = 0.6$, and $\overline{h} = 8$.