Signal Processing 33 (1993) 209–221
Elsevier

A simple method for calculating the rate distortion function of a source with an unknown parameter

Laurence B. Wolfe[1] and Chein-I Chang*
Department of Electrical Engineering, University of Maryland, Baltimore County Campus, Baltimore, MD 21228, USA

Received 26 September 1991
Revised 6 March 1992 and 19 November 1992

Abstract. The rate distortion function $R(D)$ measures the minimum information rate of a source required to be transmitted at a fidelity level D. Although Blahut developed an elegant algorithm to calculate $R(D)$ for discrete memoryless sources, computing $R(D)$ for other types of sources is still very difficult. In this paper, we study the computation of $R(D)$ for discrete sources with an unknown parameter which takes values in a continuous space. According to the well known ergodic decomposition theorem, a non-ergodic stationary source can be represented by a class of parameterized ergodic subsources with a known prior distribution. Based on this theory, a source matching approach and a simple algorithm is presented for computational purposes. The algorithm is shown to be convergent and efficient. In order to see the performance of this simple algorithm, we consider a special class of binary symmetric first-order Markov sources which has been previously studied. $R(D)$ is computed over this class of sources and compared with the bound developed in previous work by Gray and Berger. The example shows that the algorithm is very efficient and produces results close to Gray and Berger's bound. Other examples further demonstrate the efficiency of the algorithm.

Zusammenfassung. Die Raten-Verzerrungsfunktion $R(D)$ mißt die minimale Informationsrate einer Quelle, wenn bei der Übertragung ein Güteniveau D gefordert ist. Obwohl Blahut einen eleganten Algorithmus zur Berechnung von $R(D)$ für diskrete gedächtnislose Quellen entwickelte, ist die Berechnung von $R(D)$ für andere Quellentypen nach wie vor sehr schwierig. In der vorliegenden Arbeit untersuchen wir die Berechnung von $R(D)$ für diskrete Quellen mit einem unbekannten Parameter, dessen Werte in einem kontinuierlichen Raum liegen. Aufgrund des bekannten ergodischen Zerlegungssatzes kann eine nichtergodische stationäre Quelle durch eine Klasse parameterisierter ergodischer Subquellen mit bekannter a priori Verteilung dargestellt werden. Von diesem Satz ausgehend werden für Zwecke der Berechnung ein Quellenanpassungskonzept und ein einfacher Algorithmus vorgestellt. Es wird gezeigt daß der Algorithmus konvergiert und effizient ist. Um die Leistungsfähigkeit dieses einfachen Algorithmus zu illustrieren, betrachten wir eine spezielle, bereits früher untersuchte Klasse binärer symmetrischer Markov-quellen erster Ordnung. $R(D)$ wird für diese Klasse von Quellen berechnet und mit der in früheren Arbeiten von Gray und Berger entwickelten Schranke verglichen. Das Beispiel zeigt daß der Algorithmus sehr effizient ist und daß dessen Ergebnisse der Schranke von Gray und Berger nahe kommen. Die Effizienz des Algorithmus wird auch durch weitere Beispiele belegt.

Résumé. La fonction taux de distorsion $R(D)$ mesure le débit minimum nécessaire, pour transmettre unc source, avec un niveau de fidélité donné D. Bien que Blahut ait développé un élégant algorithme pour calculer $R(D)$ dans le cas d'une sourcr discrète sans mémoire, le calcul de $R(D)$ pour d'autres types de sources est toujours très difficile. Dans ce papier nous étudions le calcul de $R(D)$ pour des sources discrètes dépendant d'un paramètre inconnu, prenant ses valeurs dans un ensemble continu. En appliquant le thèorème bien connu de décomposition ergodique, une source non ergodique peut être représentée par une classe de sources ergodiques paramétrées, chaque paramètre ayant une distribution a priori connue. A partir de ce théorème une approche basée sur la recherche d'un code adapté à la source conduit à un algorithme simple, qui se prête bien au calcul numérique. L'algorithme converge et est efficace. De manière à voir les performances de cet algorithme, nous considérons une classe particulière de sources binaires, markoviennes d'ordre 1, déjà étudiées auparavant. Sur cette classe, la fonction $R(D)$ a été calculée et les résultats ont été comparés à la borne précédemment établie par Gray

Correspondence to: Dr. Chein-I Chang, Department of Electrical Engineering, University of Maryland, Baltimore County Campus, 5401 Wilkens Ave., Baltimore, MD 21228, USA. Tel.: (410) 455-3502; fax: (410) 455-3559; E-mail: cchang@umbc1.umbc.edu

[1]Presently with the US Government (GSA), Washington DC, USA.

*This work was supported in part by Minta Martin Fund from the College of Engineering, the University of Maryland, in part by a Summer Faculty Fellowship and in part by a Special Research Initiative Support from the University of Maryland, Baltimore County Campus.

et Berger. Cet exemple montre que l'algorithme est très efficace et donne des résultats proches de la borne de Gray et Berger. Les autres exemples qui suivent démontrent encore l'efficacité de l'algorithme.

Keywords. Markov chain; rate distortion function; relative entropy; source coding; sufficient statistic.

1. Introduction

The rate distortion function $R(D)$ measures the minimum amount of information about a source that must be preserved by any data compression code to allow reproduction of the compressed data with average distortion less than or equal to a given D. When a discrete ergodic memoryless source's statistics are completely known, then $R(D)$ may be calculated using Blahut's algorithm [2]. However, if the source is not memoryless or if its statistics are not completely known, then calculation of $R(D)$ may be difficult.

More precisely, the well known ergodic decomposition theorem states that a non-ergodic source cannot be completely characterized by its statistics but only be specified by a class of ergodic sources with a known prior distribution. Accordingly, this theorem can be used to derive many different definitions of $R(D)$. Among them is one defined by Sakrison in [10] which is of particular and practical interest for classes of sources. Using Sakrison's definition, we present a computational algorithm to calculate $R(D)$ for a class of sources with an unknown parameter that takes values in a continuous parameter space. The algorithm uses a source matching approach to find the Sakrison rate distortion function $R(D)$ for a compact class of ergodic sources specified by an unknown random parameter.

The main idea of the approach is to use a simple algorithm which partitions a class of sources into a finite number of subclasses. Since it is known that a source can be completely characterized by its entropy, this Simple Algorithm [4] utilizes relative entropy (i.e. Kullback distance, cross-entropy, discrimination information) as a measure of the similarity between sources in order to group within subclasses, all sources in the class whose entropies are within a previously assigned level of discrepancy. Subsequently, a finite set of representatives is selected, one from each subclass. With this finite set of representatives, an approximation to the rate distortion function and the corresponding minimax code can be found.

This paper is organized as follows. The rate distortion problem is described in Section 2. In Section 3 the Simple Algorithm [4] is adapted to the rate distortion problem for sources with an unknown parameter. Convergence of the Simple Algorithm is also shown and a complexity analysis is presented. Numerical examples are studied in Section 4 and, finally, conclusions are drawn in Section 5.

2. Defining $R(D)$

Consider first an example that demonstrates the difficulty of assigning values of $R(D)$ to a source with an unknown parameter. Let Γ be a process where nature randomly selects one of two ergodic discrete sources according to a given prior distribution. Nature makes an irrevocable selection at time $-\infty$ and the chosen source outputs symbols forever. Clearly, this process is non-ergodic and the value of $R(D)$ will depend upon which of the two sources has been chosen.

Let us specify the two possible sources as source P which almost always produces equiprobable output symbols α and β, while source Q almost always outputs symbol α. Source P is selected with probability π and Q is chosen with probability $1-\pi$. For any value of D, $R(D)=r^P(D)$ if P is chosen and $R(D)=r^Q(D)$ if Q is selected. Which value should be selected for $R(D)$: $r^P(D)$, $r^Q(D)$ or a random variable taking values in the set $\{r^P(D), r^Q(D)\}$? There are many choices and, thus, the theoretical problem becomes one of choosing an appropriate definition.

Proceeding, Shannon [11] first defined the rate distortion function $R^\theta(D)$ for a discrete ergodic memoryless source θ with alphabet $A=\{0, 1, \ldots, J-1\}$ and

reproduction letters $B=\{0, 1, \ldots, K-1\}$ as

$$R^{\theta}(D)=\min_{Q\in Q(D)} I(p_j^{\theta}; Q_{k|j}) , \tag{1}$$

where

$$I(p_j^{\theta}; Q_{k|j})=\sum_{j,k} p_j^{\theta} Q_{k|j} \log\left(\frac{Q_{k|j}}{\sum_i p_i^{\theta} Q_{k|i}}\right) \tag{2}$$

and

$$Q(D)=\left\{Q_{k|j}\left|\sum_{j,k} p_j^{\theta} Q_{k|j} d_{jk} \leqslant D\right.\right\} .$$

The source distortion matrix $[d_{jk}]$ assigns a non-negative penalty value for the reproduction of letter j by letter k and $\{Q\}$ is the set of all such matrices. Unfortunately, as shown by the previous example, Shannon's definition gives no guidance on calculating $R(D)$ for a source with a unknown parameter. To alleviate this difficulty, the Sakrison rate distortion function in [10] is a better candidate for defining the rate distortion function of a source with unknown statistics when it is known that the source is contained in a given class of sources. For a given class of sources Sakrison defined the rate distortion function as

$$R^{\Theta}(D)=\inf_{Q\in Q(D)} \sup_{\theta\in\Theta} I(p^{\theta}; Q) .$$

As shown by Sakrison in [10], if the parameter space Θ is compact, then

$$\begin{aligned} R^{\Theta}(D) &\equiv \inf_{Q\in Q(D)} \sup_{\theta\in\Theta} I(p^{\theta}; Q) \\ &= \sup_{\theta\in\Theta} \inf_{Q\in Q(D)} I(p^{\theta}; Q) \\ &= \sup_{\theta\in\Theta} R^{\theta}(D) . \end{aligned} \tag{3}$$

Apparently, Sakrison's rate distortion function gives a single code for evaluating $R^{\Theta}(D)$ over the class Θ. Another advantage is that Sakrison's definition requires no knowledge of the prior distribution W over the class Θ. Finally, $R^{\Theta}(D)$ is a reasonable design criterion for systems with a specified worst case level of fidelity.

Unfortunately, calculating $R^{\Theta}(D)$ is generally difficult because of the lack of a general computational algorithm. If the decomposition class is known and finite, Blahut's algorithm [2] may be used to calculate $R^{\theta}(D)$ for each $\theta\in\Theta$. $R^{\Theta}(D)$ may then be determined by an exhaustive search over all θ. However, if Θ is continuous, a solution of (3) is not generally available since an uncountable number of integrals must be solved. Fortunately, a source matching approach presented in the next section may be used to calculate an approximation to $R^{\Theta}(D)$.

3. A source matching approach for calculating $R^{\Theta}(D)$

A source matching problem seeks the codes which minimize the maximum redundancies over classes of sources where relative entropy (i.e. Kullback distance, cross-entropy) is adopted as a criterion to measure the redundancy. However, this source matching approach may also be adapted to select both a finite set of representatives and a least favorable distribution W^* for use in finding $R^{\Theta}(D)$.

Reviewing [5] for reference, consider a discrete memoryless source S with n output symbols and probability mass function (pmf) $P=[p_1, \ldots, p_n]$. Let $L=[l_1, \ldots, l_n]$ correspond to a complete variable length code C such that

$$\sum_{i=1}^{n} 2^{-l_i}=1 .$$

The average code length for encoding the source is $\bar{l}(L, P)=\sum_{i=1}^{n} p_i l_i$. Let $\mathscr{L}$ be the set of code length functions and L^*:

$$\bar{l}(L^*, P)=\min_{L\in\mathscr{L}} \bar{l}(L, P) , \tag{4}$$

achieve the minimum average code length for S. Define the redundancy of C by

$$r(L, P)=\bar{l}(L, P)-H(P) , \tag{5}$$

where $H(P)$ is the entropy of the source S. Notice that this measure of redundancy is justified by the double inequality of Shannon's theorem $H(P)\leqslant\bar{l}(L, P)<H(P)+1$, which is true for any decodable code. This

result is derived from the characteristic property of a decodable code given by Kraft's inequality,

$$\sum_{i=1}^{n} 2^{-l_i} \leqslant 1 .$$

Define a pmf by $Q(L) = [q_1(L), \ldots, q_n(L)]$,

$$q_i(L) = 2^{-l_i} .$$

Then (5) can be expressed as

$$\begin{aligned} r(L, P) &= \sum_{i=1}^{n} p_i l_i + \sum_{i=1}^{n} p_i \log (p_i) \\ &= \sum_{i=1}^{n} p_i \log \left(\frac{p_i}{2^{-l_i}} \right) \\ &\equiv H(P; Q(L)) . \end{aligned} \tag{6}$$

Equation (6) implies that the relative entropy (i.e. Kullback distance) $H(P; Q(L))$, can be adopted as a performance criterion to measure the discrepancy between code length function L and the best possible performance for S. Let $S = \{A, \mathscr{B}_A, P^\theta; \theta \in \Theta\}$ be a class of discrete memoryless sources rather than a single source, where $\mathscr{B}_A$ denotes the Borel σ-field. Assume that the class S contains an infinite number of discrete memoryless sources and let Q be the set of all n-dimensional pmfs. Then, according to Davisson and Leon-Garcia [5], a source matching problem seeks a best matched source with pmf Q^* which satisfies

$$\begin{aligned} R_s &= \sup_{\theta \in \Theta} H(P^\theta; Q^*) \\ &= \min_{Q(L) \in Q(\mathscr{L})} \sup_{\theta \in \Theta} H(P^\theta; Q^*) . \end{aligned} \tag{7}$$

Using the minimax principle, they proved a major source matching theorem.

SOURCE MATCHING THEOREM

$$\begin{aligned} \min_{Q(L) \in Q(\mathscr{L})} \sup_{\theta \in \Theta} H(P^\theta; Q) \\ = \sup_{W \in \Xi} \min_{Q(L) \in Q(\mathscr{L})} \mathscr{H}(W; Q) , \end{aligned} \tag{8}$$

where Ξ is the set of all prior distributions defined on θ and

$$\mathscr{H}(W; Q) = \int_\Theta H(P^\theta; Q) \, \mathrm{d}W(\theta) .$$

Rewriting (8), observe that a source matching problem is equivalent to finding the channel capacity between the parameter space Θ and the source output space A:

$$\begin{aligned} &\sup_{W \in \Xi} \min_{Q(L) \in Q(\mathscr{L})} \mathscr{H}(W; Q) \\ &= \sup_{W \in \Xi} \int_\Theta \sum_{i=1}^{M} p_i^\theta \log \left(\frac{p_i^\theta}{\int_\theta p_i^\theta \, \mathrm{d}W(\theta)} \right) \mathrm{d}W(\theta) . \end{aligned} \tag{9}$$

Clearly, solution of (9) yields a distribution W^* which may be used to find $R^\Theta(D)$. W^* is generally called the least favorable distribution over Θ. Unfortunately, there is generally no closed form solution of (9). However, a recent simple algorithm presented in [4] can be modified to calculate an approximation to $R^\Theta D)$. For simplicity of discussion, the Simple Algorithm will be given assuming that the parameter θ is a scalar parameter which lies on the real line in the closed interval $[a, b]$. Additional details are provided in [4]. The extension of the algorithm to vector parameters is straightforward, as we will see in the numeric examples of Markov sources presented in the next section. It should also be noted that, in principle, the Simple Algorithm can be applied to discrete sources, in general.

Assume that the parameter space Θ which describes a class of sources is compact and that we are given a discrete source output alphabet $A = \{1, \ldots, M\}$. Therefore, the statistics of each source in the class are determined by P^θ defined on A where $\theta \in \Theta$. The Simple Algorithm utilizes a Partition Algorithm to divide this class of sources into a finite number of subclasses and selects one representative from each subclass to calculate a prior distribution W^* and an approximation to the code that achieves channel capacity. With this finite set of representatives and W^* we can calculate $R^\Theta(D)$.

The validity of utilizing a finite set of representatives chosen from an uncountable number of sources is supported by [7, Corollary 3, p. 96] which states that for a finite output space A there is a distribution W^* over Θ that assigns a nonzero probability to only a minimal number of sources and W^* gives rise to the optimal minimax code. Also m, the size of this minimal set, can be no larger than the size of the output set M, i.e., $m \leqslant M$.

Therefore, the Simple Algorithm must select J representatives from the subclasses, where J is the size of any set of sources that contains the minimal set, i.e., $m \leqslant M \leqslant J$.

The Partition Algorithm given below groups all sources into J subclasses using relative entropy to measure the similarity between any two sources. Thus, the algorithm produces a finite set of J parameters θ_j, which partitions $[a, b]$ into J subclasses $\{S_j\}$ where $S_j = [\theta_{j-1}, \theta_j]$. The $\{S_j\}$ are chosen such that the relative entropy between any two sources in a subclass is less than a given tolerance ε. The variable r in the Partition Algorithm determines how fine the partition will be and ensures that the number of subclasses will at least equal the size of the discrete source output space alphabet, i.e. $J \geqslant M$. Given this criterion as a stopping rule, the algorithm starts with $r=0$ and an initial error tolerance of ε_0. If $J<M$, a smaller error tolerance $\varepsilon_1 = \varepsilon_0 2^{-r}$ is generated by increasing r by 1. This procedure is repeated until $J \geqslant M$. Thus, the Partition Algorithm by construction ensures that J representatives are selected for the class of sources, where $J \geqslant M$.

PARTITION ALGORITHM

1. Initialization. Set $\varepsilon_0 =$ an assigned error tolerance and $r=0$.
2. Set $\theta_0 = a$, $\varepsilon_1 = \varepsilon_0 2^{-r}$ and $J=0$, where all θ_i take values in $[a, b]$.
3. Set $J=J+1$. Find $z_J > \theta_{J-1}$ such that $H(P^{\theta_{J-1}}; P^{z_J}) = \varepsilon_1$. Find $\theta_J < z_J$ such that $H(P^{z_J}; P^{\theta_J}) = \varepsilon_1$. z_J is the representative for subclass $S_J = [\theta_{J-1}, \theta_J]$.
4. If $z_J \geqslant b$, go to Step 7.
5. If $\theta_J < b$, go to Step 3.
   Otherwise, continue.
6. If $J<M$, let $r=r+1$ and go to Step 2. Otherwise, let $\theta_J = b$ and output $\{\theta_i\}_{j=1}^{J}$ and $\{z_j\}_{j=1}^{J}$ and stop.
7. If $J<M$, let $r=r+1$ and go to Step 2. Otherwise, let $z_J = b$ and output $\{\theta_j\}_{j=1}^{J-1}$ and $\{z_j\}_{j=1}^{J}$ and stop.

The Simple Algorithm may now be given as follows.

SIMPLE ALGORITHM

1. Initialization: let $\varepsilon =$ an assigned error tolerance for Blahut's algorithm.
2. Apply the Partition Algorithm to produce $\{\theta_j\}$ and $\{z_j\}$.
3. Apply Blahut's channel capacity algorithm [2] where the output space is given by A and the input space is given by the representative set $\{z_j\}$. The probabilities $\{P^{z_j}(k)\}_{z_j \in F, k \in A}$ give the channel matrix $\{P(k|z_j)\}_{z_j \in F, k \in A}$.
4. Stop and output the least favorable distribution W^* and set $\{z_j\}$.

The Simple Algorithm discretizes the continuous parameter space Θ and chooses W^* and $\{z_j\}$. Blahut's rate distortion algorithm [2] may then be used to calculate $R^{\Theta}(D)$. To summarize:

ALGORITHM RD.SA

(1) Apply Simple Algorithm to produce $\{z_j\}$ and W^*.
(2) Apply Blahut's Rate Distortion algorithm [1] for each $\{z_j\}$.
(3) Compute and output:

$$R^{z_j}(D) = \inf_{Q \in Q(D)} I(p^{z_j}; Q)$$

and

$$R^{\Theta}(D) = \sup_j R^{z_j}(D) .$$

Clearly, Step 3 of the Partition Algorithm is an upper bound for the complexity of the Simple Algorithm and Algorithm RD.SA. Although Newton's method was used in Step 3 of the Partition Algorithm to produce the results for the examples in the next section, our analysis of the Simple Algorithm's complexity will not be predicated on any particular method but only on the computational requirements.

To determine the complexity, we begin by observing that regardless of the method used, Step 3 is equivalent to finding the roots of $2J$ problems with variables z_j and θ_j. The complexity of each of these problems is determined by the associated function (in this case the pmf). Let f^{Θ} be defined as a function of θ which is the least upper bound (i.e. worst case) complexity of all pmfs indexed by z_j. That is

$$O(f^{\theta}) \geqslant O(P^{z_j}) \quad \forall z_j \in \{z_j\}_{j=1}^{J} .$$

Since there are $2J$ problems, the complexity of the Simple Algorithm is given by

$$O(Jf^{\theta}) . \tag{10}$$

The complexity may be further reduced by observing that two discrete sets of parameters $\{\theta_j\}$ and $\{z_j\}$ are produced by the Simple Algorithm. However, only the set $\{z_j\}$ is used to represent the parameter space. Apparently, the Simple Algorithm can be modified to select only one set of parameters $\{\theta_j\}$ to represent the subclasses and to determine the corresponding W^*. This modification reduces the complexity by one-half and additional details may be found in [13].

It should also be noted that in [5] Davisson and Leon-Garcia presented a source matching algorithm for binary memoryless sources that may also be adapted to the rate distortion problem. However, this algorithm solves a problem that is different but related to that solved by the Simple Algorithm. Davisson and Leon-Garcia's algorithm seeks the finite set of localized maxima $\{\theta_j\}$ over the continuous parameter space Θ. It is this finite set that is chosen by Davisson and Leon-Garcia's algorithm to represent the class of sources S. In contrast, the Simple Algorithm seeks a partition of the class of sources S into subclasses where relative entropy is used as a criterion to group similar sources within each subclass.

Subsequently, one source is selected from each subclass by the Simple Algorithm to represent the class of sources.

Both algorithms, however, produce a discrete least favorable distribution W^* and a finite set of representatives for the parameter space Θ which may subsequently be used as inputs to Blahut's rate distortion algorithm [2] to find $R^{\Theta}(D)$.

To conclude our examination we adapt Davisson and Leon-Garcia's algorithm to the rate distortion problem and compare its complexity with the Simple Algorithm.

DAVISSON AND LEON-GARCIA'S ALGORITHM

1. Initialization. Select an error tolerance ε and arbitrary discrete parameter set $\{\theta_j\}$ of size J, where $J \geqslant M$ and M is the size of the source output space.
2. Apply Blahut's algorithm to find Q, the distribution over the output space.
3. Find $\{\theta_j^*\}$, the set of local maxima of $H(P^{\theta};Q^*)$. $\{\theta_j\} \leftarrow \{\theta_j^*\}$.
4. *If* $\mathscr{K}(W;Q) - \max_{\theta \in \Theta} H(P^{\theta};Q) \leqslant \varepsilon$, then output W^* and $\{\theta_j\}$. Else, go to Step 2.

Since W^* and a finite set of representatives $\{\theta_j\}$ are now available, the algorithm can now be modified to calculate $R^{\Theta}(D)$. To summarize these results.

ALGORITHM RD.DLG

(1) Apply Davisson and Leon-Garcia's algorithm to produce $\{\theta_j\}$ and W^*.
(2) Apply Blahut's Rate Distortion algorithm [2] for each $\{\theta_j\}$.
(3) Compute and output:

$$R^{\theta_j}(D) = \inf_{Q \in Q(D)} I(p^{\theta_j};Q)$$

and

$$R^{\Theta}(D) = \sup_j R^{\theta_j}(D) .$$

Clearly Step 3 of Davisson and Leon-Garcia's algorithm upper bounds the time complexity for this algorithm and Algorithm RD.DLG. Also, it is apparent that regardless of the approach used, the search for the optimal set of local maxima in Step 3 of Davisson and Leon-Garcia's algorithm involves the identification and evaluation of all relative local extrema. While the local maxima may equal global maxima in the simplest classes of sources, more complex sources will have additional extrema. Therefore in general, the number of extrema to be evaluated will be a function of J, say $g(J)$, where $g(J) \geqslant J$.

Although many methods are available to identify and evaluate extrema, a generalized approach will include taking the derivatives of functions (i.e. pmfs in this case) and finding the roots of the resulting equations. Accordingly, repeated iterations may be required. However, we will assume that derivatives are taken and roots found for each of the $g(J)$ problems in just one iteration. To be consistent with our previous approach we will again utilize $O(f^{\theta})$ as the complexity of finding the roots. Thus, the complexity of each of the $g(J)$

problems is $O(f^{\theta}+1)$ and the complexity of the algorithm is

$$O(|g(J)|f^{\theta}+|g(J)|)\,. \tag{11}$$

Clearly the Simple Algorithm is more efficient since

$$O(|g(J)|f^{\theta}+|g(J)|)>O(Jf^{\theta})\,. \tag{12}$$

We can also state the following theorem.

THEOREM 1. The approximation to $R^{\Theta}(D)$ calculated by the Simple Algorithm converges to the optimal $R^{\Theta}(D)$ for S and is bounded from below by the approximation to $R^{\Theta}(D)$ as calculated by Davisson and Leon-Garcia'a algorithm.

PROOF. The proof is given in Appendix A.

In concluding this section we observe yet another advantage provided by the Simple Algorithm. The Simple Algorithm by construction always selects a finite set of representatives for the continuous space regardless of the initial error threshold. However, Davisson and Leon-Garcia's algorithm may continue to iterate without stopping when hardware/software round off error prevents the algorithm from achieving the selected threshold error.

4. Examples

Example 1. Discrete first-order binary Markov sources

In this example, we examine a class of binary symmetric sources with memory where the parameter $\boldsymbol{\theta}=[\theta_0, \theta_1]$ is a two-dimensional variable in the unit square $[0, 1]\times[0, 1]$. Without loss of generality, the source alphabet is given by $A=\{0, 1\}$ and the stochastic transition matrix is given by

$$\begin{bmatrix}\theta_{00} & \theta_{01}\\ \theta_{10} & \theta_{11}\end{bmatrix}=\begin{bmatrix}1-\theta_0 & \theta_0\\ \theta_1 & 1-\theta_1\end{bmatrix},$$

where θ_{ij} for $i, j\in\{0, 1\}$ represents the transition probability from the previous state i to the present state j. For simplicity of notation, θ_{01} is denoted by θ_0 and θ_{10} is denoted by θ_1. The stationary pmf is easily shown to be

$$\boldsymbol{\pi}=\left(\frac{\theta_1}{\theta_0+\theta_1}, \frac{\theta_0}{\theta_0+\theta_1}\right)=(\pi_0, \pi_1)\,. \tag{13}$$

The distortion matrix, $[d_{ij}]$, for this example and all examples in this section is given by the probability of error matrix such that $d_{ij}=0$ if $i=j$ and 1 elsewhere.

Let $\boldsymbol{x}^N=[x_1, \ldots, x_N]$ be a message block of length N generated by the source and $p(\boldsymbol{x}^N|\boldsymbol{\theta})$ or $p^{\theta}(\boldsymbol{x}^N)$ denote the pmf of the source indexed by $\boldsymbol{\theta}$ (both notations will be used interchangeably). Then, for each Markov source indexed by $\boldsymbol{\theta}$,

$$p(\boldsymbol{x}^N|\boldsymbol{\theta})=\pi_h(1-\theta_0)^i\theta_0^j\theta_1^k(1-\theta_1)^{N-i-j-k-1}\,, \tag{14}$$

where i, j, k represent the number of stage transitions as defined by the transition matrix and π_h is the steady-state probability of x_1 in state h. Since each $\boldsymbol{\theta}^*\in\boldsymbol{\Theta}$ takes values in the entire range space (i.e. the unit square $[0, 1]\times[0, 1]$) and the class of sources is symmetric (i.e. $\theta_0=\theta_1$), we may observe by inspection that the maximum value of $R^{\theta}(D)$ which solves (2) occurs when all message blocks $\boldsymbol{x}^n$ are equiprobable. That is, when $p(\boldsymbol{x}^N|\boldsymbol{\theta}^*)=(\frac{1}{2})^N$ for some $\boldsymbol{\theta}^*\in\boldsymbol{\Theta}$. This clearly occurs at the value of $\boldsymbol{\theta}^*$ that provides maximum uncertainty about the source output $\boldsymbol{x}^n$; that is, when $\boldsymbol{\theta}^*=[0.5, 0.5]$. The resulting $R^{\Theta}(D)$ is shown in Fig. 1.

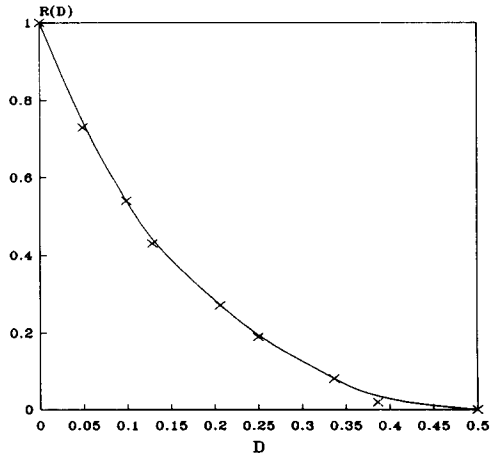

Fig. 1. $R^{\Theta}(D)$ for the class of binary symmetric Markov sources.

However, in general $R^{\Theta}(D)$ is not easily calculated as can be shown by changing this example so that each $\boldsymbol{\theta} \in \boldsymbol{\Theta}$ can take values only in $[0.1, 0.4] \times [0.2, 0.45]$. Clearly, the pmf given by (14) may be utilized with the Simple Algorithm to solve (2) and calculate an approximate value of $R^{\Theta}(D)$. The Simple Algorithm was given for scalar parameters in the previous section. However, the algorithm can easily be extended to vector parameters by performing vector calculations and comparisons. For example in this problem, Step 5 of the algorithm is modified to compare $\boldsymbol{\theta} = (\theta_0, \theta_1)$ with $\boldsymbol{b} = (b_0, b_1)$. This results in a comparison of θ_0 to the end point $b_0 = 0.4$ in the interval $[0.1, 0.4]$ and a comparison of θ_1 to the endpoint $b_1 = 0.45$ in the interval $[0.2, 0.45]$. Clearly, the comparison results in a decision that $\boldsymbol{\theta} > \boldsymbol{b}$ if either $b_0 \leqslant \theta_0$ or $b_1 \leqslant \theta_1$.

Unfortunately, this particular problem soon becomes intractable due to the exponentially increasing size of the source alphabet that occurs when the message blocks increase in length. Therefore, we briefly review and utilize a sufficient statistic that we presented in [13] to reduce the complexity of the problem. The sufficient statistic uses a result in [6, p. 43] and is based on the number of fixed length subchains (i.e. runs) of consecutive 1s that appear in each message block. For example, $\boldsymbol{x}^5 = [00011]$ has one subchain (i.e. run) of 1s of length 2.

For a message block $\boldsymbol{x}^N = [x_1, \ldots, x_N]$ with n 1s, let a_i be the number of runs of 1s of length i, where i ranges from 1 to n. Then D_n, the total number of all the runs of 1s with length up to n in $\boldsymbol{x}^N$, is upper bounded by

$$D_n \sum_{i=1}^{n} a_1 \leqslant \min(n, N-n+1) .$$

With x_1 and x_N the first and last symbols of $\boldsymbol{x}^N$ fixed. $T(\boldsymbol{x}^N) = (n, D_n, x_1, x_N)$ uniquely specifies $p(\boldsymbol{x}^N | \boldsymbol{\theta})$ and is an eligible sufficient statistic. However, it is apparent that the probability of $(n, D_n, 0, 1)$ is identical to that of $(n, D_n, 1, 0)$, as can be seen by applying (14).

Define the following three cases:

Case 1. $(n, D_n, 1) \equiv (n, D_n, 0, 0)$,

Case 2. $(n, D_n, 2) \equiv$
$(n, Dn, 0, 1)$ or $(n, Dn, 1, 0)$, (15)

Case 3. $(n, D_n, 3) \equiv (n, D_n, 1, 1)$.

Let i identify the specific case and define the sufficient statistic as

$$\hat{T}(\boldsymbol{x}^N) \equiv (n, D_n, i) ,$$

We can now utilize the sufficient statistic to define the pmf. First, let

$$\binom{i}{j} = \frac{i!}{j!(i-j)!}, \quad \text{if } i, j \text{ and } (i-j) \geqslant 0 ;$$

$$\binom{i}{j} = 0, \quad \text{elsewhere.}$$

Then

$$M_{(n,D_n,i)} \equiv \{\boldsymbol{x}^N \in \{0, 1\}^N : \hat{T}(\boldsymbol{x}^N) = (n, D_n, i)\}$$
$$= (1 + \delta_{i,2}) \binom{N-n+1}{D_n - i + 1} \binom{n-1}{D_n - 1}$$
$$+ \delta_{N-n,0} \delta_{D_n,1} + \delta_{n,0} \delta_{D_n,0} , \qquad (16)$$

and it is apparent that

$$\sum_{\substack{(n,D_n,i) \\ n<N \\ i=1,2,3}} M_{(n,D_n,i)} = 2^N .$$

Consequently, (16) may be used to yield the desired pmf,

$$p(\hat{T}(\boldsymbol{x}^N) = (n, D_n, i) \mid \theta)$$
$$= \frac{M_{(n,D_n,i)}}{(\theta_0 + \theta_1)} (1-\theta_0)^{N-n-D_n+i-2} \theta_0^{D_n}$$
$$\times \theta^{D_n - i + 2} (1-\theta_1)^{n-D_n} . \qquad (17)$$

Thus, we have reduced the problem from exponential to a size that is polynomial in N (see Table 1). 2^N probabilities are needed to completely describe a binary first-order Markov source. However, the sufficient statistic reduces the complexity to

$$\sum_{n,D_n,i} (1 - \delta_{0,M(n,D_N,i)}) .$$

Both the Simple Algorithm and Davisson and Leon-Garcia's algorithm were applied to (17) to find $R^{\theta}(D)$. However, consideration was exclusively restricted to the class of binary symmetric (i.e. $\theta_0 = \theta_1$) first-order

Table 1

First-order binary chain

N	Size of $\|A\|^N$	Size of T	Reduction ratio $\|A\|^N/\|T\|$
3	8	6	1.33
5	32	16	2.00
7	128	32	4.00
9	512	54	9.48
11	2048	82	24.98
13	8192	116	70.62

$A = \{0,1\}$

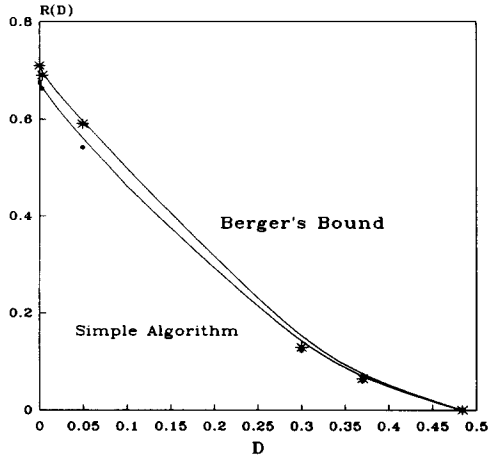

Fig. 2. Berger's bound for $R^{\Theta}(D)$.

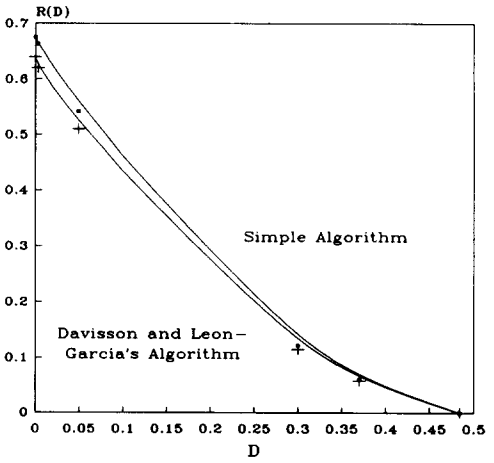

Fig. 3. $R^{\Theta}(D)$ as calculated by two algorithms.

discrete Markov sources studied by Gray [8] and Berger [1]. As an additional constraint, the range space is given by $[0.1, 0.4] \times [0.2, 0.45]$. Not only are the

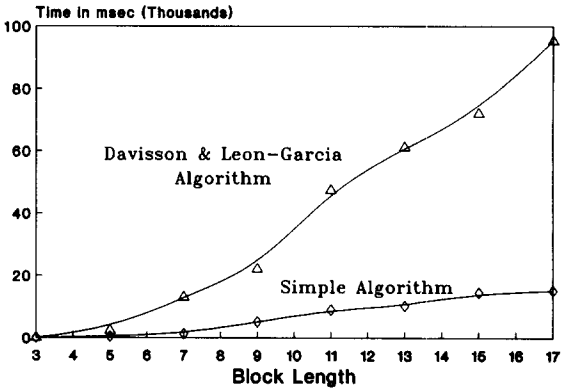

Fig. 4. Computer processing time for symmetric Markov source.

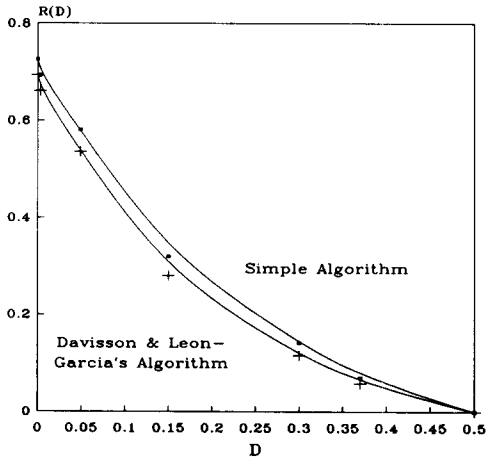

Fig. 5. $R^{\Theta}(D)$ for binary nonsymmetric Markov source.

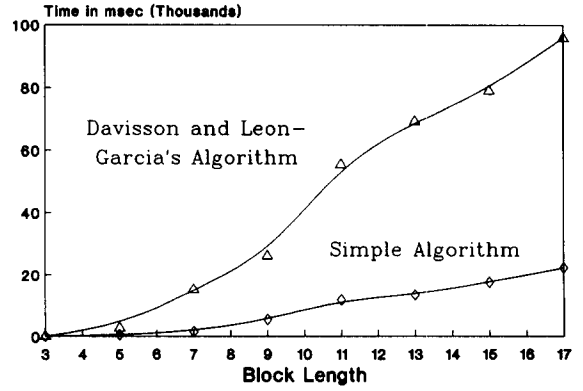

Fig. 6. Computer time for binary nonsymmetric Markov source.

results efficiently produced by the Simple Algorithm, but they are also very close to those produced by Davisson and Leon-Garcia's algorithm and those derived by applying Berger's bound [1] over the decomposition class. Figures 2–4 summarize these results.

To further demonstrate this approach, we now remove the symmetry constraint. The results are shown in Figs. 5 and 6.

Example 2. Finite alphabet, first-order Markov sources

Consider the class of stationary discrete first-order Markov sources with an alphabet of size J; that is, $A=\{1, \ldots, J\}$. The stochastic transition matrix is

$$\begin{bmatrix} \theta_{11} & \theta_{12} & \cdots & \theta_{1J} \\ \vdots & & \vdots & \vdots \\ \theta_{J1} & \theta_{J2} & \cdots & \theta_{JJ} \end{bmatrix} = \begin{bmatrix} \theta_{11} & \theta_{12} & \cdots & 1-\sum_{i\neq J}\theta_{1i} \\ \vdots & \vdots & & \vdots \\ \theta_{J1} & \theta_{J2} & \cdots & 1-\sum_{k\neq J}\theta_{Jk} \end{bmatrix}.$$

For all $i, j \in A$, θ_{ij} represents the transition probability for letter j given a preceding letter i. Therefore, in general, a finite alphabet first-order Markov source can be represented by an array of J parameter vectors $\boldsymbol{\theta}=[\boldsymbol{\theta}_j:\boldsymbol{\theta}_j=(\theta_{j1}, \ldots, \theta_{jJ}]$, where $\boldsymbol{\theta}$ takes values in the J-dimensional hypercube $[0, 1]^J$.

Following Example 1, it is clear that $R^{\Theta}(D)$ is determined by the value of $\boldsymbol{\theta}$ which yields an equiprobable source when the class of sources includes all finite alphabet first-order Markov sources such that $\boldsymbol{\theta}$ takes values in the J-dimensional hypercube $[0, 1]^J$. However. $R^{\Theta}(D)$ cannot necessarily be determined by inspection for other parameter spaces. Therefore, this example will use a parameter space given by $[0.1, 0.3]\times[0.1, 0.4]\times[0.3, 0.45]$ and a ternary alphabet $A=\{0, 1, 2\}$.

We consider the Nth extension of the first-order Markov source with a J-length alphabet A. Because this problem becomes intractable for increasing message block length N, we begin by deriving a sufficient statistic [12] for finite alphabet first-order Markov sources using a derivation of the run-length approach presented in Example 1.

Recall that a sufficient statistic was derived in Example 1 by observing that (n, D_n, x_1, x_N) uniquely specifies $p(\boldsymbol{x}^N|\boldsymbol{\theta})$, where D_n is the total number of all the runs of 1s with length up to n in $\boldsymbol{x}^N$. However, a binary alphabet was used in Example 1 such that for $i\neq 1$, and given that x^N has a run of 1s beginning with x_i, it is clear that $x_{i-1}=0$. Stated in another way, this means that there is a conjunction of a 0 and a 1 when letter x_i is succeeded by letter x_{i-1}. It is therefore clear that a sufficient statistic for first-order binary Markov sources is equivalently derived by observing the number of conjunctions C_{ij} for message blocks $\boldsymbol{x}^N$ where $i, j\in\{0, 1\}$. Apparently, this observation can be used to derive a sufficient statistic for finite alphabet first-order Markov sources by concluding that given fixed first letters x_1 and x_N, C_{ij} and C_{jk} determine the number of message blocks $\boldsymbol{x}^N$ with a fixed number of arrangements of letter j preceded by letter i and succeeded by letter k.

More precisely, let α and γ be the first and last letters of message block $\boldsymbol{x}^N=[x_1, \ldots, x_N]$ where α and γ are drawn from alphabet A. Let the probability that the initial letter $x_1=\alpha$ be given by π_α and let the number of conjunctions C_{ij} and C_{jk} also be given for all i, j, $k\in A$. Let r_{ijk} be defined as the number of occurrences of the subchain ijk in $\boldsymbol{x}^N$ where letter i precedes letter j which is succeeded by letter k. Clearly, $0\leqslant r_{ijk}\leqslant N-2$. The set $\{C_{jk}\}$ apparently defines both $\{r_{ijk}\}$ and the last letter γ. However, $X_1=\alpha$ must be given because of the boundary condition $r_{\varnothing\varnothing\alpha}$.

Apparently, $(\alpha, \{r_{ijk}\})$ defines both D_i, the number of runs of letter i as well as n_i, the total number of letters i in message block $\boldsymbol{x}^N$ since

$$D_i \equiv \sum_{\substack{j,k\in A \\ j\neq 1}} r_{ijk} + \delta_{i,\gamma},$$

where

$$0\leqslant D_i \leqslant \left\lceil \frac{N}{2} \right\rceil,$$

and

$$n_j \equiv \sum_{i,k\in A} r_{ijk} + \delta_{\alpha j} + \delta_{j,\gamma},$$

where

$0 \leqslant n_i \leqslant N$.

We may therefore conclude that each unique message block $\boldsymbol{x}^N$ is completely determined by α and the set $\{r_{ijk}\}$. Therefore, a sufficient statistic is given by

$$T(x^N) = (\alpha, \{r_{ijk}\}) .$$

Observe that for every $j \in A$ it is apparent that after accounting for boundary conditions, the magnitude of the C_{ij} when summed over i must balance with the sum over k of all C_{jk}. We conclude that the permutations of the C_{ij} and C_{jk} subchains ij and jk, respectively, determine the number of $\boldsymbol{x}^N$ with a fixed total number of subchains ijk; that is, the total numbers of all m such that $X_m = i$, $X_{m+1} = j$ and $X_{m+2} = k$, where $1 \leqslant m \leqslant N-2$. In order to account for all such $\boldsymbol{x}^N$, we must account for permutations of the $(x_1, \{C_{ij}\})$. There are two cases: Case I where $i \neq j, j \neq k$ and Case 2 where $i = j$ and (or) $j = k$. Case 1 represents distinct state changes and Case 2 represents events that remain in the same state.

For simplicity of notation, define the total number of instances where a fixed letter i is succeeded by any letter $j \in A$ which precedes a fixed letter k as

$$R_{ik} \equiv \sum_{j \in A} r_{ijk} .$$

Consider Case 1 first, where $i \neq j$, $j \neq k$. Clearly the unique number of arrangements of the (distinct) individual set elements $\{r_{ijk}\}$ out of R_{ik} can be determined in an obvious way. Now consider Case 2 where the $\{r_{ijk}\}$ are given by $i = j$ and (or) $j = k$. In this case the process remains in the same state for a specific number of transitions; that is, the chain has subchains (e.g. runs) of i of a specific length. For each $i \in A$, the number of ways to begin D_i runs of n_i (non-distinct) i's is determined in an obvious way. Combining Cases 1 and 2, the total number of $\boldsymbol{x}^N$ with $X_1 = \alpha$ and $\{r_{ijk}\}$ is

$$M_{(\alpha,\{r_{ijk}\})} = \prod_{i,k \in A} \frac{R_{ik}!}{\prod\limits_{\substack{j \in A \\ j \neq i,k}} r_{ijk}!} \prod_{m \in A} \binom{n_m - 1 + \delta_{n_m,0}}{D_m - 1 + \delta_{D_m,0}}, \tag{18}$$

where the $\delta_{(\ )}$ account for boundary conditions. It is apparent that

$$\sum_{\substack{(\alpha,\{r_{ijk}\}) \\ 0 \leqslant r_{ijk} \leqslant N-2 \\ \alpha,i,j,k \in A}} M_{(\alpha,\{r_{ijk}\})} = |A|^N ,$$

where $|A|$ denotes the size of A. The sufficient statistic reduces the calculations needed from $|A|^N$ to

$$\sum_{\substack{(\alpha,\{r_{ijk}\}) \\ 0 < r_{ijk} \leqslant N-2 \\ \alpha,i,j,k \in A}} (1 - \delta_{0,M(\alpha,\{r_{ijk}\})}) .$$

Table 2 shows examples of the reduction achieved by the sufficient statistic. Equation (18) can now be used to derive the desired source pmf by recalling the definition of r_{ijk} in terms of C_{ij} and C_{jk}. Therefore, the pmf is given by

Table 2

First-order ternary chain

N	Size of $\|A\|^N$	Size of T	Reduction ratio $\|A\|^N/\|T\|$
3	27	27	1.00
5	243	186	1.31
7	2187	840	2.60
9	19683	2784	7.07
11	177147	7476	23.70
13	1574323	17313	90.93

$A = \{0,1,2\}$

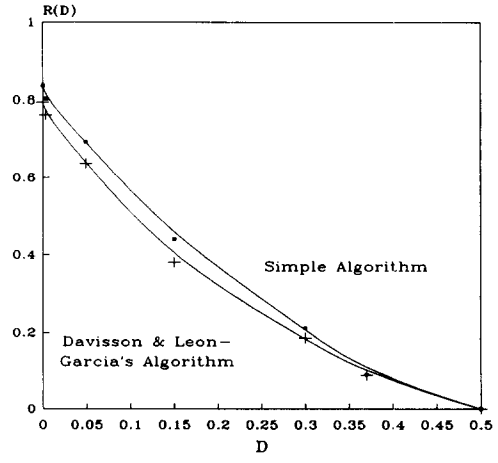

Fig. 7. $R^{\Theta}(D)$ for ternary nonsymmetric Markov source.

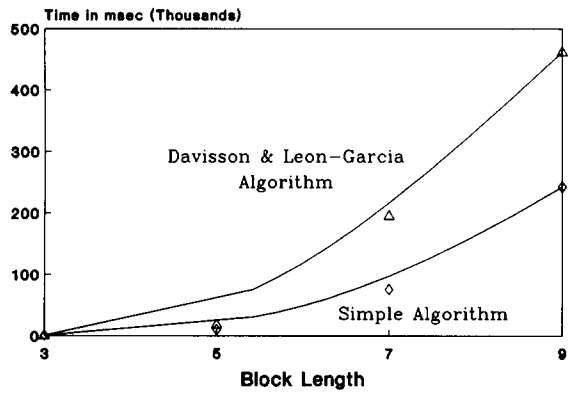

Fig. 8. Computer time for ternary nonsymmetric Markov source.

$$p(T(\boldsymbol{x}^N)=(\alpha,\{r_{ijk}\})\,|\,\boldsymbol{\theta}) = M_{(\alpha,\{r_{ijk}\})}\,\pi_\alpha \prod_{\substack{i,j\in A\\ i\neq j}} \theta_{ij}^{C_{ij}} \times \prod_{k\in A}\left(1-\sum_{\substack{m\in A\\ m\neq k}}\theta_{km}\right)^{n_k-\sum_{m\neq k}C_{km}} \tag{19}$$

Furthermore, it is apparent that this same approach can be used to derive a sufficient statistic for finite alphabet finite order Markov sources.

$R^{\Theta}(D)$ was calculated for a first-order Markov source with alphabet $A=\{0, 1, 2\}$ and parameter space $[0.1, 0.3]\times[0.1, 0.4]\times[0.3, 0.45]$. Figures 7 and 8 show the results. The results for all examples were produced on an IBM 4381 computer.

5. Conclusions

A source matching approach for calculating the rate distortion function of a source with an unknown, continuous parameter was presented in this paper. The source matching approach seeks a finite set of representatives and a least favorable distribution using a simple algorithm to discretize the continuous parameter space. Convergence was proven and the Simple Algorithm was applied to several examples including a binary symmetric first-order discrete Markov decomposition class. The results were compared with both Davisson and Leon-Garcia's algorithm [5] as well as Berger's bound [1] and show that the algorithm is both efficient and produces results with a high degree of accuracy. Sufficient statistics were introduced to reduce the size of the problem for both binary and finite alphabet examples of Markov sources. These examples similarly demonstrated the efficiency of the Simple Algorithm. Thus, the algorithm becomes very attractive for computer implementations.

Appendix A

PROOF OF THEOREM 1. The theorem states that the approximation to $R^{\Theta}(D)$ calculated by the Simple Algorithm converges to the optimal $R^{\Theta}(D)$ for S and is bounded from below by the approximation to $R^{\Theta}(D)$ as calculated by Davisson and Leon-Garcia's algorithm. To prove the first part we need only show that the minimax redundancy converges to zero for the code that achieves channel capacity between the source output space and the parameter space. This is true because [7, Theorem 4.5.1] tells us that channel capacity is achieved when

$$\mathscr{I}(W;Q^*)=H(P^{\theta_j};Q^*)$$

for $P^{\theta_j}>0$, $j\in\{1, \ldots, J\}$ and $J\geqslant M$ where M is the size of the alphabet A. Also, convergence of Davisson and Leon-Garcia's algorithm was previously proven in [3], that is,

$$\lim_{\substack{N\to\infty\\ \varepsilon\to 0}} H(P^{\theta_j};Q^*)=0.$$

However, the Simple Algorithm selects a representative z_j for the partition containing θ_j, and

$$H(P^{z_j};P^{\theta_j})\leqslant\varepsilon.$$

Thus, for every ε, there exists a J depending on ε such that

$$H(P^{z_j};Q^*)-H(P^{\theta_j};Q^*)\leqslant\varepsilon.$$

Therefore, the first part of the theorem holds because

$$\lim_{\substack{N\to\infty \\ \varepsilon\to 0}} (H(P^{z_j};Q^*) - H(P^{\theta_j};Q^*)) = 0 .$$

It is clear that the second part of the theorem holds if and only if the channel capacity between A and Θ as calculated by the Simple Algorithm converges from below. This is easily seen to be true by observing that the Simple Algorithm maps S to a new class of source S' which can be viewed as encoding S by S' and encoding Θ by $\{z_j\}$. The Data Processing Theorem [7, p. 80] tells us that the channel capacity between A and $\{z_j\}$ cannot exceed that between A and Θ and the theorem follows. □

References

[1] T. Berger, "Explicit bounds to $R(D)$ for a binary symmetric Markov source", *IEEE Trans. Inform. Theory*, Vol. IT-23, January 1977, pp. 52–59.

[2] R.E. Blahut. "Comparison of channel capacity and rate distortion functions", *IEEE Trans. Inform. Theory*, Vol. IT-18, July 1972, pp. 460–473.

[3] C.-I Chang, A generalized minimax approach to statistical decision problems with applications to information theory, Ph.D. Dissertation, University of Maryland, College Park, MD, 1987.

[4] C.-I Chang. S.C. Fan and L.D. Davisson, "A simple method of calculating channel capacity and finding minimax codes for source matching problems", 1988 *Conf. Information Sciences and Systems*, Princeton Univ., Princeton, NJ, 16–18 March 1988, pp. 362–366.

[5] L.D. Davisson and A. Leon-Garcia, "A source matching approach to finding minimax codes", *IEEE Trans. Inform. Theory*, Vol. IT-26, March 1980, pp. 166–174.

[6] W. Feller, *An introduction to Probability Theory and Its Applications*, Vol. 1, Wiley, New York, 1967, 3rd Edition.

[7] R.G. Gallagher, *Information Theory and Reliable Communications*, Wiley, New York, 1968.

[8] R.M. Gray, "Information rates of autoregressive processes", *IEEE Trans. Inform. Theory*, Vol. IT-16, July 1970, pp. 412–421.

[9] M.D. Pursley and L.D. Davisson, "Variable rate coding for non-ergodic sources and classes of ergodic sources subject to a fidelity criterion", *IEEE Trans. Inform. Theory*, Vol. IT-22, May 1976, pp. 324–337.

[10] D.J. Sakrison, "The rate distortion function for a class of sources", *Inform. and Control*, Vol. 15, 1969, pp. 165–195.

[11] C.E. Shannon, "A mathematical theory of communications". *Bell Syst. Tech. J.*, Vol. 27, July 1948, pp. 379–423.

[12] L.B. Wolfe and C.-I Chang, "A complete sufficient statistic for finite-state Markov processes with application to source coding", *Proc. 1992 Conf. Information Sciences and Systems*, Princeton, NJ, USA, 18–20 March 1992.

[13] L.B. Wolfe and C.-I Chang, "Source matching problems revisited", *Proc. Internat. Conf. Signal Process. '90*, Beijing, China, 22–26 October 1990. pp. 119–122.