Side Information Source Coding: Low Complexity Design and Source Independence *

Qian Zhao Sidharth Jaggi Michelle Effros

{qianz, jaggi, effros}@z.caltech.edu
Electrical Engineering Department,
California Institute of Technology,
Pasadena, CA 91125

Abstract

Correlated sources X and Y are drawn i.i.d. according to probability mass function (pmf) $p(x,y)$. In the side information source code (SISC) configuration: $p(x,y)$ is known a priori *to both the encoder and the decoder; the encoder knows X but not Y; the decoder knows Y but not X; the encoder encodes X and transmits the description of X to the decoder; the decoder reconstructs X using the source description and side information Y. The universal linked side information source code (ULSISC) configuration modifies the SISC configuration by assuming that $p(x,y)$ is unknown* a priori *and that an asymptotically negligible amount of communication is allowed from the decoder to the encoder. We combine SISC design with ULSISC theory to build codes for applications where the source statistics are unknown at design time. Experimental results compare ULSISC and SISC performance.*

1 Introduction

A side information source code (SISC) [1] is a data compression algorithm designed for a network with a single transmitter and a single receiver where the receiver is assumed to have access to side information that is unavailable to the encoder. In particular, consider finite alphabets $\mathcal{X}$ and $\mathcal{Y}$, and assume that $(X_1,Y_1),(X_2,Y_2),\ldots$ are drawn i.i.d. according to joint pmf $p(x,y)$ on alphabet $\mathcal{X}\times\mathcal{Y}$. An SISC comprises an encoder and a decoder. The encoder maps a sequence of source samples $X_1,\ldots,X_n$ to a binary description, and the decoder reconstructs the source sequence using both the encoder's binary description and the side information $Y_1,\ldots,Y_n$. Like traditional compression systems where no side information is available, SISCs are source dependent. That is, the optimal SISC designed for pmf $p(x,y)$ is not the same as the optimal SISC designed for a distinct pmf $q(x,y)$, in general.

When the pmf $p(x,y)$ is unknown at design time, we require more sophisticated techniques to achieve good performance. A *universal linked side information source code* (ULSISC) [2, 3] is a modified SISC that achieves asymptotically optimal performance for any pmf $p(x,y)$ on fixed alphabet $\mathcal{X}\times\mathcal{Y}$. The modification involves allowing an asymptotically negligible amount of communication from the decoder back to the encoder. This modification is critical to achieving universality in the SISC framework [2, 3].

The proof of the existence of ULSISCs given in [2, 3] is constructive. The encoder describes some fraction of the incoming data sequence to the decoder using a simple source code. The decoder then estimates $p(x,y)$ and describes its estimate to the encoder. Finally, the encoder describes X^n using an SISC matched to the pmf estimate. A careful balance between the fraction of the data sequence used in the pmf estimate and the resulting estimation accuracy allows for asymptotically optimal performance for ULSISC.

This approach is difficult to implement in practice. Using an SISC matched to the estimate of $p(x,y)$ requires that both the encoder and the decoder either have access to a family of pre-designed SISCs or are able to design SISCs to match the estimated pmf. However, optimal SISC design is NP-hard [4]. In this paper we investigate the performance associated with implementing the construction of [2, 3] using the fast but suboptimal SISC design techniques from [5, 6].

The remainder of the paper is organized as follows.

*This material is based upon work supported by NSF under Award No. CCR-0220039, the Caltech Lee Center for Advanced Networking, and the Intel Technology for Education 2000 program.

0-7803-7576-9/02$17.00 © 2002 IEEE

Section 2 gives a brief review of fast SISC design. Section 3 applies fast SISC design to ULSISC design. Section 4 contains experimental results that compare ULSISC and SISC performance, thereby quantifying the performance penalty associated with not knowing the source statistics. Section 5 gives a summary of the key contributions of the paper.

2 Fast SISC Design

Since optimal SISC design is NP-hard [4], we use a fast, suboptimal SISC design algorithm. The algorithm in [5, 6] gives an iterative descent technique for searching the space of possible codes for a solution that minimizes the rate subject to a constraint on the error probability. We here employ an unconstrained minimization of the form $R + \lambda P_e$, where R is the rate, P_e is the error probability, and λ is a non-negative Lagrangian constant. The output of the design is an instantaneous SISC with a non-zero error probability. Constraining the search to a number of operations that grows as $O(|\mathcal{X}|^6)$ yields good performance in the experiments performed to date and makes the algorithm feasible for on-line design. Some details follow.

In designing SISCs with non-zero error probabilities, we allow only errors that do not propagate. Thus the code must decode the right number of bits even when an error occurs. Given this constraint, any symbols $x, x' \in \mathcal{X}$ can have identical binary descriptions $\gamma_X(x) = \gamma_X(x')$, but only symbols $x, x' \in \mathcal{X}$ such that $p(x,y)p(x',y) = 0$ for all $y \in \mathcal{Y}$ can have descriptions $\gamma_X(x)$ and $\gamma_X(x')$ that are proper prefixes of each other.

In low complexity SISC design, we order alphabet $\mathcal{X}$ as $\mathcal{O} = \{x_1, x_2, \ldots, x_N\}$, where $N = |\mathcal{X}|$ and $i < j$ implies x_i precedes x_j in the chosen ordering. The *order-constrained SISC* for ordering $\mathcal{O}$ requires that symbols with identical description hold adjacent positions in $\mathcal{O}$, and groups of symbols whose descriptions are prefixes or siblings of each other hold adjacent positions in $\mathcal{O}$.

For a given ordering we use a dynamic programming approach to design the optimal order-constrained SISC such that $J = R + \lambda P_e$ is minimized. The complexity of this approach is $O(N^4)$. Finding the ordering that gives the globally optimal SISC is NP-hard. We therefore employ an iterative descent technique to search for a suboptimal ordering in polynomial time. Our experiments show that stopping the iterative descent technique after $O(N^2)$ orderings usually gives satisfactory performance. Thus the total complexity of the approach is $O(N^6)$.

3 ULSISC Design

Recall that in the ULSISC configuration, the sequence pair (X^n, Y^n) is distributed i.i.d. according to $p(x,y)$. The encoder sees X^n but not Y^n. The decoder sees Y^n but not X^n. Since $p(x,y)$ is unknown, we allow the decoder to send some feed-back to the encoder. The encoding/decoding process for (X^n, Y^n) follows.

The encoder uses a traditional universal code to transmit the first $m(n)$ symbols $X^{m(n)} = \{X_i\}_{i=1}^{m(n)}$ of X^n to the decoder; this description requires $L_m(X^{m(n)})$ bits. The decoder then uses the recovered $X^{m(n)}$ and known $Y^{m(n)}$ to calculate the empirical pmf $p_m(x,y) = \sum_{i=1}^{m(n)} 1((X_i, Y_i) = (x,y))/m(n)$; the decoder describes this pmf to the encoder using $\delta(m(n)) = |\mathcal{X}||\mathcal{Y}| \log(m(n)+1)$ bits. The encoder and decoder then calculate a new pmf $q(x,y)$ as a function of the estimate $p_m(x,y)$, and then independently design identical codes to match $q(x,y)$. The encoder encodes the remaining $n - m(n)$ symbols $X_{m(n)+1}^n = \{X_i\}_{i=m(n)+1}^n$ of X^n using this SISC; this description requires $L_{S,q}(X_{m(n)+1}^n)$ bits. The decoder reconstructs $X_{m(n)+1}^n$ with a number of errors $e_{U,q}(X_{m(n)+1}^n)$. The total number of bits transmitted in the above process is $L_U(X^n) = L_m(X^{m(n)}) + \delta(m(n)) + L_{S,q}(X_{m(n)+1}^n)$. The total number of errors is $e_U(X^n) = e_{U,q}(X_{m(n)+1}^n)$.

In [2], it is shown that there exist codes such that for any true underlying pmf $p(x,y)$, we can achieve

$$\begin{aligned} \lim_{n\to\infty} E_p L_U(X^n)/n &= \lim_{n\to\infty} E_p L_{S,q}(X_{m(n)+1}^n)/n \\ &= H(X|Y), \\ \lim_{n\to\infty} E_p e_U(X^n)/n &= \lim_{n\to\infty} E_p e_{U,q}(X_{m(n)+1}^n)/n \\ &= 0. \end{aligned}$$

This performance is achieved using an ensemble of random codes. To achieve similar performance in practice, we choose the estimate $q(x,y)$ as follows.

We wish to use an estimate $q(x,y)$ such that for the sequence $(X_{m(n)+1}^n, Y_{m(n)+1}^n)$, the SISC matched to $q(x,y)$ has the rate and error probability performance "close" to the rate and error probability performance achieved by the SISC matched to the true pmf $p(x,y)$. The empirical pmf $p_m(x,y)$ is a good candidate, since $p_m(x,y)$ is a good estimate of $p(x,y)$ based on $(X^{m(n)}, Y^{m(n)})$.

However, for some $(x_1, y_1) \in \mathcal{X} \times \mathcal{Y}$ such that $p(x_1,y_1) \neq 0$, it is possible that $p_m(x_1,y_1) = 0$ (e.g., if $p(x_1,y_1)$ is much smaller than $1/m(n)$, then it is very likely that the pair (x_1,y_1) will not appear in the sequence $(X^{m(n)}, Y^{m(n)})$). Therefore for two symbols $x, x' \in \mathcal{X}$, even if $p(x,y)p(x',y) > 0$ for some $y \in \mathcal{Y}$, an SISC matched to pmf $p_m(x,y)$ may still give x, x' descriptions $\gamma_X(x)$ and

$\gamma_X(x')$ that are proper prefixes of each other, and this will result in error propagation and catastrophic decoding failure.

To remedy this problem, we choose the estimate $q(x,y)$ as follows. Let Z be the number of zeros in the empirical pmf $p_m(x,y)$. Then

$$q(x,y) = \begin{cases} \epsilon_m & \text{if } p_m(x,y) = 0 \\ p_m(x,y)(1-\epsilon_m Z) & \text{otherwise} \end{cases}$$

where ϵ_m is as small as possible, e.g., we use $\epsilon_m = 10^{-6}$ in our experiments.

4 Experimental Results

The following experimental results show the performance of a code of the type described in Section 3 on two classes of randomly generated sources with $|\mathcal{X}| = |\mathcal{Y}| = 8$.

The first class of random sources assumes a uniform probability density function μ on the family of memoryless pmfs on $\mathcal{X} \times \mathcal{Y}$. A pmf $p(x,y)$ is generated by first independently choosing a random value from $[0,1)$ for each entry of $p(x,y)$ then normalizing.

The second class of random sources assumes a fixed percentage of zero entries in $p(x,y)$ and is generated by assigning zeros to a number of randomly chosen entries then choosing the remaining entries at random and normalizing.

In our experiments, we generate four groups of pmfs with 20 pmfs each. The first group comes from the first class of pmfs. The second, third and fourth groups are from the second class of pmfs and containing 25%, 50%, and 75% percent zero elements, respectively. The average entropies for these four groups of $p(x,y)$s are given in Table 1.

For each $p(x,y)$, we randomly generate 50 pairs of sequences $(X^M, Y^M)(M = 65536)$ i.i.d. according to $p(x,y)$. For each (x^M, y^M) pair, we design a sequence of independent ULSISCs with block length $n \leq M$. Thus for every length-n subsequence pair (x^n, y^n), we design a ULSISC for (x^n, y^n) as in Section 3. The performances are then averaged over the 50 pairs and are further averaged over the 20 pmfs in the same group. By varying $m(n)$, n, and λ (used in SISC design) we obtain curves of rates and error probabilities.

We compare these curves with the performance of an SISC designed for the true pmf $p(x,y)$ and the same λ used in ULSISC design. Again we average over the 50 pairs and over the 20 pmfs in the same group. We denote the average performances of ULSISC and SISC by $(R_U, P_{e,U}, J_U)$ and $(R_S, P_{e,S}, J_S)$ respectively.

For the low complexity SISC design, we use the multiple *descent neighbor* algorithm [6] allowing a total of C orderings to be searched. We use the same pseudo-random number generator at the encoder and the decoder to ensure that the decoder can independently perform the SISC design performed at the encoder.

We test on $\lambda = 2^i, 0 \leq i \leq 7$, $\lambda = 0$, and $\lambda = 10000$. As λ increases, rate R increases and error probability P_e decreases. By varying λ from zero to infinity, we trace out the curves for R against P_e. For the SISC curve, we can achieve $P_{e,S} = 0$ by setting $\lambda = \infty$. For the ULSISC curves, $P_{e,U} = 0$ is not always achievable. We focus on $P_e \in [0, 0.1]$ in the following analysis.

Table 1: Average statistics.

% zeros	0%	25%	50%	75%
$H(X)$	2.9684	2.9321	2.8668	2.7021
$H(X\|Y)$	2.7321	2.3462	1.9404	1.1699
Huffman Rate	2.9959	2.9663	2.9069	2.7518

Figure 1 shows the performance for $m(n) \in \{n/2, n^{0.9}, \sqrt{n}\log n, \sqrt{n}, \log n\}$. As expected, for $m(n)$ large (e.g., $m(n) = n/2$ or $n^{0.9}$), the estimate $q(x,y)$ is more accurate resulting in a smaller $P_{e,U}$ but more rate spent on sending these $m(n)$ symbols to the decoder. Similarly, for $m(n)$ small (e.g., $m(n) = \log n$), $P_{e,U}$ is larger but $R_{e,U}$ is smaller. We observe that $m(n) = \sqrt{n}\log n$ and $m(n) = \sqrt{n}$ achieve performance very close to that of the SISC in these experiments. (Note that $m(n) = \log n$ results in large P_e and does not appear in the figure.)

Figure 2 compares the curves for $m(n) = \sqrt{n}\log n$ as n increases. As expected, as n increases, the ULSISC performance approaches the SISC performance. When $n = M = 65536$, the observed performances are very close (but not identical).

We next investigate the trade-off between SISC design complexity and ULSISC performance. As shown in Figure 3 by increasing the design complexity of SISC from $C = N$ to $C = N^3$, the performance of ULSISC improves.

Finally, we compare the performance for the four groups of pmfs. For the first group, $p(x,y)$ contains almost no zero value elements and $H(X|Y)$ is very close to $H(X)$ (see Table 1). Thus the benefits of ULSISC is not obvious, since the theoretically achievable rate $H(X|Y)$ of ULSISC does not differ much from the theoretically achievable rate $H(X)$ of traditional universal codes. As the number of zeros in $p(x,y)$ increases, the difference between $H(X|Y)$ and $H(X)$ also increases, thus the benefit of ULSISC over traditional universal code amplifies. As shown in Figure 2, when $P_e \in [0, 0.1]$, the achievable rate of ULSISC ranges in $[1, 2.25]$ for $p(x,y)$'s with 75% zeros and increases to $[2.25, 3.25]$ for $p(x,y)$'s with 25% zeros. (Here, a rate greater than $\log|\mathcal{X}|$ is caused by the nonneglible number of bits $\delta(m(n))$ that is needed by the decoder to transmit the estimate $q(x,y)$ to the encoder when the block length n is not large enough.)

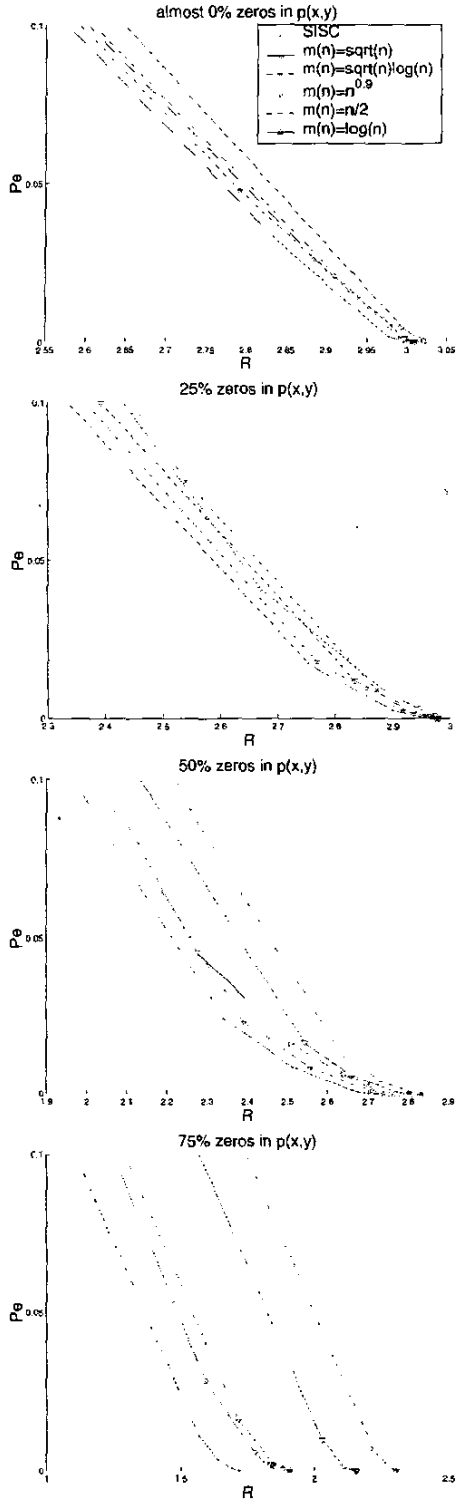

Figure 1: Observe $P_{e,U} - R_U$ curves for various $m(n)$ ($n = 65536$, $C = 512$).

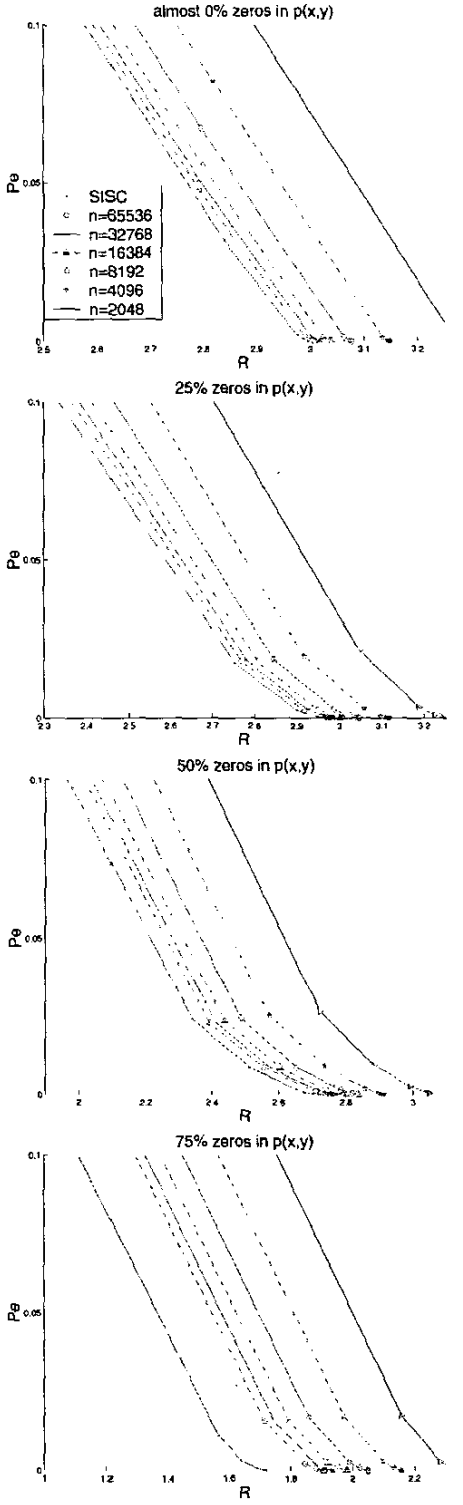

Figure 2: Observe $P_{e,U} - R_U$ curves for various n ($m(n) = \sqrt{n}\log n, C = 512$).

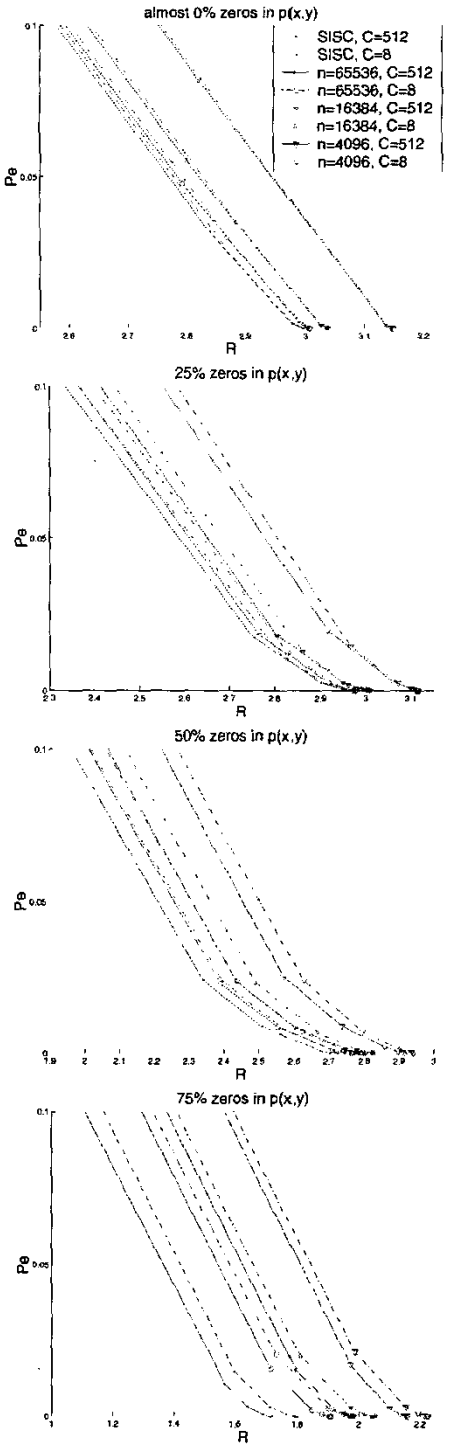

Figure 3: Observe $P_{e,U} - R_U$ curves for various C ($m(n) = \sqrt{n}\log n$).

5 Summary

In this paper, we present a low complexity ULSISC design algorithm. Our experimental results give guidelines for choosing appropriate parameters used in these algorithms.

References

[1] D.Slepian and J.K.Wolf. Noiseless coding of correlated information sources. *IEEE Transactions on Information Theory*, IT-19(4):471–480, July 1973.

[2] S. Jaggi and M. Effros. Universal linked multiple access source codes. In *Proc. of the IEEE International Symposium on Information Theory*, Lausanne, Switzerland, June 2002.

[3] S. Jaggi and M. Effros. Universal linked multiple access source codes. *IEEE Transactions on Information Theory*. Submitted 2002. In review.

[4] P. Koulgi, E. Tuncel, S. Regunathan, and K. Rose. Minimum redundancy zero-error source coding with side information. In *Proc. of the IEEE International Symposium on Information Theory*, Washington DC, USA, June 2001. IEEE.

[5] Q. Zhao and M. Effros. Lossless and near lossless source coding for multiple access networks. *IEEE Transactions on Information Theory*, January 2003. To Appear.

[6] Q. Zhao and M. Effros. Low complexity code design for lossless and near lossless side information source codes. Submitted to Data Compression Conference 2003.