Verification Codes for Deletions

Michael Mitzenmacher[1]
Harvard University
e-mail: michaelm@eecs.harvard.edu

I. Introduction

We demonstrate new deletion codes based on low-density parity-check (LDPC) codes using the *verification code* framework, introduced in [1]. (Similar ideas were suggested by Metzner [5].) Verification codes are designed to work with data organized as packets; packets could be thousands of bits (Internet packets) or simply represent a thirty-two bit integer. Deletions are assumed to occur at the packet level. Verification codes for deletions (VCDs) have polynomial time encoding and decoding algorithms and can be analyzed asymptotically using standard LDPC techniques. Variations of VCDs are also robust against transpositions, random insertions, and random substitution errors.

II. The LDPC Framework

We use the now standard framework for low-density parity-check codes, following [4]. LDPC codes are represented by bipartite graphs. The n *variable nodes* correspond to symbols in the codeword; the m *check nodes* correspond to constraints on the adjacent variable nodes; and the design rate R is given by $R = \frac{n-m}{n}$. Symbols are interpreted as numbers modulo q, with constraints being on the sum of the variable nodes modulo q. With irregular codes, introduced in [3, 4], the fraction of edges connected to variable nodes of degree i (with $2 \le i \le d_v$) is λ_i , and similarly the fraction of edges connected to check nodes of degree i (with $2 \le i \le d_c$) is ρ_i. We define the polynomials $\lambda(x) := \sum_{i=2}^{d_v} \lambda_i x^{i-1}$ and $\rho(x) := \sum_{i=2}^{d_c} \rho_i x^{i-1}$. Following standard LDPC techniques, $\lambda(x)$ and $\rho(x)$ will determine the asymptotic behavior of the codes we describe; this is proven formally using martingale arguments [2, 3].

Without loss of generality we assume henceforth that the constraints are such that the sum of the symbols associated with the variable nodes adjacent to the ith check node is i. This is a useful deviation from the standard LDPC approach. Linear time encoding schemes still apply.

III. Random errors and random deletions

We describe the case where data being sent is random and each packet is deleted with constant probability p, independent of other packets.

We describe our algorithm PlaceNodes for the case of regular graphs (with degrees d_v and d_c); it generalizes trivially. All nodes begin as *unverified*; unverified nodes do not have a fixed fixed final value. When a node is *verified*, it is given a fixed value for the remainder of the algorithm. With high probability, the algorithm never assigns a verified node an incorrect value.

If a out of n node values arrive, the decoding algorithm considers all $\binom{a}{d_c}$ combinations of d_c nodes that arrived and applies the following rules in any order as much as possible:

1. If there exists a combination of d_c variable nodes whose sum equals i, then associate these nodes with the neighbors of the ith check node.
2. If all but one of the neighbors of the ith check node is verified, it becomes verified, with its final value being set to satisfy the appropriate constraint.

Check nodes may *verify* the appropriate position of a collection of nodes, according to first rule. Concretely, we have the following lemma:

Lemma 1 *The probability that any set of d_c variable nodes not associated with the ith check node sum to i is at most $1/q$. Hence the probability of a false verification at any point in the decoding process is at most $\binom{n}{d_c}/q$.*

For example, if symbols are b bits, then $q = 2^b$; hence the packet size b needs to be only $\Omega(\log n)$ bits so that the probability of failure due to a false verification is polynomially small in n. The other role of a check node is to correct a neighboring variable node that was received incorrectly, according to Rule 2. A check node can correct a neighbor after all other neighbors have been verified and therefore are known to have the correct value with high probability.

The algorithm above is easily modified to a message-passing algorithm, which allows us to apply standard analysis techniques. More details are available in the submission version at the author's web site. Results include the following theorem:

Theorem 1 *For any rate R, with $0 < R < 1$, and a given $\epsilon > 0$, there exists a family of verification deletion codes of rate R that correct a $1 - \frac{R}{2} - \frac{\sqrt{4R-3R^2}}{2} - \epsilon$ fraction of deletions with high probability.*

As a perhaps more specific example, a 3-6 regular LDPC code yields a VCD code of rate $1/2$ that runs in $O(n^6)$ time and can handle over 13% of the symbols being deleted.

References

[1] M. Luby and M. Mitzenmacher. Verification Codes. In *Proc. of the 40th Annual Allerton Conference*, 2002.

[2] M. Luby, M. Mitzenmacher, and M. A. Shokrollahi. Analysis of Random Processes via And-Or Tree Evaluation. In *Proceedings of the 9th Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 364-373, 1998.

[3] M. Luby, M. Mitzenmacher, M. A. Shokrollahi, and D. Spielman. Efficient Erasure Correcting Codes. *IEEE Transactions on Information Theory*, 47(2), pp. 569-584, 2001.

[4] M. Luby, M. Mitzenmacher, M. A. Shokrollahi, and D. Spielman. Improved Low-Density Parity-Check Codes Using Irregular Graphs. *IEEE Transactions on Information Theory*, 47(2), pp. 585-598, 2001.

[5] J. J. Metzner. Majority-Logic-Like Decoding of Vector Symbols. *IEEE Transactions on Communications*, 44(10):1227-1230, 1996.

[1]Supported in part by an Alfred P. Sloan Research Fellowship and NSF grants CCR-9983832, CCR-0118701, and CCR-0121154.