Eighth International Workshop on Optimal Codes and Related Topics
July 10-14, 2017, Sofia, Bulgaria *pp. 110-115*

On the success probability of decoding (partial) unit memory codes[1]

Sven Puchinger sven.puchinger@uni-ulm.de
Sven Müelich sven.mueelich@uni-ulm.de
Martin Bossert martin.bossert@uni-ulm.de
Institute of Communications Engineering, University of Ulm, Germany

Abstract. In this paper, we derive analytic expressions for the success probability of decoding (Partial) Unit Memory codes in memoryless channels. An applications of this result is that these codes outperform individual block codes in certain channels.

1 Introduction

(Partial) Unit Memory ((P)UM) codes, introduced in [6] and [5], are convolutional codes, defined using block codes. Several (P)UM code constructions and a decoder based on the underlying block codes were proposed in [1–3]. Since these publications, there have been results on improving the decoding algorithm [9], extension to rank-metric codes [10], and applications to random linear network coding [7] and the streaming scenario [4]. In these applications, the codes were evaluated numerically in probabilistic channels and it was observed that (P)UM codes often outperform individual block codes in these scenarios.

In this paper, we derive analytic expressions for the probability of successfully recovering an information block that is encoded with a (P)UM code, in memoryless channels. Using these new expressions, we are able to partly explain the numerical observations in [7] and [4] analytically.

2 (Partial) Unit Memory Codes

We use the description of (P)UM codes as in [1]. Let $k \leq n$ and $k_1 \leq \min\{k, n-k\}$ be non-negative integers. We choose matrices $\boldsymbol{G}_0$ and $\boldsymbol{G}_1$ of the form

$$\boldsymbol{G}_0 = \begin{bmatrix} \boldsymbol{G}_0^* \\ \boldsymbol{G}_0^\diamond \end{bmatrix}, \quad \boldsymbol{G}_1 = \begin{bmatrix} \boldsymbol{G}_1^* \\ \boldsymbol{0} \end{bmatrix},$$

where the row spaces of the three matrices $\boldsymbol{G}_0^*, \boldsymbol{G}_1^* \in \mathbb{F}^{k_1 \times n}$ and $\boldsymbol{G}_0^\diamond \in \mathbb{F}^{k-k_1 \times n}$ pairwise intersect only in the zero codeword. Let $\langle \boldsymbol{G} \rangle$ denote the row space of

[1] Due to space limitation, some details are moved to an extended version [8].

a matrix $\boldsymbol{G}$. We define the following codes:

$$\mathcal{C}_\alpha := \left\langle \begin{bmatrix} \boldsymbol{G}_0^* \\ \boldsymbol{G}_0^\diamond \\ \boldsymbol{G}_1^* \end{bmatrix} \right\rangle, \quad \mathcal{C}_0 := \left\langle \begin{bmatrix} \boldsymbol{G}_0^* \\ \boldsymbol{G}_0^\diamond \end{bmatrix} \right\rangle, \quad \mathcal{C}_1 := \left\langle \begin{bmatrix} \boldsymbol{G}_0^\diamond \\ \boldsymbol{G}_1^* \end{bmatrix} \right\rangle, \quad \mathcal{C}_{01} := \langle \boldsymbol{G}_0^\diamond \rangle \,.$$

The codes have parameters $\mathcal{C}_\alpha(n, k+k_1)$, $\mathcal{C}_0(n,k)$, $\mathcal{C}_1(n,k)$, and $\mathcal{C}_{01}(n, k-k_1)$.

2.1 Encoding

Given the generator matrices $\boldsymbol{G}_0$ and $\boldsymbol{G}_1$, we encode a sequence of information vectors $\boldsymbol{i}_t \in \mathbb{F}^k$ ($t = 0, \ldots, L$, where we choose $\boldsymbol{i}_0 = \boldsymbol{i}_L = \boldsymbol{0}$) into a code sequence

$$\boldsymbol{c}_t = \boldsymbol{i}_t \cdot \boldsymbol{G}_0 + \boldsymbol{i}_{t-1} \cdot \boldsymbol{G}_1 \quad \text{for } t = 1, \ldots, L.$$

Note that we can re-write this relation into

$$\boldsymbol{c}_t = \boldsymbol{i}_t^* \cdot \boldsymbol{G}_0^* + \boldsymbol{i}_t^\diamond \cdot \boldsymbol{G}_0^\diamond + \boldsymbol{i}_{t-1}^* \cdot \boldsymbol{G}_1^*. \tag{1}$$

The information vectors and codewords corresponding to an index t are called t-th *block*. If $k < k_1$, the resulting code is called $(n, k|k_1)$ *partial unit memory code*. If $k = k_1$, the code is called (n,k) *unit memory* code. In the latter case, $\boldsymbol{i}_{t-1}$ can be completely recovered by knowing $\boldsymbol{c}_t$ or $\boldsymbol{c}_{t-1}$.

2.2 Decoding

The sequence of received words is of the form $\boldsymbol{r}_t = \boldsymbol{c}_t + \boldsymbol{e}_t$ for $t = 1, \ldots, L$. We choose some[2] metric $\mathrm{d}(\cdot,\cdot) : \mathbb{F}^n \times \mathbb{F}^n \to \mathbb{R}_{\geq 0}$, and the corresponding weight $\mathrm{wt}(\cdot) = \mathrm{d}(\cdot, \boldsymbol{0})$, for which we know decoders of the codes $\mathcal{C}_\alpha, \mathcal{C}_0, \mathcal{C}_1, \mathcal{C}_{01}$ that can find all codewords with distance to the received word at most $\tau_\alpha, \tau_0, \tau_1, \tau_{01}$, respectively. We assume that $\tau_\alpha < \tau_0 = \tau_1 < \tau_{01}$ in this paper.[3] For notational convenience, we say that t *errors occurred* if the error word has weight t.

We use the description of decoding as in [3]. There, the Hamming metric in combination with bounded-minimum-distance decoders was used. However, the decoder also works with list decoders in the Hamming metric [9], with rank-metric PUM codes [10], or with erasures [4]. First, candidates for the codewords $\boldsymbol{c}_t$ are found in 4 steps (see below). Afterwards, the most likely sequence $\boldsymbol{c}_1, \ldots, \boldsymbol{c}_L$ is found among these candidates using the Viterbi algorithm. In this paper, we say that decoding is successful at the t-th position if the sent codeword $\boldsymbol{c}_t$ is among the candidates. Finding the candidates works in 4 steps:

1. Each received word $\boldsymbol{r}_t = \boldsymbol{c}_t + \boldsymbol{e}_t$ is decoded independently using the decoder of $\mathcal{C}_\alpha$ (note $\boldsymbol{c}_t \in \mathcal{C}_\alpha$). We can decode up to τ_α errors in this step.

[2] This can e.g. be the Hamming metric as in [2] and [4], or the rank metric as in [10].

[3] This is not a major restriction since most known PUM constructions, e.g. based on Reed–Solomon, BCH [2], or Gabidulin codes [10], provide codes $\mathcal{C}_0, \mathcal{C}_1$ of the same minimum distance.

2. Using the information fragment $\boldsymbol{i}^*_{t-1}$ given by a candidate codeword $\boldsymbol{c}_{t-1}$, we can successfully decode the right neighbor $\boldsymbol{c}_t$ in the code $\mathcal{C}_0$ if $\mathrm{wt}(\boldsymbol{e}_t) \leq \tau_0$, using the following relation (note that the left-hand side is known)
$$\boldsymbol{r}_t - \boldsymbol{i}^*_{t-1} \cdot \boldsymbol{G}^*_1 = \underbrace{\boldsymbol{i}^*_t \cdot \boldsymbol{G}^*_0 + \boldsymbol{i}^\diamond_t \cdot \boldsymbol{G}^\diamond_0}_{\in \mathcal{C}_0} + \boldsymbol{e}_t.$$
We can repeat this so-called *forward* step iteratively for all candidates.

3. Similar to Step 2, we can go in *backward* direction by decoding
$$\boldsymbol{r}_t - \boldsymbol{i}^*_t \cdot \boldsymbol{G}^*_0 = \underbrace{\boldsymbol{i}^\diamond_t \cdot \boldsymbol{G}^\diamond_0 + \boldsymbol{i}^*_{t-1} \cdot \boldsymbol{G}^*_1}_{\in \mathcal{C}_1} + \boldsymbol{e}_t,$$
in the code $\mathcal{C}_1$, which is successful if the number of errors is at most τ_1.

4. Using $\mathcal{C}_{01}$, we can find the $\boldsymbol{c}_t$ in positions t, where both neighbor blocks $t-1$ and $t+1$ have been successfully decoded and $\mathrm{wt}(\boldsymbol{e}_t) \leq \tau_{01}$, using
$$\boldsymbol{r}_t - \boldsymbol{i}^*_t \cdot \boldsymbol{G}^*_0 - \boldsymbol{i}^*_{t-1} \cdot \boldsymbol{G}^*_1 = \underbrace{\boldsymbol{i}^\diamond_t \cdot \boldsymbol{G}^\diamond_0}_{\in \mathcal{C}_{01}} + \boldsymbol{e}_t,$$

3 New Expressions for the Success Probability

Let the PUM code and constituent decoders with decoding radii $\tau_\alpha, \tau_0, \tau_1, \tau_{01}$ be given. We assume that the error words $\boldsymbol{e}_t$ are drawn i.i.d. at random according to an arbitrary distribution (memoryless channel). Let $X_1, \ldots, X_L$ be the random variables describing the error weight, i.e., $X_t := \mathrm{wt}(\boldsymbol{e}_t)$. Thus, the X_t are also independently and identically distributed as some random variable X.

In the following, we derive an expression for the probability P_t that the t-th information word $\boldsymbol{i}_t$ of the PUM code is successfully recovered (i.e., among the candidates), only depending on the distribution of X and the position t. The expression depends on the probabilities

$$p_a := \mathrm{P}(0 \leq X \leq \tau_\alpha), \qquad p_b := \mathrm{P}(\tau_\alpha < X \leq \tau_0),$$
$$p_c := \mathrm{P}(\tau_0 < X \leq \tau_{01}), \qquad p_d := \mathrm{P}(\tau_{01} < X).$$

Note that $p_a + p_b + p_c + p_d = 1$. Let Q_t denote the probability that the t-th block is correctly decoded by Step 1 or 2 (individually or in forward direction). Similarly, by R_t we define the probability that it is found by Step 1 or 3.

Lemma 1. *For all $t = 1, \ldots, L$, we have*
$$Q_t = \tfrac{p_a}{1-p_b} + p_b^t \cdot \left(\tfrac{1-p_a-p_b}{1-p_b}\right) \text{ and } R_t = \tfrac{p_a}{1-p_b} + p_b^{L-t+1} \cdot \left(\tfrac{1-p_a-p_b}{1-p_b}\right).$$

Proof. We prove the claim by induction. Since the information word $\boldsymbol{i}_0 = \boldsymbol{0}$ is known, we can directly decode the first codeword $\boldsymbol{c}_1$ in $\mathcal{C}_0$ and obtain
$$Q_1 = \mathrm{P}(X_1 \leq \tau_0) = p_a + p_b = \tfrac{p_a + p_b - p_a p_b - p_b^2}{1-p_b} = \tfrac{p_a}{1-p_b} + p_b^1 \cdot \left(\tfrac{1-p_a-p_b}{1-p_b}\right).$$

The probability that the t-th block is found in forward direction is given by the sum of the probability that it is found individually and the probability that Step 1 fails, but it is successfully recovered in forward direction, i.e.,

$$\begin{aligned} Q_t &= \mathrm{P}(X_t \le \tau_\alpha) + \mathrm{P}(\tau_\alpha < X_t \le \tau_0) \cdot Q_{t-1} \\ &= p_a + p_b \cdot \left(\tfrac{p_a}{1-p_b} + p_b^{t-1} \cdot \left(\tfrac{1-p_a-p_b}{1-p_b} \right) \right) = \tfrac{p_a}{1-p_b} + p_b^t \cdot \left(\tfrac{1-p_a-p_b}{1-p_b} \right). \end{aligned}$$

The proof of for R_t is equivalent using the base case $R_L = p_a + p_b$. □

3.1 Partial Unit Memory Codes ($k_1 < k$)

In the case of PUM codes, the correct information vector $\boldsymbol{i}_t$ is found if and only if $\boldsymbol{c}_t$ is found. Hence, we can state the following result.

Theorem 1. *For any $t = 1, \dots, L$, we have*

$$P_t = p_a + \tfrac{p_a}{(1-p_b)^2} \left[p_b(2 - p_a - 2p_b) + p_a p_c \right] + \varepsilon(t, L),$$

where $\varepsilon(t, L) \ge 0$ and $\varepsilon(t, L) \in O(\max\{p_b^t, p_b^{L-t}\})$ (i.e., the term $\varepsilon(t, L)$ is negligible if t is sufficiently far away from 0 and L).

Proof. We can write

$$\begin{aligned} P_t &= \mathrm{P}(\boldsymbol{i}_t \text{ is found}) = \mathrm{P}(\boldsymbol{c}_t \text{ is found}) \\ &= \underbrace{\mathrm{P}(X_t \le \tau_\alpha)}_{\text{found in Step 1}} + \underbrace{\mathrm{P}(\tau_\alpha < X_t \le \tau_0) \cdot (Q_{t-1} + R_{t+1} - Q_{t-1}R_{t+1})}_{\text{found only in forward or backward direction}} \\ &\quad + \underbrace{\mathrm{P}(\tau_0 < X_t \le \tau_{01}) \cdot Q_{t-1} \cdot R_{t+1}}_{\text{found in Step 4}} \\ &= p_a + p_b(Q_{t-1} + R_{t+1} - Q_{t-1}R_{t+1}) + p_c Q_{t-1} R_{t+1}. \end{aligned}$$

Let $A := \frac{p_a}{1-p_b}$, $B := \frac{1-p_a-p_b}{1-p_b} p_b^{t-1}$, and $C := \frac{1-p_a-p_b}{1-p_b} p_b^{L-t}$. Then, $A, B, C \ge 0$ and $B \in O(p_b^t)$ and $C \in O(p_b^{L-t})$. Also, $Q_{t-1} = A + B$ and $R_{t+1} = A + C$, so

$$\begin{aligned} P_t &= p_a + p_b(A + B + A + C - (A + B)(A + C)) + p_c(A + B)(A + C) \\ &= p_a + \tfrac{p_a}{(1-p_b)^2} \left[p_b(2 - p_a - 2p_b) + p_a p_c \right] + \varepsilon(t, L), \text{ where} \\ \varepsilon(t, L) &= p_b \underbrace{(B + C - AB - AC - BC)}_{\ge (A+B)(B+C) - A(B+C) - BC = B^2 \ge 0} + p_c(AB + AC + BC) \ge 0. \end{aligned}$$

Since all terms depend on B or C, we have $\varepsilon(t, L) \in O(\max\{p_b^t, p_b^{L-t}\})$. □

3.2 Unit Memory Codes ($k = k_1$)

Unit memory codes have the advantage that we can obtain $\boldsymbol{i}_t$ from either $\boldsymbol{c}_t$ or $\boldsymbol{c}_{t+1}$. In UM codes, $\mathcal{C}_{01}$ has dimension $k - k_1 = 0$, and hence, Step 4 is not useful. On the other hand, we can define $\tau_{01} := \infty$, so $p_d = 0$ and $p_c = 1 - p_a - p_b$.

Theorem 2. *For any t, there is a $\delta(t,L) \geq 0$ with $\delta(t,L) \in O(\max\{p_b^t, p_b^{L-t}\})$:*

$$P_t = 1 - \left(\tfrac{p_c}{1-p_b}\right)^2 + \delta(t,L).$$

Proof. We define A, B, C as in the proof of Theorem 1. Then, we can write

$$\begin{aligned} P_t &= Q_t + R_{t+1} - Q_t R_{t+1} = A + B + A + C + (A+B)(A+C) \\ &= A(2-A) + \underbrace{B + C - A(B+C) - BC}_{=:\,\delta(t,L)} = 1 - \left(\tfrac{p_c}{1-p_b}\right)^2 + \delta(t,L), \end{aligned}$$

where $\delta(t,L)$ has the desired properties. □

4 Applications

4.1 Fast Code Design

Based on the results in Section 3, it is possible to determine the failure probability of decoding a PUM code block only from the probability density function (pdf) of the error weight in a block (which is given by the channel and the block length n). Hence, as soon as this pdf is determined (either theoretically or numerically), one can compute the decoding failure probability for any code parameter set, i.e., variations of k, k_1, τ_α, τ_0, τ_1, and τ_{01}, without the need for computationally expensive Monte-Carlo simulations (a numerical example is included in the extended version [8]). This allows to optimize code parameters (e.g. k_1 for given k) quickly.

4.2 (P)UM Codes vs. Independent Block Codes

Let $\mathcal{C}(n,k)$ be a linear block code with the same rate as the PUM code. For a fair comparison, we assume that decoding in $\mathcal{C}$ is possible up to τ_0 (i.e., as the decoding radius of the code $\mathcal{C}_0$ in the PUM coding scheme). Consider a channel in which a position independently adds 1 to the error weight of the block with probability p (i.e., the error weight is binomially distributed with parameters n and p). Using our results from Section 3, it is possible to show that for $p \to 0$, the failure probability of decoding (P)UM codes gets below the one of encoding/decoding each information block independently in $\mathcal{C}$ (see the extended version [8] for a formal statement and proof). This result proves that (P)UM codes can outperform independent block codes in some scenarios, which was observed experimentally before (cf. [4]).

4.3 Rank-Metric (P)UM Codes in Network Coding

In [10] and [7], (P)UM codes in the rank metric were used for error correction in variants of random linear network coding. It was observed numerically that

(P)UM codes result in lower failure probabilities compared to independent rank-metric codes in this scenario. The results in this paper might provide a basis for an analytical explanation of this observation. Although the channel model in [7] is quite complex, it is reasonable that bounds on the tail probabilities of the error weight can be derived, resulting in similar results as in Section 4.2.

5 Conclusion

In this paper, we have derived analytic expressions for the success probability of (P)UM codes in memoryless channels and have shown applications for them. Besides the already mentioned future work, the results should be generalized—if possible—to certain channels with memory (e.g. burst channels).

Acknowledgement: This work was supported by the German Research Foundation (DFG), grant BO 867/29-3.

References

[1] U. Dettmar, Partial Unit Memory Codes, *diss.*, TU Darmstadt, 1994.

[2] U. Dettmar and U. Sorger, New Optimal Partial Unit Memory Codes based on Extended BCH Codes, *Electronics Letters,* **29**, 2024–2025, 1993.

[3] U. Dettmar and U. Sorger, Bounded Minimum Distance Decoding of Unit Memory Codes, *IEEE Trans. Inf. Theory,* **41**, 591–596, 1995.

[4] M. Kuijper and M. Bossert, On (Partial) Unit Memory Codes based on Reed-Solomon Codes for Streaming, in *IEEE ISIT,* 2016, 920-924.

[5] G. Lauer, Some optimal partial-unit-memory codes (Corresp.), *IEEE Trans. Inf. Theory,* **25**, 240–243, 1979.

[6] L. Lee, Short Unit-Memory Byte-oriented Binary Convolutional Codes having maximal free distance (Corresp.), *IEEE Trans. Inf. Theory,* **22**, 349–352, 1976.

[7] S. Puchinger, M. Cyran, R. F. H. Fischer, M. Bossert, J. B. Huber, Error Correction for Differential Linear Network Coding in Slowly-Varying Networks, in *ITG SCC* 2015, 1-6.

[8] Extended version of this paper, *arXiv preprint, arXiv:1705.08652.*

[9] S. Puchinger, A. Wachter-Zeh and M. Bossert, Improved Decoding of Partial Unit Memory Codes Using List Decoding of Reed-Solomon Codes, in *IZS,* 2014, 87-90.

[10] A. Wachter-Zeh, M. Stinner and V. Sidorenko, Convolutional Codes in Rank Metric with Application to Random Network Coding, *IEEE Trans. Inf. Theory,* **61**, 3199–3213, 2015.