Short Blocklength Distribution Matching by Linear Programming

Shuangyu Dong[(1)], Honglin Ji[(1)], Zhaopeng Xu[(1)], Jingge Zhu[(1)], William Shieh[(1)]

[(1)] Department of Electrical and Electronic Engineering, The University of Melbourne, VIC 3010, Australia
shuangyud@student.unimelb.edu.au

Abstract *A low-complexity distribution matching algorithm that aims to achieve high information rate for short blocklength probabilistic shaping by linear programming is proposed. At AIR of 4 bits/symbol and SNR of 11.87 dB, the distribution matching blocklength is shortened by 256 times compared with CCDM.*

Introduction

Probabilistic constellation shaping has attracted extensive research interests in optical communications for its energy efficiency gains and fine-grained rate adaptability. It is practically enabled by a scheme called probabilistic amplitude shaping (PAS)[1] which elegantly combines forward error correction (FEC) and probabilistic shaping. Distribution matching (DM), a key element of probabilistic shaping, maps uniform input bits into capacity-achieving output symbols. Constant composition distribution matching (CCDM)[2] is a well-known DM algorithm. It can achieve channel capacity when DM output symbol blocklength goes to infinity. However, due to the rate loss caused by the feature of constant composition, its performance declines for short DM blocklengths. Besides, its implementation complexity increases for long DM blocklengths. Recently, probabilistic shaping has been investigated for short-reach links, such as data center interconnects and 5G fronthaul networks, which require low power consumption, low complexity and minimum latency[3]. To meet these requirements, short blocklength DM with high performance is desirable. Algorithms were proposed to reduce DM blocklengths[4]–[6] while maintaining a high performance. They are all fixed-to-fixed (F2F) DM. A variable-to-fixed (V2F) DM algorithm was implemented by Geometric Huffman Coding (GHC)[7] in 2011, which was developed from the well-known algorithm, Huffman Coding, in data compression.

In this paper, we propose a novel distribution matching by linear programming (DMLP) algorithm which significantly reduces the implementation complexity while maintaining high performance. It achieves high information rate for short blocklength probabilistic shaping by using linear programming to minimize the normalized information divergence between the targeted capacity-achieving distribution and the empirical distribution of channel input symbols. Compared with GHC[7], DMLP is a more general method that can be used for both F2F and V2F DM. At an achievable information rate (AIR) of 4 bits/symbol, and SNR of 11.87 dB, the blocklength of V2F DMLP is shortened by 256 times compared with CCDM. This outperforms the prior F2F DM algorithms with blocklengths 3, 5.5 and 13 times shorter than CCDM in references[4]–[6] respectively.

978-1-6654-3868-1/21/$31.00 ©2021 IEEE

Principle of DMLP

Linear programming is an optimization problem in which the objective and all constraint functions are linear[8]. The purpose of DM is to map uniform input bits B into capacity achieving output symbols A. After source encoding, information bits are independent and uniformly distributed. Conventionally without DM, channel input symbols X also have a near uniform distribution after linear channel coding. Channel capacity is the maximum mutual information of channel input and output, taken over all possible channel input symbol distributions[9]. To reach the channel capacity, DM is needed to shape channel input symbols into the targeted capacity achieving distribution.

The architecture of DMLP is shown in Fig.1. F2F DMLP maps fixed-length bit sequences B^m of length m to fixed-length symbol codewrods A^n of length n. While V2F DMLP maps variable-length bit sequences B^*, with the length ranging from $m_{\min}$ to $m_{\max}$, to fixed-length A^n. The output dimension of DM (n) is called DM blocklength. Symbol A is treated as a random variable on the alphabet $\mathcal{A} = \{a_1, ..., a_{|\mathcal{A}|}\}$, where $|\mathcal{A}|$ stands for set dimension. In Fig.1, we take $n = 4$ and $|\mathcal{A}|$ = 4 as an example.

As is shown in Fig.1 (a) and (b), both F2F and

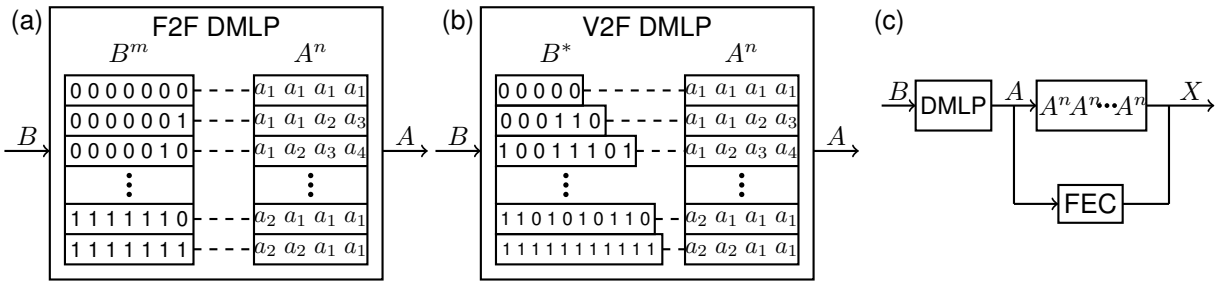

Fig. 1: The architecture of DMLP: (a) fixed-to-fixed DMLP; (b) variable-to-fixed DMLP; (c) DM output concatenation for FEC.

Tab. 1: Optimization Problem Statement for F2F and V2F DMLP

Optimization	**Fixed to Fixed**	**Variable to Fixed**
Objective Function	$\min D_n = -\mathbb{E}[R] - \sum_{j=1}^{\lvert\mathcal{A}\rvert} P_{\overline{A}}(a_j)\log_2 P_A(a_j)$ (1)	
$\mathbb{E}[R]$	$\mathbb{E}[R] = \frac{m}{n}$ (2)	$\mathbb{E}[R] = \sum_m 2^{-m}\frac{m}{n}\sum_i x_i[m]$ (3)
$P_{\overline{A}}(a_j)$	$P_{\overline{A}}(a_j) = \frac{\sum_i 2^{-m}x_i T_i[j]}{n}$ (4)	$P_{\overline{A}}(a_j) = \frac{\sum_i\sum_m 2^{-m}x_i[m]T_i[j]}{n}$ (5)
Constraints	$x_i \le \mu_i, \forall i$ (6)	$\sum_m x_i[m] \le \mu_i, \forall i$ (7)
	$\sum_i 2^{-m}x_i = 1$ (8)	$\sum_m 2^{-m}\sum_i x_i[m] = 1$ (9)

V2F DMLP are implemented with look up table (LUT) drawn inside the DMLP square blocks. The left side of the LUT containing bit sequences is called "sequence codebook". Similarly, the right side is called "symbol codebook". The dashed lines show the mapping between B^m (or B^*) and A^n. F2F DMLP is simply implementable by chunking input bit streams into length m sequences and then following the LUT. As for V2F DMLP, the implementation takes two steps. The first step is to partition input bit stream into sequences that are the same as B^* in the sequence codebook. To avoid confusion during the partition, B^* sequence needs to be prefix-free[10]. It can be shown that any input bit stream can be partitioned into the pre-designed B^* sequence. V2F DMLP first checks whether input bit stream of length $m_{\min}$ exists in the sequence codebook by searching all B^* of length $m_{\min}$. If there is no B^* of length $m_{\min}$ the same as the input stream, it will include one more bit of the input stream and continue to search B^* of length $m_{\min}+1$. The sequence codeword searching stops when the input bit stream is the same as B^* in the sequence codebook. The second step of V2F DMLP implementation is simply finding the corresponding A^n in the LUT. To combine DMLP with FEC following PAS scheme[1], a concatenation of DM output symbol blocks is needed as is shown in Fig.1(c).

Information divergence measures the difference between two distributions in information theory[9]. The normalized information divergence between $P_{\tilde{A}^n}$, the empirical codeword distribution generated by DM, and P_A^n, the target codeword distribution, is given by $D_n = \frac{\mathbb{D}(P_{\tilde{A}^n}\|P_A^n)}{n}$[2]. The key of DMLP is to design the LUT such that D_n is minimized. Conventionally without DM, the original symbol codebook contains $|\mathcal{A}|^n$ symbol codewords. Intuitively, for F2F DMLP, D_n can be reduced by removing some A^n from the original codebook so that the remaining symbol codewords have a distribution closer to the target. For V2F DMLP, the probability of each A^n is weighted by the length of its matched B^*. Intuitively, A^n that helps to reduce D_n can be assigned a shorter B^*.

The LUTs are generated for F2F and V2F DM by solving an optimization problem stated in Tab.1. $P_{\overline{A}}$ and P_A are empirical and target distributions of symbol A respectively. R is the DM code rate. We can show that the objective function to minimize D_n can be further derived into Eq. (1). Since P_A is fixed, Eq. (1) is a linear function of $\mathbb{E}[R]$ and $P_{\overline{A}}$. The optimization problem can then be solved by a linear programming (LP) solver. The calculations of $\mathbb{E}[R]$ and $P_{\overline{A}}$ (Eq. (2-5)) are different between F2F and V2F DMLP. This is mainly because $P_{\tilde{A}^n}$ is the same for all A^n in F2F DMLP, whereas it has different values in V2F DMLP as it depends on the lengths of B^*.

Type is a terminology in information theory[9]. Here a type of a symbol codeword A^n means the composition of A^n. For example, a combination of $3a_1, 1a_2$, and $2a_3$ is one type for A^6. By classifying codewords A^n according to their types, the

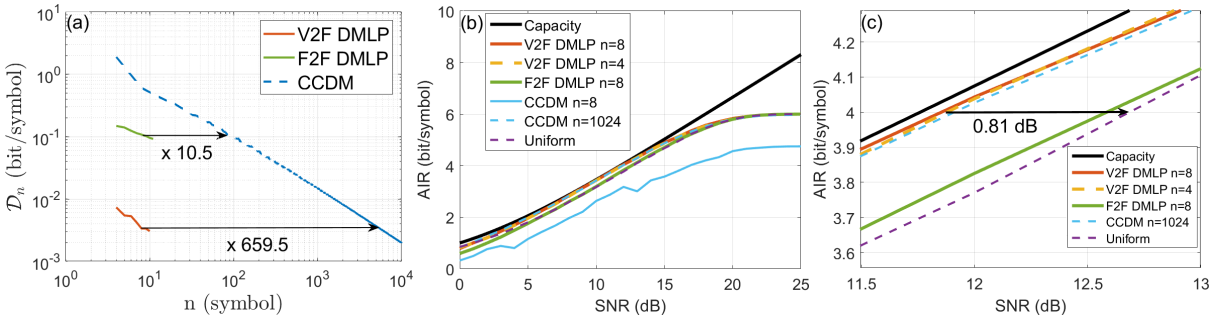

Fig. 2: Performance comparison of DMLP and CCDM: (a) Normalized information divergence D_n over DM blocklength n; AIR for 64QAM over SNR (b) from 0 to 25 dB; (c) from 11.5 to 13 dB.

number of optimization variables are significantly reduced. The optimization variables x and parameters i, j, m, T, μ in Tab.1 are explained as follows. Index i represents different types of codewords A^n. Parameter μ_i is the maximum number of realizations of A^n for type i. Parameter $T_i[j]$ defines the exact composition of A^n with type i. j is the index of symbol A realization a_j. For example, $T_2[3] = 1$ means A^n of type 2 has 1 a_3 in the composition. Variable $x_i[m]$ for V2F DMLP is the total number of A^n of type i matched to B^* of length m in the LUT. For F2F DMLP, all sequences have the same length so variable x only has index i and doesn't have index m. The task of the LP solver is to find the values for variable $x_i[m]$ or x_i. Length m of bit sequences is a free parameter. The best (range of) m can be found by simulation.

Optimization constraints are shown in Eq. (6-9). Constraints of Eq. (6-7) follow the restrictions that the total number of A^n with type i should not exceed its maximum μ_i. Constraints of Eq. (8-9) are set to ensure all possible realizations of input bit streams can be divided into sequences that are the same as B^m or B^* in the sequence codebook, and to ensure B^* sequence is prefix-free.

Simulation Results and Discussion

We compare the performance of DMLP and CCDM over an AWGN channel through simulations. Following the PAS scheme[1], the target distribution for DM output symbols is Maxwell-Boltzmann distribution. DMLP codebook design is implemented using Pyomo optimization package and GLPK linear programming solver under Python environment. Other parts of the simulations are conducted in MATLAB. Normalized information divergence D_n and achievable information rate (AIR)[4] are used as the performance metrics.

Fig.2(a) shows the normalized information divergence D_n of DMLP and CCDM over DM blocklength n for symbol alphabet $\mathcal{A}$ = {1, 3, 5, 7} and target distribution P_A = (0.4415, 0.3209, 0.1654, 0.0722). At D_n = 0.0033, V2F DMLP has n of 8 whereas CCDM has n of 5276. Similar reductions hold for other blocklengths. Compared with CCDM, V2F and F2F DMLP significantly reduce the blocklength by a factor of 659.5 and 10.5 respectively. It is because CCDM only has codewords of a single composition, whereas DMLP has codewords of various types.

Fig.2 (b) and (c) show the AIR of 64QAM DMLP and CCDM over SNR. AWGN channel capacity and uniform 64QAM without DM are also included as reference. At n=8 and AIR=4 bits/symbol, DMLP improves SNR sensitivity by 4.83 dB compared with CCDM. V2F DMLP with n=4 and CCDM with n=1024 have similar performances. At AIR=4 bits/symbol, they both are around 0.81 dB more power-efficient than the uniform 64QAM and are within 0.16 dB of the channel capacity. The blocklength of V2F DMLP is reduced by 256 times compared with CCDM. This factor of reduction is much larger than 3[4], 5.5[5], and 13[6] among the existing algorithms.

From the comparison above, we can see that V2F DMLP has much better performance than F2F DMLP. The LUT size for both DMLP is reasonably small. At n=4 and SNR=14, the number of symbol codewords in the LUT is 128 and 256 for F2F and V2F DMLP respectively. As for implementation complexity, V2F is more complex than F2F. However, it is still acceptable as the expected number of sequence codeword searching operations is just around 6.3 per input bit.

Conclusions

We proposed a novel linear programming based distribution matching algorithm that achieves high information rate for short blocklength probabilistic shaping by minimizing the normalized information divergence. At AIR of 4 bits/symbol and SNR of 11.87 dB, the blocklength is shortened by a factor of 256 compared with CCDM, which is desirable for short-reach applications.

References

[1] G. Bocherer, F. Steiner, and P. Schulte, "Bandwidth efficient and rate-matched low-density parity-check coded modulation", *IEEE Trans. Commun.*, vol. 63, no. 12, pp. 4651–4665, 2015.

[2] P. Schulte and G. Bocherer, "Constant composition distribution matching", *IEEE Trans. Inf. Theory*, vol. 62, no. 1, pp. 430–434, 2016.

[3] K. Zhong *et al.*, "Digital signal processing for short-reach optical communications: A review of current technologies and future trends", *J. Lightw. Technol.*, vol. 36, no. 2, pp. 377–400, 2018.

[4] T. Fehenberger *et al.*, "Multiset-partition distribution matching", *IEEE Trans. Commun.*, vol. 67, no. 3, pp. 1885–1893, 2019.

[5] P. Schulte and F. Steiner, "Divergence-optimal fixed-to-fixed length distribution matching with shell mapping", *IEEE Wireless Commun. Lett.*, vol. 8, no. 2, pp. 620–623, 2019.

[6] A. Amari *et al.*, "Introducing enumerative sphere shaping for optical communication systems with short blocklengths", *J. Lightw. Technol.*, vol. 37, no. 23, pp. 5926–5936, 2019.

[7] G. Bocherer and R. Mathar, "Matching dyadic distributions to channels", in *Proc. Data Compress. Conf.*, 2011, pp. 23–32.

[8] S. P. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, 2004.

[9] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. J. Wiley, 2005.

[10] J. Cho, "Prefix-free code distribution matching for probabilistic constellation shaping", *IEEE Trans. Commun.*, vol. 68, no. 2, pp. 670–682, 2020.