Technische Universität München
Professur für
Leitungsgebundene Übertragungstechnik
Prof. Dr.-Ing. Norbert Hanik

Research Internship

Probabilistic Constellation Shaping with Autoencoders

Vorgelegt von:
David de Andrés Hernández

München, Month Year

Betreut von:
M.Sc. Francesca Diedolo

Research Internship an der
Professur für Leitungsgebundene Übertragungstechnik (LÜT)
der Technischen Universität München (TUM)
Titel : Probabilistic Constellation Shaping with Autoencoders
Autor : David de Andrés Hernández

David de Andrés Hernández Boschetsriederstr. 55A
81379 München
deandres.hernandez@tum.de

Ich versichere hiermit wahrheitsgemäß, die Arbeit bis auf die dem Aufgabensteller bereits bekannte Hilfe selbständig angefertigt, alle benutzten Hilfsmittel vollständig und genau angegeben und alles kenntlich gemacht zu haben, was aus Arbeiten anderer unverändert oder mit Abänderung entnommen wurde.

München, xx.xx.20xx

. . . .

Ort, Datum (David de Andrés Hernández)

Contents

Abstract

The abstract comes here.

1 Introduction

The constant demand for higher capacity digital links has motivated the development of communication schemes which approach closer and closer the analytical limit of the channel capacity. According to the definition of channel capacity, sending a single bit per time-frequency slot is inefficient. For this reason, higher-order modulations like Amplitude shift keying (ASK) or Quadrature amplitude modulation (QAM) are used for better efficiency. Under these modulation schemes, the receiver handles more than two signal points per real dimension; the set of signal points is known as constellation. However, these schemes present a constant-width gap to the capacity limit. This is due to the usage of both uniform and discrete probability distributions for the occurrence of the constellation points. While it is not possible to move away from the discrete distributions because of the digital nature of the communication, specific non-uniform distributions can lead to further capacity improvements.

Constellation shaping thus, is a technique which seeks to optimize the distribution of the transmit symbols. Furthermore, this optimization unfolds into the improvement of the constellation points' location, their occurrence probability or both simultaneously. The first is known as geometric shaping while the latter is known as probabilistic shaping. In both cases, the goal is to maximize the Mutual information (MI) $I(X;Y)$ of the channel input X and output Y by optimizing the constellation. This optimization problem arises from the definition of channel capacity C:

$$C = \max_{p(X)} I(X;Y) \tag{1.1}$$

Currently, the optimal $p(x)$ has only be found for specific channels, such as the Additive white gaussian noise (AWGN), as knowledge of the channel distribution $p(y|x)$ is required. Still, solving 1.1 can become mathematically intractable despite knowing $p(y|x)$.

Here is where the application of deep-learning helps to find constellations which maximize $I(X;Y)$, without analytical knowledge of the channel. As shown in [OH17], the complete communication systems can be interpreted as an autoencoder. This approach tackles the physical layer by optimizing the end-to-end performance instead of the performance of the individual components by carefully choosing the loss function. Geometric shaping

under the autoencoder framework has been already performed in [OH17], [JEY$^+$18]. Furthermore, geometric shaping and probabilistic shaping have been jointly implemented in [SAAH19], [AH20] and [AC21]. The results show that the joint application of probabilistic and geometric shaping outperform the PS-QAM scheme from [BSS15] and approach the limit to within 0.1 dB in the AWGN channel. Yet the existence of multiple and apparently different architectures is intriguing. Consequently, in this work we study the differences in the implementations proposed by [SAAH19] and [AC21].
The rest of this document is organized as follows. Chapter 2, Preliminaries, presents the fundamentals of classical Probabilistic constellation systems as well as of autoencoder systems. Chapter 3, presents an implementation of both architectures with Pytorch [PGM$^+$19] and analyses their main differences. Chapter 4 wraps-up this work.

2 Preliminaries

2.1 Probabilistic Constellation Shaping

In this section our goal is to present the capacity limitations of the commonly used ASK and QAM modulation schemes. These schemes are penalized for two reasons:

1. They use uniform probability densities.

2. The constellation points are equidistant.

In the following we explain the nature of these penalties.

2.1.1 Introduction

We begin with an important result from Information theory. Under a second-moment constraint, also known as power constraint, the probability distribution which maximizes the differential entropy is the Gaussian distribution, p_G. We thus have

$$h(X) \leq \frac{1}{2}\log\left(2\pi e\sigma^2\right) \tag{2.1}$$

where $\sigma^2 = \mathbb{E}[X^2]$, and with equality if and only if X is Gaussian-distributed. More generally in the multi-dimensional case we have

$$h(\underline{X}) \leq \frac{1}{2}\log\left((2\pi e)^n|\mathbf{Q}_{\underline{X}}|\right) \tag{2.2}$$

where we have considered a random column vector $\underline{X}$ of dimension n, mean $\mathbb{E}[\underline{X}] = \underline{m}$ and covariance matrix

$$\mathbf{Q}_{\underline{X}} = \mathbb{E}[(\underline{X} - \underline{m})(\underline{X} - \underline{m})^\intercal] \tag{2.3}$$

and equality in 2.2 if only if $\underline{X}$ are jointly Gaussian.
Lets now consider an AWGN channel with input X, of zero mean and variance P, noise

Z and output Y; i.e. $Y = X + Z$. Furthermore, the capacity of the AWGN is

$$\begin{aligned} C(P) &= \max_{P_X:\mathbb{E}[X^2]\leq P} \mathbb{I}(X;Y) && (2.4)\\ &= \max_{P_X:\mathbb{E}[X^2]\leq P} [h(Y) - h(Y|X)] && (2.5)\\ &= \frac{1}{2}\log(2\pi e(P+N)) - \frac{1}{2}\log(2\pi e N) && (2.6)\\ &= \frac{1}{2}\log\left(1+\frac{P}{N}\right). && (2.7)\end{aligned}$$

We can analyse the mutual information

$$\mathbb{I}(X;Y) = h(Y) - h(Y|X) \tag{2.8}$$

in two parts, the differential entropy of the output and the conditional differential entropy of the output given the input. We expand the second term as

$$\begin{aligned} h(Y|X) &= h(Y-X|X) && (2.9)\\ &= h(Z|X) && (2.10)\\ &= h(Z) = \frac{1}{2}\log\left(2\pi e\sigma^2\right) && (2.11)\end{aligned}$$

and observe that the term $h(Y|X)$ does not depend on how X is distributed. In contrast, $h(y)$ does depend on how X is distributed by

$$p_Y(y) = \int_{-\infty}^{\infty} p_X(X) p_Z(y-x)\,dx = (p_X \star p_Z)(y). \tag{2.12}$$

To circumvent the fact that it is difficult to find a closed-form expression of $h(Y)$, we make use of the information divergence as

$$h(Y) \overset{\text{(a)}}{=} h(Y_G) - \mathbb{D}(p_Y \| p_G) \tag{2.13}$$

where (a) arises from the fact that $\mathbb{X}(p_X\|p_G) = h(Y_G)$ if and only if p_X has zero mean and variance P as p_G. 2.13 is very useful as it allows us to express the differential entropy of the output in terms of the difference between p_G and any other distribution by means of the cross entropy.

Now we can rewrite 2.8 as

$$
\begin{aligned}
\mathbb{I}(X;Y) &= h(Y) - h(Y|X) && (2.14)\\
&= h(Y) - h(Z) && (2.15)\\
&= h(Y_G) - \mathbb{D}(p_Y \| p_G) - h(Z) && (2.16)\\
&= [h(Y_G) - h(Z)] - \mathbb{D}(p_Y \| p_G) && (2.17)\\
&= C(P/\sigma^2) - \mathbb{D}(p_Y \| p_G) && (2.18)
\end{aligned}
$$

This last result indicates that the loss of MI when using a distribution P_X different than P_G is the informational divergence $\mathbb{D}(p_Y \| p_G)$. In other words, if the gaussian distribution is not used, the capacity penalty is characterized by $\mathbb{D}(p_Y \| p_G)$.

2.1.2 Capacity Gap for Uniform Continuous Input

We would like now to understand how far a uniform distribution is from 2.1. To do this, we will follow the approach presented in [Bö18] to lower bound the MI. Start by defining X_u as a uniformly distributed input on the interval $[-A, A]$ where A is carefully chosen so that $\mathbb{E}[{X_u}^2] = P$. The corresponding output is Y_u and we proceed

$$
\begin{aligned}
\mathbb{I}(X_u;Y_u) &= C(\text{snr}) - \mathbb{D}(p_{Y_u} \| p_{Y_G}) && (2.19)\\
&\geq C(\text{snr}) - \mathbb{D}(p_{X_u} \| p_{X_G}) && (2.20)\\
&= C(\text{snr}) - [h(X_G) - h(X_u)] && (2.21)\\
&= C(\text{snr}) - \frac{1}{2}\log_2\left(\frac{\pi e}{6}\right). && (2.22)
\end{aligned}
$$

In figure 2.1 we display the derived lower bound and observe that the capacity loss because of using a uniform input density is never more than $\frac{1}{2}\log_2\frac{\pi e}{6}$ independent of the Signal-to-noise ratio (SNR). To show that the shaping gap is tight, it is necessary to proof an upper bound for $\mathbb{I}(X_u;Y_u)$ that approaches 2.22 with increasing SNR. We refer the reader to [Bö18], section 4.3, for this proof.

2.1.3 Uniform Discrete Input Bound

We now show the penalty received for the use of an equidistant M-ASK constellation. We denote the channel input as

$$
\mathcal{X} = \{\pm1, \pm3, \dots, (M-1)\}. \qquad (2.23)
$$

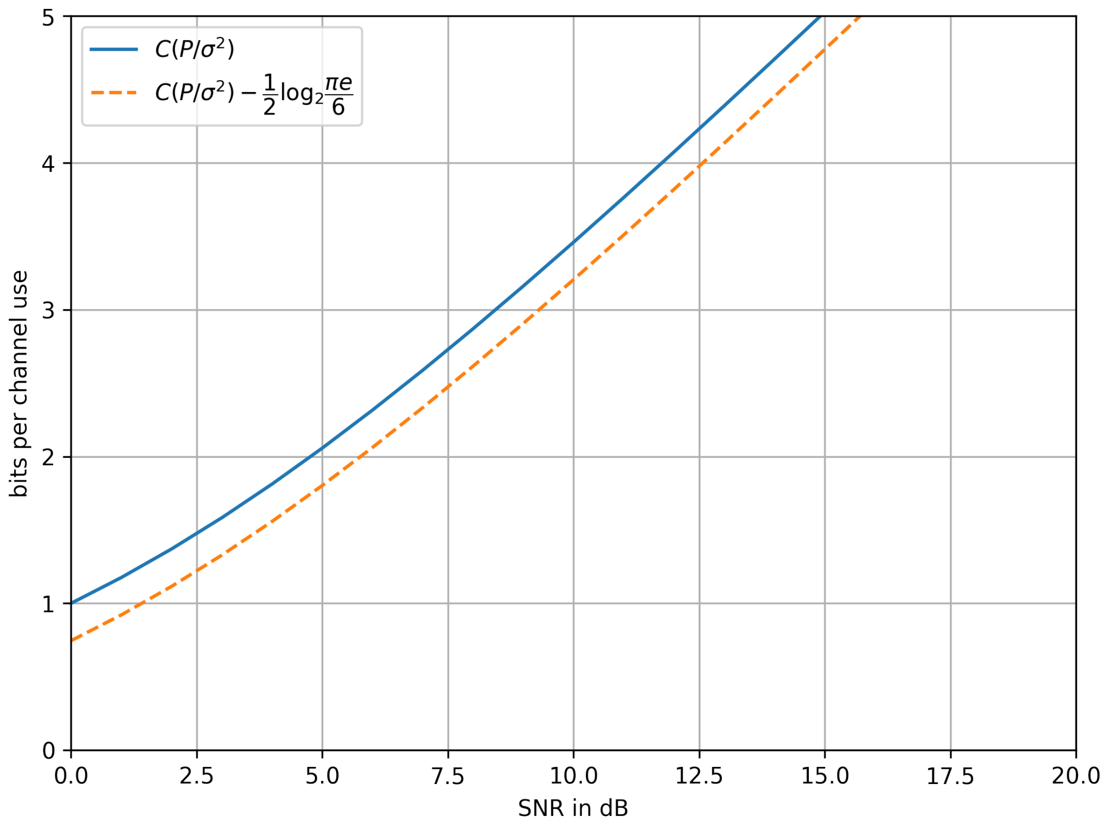

Figure 2.1: AWGN channel capacity gap.

Theorem 1 (Uniform Discrete Input Bound)**.** *The mutual information by X_M is lowered bounded by*

$$\mathbb{I}(X_M; Y_M) \geq \frac{1}{2}\log_2\left(\frac{M^2}{M^2-1}\right) - \frac{1}{2}\log_2\left[2\pi e\left(\frac{P}{M^2-1} + \frac{P}{1+P/\sigma^2}\right)\right] \tag{2.24}$$

$$< C(snr) - \frac{1}{2}\log_2\left(\frac{\pi e}{6}\right) - \frac{1}{2}\log_2\left[1 + \left(\frac{2^{C(snr)}}{M}\right)^2\right] \tag{2.25}$$

$$< C(snr) - \mathit{Penalty}(\mathit{Uniform\ dist.}) - \mathit{Penalty}(\mathit{Equidistant\ dist.}) \tag{2.26}$$

where $snr = P/\sigma^2$.

We refer the reader to [Bö18], section 4.5, for the proof. Theorem 1 shows our goal, namely that both the usage of a uniform distribution and an equidistant constellation penalizes the capacity. We can additionally compute the relation between the constellation size, M, and $C(\text{snr})$ so that the resulting mutual information is within a constant gap of capacity. To make this result even more attractive, we increase the constraint for the gap to match the order of the distribution loss (0.255 bits). We obtain

$$-\frac{1}{2}\log_2\left[1 + \left(\frac{2^{C(\text{snr})}}{M}\right)^2\right] \leq -\frac{\log_2 e}{2}\left(\frac{2^{C(\text{snr})}}{M}\right)^2 = \frac{1}{4} \tag{2.27}$$

$$\Leftrightarrow M = 2^{C(\text{snr}) + \frac{1}{2} + \frac{1}{2}\log_2\log_2 e} \tag{2.28}$$

by using $\log_e(x) \leq (1-x)$. So if

$$\log_2 M \approx C(\text{snr}) + 0.77, \tag{2.29}$$

then the mutual information is within 0.5 bit of capacity.

2.1.4 PAS architecure

2.2 Autoencoders

In this section we present the basics of autoencoders in the context of Deep learning and communication systems, which was pioneered in **??**. The idea behind an autoencoder is to transmit a particular representation of the input data so that at the output, it can be reconstructed with minimal error. This means that the desired representations must be robust with respect to the channel impairments (i.e. noise, fading, distortion, etc.). To find such representations and the corresponding mappings $\mathbf{x}$ to $\mathbf{y}$ we train deep

neural networks (NNs). Because NNs are universal function approximators [HSW89], this technique is particularly interesting for training over channels without a mathematically tractable model. A representation of an autoencoder system is shown in Figure **??**.
To find fitting sets of parameters θ, the most used algorithm is Stochastic gradient descent (SGD) which starts with a random set of initial values and then updates θ with each iteration as

$$\theta_{new} = \theta_{old} + \epsilon \frac{\partial}{\partial \theta_{old}} L(\theta_{old}) \tag{2.30}$$

One particular requirement for using SGD to train the autoencoders is that the loss gradient needs to be backpropagated all the way through receiver and channel to the transmitter. Otherwise, the transmitter parameters cannot be updated. This in turn means, that channel and receiver must be available as differentiable functions during training.
We can now set up the problem which we would like to address in this work. Namely, to train a deep NN-based autoencoder system to find a parametric distribution $p_\theta(s)$. By maximizing the MI during training, the output distribution must satisfy 2.18, and thus, approach the channel capacity.

3 contribution

3.1 Notation

In the following we will use the notation:

$$\mathbb{I}\left(X, Y; D, P_M, C_M\right)$$

which expresses the mutual information between X and Y and D, P_M, C_M, separated by a semi-colon, are the trainable parameters of the system. The parameters can be seen as additional input to a function.

3.2 Optimization of trainable parameters

As we have seen in chapter 2, the goal of probabilistic constellation shaping is to maximize the mutual information

$$\max_{D, P_M, C_M} \mathbb{I}\left(X, Y; D, P_M, C_M\right) \overset{\text{(a)}}{=} \mathbb{H}(X) - \mathbb{X}(P_{X|Y} \| Q_{X|Y}; D, P, C) \tag{3.1}$$

where the entropy is maximized when the probabilities of the constellation points follow a Gaussian distribution; and the cross-equivocation is minimum when $Q_{X|Y} = P_{X|Y}$.
Typically, the gradient descent (ascent) allows us to solve the optimization problem by adjusting the trainable parameters as:

$$\theta_{new} = \theta_{old} + \epsilon \frac{\partial}{\partial \theta_{old}} \mathbb{I}\left(X, Y; \theta_{old}\right) \tag{3.2}$$

for all trainable parameters $\theta \in P, C, D$. However, [work in progress...]
We will need to numerically compute the terms in 3.1. For this reason, we will use the following expansions:

$$\mathbb{X}\left(P_{X|Y} \| Q_{X|Y} | Y = b\right) = \sum_{a \in Supp(P_{X|Y}(\cdot|b))} P_{X|Y}(a|b) \log_2(Q_{X|Y}(a|b)) \tag{3.3}$$

$$\mathbb{X}\left(P_{X|Y}\|Q_{X|Y}\right) = \sum_{b\in Supp(P_Y)} P_Y(b)\mathbb{X}\left(P_{X|Y}\|Q_{X|Y}|Y=b\right) \tag{3.4}$$

which combined together and applying Bayes' theorem yields

$$\mathbb{X}\left(P_{X|Y}\|Q_{X|Y}\right) = \sum_{(a,b)\in Supp(P_{XY})} P_X(a)P_{Y|X}(b|a)\log_2(Q_{X|Y}(a|b)). \tag{3.5}$$

Precisely, we must compute the derivative terms of 3.5 with respect to $P_X(j), j \in \mathcal{X}$. This is

$$\frac{\partial}{\partial P_X(j)}\mathbb{X}\left(P_{X|Y}\|Q_{X|Y}\right) = \sum_{b \text{ if } x=j} P_{Y|X}(b|j)\log_2 Q_{X|Y}(j|b) \tag{3.6}$$

$$+ \sum_{(a,b)\in Supp(P_{XY})} P_{XY}(a,b)\frac{\partial}{\partial P_X(j)}\log_2 Q_{X|Y}(a|b) \tag{3.7}$$

which can be rewritten using the expectation operator as

$$\frac{\partial}{\partial P_X(j)}\mathbb{X}\left(P_{X|Y}\|Q_{X|Y}\right) = \mathbb{E}_{Y|X}[\log_2 Q_{X|Y}(j|b)|X=j] \tag{3.8}$$

$$+\,\mathbb{E}_{XY}[\frac{\partial}{\partial P_X(j)}\log_2 Q_{X|Y}(a|b)]. \tag{3.9}$$

The terms can now be numerically computed as

$$\mathbb{E}_{Y|X}[\log_2 Q_{X|Y}(j|b)|X=j] \approx \frac{1}{BP_X(j)}\sum_{b \text{ if } x=j}\log_2 Q_{X|Y}(j|b) \tag{3.10}$$

$$\mathbb{E}_{XY}[\frac{\partial}{\partial P_X(j)}\log_2 Q_{X|Y}(a|b)] \approx \frac{1}{B}\sum_{(a,b)\in Supp(P_{XY})}\log_2 Q_{X|Y}(a|b) \tag{3.11}$$

Appendix A

First appendix goes here.

Appendix B

Second appendix goes here.

Bibliography

[AC21] V. Aref and M. Chagnon, “End-to-end learning of joint geometric and probabilistic constellation shaping,” 2021. [Online]. Available: https://arxiv.org/abs/2112.05050

[AH20] F. A. Aoudia and J. Hoydis, “Joint learning of probabilistic and geometric shaping for coded modulation systems,” in *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, 2020, pp. 1–6.

[BSS15] G. Böcherer, F. Steiner, and P. Schulte, “Bandwidth efficient and rate-matched low-density parity-check coded modulation,” *IEEE Transactions on Communications*, vol. 63, no. 12, pp. 4651–4665, 2015.

[Bö18] G. Böcherer, “Principles of coded modulation. [online] available,” 2018. [Online]. Available: http://www.georg-boecherer.de/bocherer2018principles.pdf

[HSW89] K. Hornik, M. Stinchcombe, and H. White, “Multilayer feedforward networks are universal approximators,” *Neural Networks*, vol. 2, no. 5, pp. 359–366, 1989. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0893608089900208

[JEY+18] R. T. Jones, T. A. Eriksson, M. P. Yankov, B. J. Puttnam, G. Rademacher, R. S. Luis, and D. Zibar, “Geometric constellation shaping for fiber optic communication systems via end-to-end learning,” 2018. [Online]. Available: https://arxiv.org/abs/1810.00774

[OH17] T. O’Shea and J. Hoydis, “An introduction to deep learning for the physical layer,” *IEEE Transactions on Cognitive Communications and Networking*, vol. 3, no. 4, pp. 563–575, 2017.

[PGM+19] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, “Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 8024–8035. [Online]. Available: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

[SAAH19] M. Stark, F. Ait Aoudia, and J. Hoydis, "Joint learning of geometric and probabilistic constellation shaping," in *2019 IEEE Globecom Workshops (GC Wkshps)*, 2019, pp. 1–6.