NUMERICAL COMPUTATION OF THE CAPACITY OF CONTINUOUS MEMORYLESS CHANNELS

Justin Dauwels
Dept. of Information Technology and Electrical Engineering
ETH, CH-8092 Zürich, Switzerland
dauwels@isi.ee.ethz.ch

We extend the Blahut-Arimoto algorithm to continuous memoryless channels by means of sequential Monte Carlo integration in conjunction with gradient methods. We apply the algorithm to a Gaussian channel with an average- and/or peak-power constraint.

INTRODUCTION

We consider the problem of computing the capacity

$$C(X;Y) \triangleq \sup_{p(x)} \int_x \int_y p(x)p(y|x) \log \frac{p(y|x)}{p(y)} dxdy \triangleq \sup_{p(x)} I(X;Y) \quad (1)$$

between the input X and the output Y of a memoryless channel $p(y|x)$ with $p(y) \triangleq \int_x p(x)p(y|x)dx$. Both X and Y may be discrete or continuous. If x is discrete, $\int_x g(x)dx$ stands for the summation of $g(x)$ over its support.

For memoryless channels with *finite* input and output alphabets $\mathcal{X}$ and $\mathcal{Y}$ respectively, the capacity (1) can be computed by the Blahut-Arimoto algorithm [1] [2]. Recently, Matz et al. proposed two Blahut-Arimoto-type algorithms that often converge significantly faster than the standard Blahut-Arimoto algorithm [3]. Moreover, the Blahut-Arimoto algorithm was recently extended to channels *with memory* and *finite* input alphabets and state spaces [4].

For memoryless channels with *continuous* input and/or output alphabets, the Blahut-Arimoto algorithm is not directly applicable. In this paper, we propose an algorithm for computing the capacity of such channels. It is similar in spirit as the algorithm we proposed in [5] for computing information rates of continuous channels with memory; the key idea is again the use of sequential Monte-Carlo integration methods (a.k.a "particle filters") [6]. In those methods, probability distributions are represented as lists of samples ("particles"). In the proposed algorithm, the lists of samples are recursively updated in two steps: first, the weights

of the particles are computed by the Blahut-Arimoto algorithm; then, one determines the corresponding output distribution $p(y)$, and moves the particles $\hat{x}$ in the input space with the aim of increasing the relative entropy $D(p(y|x)\|p(y))$ (which is a function of x) between $p(y|x)$ and $p(y)$; this can be done by several iterations of some gradient method such as steepest descent or the Newton method [9].
The proposed algorithm is related to the algorithm proposed in 1988 by Chang et al. [7], where, after the weight updates, one determines *all* local maxima $\hat{x}_{\max}$ of the relative entropy $D(p(y|x)\|p(y))$; those maxima are then appended to the list of samples. That algorithm is rather impractical: finding *all* local maxima of a function is often infeasible, especially in high dimensions. In addition, there may be an infinite number of local maxima. Lafferty et al. [8] proposed an alternative algorithm based on Markov-chain-Monte-Carlo methods. Its complexity grows *quadratically* in the number of iterations, whereas the complexity of our algorithm depends *linearly* on the number of iterations.
This paper is structured as follows. First, we briefly review the Blahut-Arimoto algorithm [1] [2] and two recently proposed extensions [3]. We then outline our particle-based algorithm and shortly discuss a few numerical examples.

REVIEW OF BLAHUT-ARIMOTO-TYPE ALGORITHMS

The Blahut-Arimoto algorithm [1] [2] is an alternating-maximization algorithm for computing the capacity (1). One starts with an arbitrary probability function $p^0(x)$. At each iteration k, the probability function $p^{(k)}(x)$ is obtained as

$$p^{(k)}(x) = \frac{1}{Z^{(k)}} p^{(k-1)}(x) \exp\left(D\left(p(y|x)\|p^{(k-1)}(y)\right)\right), \tag{2}$$

where the expression $p^{(k)}(y)$ is defined as

$$p^{(k)}(y) \triangleq \int_{x\in\mathcal{X}} p^{(k)}(x)p(y|x)dx, \tag{3}$$

and $Z^{(k)}$ is a scaling factor

$$Z^{(k)} \triangleq \int_{x\in\mathcal{X}} p^{(k-1)}(x) \exp\left(D\left(p(y|x)\|p^{(k-1)}(y)\right)\right) dx. \tag{4}$$

If after n iterations

$$\max_{x\in\mathcal{X}} D\left(p(y|x)\|p^{(n)}(y)\right) - I^{(n)} < \varepsilon, \tag{5}$$

with

$$I^{(n)} \triangleq \int_{x\in\mathcal{X}} p^{(n)}(x) D\left(p(y|x) \| p^{(n)}(y)\right) dx, \tag{6}$$

and ε a "small" positive real number (e.g., $\varepsilon = 10^{-5}$), then one concludes with the estimate $\hat{C} = I^{(n)}$.

In 2004, Matz et al. [3] proposed two related algorithms for computing the capacity of memoryless channels, i.e., the so-called "natural-gradient-based algorithm" and the "accelerated Blahut-Arimoto algorithm"; they often converge significantly faster than the standard Blahut-Arimoto algorithm.

In the natural-gradient-based algorithm, the density $p^{(k)}(x)$ is recursively updated by the rule

$$p^{(k)}(x) = p^{(k-1)}(x)\left[1 + \mu^{(k)}\left(D\left(p(y|x) \| p^{(k-1)}(y)\right) - I^{(k-1)}\right)\right], \tag{7}$$

where $\mu^{(k)}$ is a step-size parameter. Note that $p^{(k)}(x)$ in (7) is guaranteed to be normalized. In the accelerated Blahut-Arimoto algorithm, the update rule is given by:

$$p^{(k)}(x) = \frac{1}{Z^{(k)}} p^{(k-1)}(x) \exp\left(\mu^{(k)} D\left(p(y|x) \| p^{(k-1)}(y)\right)\right), \tag{8}$$

where $Z^{(k)}$ is a normalization constant.

Many channels have an associated expense of using each of the input symbols. A common example is the power associated with each input symbol. A constrained channel is a channel with the requirement that the average expense be less than or equal to some specified number E. The capacity at expense E is defined as [2]

$$C(X;Y) \triangleq \sup_{p(x)\in P_E} \int_x \int_y p(x)p(y|x) \log \frac{p(y|x)}{\int_x p(x)p(y|x)} \triangleq \sup_{p(x)\in P_E} I(X;Y), \tag{9}$$

where

$$P_E \triangleq \left\{p : \mathbb{K}^m \to \mathbb{R} : \int_x p(x)dx = 1, p(x) \geq 0, \int_x p(x)e(x)dx \leq E\right\}, \tag{10}$$

and $\mathbb{K} = \mathbb{R}$ or $\mathbb{C}$. The Blahut-Arimoto can be extended to constrained channels by replacing recursion (2) by

$$p^{(k)}(x) = \frac{1}{Z^{(k)}} p^{(k-1)}(x) \exp\left(D\left(p(y|x) \| p^{(k-1)}(y)\right) - se(x)\right), \tag{11}$$

with s a positive real number [2], that is adjusted after each Blahut-Arimoto update (11) to keep the average expense smaller or equal to E. The two Blahut-Arimoto-type algorithms of [3] can similarly be extended to constrained memoryless channels.

A PARTICLE-BASED BLAHUT-ARIMOTO ALGORITHM

When X and Y are discrete, the Blahut-Arimoto-type algorithms reviewed in the previous section are practical. Otherwise, the Blahut-Arimoto updates (cf. (2), (7), and (8)) can not be carried out as such. We propose the following algorithm to compute the capacity of *continuous* memoryless channels:

1. Start with some initial list $\mathcal{L}_X^{(0)}$ of particles $\hat{x}^{(0)} \in \mathcal{X}$ with uniform weights $w^{(0)}$.
2. Compute the weights $w^{(k)}$ of the particles $\hat{x}^{(k)}$ by running a Blahut-Arimoto-type algorithm [2]–[3] (until convergence) with the particles $\hat{x}^{(k)}$ as input alphabet.
3. Determine the output distribution $p^{(k)}(y)$ corresponding to the list $\mathcal{L}_X^{(k)}$ of particles $\hat{x}^{(k)}$ with weights $w^{(k)}$.
4. Move the particles $\hat{x}^{(k)}$ in order to increase the relative entropy $D(p(y|x)\|p(y))$ between $p(y|x)$ and $p^{(k)}(y)$ while keeping $p^{(k)}(y)$ fixed; this may be carried out by several iterations of steepest descent or the Newton method [9].
5. Iterate 2–4 until convergence or until the available time is over.

After n iterations, an approximation for the capacity (1) may be obtained by numerical integration (w.r.t. Y):

$$\hat{C} = I^{(n)} \triangleq h \sum_{\hat{x}^{(n)},\hat{y}} w^{(n)} p\big(\hat{y}|\hat{x}^{(n)}\big) \left[\log p\left(\hat{y}|\hat{x}^{(n)}\right) - \log p^{(n)}(\hat{y})\right], \tag{12}$$

where $\hat{y} \in \mathcal{Y}$ are quantization levels. The output distribution $p^{(n)}(y)$ is then approximated by a histogram with bins of width h, centered at the quantization levels $\hat{y}$. Note that numerical integration is only feasible for low-dimensional systems. Alternatively, the capacity (1) may be computed by Monte Carlo integration, which is also applicable to high-dimensional systems:

$$\hat{C} = I^{(n)} \triangleq \sum_{\hat{x}^{(n)},\hat{y}^{(n)}} w^{(n)} \left[\log p\left(\hat{y}^{(n)}|\hat{x}^{(n)}\right) - \log p^{(n)}(\hat{y}^{(n)})\right], \tag{13}$$

where $\hat{y}^{(n)}$ are samples from the conditional pdfs $p(y|\hat{x}^{(n)})$. The output distribution $p^{(n)}(y)$ is then represented as a list of samples. The required evaluation of $p^{(n)}(y)$ may be performed based on some continuous approximation of $p^{(n)}(y)$ such as a density tree [10] generated from the samples $\hat{y}^{(n)}$. The relative entropies occurring in the Blahut-Arimoto-type updates of Step 2 (cf. (2), (7), and (8)) may also be computed by numerical or Monte Carlo integration (w.r.t Y).

If one makes sure that the relative entropy $D(p(y|x)\|p(y))$ does not decrease while moving the particles in Step 4 (by using the Armijo rule [9] for example for determining the step size), then the sequence $I^{(n)}$ is non-decreasing, i.e., $I^{(n+1)} \geq I^{(n)}$, and the list of particles converges to stationary points of the (non-convex) relative entropy $D(p(y|x)\|p(y))$. In the limit of an infinite number of particles, the sequence $I^{(n)}$ converges to the channel capacity.

NUMERICAL EXAMPLE: GAUSSIAN CHANNEL

By means of the above algorithm, we computed the capacity of a Gaussian channel described by the equation

$$Y_k = X_k + N_k, \tag{14}$$

where X_k, Y_k, and $N_k \in \mathbb{R}$; N_k is independent of X_k and is drawn independently and identically (IID) from a zero-mean Gaussian distribution with variance σ_0^2

$$N_k \sim \text{IID}\, \mathcal{N}_{\mathbb{R}}(0, \sigma_0^2). \tag{15}$$

We considered the average-power constraint $\text{E}[X^2] \leq P$ and the peak-power constraint $\Pr[|X| > A] = 0$.

In the simulations, we used 100 particles $\hat{x}$ and between 100 and 1000 main iterations (Step 5), depending on the SNR; each such iteration consisted of 1000 Blahut-Arimoto iterations (Step 2) and 20 steepest descent updates (Step 4). Other settings of the parameters could be chosen, we spent only little effort in optimizing the algorithm. Since the problem is one-dimensional, we used numerical integration (w.r.t. Y) to compute the integrals involved in Step 2 and in the evaluation of the capacity (cf. (12)). In the future, we plan to tackle higher-dimensional problems by combining Monte-Carlo integration with density trees, as explained in the previous section.

The capacity C (in bits) of the Gaussian channel with average-power constraint is well-known:

$$C = \frac{1}{2}\log_2\left(1 + \frac{P}{\sigma_0^2}\right). \tag{16}$$

Fig. 1(a) shows the expression (16) together with the capacity values computed by our algorithm. We defined the signal-to-noise ratio (SNR) as

$$\text{SNR[dB]} \triangleq 10\log_{10}\left(\frac{P}{\sigma_0^2}\right). \tag{17}$$

The capacity-achieving input distribution is a zero-mean Gaussian distribution $\mathcal{N}_{\mathbb{R}}(0, P)$ with variance P. In Fig. 1(b), this distribution is shown together with the particle-based approximation. Notice that in Fig. 1(a) and Fig. 1(b), the exact and the numerical results are practically coinciding. The deviation between the numerical and the exact values of the capacity is about 10^{-5} bits/channel use. The accuracy could in fact be improved by increasing the number of particles and iterations.
Fig. 2 and Fig. 3 show the results for the peak-power constrained Gaussian channel ($A = 1$). Fig. 2(a) shows the capacity of the channel for several SNR-values, where the SNR is now defined as $\text{SNR[dB]} \triangleq 10\log_{10}\left(\frac{A^2}{\sigma_0^2}\right)$. The capacity-achieving input distribution for this channel is discrete (see e.g., [11] and references therein). Note that in our algorithm, we did not (need to) exploit the fact that the input distribution is discrete. Fig. 2(b) shows the input probability mass function for several SNR levels: the dots are the constellation points, their probability mass is encoded in the grayscale (white: $p = 0$; black: $p = 0.5$). As an illustration, the capacity-achieving cumulative input distribution $F(x)$ is depicted in Fig. 3(a) for SNR = 13dB. Fig. 3(b) shows how the particles explore the input space during the iterations: initially, the particles are uniformly distributed in the interval $[-1, 1]$; they gradually move towards the signal points of the capacity achieving input distribution.
Fig. 4 shows the results for the average- and peak-power constrained Gaussian channel ($A = 1$ and $P = 0.5$). Fig. 4(a) shows the channel capacity for several SNR-values, where the SNR is given by (17). In Fig. 4(b), the capacity-achieving input constellations are depicted.
We also have numerical results for some optical channel models (e.g., Poisson channel).

ACKNOWLEDGMENTS

The author wishes to thank Frank Kschischang, Sascha Korl, and Hans-Andrea Loeliger for inspiring discussions. This project was supported in part by the Swiss National Science Foundation grant 200021-101955.

REFERENCES

[1] S. Arimoto, "An algorithm for computing the capacity of arbitrary discrete memoryless channels," *IEEE Trans. Information Theory*, vol. 18, pp. 14–20, 1972.

[2] R. Blahut, "Computation of channel capacity and rate-distortion functions," *IEEE Trans. Information Theory*, vol. 18, pp. 460–473, 1972.

[3] G. Matz and P. Duhamel, "Information geometric formulation and interpretation of accelerated Blahut-Arimoto-type algorithms," *Proc. 2004 IEEE Information Theory Workshop,* San Antonio, TX, USA, Oct. 24–29, 2004.

[4] P. O. Vontobel, A. Kavčić, D. Arnold, and H.-A. Loeliger, "Capacity of Finite-State Machine Channels," submitted to *IEEE Trans. Inform. Theory*, Nov. 2004.

[5] J. Dauwels and H.-A. Loeliger, "Computation of information rates by particle methods," *Proc. 2004 IEEE Int. Symp. Information Theory*, Chicago, IL, USA, June 27 – July 2, 2004, p. 178.

[6] A. Doucet, J. F. G. de Freitas, and N. J. Gordon, eds., *Sequential Monte Carlo methods in practice.* New York: Springer-Verlag, 2001.

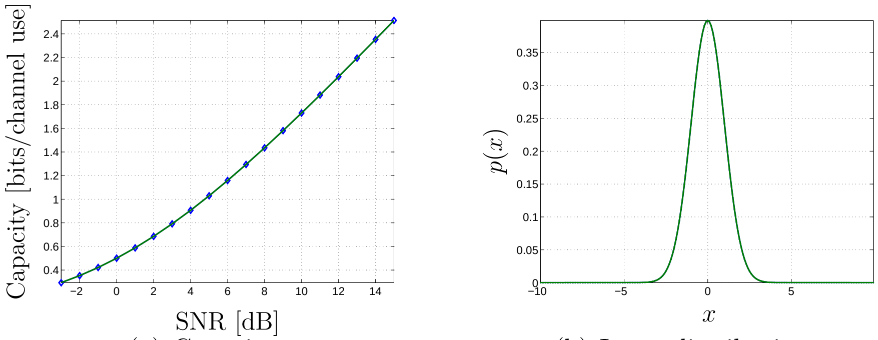
[7] C. Chang and L. D. Davisson, "On calculating the capacity of an infinite-input finite (infinite)-output channel," *IEEE Trans. Information Theory,* vol. 34, Sept. 1998, pp. 1004–1010.

[8] J. D. Lafferty and L. A. Wasserman, "Iterative Markov chain Monte Carlo computation of reference priors and minimax risk," *Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence,* August 02–05, 2001, pp. 293–300.

[9] D. P. Bertsekas, *Nonlinear programming,* Athena Scientific, Belmont, MA, 1995.

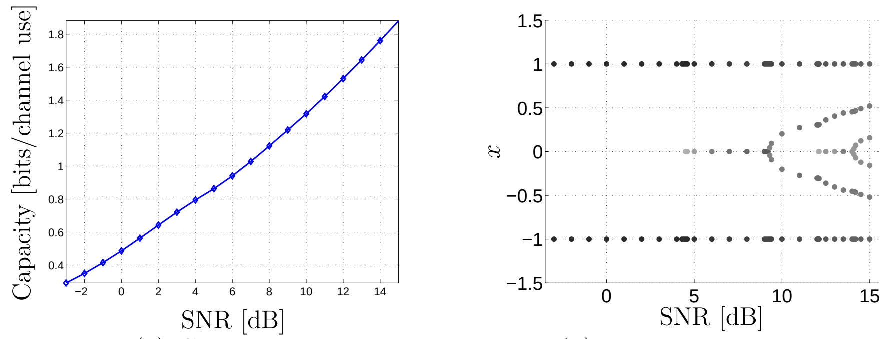
[10] J. H. Friedman, J. L. Bentley, and R. A. Finkel, "An algorithm for finding best matches in logarithmic expected time," *ACM Transactions on Mathematical Software,* vol. 3, Sept. 1977, pp. 209–226.

[11] T. Chan, S. Hranilovic and F. Kschischang, "Capacity-achieving probability measure for conditionally Gaussian channels with bounded inputs," submited to IEEE Trans. Information Theory, Dec. 2004.



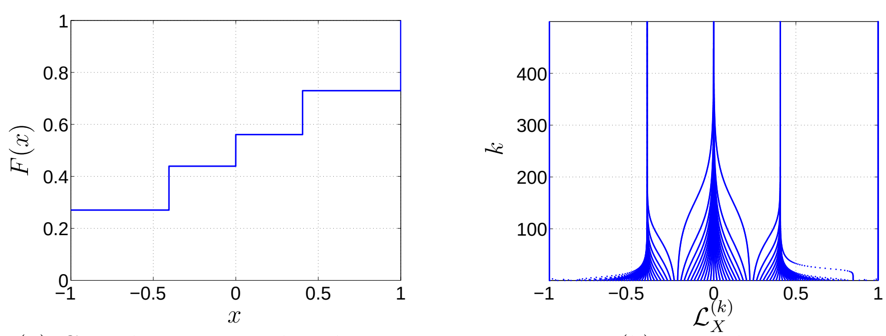
(a) Capacity (b) Input distribution

Figure 1: Gaussian channel with average-power constraint.



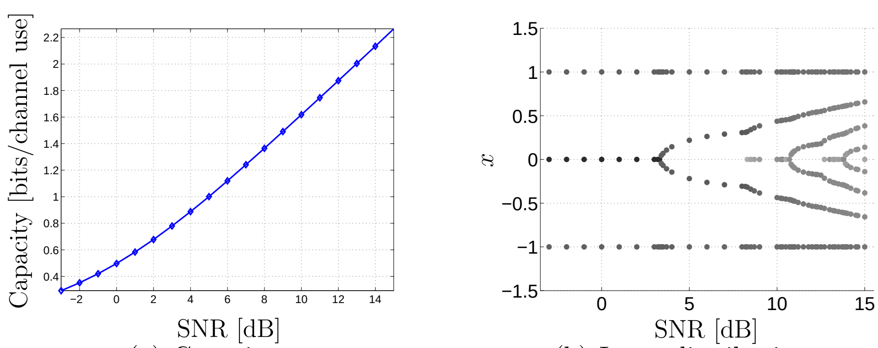
(a) Capacity (b) Input distribution

Figure 2: Gaussian channel with peak-power constraint.



(a) Cumulative input distribution (b) Particles

Figure 3: Gaussian channel with peak-power constraint (SNR $=$ 13dB).



(a) Capacity (b) Input distribution

Figure 4: Gaussian channel with average- and peak-power constraint.