Iterating the Arimoto-Blahut Algorithm for Faster Convergence

Jossy Sayir[1]
FTW, Maderstr. 1, A-1040 Vienna.
`j.sayir@ieee.org`

I. Introduction

The Arimoto-Blahut algorithm ([1], [2]) determines the capacity of a discrete memoryless channel through an iterative process in which the input probability distribution is adapted at each iteration. While it converges towards the capacity-achieving distribution for any discrete memoryless channel, the convergence can be slow when the channel has a large input alphabet. This is unfortunate when only a small number of the input letters are assigned non-zero probabilities in the capacity-achieving distribution. If we knew which input letters will end up with a probability of zero, we could eliminate these letters and operate the algorithm on a subset of the input alphabet. The algorithm would converge towards the same solution faster.

We present an algorithm which makes use of this fact to speed up the convergence of the Arimoto-Blahut algorithm in such situations. The algorithm starts with an input alphabet consisting of two symbols, then grows the alphabet by one symbol at every iteration until it includes all the symbols with non-zero probabilities. At every iteration, the Arimoto-Blahut algorithm is used to compute a capacity relative to a partial input alphabet. When our algorithm terminates, it will have found the same solution as the Arimoto-Blahut algorithm applied to the complete input alphabet. However, we cannot guarantee that our algorithm will include only symbols with non-zero probabilities in the partial alphabet it ends up with.

II. The Algorithm

Let X be the input random variable to a discrete memoryless channel, and let X take on values over the finite alphabet $\mathcal{A}$. Let Y be the output random variable of the channel and let C be its capacity. We define

$$I(X=x;Y)=\sum_y P_{Y|X}(y|x)\log\frac{P_{Y|X}(y|x)}{\sum_{x'}P_{Y|X}(y|x')P_X(x')}$$

as in [3]. Let $C_{\mathcal{A}'}$ denote the capacity of the discrete memoryless channel induced when all but the letters in the subset $\mathcal{A}'$ of $\mathcal{A}$ are forcibly assigned a probability of zero. We give an outline of our algorithm:

1. Determine $(x,y)\in\mathcal{A}^2$ which maximizes $C_{\{x,y\}}$ over all choices of x and y. Define $\mathcal{A}'=\{x,y\}$ and $C'=C_{\mathcal{A}'}$.
2. If $\mathcal{A}'=\mathcal{A}$, then $C=C'$ and the algorithm terminates. Otherwise, for all $x\in\mathcal{A}\setminus\mathcal{A}'$, compute $I(X=x;Y)$. If the values computed are all smaller or equal to C', then $C=C'$ by [3, Theorem 4.5.1] and the algorithm can be terminated at this point.
3. Add the symbol x that maximized $I(X=x;Y)$ in step 2 to the set $\mathcal{A}'$. Recompute $C'=C_{\mathcal{A}'}$ using the Arimoto-Blahut algorithm. Return to step 2.

The algorithm is certain to obtain the correct solution for the following reasons:

- When the algorithm exits in step 2 because all the values of $I(X=x;Y)$ are smaller or equal to C', the solution is guaranteed to be correct by [3, Theorem 4.5.1].
- The algorithm must eventually exit. In the worst case, it will end up including all the symbols of $\mathcal{A}$ into $\mathcal{A}'$. In this case, the last occurrence of step 3 applies the Arimoto-Blahut algorithm to the complete input alphabet. This will be the case when our iterated algorithm is applied to channels whose capacity-achieving input distributions have only non-zero terms.

As already mentioned, there is no guarantee that the algorithm will only include symbols whose probabilities in the capacity-achieving distribution are non-zero into its partial alphabet $\mathcal{A}'$. However, the practical examples for which the algorithm was tested seem to indicate that it is highly unlikely.

III. Practical Implementation and Conclusion

The practical need for such an algorithm arose in an attempt to compute the optimal coding distributions for universal lossless source coding over sets of discrete memoryless sources with monotone non-increasing probability distributions with a fixed average. The problem of determining the optimal coding distribution for universal coding over a set of probability distributions is equivalent to the problem of computing the capacity of a discrete memoryless channel [4].

For an alphabet size of 256, the problem of determining the optimal coding distribution for the set of monotone non-increasing distributions with a fixed average corresponds to the computation of the capacity of a channel with an input alphabet of up to 16'000 letters. Of those 16'000 letters, only about 100 letters have non-zero probabilities in the capacity-achieving distribution. Therefore, the algorithm presented here allowed a considerable acceleration of the convergence when compared to a conventional implementation of the Arimoto-Blahut algorithm. A detailed presentation of this application along with more information on the implementation of the algorithm are given in [5].

References

[1] Suguru Arimoto, “An Algorithm for Computing the Capacity of Arbitrary Discrete Memoryless Channels,” *IEEE Trans. on Inform. Theory,* **IT-18**, pp. 14-20, January 1972.

[2] Richard E. Blahut, “Computation of Channel Capacity and Rate-Distortion Functions,” *IEEE Trans. on Inform. Theory,* **IT-18**, pp. 460-473, July 1972.

[3] Robert G. Gallager (1968), *Information Theory and Reliable Communications,* John Wiley and Sons, New York, ISBN 0-471-29048-3.

[4] Robert G. Gallager, “Source Coding with Side Information and Universal Coding,” **unpublished,** submitted to the *IEEE Trans. on Information Theory,* September 1976.

[5] Jossy Sayir (1999), *On Coding by Probability Transformation,* ETH Zürich, PhD Dissert. Nr. 13099, Hartung-Gorre Verlag Konstanz, Germany, ISBN 3-89649-444-9.

---

[1] This paper documents work performed at the Signal & Inf. Proc. Lab., ETH Zürich, Switzerland.