On Numerical Approximation of the DMC Channel Capacity

(BFA'2017 Workshop)

Yi LU, Bo SUN, Ziran TU, Dan ZHANG
<Yi.Lu,Bo.Sun,Ziran.Tu,Dan.Zhang>@UiB.NO

Selmer Center for Secure and Reliable Communications,
Department of Informatics, University of Bergen (UiB), Norway

(5^{th} July, 2017)

Outline

Walsh Spectrum Characterization on Sampling Distributions

- Following a rump talk by Yi LU at FSE'2017 in Japan, it is proposed as a suitable topic for submission to the Nature journal.

Walsh Spectrum Characterization on Sampling Distributions

- Following a rump talk by Yi LU at FSE'2017 in Japan, it is proposed as a suitable topic for submission to the Nature journal.
- Main problem statement is as follows.

Walsh Spectrum Characterization on Sampling Distributions

- Following a rump talk by Yi LU at FSE'2017 in Japan, it is proposed as a suitable topic for submission to the Nature journal.
- Main problem statement is as follows.

  Consider the sampling problem for a fixed, yet *unknown* source distribution D (or the so-called signal source). A few parameters:

Walsh Spectrum Characterization on Sampling Distributions

- Following a rump talk by Yi LU at FSE'2017 in Japan, it is proposed as a suitable topic for submission to the Nature journal.
- Main problem statement is as follows.

  Consider the sampling problem for a fixed, yet *unknown* source distribution D (or the so-called signal source). A few parameters: 1) the sample number is denoted by S,

Walsh Spectrum Characterization on Sampling Distributions

- Following a rump talk by Yi LU at FSE'2017 in Japan, it is proposed as a suitable topic for submission to the Nature journal.
- Main problem statement is as follows.

  Consider the sampling problem for a fixed, yet *unknown* source distribution D (or the so-called signal source). A few parameters:
  1) the sample number is denoted by S,
  2) the dimension of the signal source is denoted by 2^n,

Walsh Spectrum Characterization on Sampling Distributions

- Following a rump talk by Yi LU at FSE'2017 in Japan, it is proposed as a suitable topic for submission to the Nature journal.
- Main problem statement is as follows.

  Consider the sampling problem for a fixed, yet *unknown* source distribution D (or the so-called signal source). A few parameters:
  1) the sample number is denoted by S,
  2) the dimension of the signal source is denoted by 2^n,
  3) the Walsh spectrum of the source distribution is denoted by the three valued set $\{0, +d, -d\}$, where the value d and the number k of nonzero coefficients are unknown variables.

Walsh Spectrum Characterization on Sampling Distributions (cont'd)

- Given an input array $\mathbf{x} = (x_0, x_1, \ldots, x_{2^n-1})$ of 2^n reals in the time domain, the Walsh transform $\mathbf{y} = \widehat{\mathbf{x}} = (y_0, y_1, \ldots, y_{2^n-1})$ of $\mathbf{x}$ is

$$y_i \stackrel{\text{def}}{=} \sum_{j \in GF(2)^n} (-1)^{\langle i,j \rangle} x_j, \text{ for } i \in GF(2)^n.$$

Walsh Spectrum Characterization on Sampling Distributions (cont'd)

- Given an input array $\mathbf{x} = (x_0, x_1, \ldots, x_{2^n-1})$ of 2^n reals in the time domain, the Walsh transform $\mathbf{y} = \widehat{\mathbf{x}} = (y_0, y_1, \ldots, y_{2^n-1})$ of $\mathbf{x}$ is

$$y_i \stackrel{\text{def}}{=} \sum_{j \in GF(2)^n} (-1)^{\langle i,j \rangle} x_j, \text{ for } i \in GF(2)^n.$$

- The main problem asks to obtain *as precise and much knowledge as possible* about the signal source D from the sampling distribution D' using S samples.

Walsh Spectrum Characterization on Sampling Distributions (cont'd)

- Given an input array $\mathbf{x} = (x_0, x_1, \ldots, x_{2^n-1})$ of 2^n reals in the time domain, the Walsh transform $\mathbf{y} = \widehat{\mathbf{x}} = (y_0, y_1, \ldots, y_{2^n-1})$ of $\mathbf{x}$ is

$$y_i \stackrel{\text{def}}{=} \sum_{j \in GF(2)^n} (-1)^{\langle i,j \rangle} x_j, \text{ for } i \in GF(2)^n.$$

- The main problem asks to obtain *as precise and much knowledge as possible* about the signal source D from the sampling distribution D' using S samples.
- The main goal is to find out some large or even the largest nontrivial Walsh coefficient(s) and the index position(s) for D.

Important Comments

- This work is the follow-up result of [Lu-Desmedt'2016], [Lu'2016] and has origins in linear cryptanalysis (cf. [Lu-Vaudenay'2008], [Molland-Helleseth'2004]).

Important Comments

- This work is the follow-up result of [Lu-Desmedt'2016], [Lu'2016] and has origins in linear cryptanalysis (cf. [Lu-Vaudenay'2008], [Molland-Helleseth'2004]).
- Note that usually we have $S \ll 2^n$ and are dealing with the case of sparse large-dimensional signal in the time domain.

Important Comments

- This work is the follow-up result of [Lu-Desmedt'2016], [Lu'2016] and has origins in linear cryptanalysis (cf. [Lu-Vaudenay'2008], [Molland-Helleseth'2004]).
- Note that usually we have $S \ll 2^n$ and are dealing with the case of sparse large-dimensional signal in the time domain.
- In real life, three kinds of source distribution D are most interesting:

Important Comments

- This work is the follow-up result of [Lu-Desmedt'2016], [Lu'2016] and has origins in linear cryptanalysis (cf. [Lu-Vaudenay'2008], [Molland-Helleseth'2004]).
- Note that usually we have $S \ll 2^n$ and are dealing with the case of sparse large-dimensional signal in the time domain.
- In real life, three kinds of source distribution D are most interesting:

  1) the dimension 2^n is very large (e.g., 2^{64}),

Important Comments

- This work is the follow-up result of [Lu-Desmedt'2016], [Lu'2016] and has origins in linear cryptanalysis (cf. [Lu-Vaudenay'2008], [Molland-Helleseth'2004]).
- Note that usually we have $S \ll 2^n$ and are dealing with the case of sparse large-dimensional signal in the time domain.
- In real life, three kinds of source distribution D are most interesting:

  1) the dimension 2^n is very large (e.g., 2^{64}),

  2) Walsh spectrum is not just a three valued set,

Important Comments

- This work is the follow-up result of [Lu-Desmedt'2016], [Lu'2016] and has origins in linear cryptanalysis (cf. [Lu-Vaudenay'2008], [Molland-Helleseth'2004]).
- Note that usually we have $S \ll 2^n$ and are dealing with the case of sparse large-dimensional signal in the time domain.
- In real life, three kinds of source distribution D are most interesting:

  1) the dimension 2^n is very large (e.g., 2^{64}),
  2) Walsh spectrum is not just a three valued set,
  3) D is an un-normalized distribution.

Important Comments

- This work is the follow-up result of [Lu-Desmedt'2016], [Lu'2016] and has origins in linear cryptanalysis (cf. [Lu-Vaudenay'2008], [Molland-Helleseth'2004]).
- Note that usually we have $S \ll 2^n$ and are dealing with the case of sparse large-dimensional signal in the time domain.
- In real life, three kinds of source distribution D are most interesting:
  1) the dimension 2^n is very large (e.g., 2^{64}),
  2) Walsh spectrum is not just a three valued set,
  3) D is an un-normalized distribution.
- The proposed problem incorporates the case that the source distribution D has zeros in the time domain.

Outline

Background

Channel Capacity Calculation

Further Discussions

Conclusion

Motivation on Studying Channel Capacity

- Inspired by the idea of compressive sensing, [Lu'2015] first constructed *imaginary channel transition matrices* $T \stackrel{\text{def}}{=} p(y|x)$ of size 2×2 and $2 \times M$, and introduced Shannon's channel coding problem to statistical cryptanalysis.

Motivation on Studying Channel Capacity

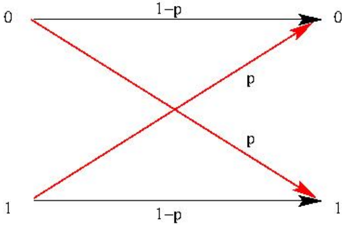
- Inspired by the idea of compressive sensing, [Lu'2015] first constructed *imaginary channel transition matrices* $T \stackrel{\text{def}}{=} p(y|x)$ of size 2×2 and $2 \times M$, and introduced Shannon's channel coding problem to statistical cryptanalysis.
- Case One: BSC (Binary Symmetric Channel)

$$T = \left[\begin{array}{cc} 1-p & p \\ p & 1-p \end{array} \right]$$

Motivation on Studying Channel Capacity

- Inspired by the idea of compressive sensing, [Lu'2015] first constructed *imaginary channel transition matrices* $T \stackrel{\text{def}}{=} p(y|x)$ of size 2×2 and $2 \times M$, and introduced Shannon's channel coding problem to statistical cryptanalysis.
- Case One: BSC (Binary Symmetric Channel)

$$T = \begin{bmatrix} 1-p & p \\ p & 1-p \end{bmatrix}$$



Motivation on Studying Channel Capacity (cont'd)

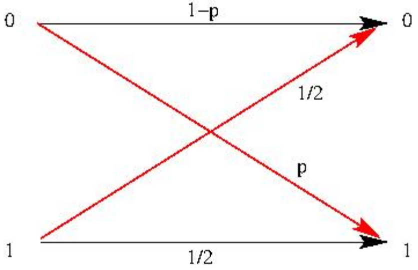
- Case Two: Non-Symmetric Binary Channel

$$T = \left[\begin{array}{cc} 1-p & p \\ 1/2 & 1/2 \end{array} \right]$$

Motivation on Studying Channel Capacity (cont'd)

- Case Two: Non-Symmetric Binary Channel

$$T = \begin{bmatrix} 1-p & p \\ 1/2 & 1/2 \end{bmatrix}$$



Motivation on Studying Channel Capacity (cont'd)

- Case Three: Non-Binary Non-square Channel

$$T = \begin{bmatrix} D \\ U \end{bmatrix},$$

D, U denote the source distribution and the uniform distribution over the binary vector space of dimension n respectively.

Motivation on Studying Channel Capacity (cont'd)

- Case Three: Non-Binary Non-square Channel
$$T = \begin{bmatrix} D \\ U \end{bmatrix},$$
D, U denote the source distribution and the uniform distribution over the binary vector space of dimension n respectively.
- Recall that the Channel Capacity with the transition matrix T, denoted by $C(T)$, invented by Shannon, describes the maximum rate (i.e., bits/transmission) to send information through the channel with an arbitrarily low error probability.

Motivation on Studying Channel Capacity (cont'd)

- Case Three: Non-Binary Non-square Channel
$$T = \left[\begin{array}{c} D \\ U \end{array} \right],$$
D, U denote the source distribution and the uniform distribution over the binary vector space of dimension n respectively.
- Recall that the Channel Capacity with the transition matrix T, denoted by $C(T)$, invented by Shannon, describes the maximum rate (i.e., bits/transmission) to send information through the channel with an arbitrarily low error probability.
- In above Case Three, $C(T)$ gives a perfect answer to the key question in cryptanalysis: What is the minimum number of data samples to distinguish one biased distribution from the uniform distribution?

The Famous Blahut-Arimoto Algorithm

- Due to independent works of [Arimoto'1972] and [Blahut'1972], the Blahut-Arimoto algorithm is known to efficiently calculate the capacity for the discrete memoryless channel (DMCs).

The Famous Blahut-Arimoto Algorithm

- Due to independent works of [Arimoto'1972] and [Blahut'1972], the Blahut-Arimoto algorithm is known to efficiently calculate the capacity for the discrete memoryless channel (DMCs).
- For the desired absolute accuracy ϵ of the capacity, Blahut-Arimoto algorithm solves the problem with transition matrix size $N \times M$ within time $O(MN^2 \log N/\epsilon)$.

The Famous Blahut-Arimoto Algorithm

- Due to independent works of [Arimoto'1972] and [Blahut'1972], the Blahut-Arimoto algorithm is known to efficiently calculate the capacity for the discrete memoryless channel (DMCs).
- For the desired absolute accuracy ϵ of the capacity, Blahut-Arimoto algorithm solves the problem with transition matrix size $N \times M$ within time $O(MN^2 \log N/\epsilon)$.
- Note that the most recent work [Sutter et al'2014] has the complexity $O(M^2 N \sqrt{\log N}/\epsilon)$ for the same problem.

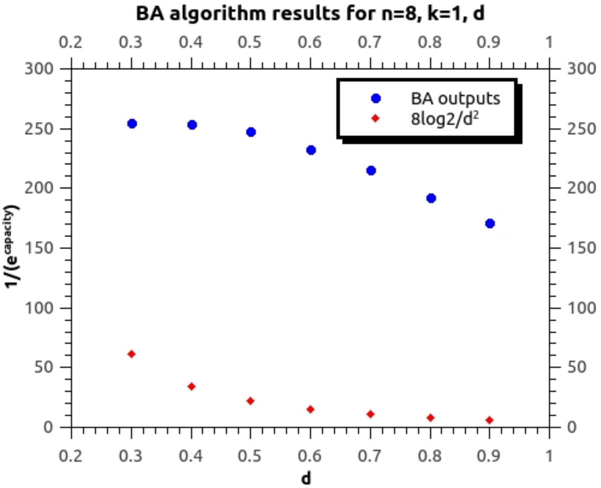
Blahut-Arimoto Algorithm in Pseudo-Codes

Input:

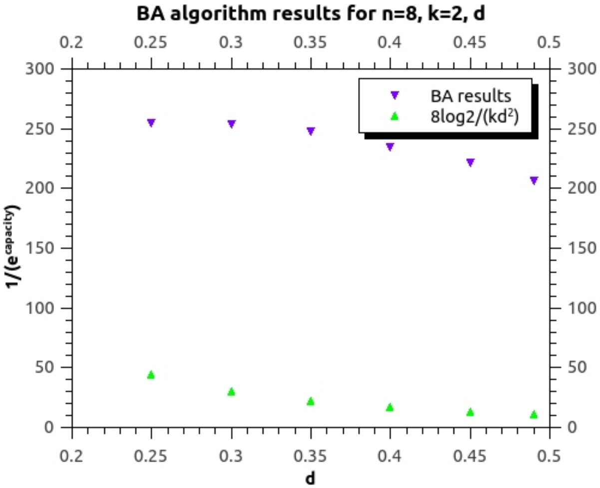
$Q_{k|j}$: transition matrix of size 2×2^n
(p_0, p_1): input distribution vector
ϵ : the desired absolute accuracy

initialize the values of $Q_{k|j}$ and p_0, p_1
repeat
  $c_0 \leftarrow exp\left(\sum_{k=0}^{2^n-1} Q_{k|0} \log \frac{Q_{k|0}}{p_0 Q_{k|0} + p_1 Q_{k|1}}\right)$
  $c_1 \leftarrow exp\left(\sum_{k=0}^{2^n-1} Q_{k|1} \log \frac{Q_{k|1}}{p_0 Q_{k|0} + p_1 Q_{k|1}}\right)$
  $I_L \leftarrow \log(p_0 c_0 + p_1 c_1)$
  $I_U \leftarrow \log \max(c_0, c_1)$
  update p_0 by $p_0 c_0 / (p_0 c_0 + p_1 c_1)$
  update p_1 by $p_1 c_1 / (p_0 c_0 + p_1 c_1)$
until $|I_U - I_L| < \epsilon$
output I_L

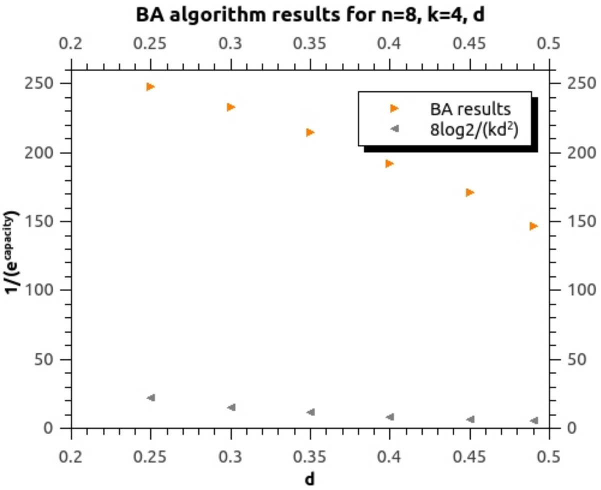
Capacity Results for $n = 8, k = 1$



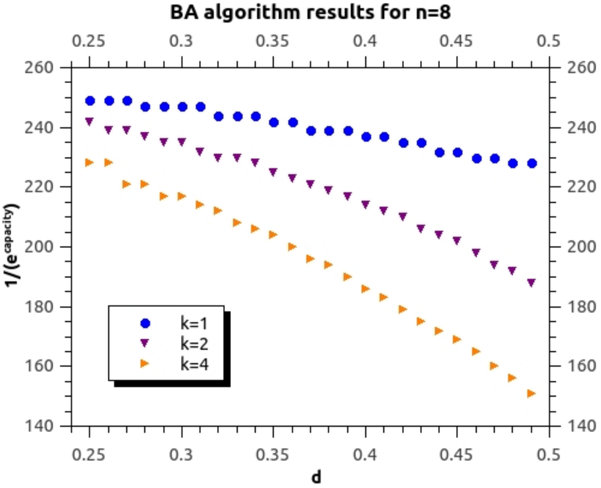
Capacity Results for $n = 8, k = 2$ (cont'd)



Capacity Results for $n = 8, k = 4$ (cont'd)



Capacity Results for $n = 8, \epsilon = 0.01$ (cont'd)



Outline

About High-Precision Numerical Computation Software

- From well-proved *paper* formulas/algorithms to *correct and efficient* computer implementations, we have a long road to go.

About High-Precision Numerical Computation Software

- From well-proved *paper* formulas/algorithms to *correct and efficient* computer implementations, we have a long road to go.
- In the new era of big data, high-precision numerical computation software is badly needed.

About High-Precision Numerical Computation Software

- From well-proved *paper* formulas/algorithms to *correct and efficient* computer implementations, we have a long road to go.
- In the new era of big data, high-precision numerical computation software is badly needed.
- Current available software and libraries with the feature:
  - MATHEMATICA
  - MATLAB
  - GNU Multiple Precision Arithmetic Library (GMP)
  - GNU Scientific Library (GSL)
  - etc.

Blahut-Arimoto Algorithm in Pseudo-Codes

Input:

$Q_{k|j}$: transition matrix of size 2×2^n
(p_0, p_1): input distribution vector
ϵ : the desired absolute accuracy

initialize the values of $Q_{k|j}$ and p_0, p_1
repeat
  $c_0 \leftarrow exp\left(\sum_{k=0}^{2^n-1} Q_{k|0} \log \frac{Q_{k|0}}{p_0 Q_{k|0} + p_1 Q_{k|1}}\right)$
  $c_1 \leftarrow exp\left(\sum_{k=0}^{2^n-1} Q_{k|1} \log \frac{Q_{k|1}}{p_0 Q_{k|0} + p_1 Q_{k|1}}\right)$
  $I_L \leftarrow \log(p_0 c_0 + p_1 c_1)$
  $I_U \leftarrow \log \max(c_0, c_1)$
  update p_0 by $p_0 c_0 / (p_0 c_0 + p_1 c_1)$
  update p_1 by $p_1 c_1 / (p_0 c_0 + p_1 c_1)$
until $|I_U - I_L| < \epsilon$
output I_L

Inspection on BA Capacity Calculations with $n = 8, k = 1, d = 0.25, \epsilon = 0.1$

- With $p_0 = 0.8, p_1 = 0.2$, BA algorithm *luckily* terminates with only one iteration for $n = 8, k = 1, d = 0.25, \epsilon = 0.1$.

Inspection on BA Capacity Calculations with $n = 8, k = 1, d = 0.25, \epsilon = 0.1$

- With $p_0 = 0.8, p_1 = 0.2$, BA algorithm *luckily* terminates with only one iteration for $n = 8, k = 1, d = 0.25, \epsilon = 0.1$.
- This encourages us to inspect the calculation details in order to check the precision of the results.

Inspection on BA Capacity Calculations with $n = 8, k = 1, d = 0.25, \epsilon = 0.1$

- With $p_0 = 0.8, p_1 = 0.2$, BA algorithm *luckily* terminates with only one iteration for $n = 8, k = 1, d = 0.25, \epsilon = 0.1$.
- This encourages us to inspect the calculation details in order to check the precision of the results.
- Check the value of c_1:

$$\log(c_1) = -8\log(2) - 2^{-8} \approx -5.54\underline{9}.$$

Inspection on BA Capacity Calculations with $n = 8, k = 1, d = 0.25, \epsilon = 0.1$

- With $p_0 = 0.8, p_1 = 0.2$, BA algorithm *luckily* terminates with only one iteration for $n = 8, k = 1, d = 0.25, \epsilon = 0.1$.
- This encourages us to inspect the calculation details in order to check the precision of the results.
- Check the value of c_1:

$$\log(c_1) = -8\log(2) - 2^{-8} \approx -5.54\underline{9}.$$

- Check the value of $c_0 = exp(TMP1 - TMP2)$:

$$TMP1 = \frac{3}{8}\log(\frac{3}{1024}) + \frac{5}{8}\log(\frac{5}{1024}) \quad (1)$$

$$TMP2 = \frac{42 \times 0.8}{8 \times 1024} = \frac{4.2}{2^{10}} \quad (2)$$

Inspection on BA Capacity Calculations with $n = 8, k = 1, d = 0.25, \epsilon = 0.1$ (cont'd)

To finalize,

- check the value of I_U:

$$\begin{aligned} \log c_0 &= TMP1 - TMP2 = -5.51\underline{3} \\ I_U &= \max(-5.51\underline{3}, -5.54\underline{9}) = -5.51\underline{3} \end{aligned}$$

Inspection on BA Capacity Calculations with $n = 8, k = 1, d = 0.25, \epsilon = 0.1$ (cont'd)

To finalize,

- check the value of I_U:

$$\begin{aligned} \log c_0 &= TMP1 - TMP2 = -5.51\underline{3} \\ I_U &= \max(-5.51\underline{3}, -5.54\underline{9}) = -5.51\underline{3} \end{aligned}$$

- check the value of I_L:

$$I_L = \log(0.8 \times e^{-5.51\underline{3}} + 0.2 \times e^{-5.54\underline{9}}) = \log(e^{-5.5\underline{X}}) = -5.5\underline{X}\,, \quad (3)$$

as $\log(\cdot)$ and $exp(\cdot)$ both increase with the input.

Inspection on BA Capacity Calculations with $n = 8, k = 1, d = 0.25, \epsilon = 0.1$ (cont'd)

To finalize,

- check the value of I_U:

$$\begin{aligned} \log c_0 &= TMP1 - TMP2 = -5.51\underline{3} \\ I_U &= \max(-5.51\underline{3}, -5.54\underline{9}) = -5.51\underline{3} \end{aligned}$$

- check the value of I_L:

$$I_L = \log(0.8 \times e^{-5.51\underline{3}} + 0.2 \times e^{-5.54\underline{9}}) = \log(e^{-5.5\underline{X}}) = -5.5\underline{X}\,, \quad (3)$$

  as $\log(\cdot)$ and $exp(\cdot)$ both increase with the input.
- As $|I_U - I_L| < 0.1$, we now know $I_L = -5.5\underline{X}$.

Inspection on BA Capacity Calculations with $n = 8, k = 1, d = 0.25, \epsilon = 0.1$ (cont'd)

To finalize,

- check the value of I_U:

$$\begin{aligned} \log c_0 &= TMP1 - TMP2 = -5.51\underline{3} \\ I_U &= \max(-5.51\underline{3}, -5.54\underline{9}) = -5.51\underline{3} \end{aligned}$$

- check the value of I_L:

$$I_L = \log(0.8 \times e^{-5.51\underline{3}} + 0.2 \times e^{-5.54\underline{9}}) = \log(e^{-5.5\underline{X}}) = -5.5\underline{X}\,, \quad (3)$$

  as $\log(\cdot)$ and $exp(\cdot)$ both increase with the input.
- As $|I_U - I_L| < 0.1$, we now know $I_L = -5.5\underline{X}$.
- Meanwhile, the computer running BA algorithm also outputs I_L: "-5.5", i.e., to be interpreted as $]-5.5-0.1, -5.5+0.1[$.

Comments

- With previous parameters, we have justified that capacity $\in]-5.6, -5.4[$.

Comments

- With previous parameters, we have justified that capacity $\in]-5.6, -5.4[$.
- As the number of transmissions per bit with arbitrarily small error probability is a critical quantity, we are mostly concerned with the value of

  $$1/(e^{\text{capacity}}) \in]244 - 23, 244 + 26[$$

  due to $e^{5.4} = 221.\underline{X}$, $e^{5.5} = 244.\underline{X}$, $e^{5.6} = 270.\underline{X}$.

Comments

- With previous parameters, we have justified that capacity $\in]-5.6, -5.4[$.
- As the number of transmissions per bit with arbitrarily small error probability is a critical quantity, we are mostly concerned with the value of

$$1/(e^{\text{capacity}}) \in]244-23, 244+26[$$

due to $e^{5.4} = 221.\underline{X}$, $e^{5.5} = 244.\underline{X}$, $e^{5.6} = 270.\underline{X}$.
- For lower value of ϵ and $k > 1$, manual checking becomes harder for (1-3).

Comments

- With previous parameters, we have justified that capacity $\in]-5.6, -5.4[$.
- As the number of transmissions per bit with arbitrarily small error probability is a critical quantity, we are mostly concerned with the value of

$$1/(e^{\text{capacity}}) \in]244 - 23, 244 + 26[$$

due to $e^{5.4} = 221.\underline{X}$, $e^{5.5} = 244.\underline{X}$, $e^{5.6} = 270.\underline{X}$.

- For lower value of ϵ and $k > 1$, manual checking becomes harder for (1-3).
- Open Question:
  Evaluate the output precision of a composite function, which has exact values of inputs *initially*.

Conclusion

- We have implemented the efficient BA capacity calculation algorithm for the transition matrix of size $2 \times M$.

Conclusion

- We have implemented the efficient BA capacity calculation algorithm for the transition matrix of size $2 \times M$.
- Our implementation allows to solve a lower-bound for distinguishing two distributions with arbitrarily small error probability.

Conclusion

- We have implemented the efficient BA capacity calculation algorithm for the transition matrix of size $2 \times M$.
- Our implementation allows to solve a lower-bound for distinguishing two distributions with arbitrarily small error probability.
- We have done experiments in the setting of Sparse Walsh Spectrum with $M = 2^8, \epsilon = 0.01, k = 1, 2, 4$ and one distribution is a uniform distribution.

Conclusion (cont'd)

- In typical Crypto setting, we notice that the capacity is a *negative* value, which differs from the real world communication channels.

Conclusion (cont'd)

- In typical Crypto setting, we notice that the capacity is a *negative* value, which differs from the real world communication channels.
- We have examined the important issue of calculation precision with $M = 2^8, \epsilon = 0.1, k = 1$.

Conclusion (cont'd)

- In typical Crypto setting, we notice that the capacity is a *negative* value, which differs from the real world communication channels.
- We have examined the important issue of calculation precision with $M = 2^8, \epsilon = 0.1, k = 1$.
- We are carrying out challenging large-scale experiments with larger M and more values of k.

References

- S. Arimoto, "An Algorithm for Computing the Capacity of Arbitrary Discrete Memoryless Channels," IEEE Trans. Inform. Theory, IT-18: 14-20, 1972.
- R. Blahut, "Computation of Channel Capacity and Rate Distortion Functions," IEEE Trans. Inform. Theory, IT-18: 460-473, 1972.
- X. Chen, D. Guo, "Robust Sublinear Complexity Walsh-Hadamard Transform with Arbitrary Sparse Support", in Proc. IEEE Int. Symp. Information Theory, 2015.
- T. M. Cover, J. A. Thomas. Elements of Information Theory. John Wiley & Sons, Second Edition, 2006.
- X. Li, J. K. Bradley, S. Pawar, K. Ramchandran, "SPRIGHT: A Fast and Robust Framework for Sparse Walsh-Hadamard Transform", arXiv:1508.06336, 2015.
- Y. Lu, Y. Desmedt, "Walsh-Hadamard Transform and Cryptographic Applications in Bias Computing", `https://eprint.iacr.org/2016/419`, 2016.
- Y. Lu, "Walsh Sampling with Incomplete Noisy Signals", arXiv preprint, `arxiv.org/abs/1602.00095`, 2016.
- Y. Lu, "Practical Tera-scale Walsh-Hadamard Transform", `http://ieeexplore.ieee.org/document/7821757/`, 2016.
- R. Scheibler, S. Haghighatshoar, M. Vetterli, "A Fast Hadamard Trans- form for Signals With Sublinear Sparsity in the Transform Domain", IEEE Transactions on Information Theory, vol. 61, no. 4, 2015.
- D. Sutter, P. M. Esfahani, T. Sutter, J. Lygeros, "Efficient Approximation of Discrete Memoryless Channel Capacities," IEEE Int. Symp. Information Theory, pp. 2904 - 2908, 2014.
- S. Vaudenay, "A Direct Product Theorem," draft.
- GSL - GNU Scientific Library (version 2.3), `https://www.gnu.org/software/gsl/`.
- GNU MP - The GNU Multiple Precision Arithmetic Library (version 6.0.0), `https://gmplib.org/`.