An Algorithm for Computing the Capacity of Symmetrized KL Information for Discrete Channels

Haobo Chen
Department of ECE
University of Florida
Gainesville, US
haobo.chen@ufl.edu

Gholamali Aminian
The Alan Turing Institute
London, UK
gaminian@turing.ac.uk

Yuheng Bu
Department of ECE
University of Florida
Gainesville, US
buyuheng@ufl.edu

2024 60th Annual Allerton Conference on Communication, Control, and Computing (Allerton) | 979-8-3315-4103-3/24/$31.00 USG | DOI: 10.1109/ALLERTON63246.2024.10735330

***Abstract*—Symmetrized Kullback-Leibler (KL) information (I_{SKL}), which symmetrizes the traditional mutual information by integrating Lautum information, has been shown as a critical quantity in communication [1] and learning theory [2]. This paper considers the problem of computing the capacity in terms of I_{SKL} for a fixed discrete channel. Such a maximization problem is reformulated into a discrete quadratic optimization with a simplex constraint. One major challenge here is the non-concavity of Lautum information, which complicates the optimization problem. Our method involves symmetrizing the KL divergence matrix and applying iterative updates to ensure a non-decreasing update while maintaining a valid probability distribution. We validate our algorithm on Binary symmetric Channels and Binomial Channels, demonstrating its consistency with theoretical values. Additionally, we explore its application in machine learning through the Gibbs channel, showcasing the effectiveness of our algorithm in finding the worst-case data distributions.**

***Index Terms*—Symmetrized KL information, channel capacity, quadratic optimization, Gibbs channel**

I. Introduction

Kullback-Leibler (KL) divergence, as a widely adopted measure in information theory, quantifies the difference between two probability distributions. Specifically, the KL divergence between two probability measures P and Q over a space $\mathcal{X}$, where P is absolutely continuous with respect to Q, is defined as

$$D(P\|Q) = \int_{\mathcal{X}} P(x) \log \frac{P(x)}{Q(x)}\, dx. \tag{1}$$

However, KL divergence is not symmetric in general, i.e., $D(P\|Q) \neq D(Q\|P)$, which limits its applicability in some contexts. To address this issue, the symmetrized KL divergence, or Jeffrey's divergence [3], has been introduced to provide a symmetric measure of distribution similarity, which is defined as

$$D_{\mathrm{SKL}}(P\|Q) \triangleq D(P\|Q) + D(Q\|P). \tag{2}$$

This symmetrized version of KL divergence exhibits inherent symmetry and simplicity, making it applicable in various scenarios as a measure of similarity for distributions, e.g. see [1], [2], [4], [5].

The mutual information between two random variables X and Y is defined as the KL divergence between their joint distribution and the product of their marginal distributions:

$$I(X;Y) \triangleq D(P_{XY}\|P_X P_Y). \tag{3}$$

Similarly, the symmetrized KL information (I_{SKL}) between X and Y is given by:

$$I_{\mathrm{SKL}}(X;Y) \triangleq D_{\mathrm{SKL}}(P_{XY}\|P_X P_Y) = I(X;Y) + L(X;Y),$$

where $L(X;Y) \triangleq D(P_X P_Y\|P_{XY})$ denotes the Lautum information [6].

This paper explores the capacity problem with symmetrized KL information, i.e., maximizing I_{SKL} over a fixed channel, which can be applied to bound channel capacity [1] and understand the worst-case generalization error as in [2].

Maximizing I_{SKL} poses significant challenges due to the non-concave nature of the Lautum information with respect to the input distribution. Therefore, traditional methods for maximizing mutual information, e.g., Arimoto-Blahut algorithm [7], which rely on the concavity of the problem, are not directly applicable here.

To address the non-concavity issue, we reformulate the problem into a quadratic matrix optimization over probability simplex. Specifically, in Section IV-A, we derive a matrix representation of I_{SKL} in the discrete case. We further symmetrize this matrix representation and apply iterative updates with element-wise normalization, ensuring adherence to the simplex constraints. This iterative algorithm, detailed in Section IV-B, is designed to handle the non-concave nature of the problem, guaranteeing a monotonic update in terms of I_{SKL}.

We apply our algorithm to various channel models, including the Binary Symmetric Channel (BSC) and Binomial Channel, validating the proposed algorithm with empirical data. The results presented in Section V show a strong alignment between the calculated capacities and theoretical values, demonstrating the effectiveness of our approach. Additionally, we explore the implications of our method in a machine-learning context through the Gibbs channel, showcasing its versatility in finding the worst-case data distributions as in Section VI.

II. Preliminaries

A. Information measures

We note that the symmetrized KL divergence also belongs to the f-divergence family [8], with $f(x) = x\log x - \log x$. As an f-divergence, symmetrized KL divergence has the following variational representation,

$$D_f(P\|Q) = \sup_h \mathbb{E}_P[h(X)] - \mathbb{E}_Q[f^*(h(X))], \tag{4}$$

where $f^*(\cdot)$ denotes the Legendre transform of function f.

Total variation (TV) is another important measure in information theory and statistics that quantifies the difference between two probability distributions. It is defined as:

$$\mathrm{TV}(P,Q) \triangleq \sup_A |P(A) - Q(A)|, \tag{5}$$

where the supremum is taken over all measurable sets A.

B. Capacity

The concept of channel capacity is crucial in information theory, as it captures the fundamental limits of communication. It is well-known that channel capacity can be characterized by maximizing mutual information for the channel $P_{Y|X}$:

$$C = \max_{P_X} I(X;Y), \tag{6}$$

where the maximization is over all possible input distributions defined over $\mathcal{X}$.

Analogous to channel capacity, the capacity using symmetrized KL information for a fixed channel $P_{Y|X}$ is

$$C_{\mathrm{SKL}} = \max_{P_X \in \mathcal{P}} I_{\mathrm{SKL}}(X;Y), \tag{7}$$

where the maximization is over some convex set of input distributions $\mathcal{P}$. Such a capacity has different interpretations in various fields.

C_{SKL} plays a significant role in bounding channel capacity. It has been shown [1] that for any channel the capacity C can be upper bounded by:

$$C \leq C_{\mathrm{SKL}}. \tag{8}$$

In [1], the authors demonstrated the similarity between the capacities defined by mutual information and symmetrized KL information in various channels, including the Binary Symmetric Channel and point-to-point Gaussian channels. Their study also shows that the capacity of memoryless Poisson channels, based on symmetrized KL divergence, is computable. Furthermore, numerical results suggest that this upper bound is effective for channels in molecular communication with small capacities. By rigorously analyzing the point-to-point Gaussian channel and the Poisson channel, they demonstrated the broad applicability of this upper bound across different types of communication channels.

Besides its role in bounding channel capacity, C_{SKL} can be interpreted as the worst-case generalization error for the Gibbs channel within the context of learning theory. It is shown in [2], [5] that the generalization error can be characterized using the I_{SKL} between the input training data and the learned weights. Therefore, the capacity-achieving input distribution for I_{SKL} can be interpreted as the worst-case data input distribution. This concept aligns with findings in the literature that generalization behavior is highly data-dependent, and a single learning algorithm does not work for all kinds of training data. Thus, examining the capacity of symmetrized KL information provides significant insights into understanding the data-dependent nature of generalization in learning theory.

C. Related Works

Channel Capacity Computation: In the context of maximizing mutual information, it is crucial to design an effective capacity computation algorithm for various scenarios. For discrete channels, most algorithms rely on iterative numerical methods. These include the famous Arimoto-Blahut algorithm [7], [9], linearly constrained optimization approaches [10], [11], and simulation-based numerical algorithms [12], [13]. These methods often require an exact characterization of the discrete channel. Therefore, these conventional algorithms cannot be easily extended to general continuous channels, where only pairs of training samples for channel input and output are available. To resolve such a limitation, a sample-based mutual information estimator is essential for handling such capacity estimation problems.

Channel Capacity Estimation: Channel capacity estimation can be viewed as two fundamental tasks: (a) estimating the mutual information using samples of the channel input and output and (b) maximizing this mutual information with respect to the channel input distribution. Traditional methods for the former task include binning [14], non-parametric kernel estimation [15], [16], and Gaussian distribution approximation [17]. However, these traditional methods lack scalability and struggle with large sample sizes and dimensions, particularly for high-dimensional data. The latter task can be addressed using gradient descent methods for differentiable mutual information estimators. With advancements in deep learning, significant progress has been made in mutual information estimation. Recent work combines variational methods with neural networks to construct neural network-based estimators of mutual information [18]–[25]. These methods have also been applied to channel capacity estimation, leveraging high-dimensional encoders with neural network [26]–[28].

III. Challenges

Both discrete algorithm and neural network-based continuous methods for computing channel capacity rely on the concavity of mutual information over input distribution for a fixed channel. In [6], it was claimed that Lautum information is concave for a fixed $P_{Y|X}$.

However, upon re-examining the proof steps, we show that Lautum information is not concave.

Theorem 1. *For a fixed channel $P_{Y|X}$, Lautum information $L(X;Y)$ is not concave with respect to the input distribution P_X.*

The non-concavity of Lautum information is due to the fact that the chain rule for mutual information, which is crucial for demonstrating concavity, does not hold for Lautum information [6]. The detailed proof of this result is provided in Appendix A. Therefore, the C_{SKL} problem, which integrates mutual information with Lautum Information, is not concave, presenting significant challenges.

We also note that variational methods that maximize mutual information using the variational representation of f-divergence cannot be directly applied to symmetrized KL information. In particular, the Legendre transform of $f(x) = x\log x - \log x$ is $f^*(\cdot) = \exp(t + \mathrm{LambertW}(\exp(1-t)) - 1)$, which relies on the Lambert W function [29], presents computational challenges for iterative updates using neural networks [18].

IV. Proposed Method

In this section, we tackle the challenges associated with the capacity problem of symmetrized KL information for discrete channels. We propose a novel approach, transforming the problem into a quadratic optimization problem and introducing an iterative algorithm to ensure that the objective function is always non-decreasing.

A. Matrix Representation of I_{SKL} in the Discrete Case

To simplify the capacity problem in (7), we consider an alternative expression for I_{SKL} as shown in the following proposition.

Proposition 1. *For fixed channel $P_{Y|X}$, I_{SKL} can be expressed equivalently as*

$$I_{\mathrm{SKL}}(X;Y) = \sum_{x,\tilde{x}} P_X(x)P_X(\tilde{x})D\left(P_{Y|X=x}\|P_{Y|X=\tilde{x}}\right), \quad (9)$$

where $\tilde{X}$ denotes an independent copy of X, sharing the same distribution P_X.

The detailed proof can be found in [30] Appendix B. Using Proposition 1, we can reformulate the capacity problem in (7) into a quadratic matrix optimization problem subject to simplex constraints. To see this, we begin by defining the vector $\boldsymbol{X} \in \Delta^{d-1}$, where Δ^{d-1} denotes the probability simplex with dimension $d-1$. If the size of the space $|\mathcal{X}| = d$, then each element $\boldsymbol{X}_i$ in vector $\boldsymbol{X}$ represents the probability $P_X(x_i)$ for $x_i \in \mathcal{X}$. Next, we define the matrix $\boldsymbol{C}$, which is an $d \times d$ non-negative matrix where each entry C_{ij} represents the KL divergence $D(P_{Y|X=x_i}\|P_{Y|X=x_j})$.

With these notations, by Proposition 1, I_{SKL} can be written in matrix form,

$$I_{\mathrm{SKL}}(X;Y) = \boldsymbol{X}^\top \boldsymbol{C}\boldsymbol{X}. \quad (10)$$

To ensure that $\boldsymbol{X} \in \Delta^{d-1}$, which represents a valid probability distribution, we impose the following simplex constraints: $\boldsymbol{X}^\top \mathbf{1} = 1$ and $\boldsymbol{X} \geq \mathbf{0}$, where $\mathbf{1}$ is an d-dimensional all-one vector.

Combining all these components, we obtain the following equivalent quadratic matrix optimization problem:

$$\begin{aligned} \max_{\boldsymbol{X}\in\mathbb{R}} \quad & \boldsymbol{X}^\top \boldsymbol{C}\boldsymbol{X} \\ \text{subject to} \quad & \boldsymbol{X}^\top \mathbf{1} = 1, \\ & \boldsymbol{X} \geq \mathbf{0}. \end{aligned} \quad (11)$$

This reformulation leverages the properties of the KL divergence and probability distributions, providing a clear path for optimization. Noteworthy observations include:

(a) The matrix $\boldsymbol{C}$ is not symmetric and contains non-negative elements, since the KL divergence $D(P_{Y|X=x_i}\|P_{Y|X=x_j})$ are asymmetric and non-negative;
(b) As $D(P_{Y|X=x}\|P_{Y|X=x}) = 0$, all diagonal elements of $\boldsymbol{C}$ are zero;
(c) The matrix $\boldsymbol{C}$ is not definite, as the capacity problem for I_{SKL} is not concave nor convex.

Therefore, this quadratic optimization problem is not directly solvable, necessitating the exploration of specific problem structures to design an appropriate algorithm.

B. Max-SKL Algorithm

In this section, we provide a novel iterative algorithm for solving the optimization problem in (11), which ensures that the objective function is always non-decreasing.

First, we symmetrize the KL divergence matrix $\boldsymbol{C}$ in (11) by averaging it with its transpose:

$$\boldsymbol{C}_{\mathrm{sym}} = \frac{1}{2}(\boldsymbol{C} + \boldsymbol{C}^\top). \quad (12)$$

It is straightforward to verify that $\boldsymbol{X}^\top \boldsymbol{C}\boldsymbol{X} = \boldsymbol{X}^\top \boldsymbol{C}_{\mathrm{sym}}\boldsymbol{X}$, and we will only work with symmetric $\boldsymbol{C}_{\mathrm{sym}}$ in the following.

Here, we notice that this non-definite simplex-constrained quadratic problem is related to the variational representation of the eigen-problem [31]. In particular, for any symmetric matrix $\boldsymbol{A}$, the maximum of the following optimization problem with a norm constraint

$$\begin{aligned} \max_{\boldsymbol{X}} \quad & \boldsymbol{X}^\top \boldsymbol{A}\boldsymbol{X} \\ \text{subject to} \quad & \|\boldsymbol{X}\|_2 = 1, \end{aligned} \quad (13)$$

is the largest eigenvalue of $\boldsymbol{A}$, which is achieved when $\boldsymbol{X}$ equals to the corresponding eigenvector. Furthermore, as $\boldsymbol{C}_{\mathrm{sym}}$ is a non-negative symmetric matrix, Perron-Frobenius theorem [31, Theorem 8.2.2] states that its dominant eigenvector is element-wise non-negative, and the constraint $\boldsymbol{X} \geq \mathbf{0}$ is automatically satisfied.

Therefore, the optimization problem in (11) can be viewed as replacing the norm constraint in (13) with a simplex constraint. We anticipate that existing algorithms for finding the largest eigenvector can be adapted to solve (11). We then introduce our algorithm for updating probability distributions within the simplex constraint, which is motivated by the power iteration method.

Power iteration is a well-known algorithm used to find the dominant eigenvector of a matrix through iterative updates and normalization [32]. Specifically, the power iteration algorithm proceeds as follows. Given a matrix $\boldsymbol{A}$, the algorithm starts with an initial vector $\boldsymbol{x}_0$ and iteratively updates it with the following rule

$$\boldsymbol{x}_{k+1} = \frac{\boldsymbol{A}\boldsymbol{x}_k}{\|\boldsymbol{A}\boldsymbol{x}_k\|}. \tag{14}$$

This process will converge when $\boldsymbol{x}_k$ reaches the dominant eigenvector of $\boldsymbol{A}$. However, this normalization by the vector norm only guarantees the norm constraint in (13) but cannot ensure that the sum of all elements in $\boldsymbol{x}_k$ is one.

Our algorithm leverages a similar iterative process but is specifically designed for the simplex constraint. In the proposed iterative update, the following operations are performed:

(a) Compute $\boldsymbol{C}_{\text{sym}}\boldsymbol{X}_k$, which is the product of the symmetrized matrix and the current probability distribution vector.
(b) Update the probability distribution by multiplying the current distribution $\boldsymbol{X}_k$ *element-wise* with $\boldsymbol{C}_{\text{sym}}\boldsymbol{X}_k$.
(c) Normalize the updated probability distribution to ensure that the probabilities sum to 1.

In particular, starting with an initial $\boldsymbol{X}_0 \in \Delta^{d-1}$, we consider the element-wise update rules

$$[\boldsymbol{X}_{k+1}]_i = \frac{[\boldsymbol{X}_k]_i \cdot [(\boldsymbol{C}_{\text{sym}}\boldsymbol{X}_k)]_i}{\boldsymbol{X}_k^\top \boldsymbol{C}_{\text{sym}}\boldsymbol{X}_k}, \quad i = 1, \ldots, d, \tag{15}$$

where $[\cdot]_i$ denotes the i-th element of the vector. As both $\boldsymbol{X}_k$ and $\boldsymbol{C}_{\text{sym}}$ are non-negative, the resulting solution will satisfy $\boldsymbol{X} \geq \boldsymbol{0}$ automatically. This approach leverages the structure of the symmetrized matrix $\boldsymbol{C}_{\text{sym}}$ to find a solution that satisfies the non-negativity and simplex constraints.

In addition, the following lemma from [33] ensures the monotonicity and convergence of the proposed iterative process, which guarantees that $\boldsymbol{X}_{k+1}^\top \boldsymbol{C}_{\text{sym}}\boldsymbol{X}_{k+1} \geq \boldsymbol{X}_k^\top \boldsymbol{C}_{\text{sym}}\boldsymbol{X}_k$.

Proposition 2 ([33, Section 3])**.** *Consider a symmetric $d \times d$ matrix $\boldsymbol{C}_{sym} = (C_{ij})$ with non-negative elements $C_{ij} \geq 0$. Let x_i^k be the probability of the i-th element $(i = 1, 2, \ldots, d)$ of $\boldsymbol{X}_k$ in iteration k, such that $\sum_{i=1}^d x_i = 1$ and $0 < x_i < 1$.*

Let

$$I_{\text{SKL}}^k = \boldsymbol{X}_k^\top \boldsymbol{C}_{sym}\boldsymbol{X}_k = \sum_{i=1}^d \sum_{j=1}^d C_{ij} x_i^k x_j^k, \tag{16}$$

$$I_{\text{SKL}}^{k+1} = \boldsymbol{X}_{k+1}^\top \boldsymbol{C}_{sym}\boldsymbol{X}_{k+1}. \tag{17}$$

Then we have the following inequality,

$$I_{\text{SKL}}^{k+1} - I_{\text{SKL}}^k \geq 0. \tag{18}$$

This Proposition validates the convergence and correctness of the update rules used in our algorithm, ensuring that the objective is always non-deceasing under the given constraints. Our algorithm will terminate when the total variation between $\boldsymbol{X}_k$ and $\boldsymbol{X}_{k+1}$ is smaller than a threshold $\epsilon > 0$. The overall description of the proposed Max-SKL algorithm can be found in Algorithm 1.

Algorithm 1 Max-SKL algorithm

$\boldsymbol{C} \leftarrow$ Compute the KL divergence matrix with all elements $C_{ij} = D(P_{Y|X=x_i} \| P_{Y|X=x_j})$
$\boldsymbol{C} \leftarrow \frac{1}{2}(\boldsymbol{C} + \boldsymbol{C}^\top)$
$\boldsymbol{X}_0 \leftarrow$ Initialize the probability distribution over $\mathcal{X}$
for $k \leftarrow 0$ **to** $maxIter - 1$ **do**
    $CP \leftarrow \boldsymbol{C} \cdot \boldsymbol{X}_k$ ▷ Matrix-vector product
    $\boldsymbol{X}_k^{'} \leftarrow \boldsymbol{X}_k * CP$ ▷ Element-wise product
    $\boldsymbol{X}_{k+1} \leftarrow \boldsymbol{X}_k^{'} / (\boldsymbol{X}_k^\top \cdot CP)$
    if $\text{TV}(\boldsymbol{X}_{k+1}, \boldsymbol{X}_k) \leq \epsilon$ **then**
        break ▷ Convergence check
    end if
end for
return $\boldsymbol{X}_{k+1}$, $\boldsymbol{X}^T \boldsymbol{C}_{\text{sym}} \boldsymbol{X}$

Remark 1 (Comparison with Power iteration algorithm)**.** *Our algorithm shares similarities with the power iteration method but also has distinct differences. The power iteration algorithm aims to find the dominant eigenvector of a matrix, utilizing direct matrix-vector multiplication and normalization using vector norm. However, it does not guarantee that the sum of the elements of the vector remains one after each iteration. In contrast, our proposed algorithm ensures that the sum of the probability distribution vector elements is one by using a normalization based on the element-wise product.*

V. Experiments with Discrete Channels

To demonstrate the efficacy of our Algorithm 1, we conduct various experiments under different settings. These experiments showcase the algorithm's ability to handle different types of channels and input distributions, providing a reliable tool for optimizing symmetrized KL information in practical applications.

A. Binary Symmetric Channel

To validate the proposed Algorithm 1 for finding the symmetric KL capacity, we conduct an experiment for the Binary Symmetric Channel (BSC). As shown in [1], the theoretical capacity of a BSC with crossover probability of $p \in [0, 1]$ is given by:

$$C_{\text{SKL}}(p) = -\log_2\left(\sqrt{p(1-p)}\right) - h(p), \tag{19}$$

where $h(p)$ is the binary entropy function. The theoretical C_{SKL} reaches its maximum at a uniform input probability distribution. Our goal is to validate our algorithm by computing the capacity and comparing it with theoretical values.

In Figure 1, We compare the calculated capacities by Algorithm 1 with the theoretical values $C_{\text{SKL}}(p)$ using (19) for different crossover probabilities p ranging from 0.1 to 0.9. Figure 1 shows an excellent agreement between the two, which confirms that our algorithm effectively maximizes the symmetrized KL information.

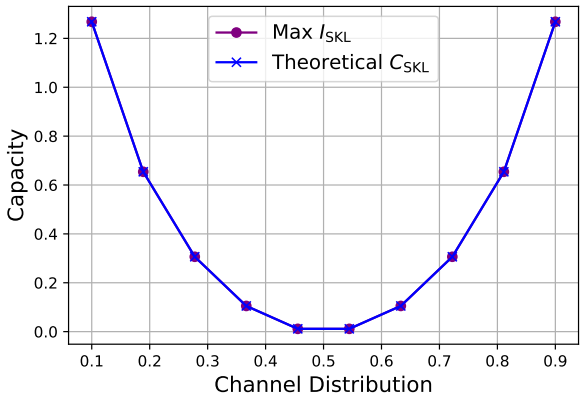

Fig. 1. Comparison of theoretical values and calculated symmetric KL capacities using Max-SKL algorithm for the BSC. The experiment varies the channel distribution p to validate that our algorithm can accurately compute the theoretical I_{SKL}.

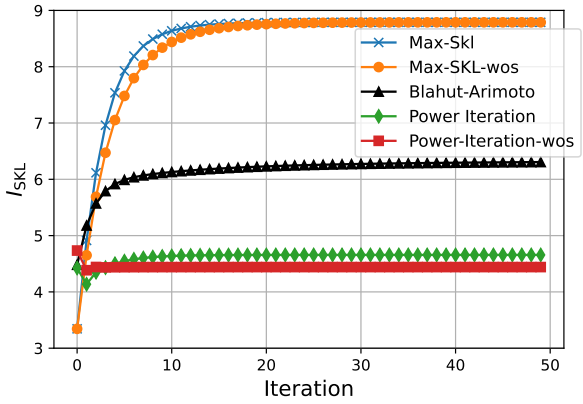

Fig. 2. Convergence of Max-SKL and Power Iteration, and comparison with the result of the Blahut-Arimoto algorithm to calculate I_{SKL}. Our algorithm shows successful convergence in the binomial channel, with the Max-SKL using a symmetrizing step demonstrating the best performance.

The capacity-achieving input distribution (caid) of C_{SKL} identified by our algorithm is a uniform distribution, which corresponds to the theoretical caid for C_{SKL} and is the same as the caid of traditional capacity with mutual information. To further validate our algorithm, we conduct additional experiments on Binary Asymmetric Channel (BAC) in [30] Appendix C to show that the capacity-achieving input distributions can be different for different capacities.

B. Binomial Channel

We consider a binomial channel [1] where the input of the channel X is a continuous value in the range $\mathcal{X} = [0, 1]$ and the output Y is a discrete variable taking values in $\{0, 1, \ldots, n\}$. The relationship between X and Y is described by the binomial distribution:

$$P(Y = y|X = x) = \binom{n}{y} x^y (1 - x)^{n-y}. \tag{20}$$

For our experiments, we quantize the input space $\mathcal{X}$ ranging from 0.1 to 0.9 with increments of 0.1 so that it is a discrete channel. The channel $P(Y|X)$ is calculated using the binomial distribution in (20) with $n = 10$.

We consider two baseline algorithms in the comparison:

- Arimoto-Blahut algorithm, which is designed to maximize the mutual information over fixed channel;
- Power iteration designed for solving eigen problem.

In addition, we also compare the Algorithm 1 with its counterpart without the symmetrizing step in (12), denoted as Max-SKL and Max-SKL-wos, respectively.

Using the Arimoto-Blahut algorithm, caid for mutual information is given by:

$$P(X) = [0.36, 0.00, 0.00, 0.05, 0.18, 0.05, 0.00, 0.00, 0.36].$$

Using the Max-SKL algorithm, the caid for I_{SKL} is:

$$P(X) = [0.5, 0, 0, 0, 0, 0, 0, 0, 0.5]$$

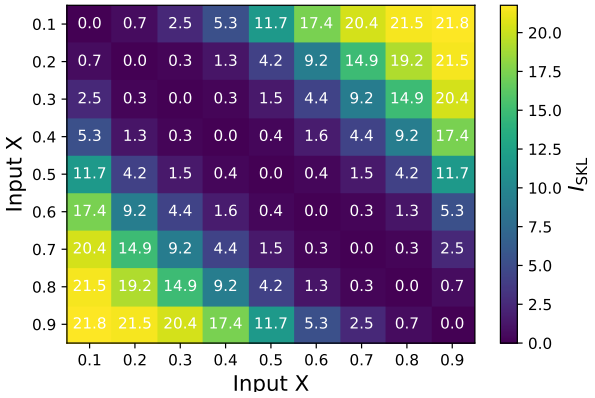

Fig. 3. Matrix of KL Divergence C_{sym} for $n = 10$. The entries at 0.1 and 0.9 are the largest, leading to a concentrated distribution at these points to maximize $X^T CX$.

The convergence of I_{SKL} over iterations are illustrated in Figure 2. Our algorithm successfully converges in the binomial channel, and the Max-SKL demonstrated the best performance. We also apply the Blahut-Arimoto algorithm to maximize the mutual information and use its caid to calculate the I_{SKL} for comparison with our results. We found that the Lautum information term plays a significant role in maximizing I_{SKL}. Additionally, the caids for mutual information and I_{SKL} are quite different. The caid for $I(X;Y)$ is more spread out, whereas for I_{SKL}, the distribution is concentrated at specific points.

To better understand the caid for I_{SKL}, we visualize the KL divergence matrix C_{sym} in Figure 3. When $n = 10$, we observe that the entries at 0.1 and 0.9 in C_{sym} are the largest. Therefore, to maximize $X^T CX$, the distribution should concentrate on these two boundary points. However, when $n = 100$ (see Figure 9 in [30] Appendix D), the caid spreads out more evenly across other points.

VI. Experiments with the Gibbs Channel

In this section, we demonstrate the effectiveness of our algorithm for a specific channel, where I_{SKL} can be interpreted as the generalization error of a supervised learning problem, highlighting the significance of the capacity problem in learning theory. Before delving into the experimental setup, we first provide the context for the Gibbs channel.

A. Dataset and Loss Function

We consider a supervised learning problem with training dataset $S = \{(Z_i)\}_{i=1}^n$, where each data $Z_i = (X_i, Y_i) \in \mathcal{S}$ is i.i.d. generated from the data distribution P_Z. The performance of different machine learning models $W \in \mathcal{W}$ is measured by a loss function $\ell\colon \mathcal{W} \times \mathcal{Z} \to \mathbb{R}^+$. Here, we consider the following Mean Squared Error (MSE) loss function

$$\ell(w, z) = \|y - \hat{y}\|_2^2 = \|y - x^\top w\|_2^2, \tag{21}$$

with $\hat{y} = x^\top w$, i.e., we adopt a linear model. The goal of supervised learning is to find a w that minimizes the following population risk

$$L_P(w, P_Z) = \mathbb{E}_{P_Z}[\|Y - X^\top w\|_2^2]. \tag{22}$$

However, as data distribution P_Z is unknown, we can only minimize the empirical risk, which can be written as:

$$L_E(w, S) = \frac{1}{n}\sum_{i=1}^{n}(Y_i - X_i^\top w)^2 = \frac{1}{n}\|\mathbf{Y} - \mathbf{X}w\|_2^2,$$

where $\mathbf{X}, \mathbf{Y}$ stacks the n i.i.d samples in vector form.

B. Gibbs Channel

Any learning algorithm can be viewed as a channel that randomly maps the training dataset S onto a model W according to the probability transition matrix $P_{W|S}$. Thus, the generalization error quantifying the degree of over-fitting can be written as

$$\mathrm{gen}(P_{W|S}, P_S) \triangleq \mathbb{E}_{P_{W,S}}[L_P(W, P_Z) - L_E(W, S)]. \tag{23}$$

Here, we consider the Gibbs distribution as a channel, which is well-studied in Bayesian learning because it provides a principled way to incorporate prior knowledge and quantify uncertainty [34], [35]. Specifically, the $(\gamma, \pi(w), L_E(w, S))$-Gibbs algorithm is given by

$$P_{W|S}^{\gamma}(w|S) = \frac{\pi(w)e^{-\gamma L_E(w,S)}}{\int_{\mathcal{W}} \pi(w)e^{-\gamma L_E(w,S)}dw}, \tag{24}$$

where $\pi(w)$ is any prior distribution and $\gamma > 0$ denotes inverse temperature.

In addition, as shown in [2], [5], the Gibbs channel has the nice property that its generalization error equals the symmetrized KL information between the input training samples and the learned model

$$\mathrm{gen}(P_{W|S}^{\gamma}, P_S) = \frac{I_{\mathrm{SKL}}(W; S)}{\gamma}. \tag{25}$$

Therefore, for any fixed Gibbs channel $P_{W|S}^{\gamma}$, our Max-SKL algorithm can be used to identify the worst-case data input that maximizes the generalization error.

C. Experimental Setup

In the following, we provide two toy examples to show how our algorithm can be used to find the discrete worst-case data distribution for the Gibbs channel. Using the MSE loss function and Gaussian prior distribution $\pi(w) \sim \mathcal{N}\left(0, \frac{1}{n}\mathbf{I}\right)$, the Gibbs channel can be modeled as a Gaussian posterior distribution. For our setup, we set the parameter γ equal to the number of samples n, ensuring that the posterior appropriately reflects the sample size.

Since our Max-SKL algorithm is designed for discrete input distributions, we make the following simplification:

1) We restrict the feature X to contain only two binary features $X \in \{-1, 1\}^2$ with a binary label $Y \in \{-1, 1\}$;
2) Instead of solving the worst case distribution P_S for n samples directly, where the alphabet size will grow exponentially with n, we focus on the case where the prior distribution $\pi_n(w)$ is pre-trained with n samples generated from P_{S_0}, and our goal is to identify the worst case distribution P_{S_1} for the next $n+1$-th sample.

Thus, in this toy example, we only need to find a distribution with an alphabet size of 8.

In the subsequent sections, we will discuss two specific cases where the pre-trained prior $\pi_n(w)$ differs because it is trained using the samples generated from different P_{S_0}.

D. Case 1: Linearly Separable

1) Motivation and Setup: In this case, the input features X and the target outcomes Y of P_{S_0} are linearly separable, meaning that a single hyperplane can accurately classify the data points into their respective classes. This setup helps us understand the impact of the worst-case data distribution on a model that initially performs well.

We consider the joint distribution P_{S_0} defined uniformly over the following four points,

$$P_{S_0}([1,1],1) = 0.25, \quad P_{S_0}(([1,-1],-1) = 0.25,$$
$$P_{S_0}([-1,1],1) = 0.25, \quad P_{S_0}([-1,-1],-1) = 0.25.$$

2) Pre-training Process: Figure 4 illustrates P_{S_0} of four data points with coordinates $[1,1]$, $[1,-1]$, $[-1,1]$, and $[-1,-1]$:

- Class 1 ($y = 1$): $[1,1]$, $[-1,1]$
- Class -1 ($y = -1$): $[1,-1]$, $[-1,-1]$.

It can be seen that $y = x_2$, i.e., $w = [0,1]^\top$ effectively fits the data points generated from P_{S_0} into their correct classes.

This pre-training uses the $n = 100$ samples generated from P_{S_0}, which gives us the following prior for the $n+1$-th sample

$$\pi_n(w) = \frac{\pi(w)e^{-nL_E(w,s_n)}}{\mathbb{E}_\pi\left[e^{-nL_E(W,s_n)}\right]}. \tag{26}$$

3) Worst-Case Distribution Impact: Using Algorithm 1, we identify the worst-case data distribution for the $(n, \pi_n(w), L_E(w, Z_{n+1}))$-Gibbs algorithm. Figure 4 visualizes the worst-case data distribution P_{S_1}. While it is defined in the same coordinates as P_{S_0}, the target labels have changed, resulting in altered class assignments. In P_{S_1}:

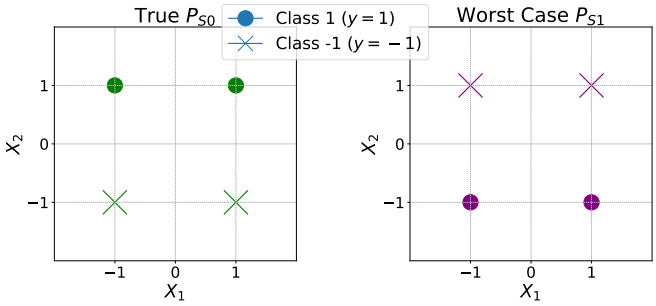

Fig. 4. Linearly Separable Data Points (Case 1). The plots show the data points under two distributions: the initial distribution P_{S_0} and the worst-case distribution P_{S_1}. The initial model correctly fits the data points, but under the worst-case distribution, the fitting plane shifts, leading to misclassification.

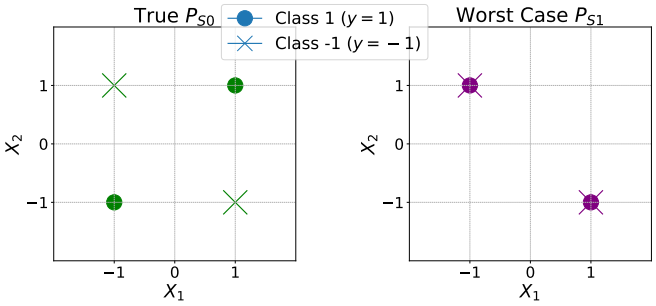

Fig. 5. Linearly Non-separable Data Points (Case 2). The initial data distribution P_{S_0} is represented on the left. Under the worst-case data distribution P_{S_1} on the right, the class labels have shifted.

- Class 1 ($y = 1$): $[1, -1]$, $[-1, 1]$
- Class -1 ($y = -1$): $[1, 1]$, $[-1, -1]$

In Figure 4, circles represent Class 1 ($y = 1$) and crosses represent Class -1 ($y = -1$). The worst-case distribution P_{S_1} alters the fitting plane to $y = -x_2$. This change results in the projection of the input data to the opposite class, thus degrading the model's performance.

E. Case 2: Linearly Non-separable

1) Motivation and Setup: In this case, the input features X and the target outcomes Y are linearly non-separable, meaning no single hyperplane can correctly classify all the data points. This scenario is designed to understand the impact of the worst-case data distribution on a model that already faces challenges due to non-separability.

We consider the joint distribution P_{S_0} defined uniformly over the following four points:

$$P_{S_0}([1,1],1) = 0.25, \quad P_{S_0}([1,-1],-1) = 0.25,$$
$$P_{S_0}([-1,1],-1) = 0.25, \quad P_{S_0}([-1,-1],1) = 0.25.$$

2) Pre-training Process: Figure 5 illustrates the distribution P_{S_0} for the data points with coordinates $[1, 1]$, $[1, -1]$, $[-1, 1]$, and $[-1, -1]$:

- Class 1 ($y = 1$): $[1, 1]$, $[-1, -1]$
- Class -1 ($y = -1$): $[1, -1]$, $[-1, 1]$.

Due to the non-separability, the initial fitting plane cannot perfectly classify the two classes and can only reduce the MSE error value to prevent it from being too large.

This pre-training also uses the $n = 100$ samples generated from P_{S_0}. In this case, the prior has the same form as in equation (26).

3) Worst-Case Distribution Impact: Using Algorithm 1, we identify the worst-case data distribution for the $(n, \pi_n(w), L_E(w, Z_{n+1}))$-Gibbs algorithm. Figure 5 visualizes the worst-case data distribution P_{S_1}. While the coordinates remain the same as in P_{S_0}, the target labels are altered, affecting class assignments. In P_{S_1}:

- Class 1 ($y = 1$): $[1, -1]$, $[-1, 1]$
- Class -1 ($y = -1$): $[1, 1]$, $[-1, -1]$

This shift in the target outcomes effectively changes the fitting plane. The details of the posterior are available in [30] Appendix E. Under the worst-case data distribution, a single data point can have two labels with the same probability, making it difficult to distinguish. This ambiguity causes a significant increase in the model's variance, resulting in a more spread-out posterior distribution and increased uncertainty.

VII. Conclusion and Future Works

This paper presents a novel approach to computing the capacity of discrete channels in terms of symmetrized Kullback-Leibler (KL) information (I_{SKL}). By transforming the problem into a discrete quadratic optimization with simplex constraints, we developed the Max-SKL algorithm. This algorithm symmetrizes the KL divergence matrix and employs iterative updates to ensure non-decreasing objective values while maintaining valid probability distributions. In summary, our work enhances the understanding of channel capacity and the impact of input distributions, providing a valuable tool for both theoretical analysis and practical applications.

Future research could extend this algorithm to more complex channel models and higher-dimensional data distributions, enhancing its applicability and performance. Specifically, our Max-SKL algorithm, currently designed for discrete inputs, is undergoing further development to accommodate continuous inputs using Random Matrix Theory based on mean-field theory.

Appendix

We start with the original proof in [6] showing that Lautum information $L(X;Y)$ is concave with respect to the input distribution P_X.

Original proof: The proposition was based on the following formulation:

$$L(X;Y) = D\left(\int P_{Y|X} dP_X \| P_{Y|X} | P_X\right) \tag{27}$$

To demonstrate concavity, the proof considered a binary random variable U with values:

$$U = \begin{cases} 1, & \text{with probability } \alpha, \\ 0, & \text{with probability } 1-\alpha, \end{cases} \tag{28}$$

where $\alpha \in [0,1]$ represents the probability that $U = 1$.

Let X_0 and X_1 be two independent random variables. The mixed distribution is defined as:

$$P_{X_U} = (1-\alpha)P_{X_0} + \alpha P_{X_1}. \tag{29}$$

Using the Markov chain $(U, X_0, X_1) - X_U - Y$, the data processing inequality was applied to suggest:

$$\begin{aligned} L(X_U;Y) &\geq L(U, X_0, X_1; Y) \\ &= L(X_0, X_1; Y|U) + L(U;Y) \\ &\geq L(X_0, X_1; Y|U) \\ &= (1-\alpha)L(X_0;Y) + \alpha L(X_1;Y), \end{aligned} \tag{30}$$

implying that $L(X;Y)$ is concave in P_X.

Critical Review of the Chain Rule: Upon re-evaluation, we found an issue with the application of the chain rule for mutual information in the context of Lautum information. Specifically, **the chain rule does not hold for Lautum information.** The detailed steps are as follows:

$$L(U, X_0, X_1; Y) = \mathbb{E}_{P_{U,X_0,X_1}P_Y}\left[\log \frac{P_{U,X_0,X_1}P_Y}{P_{U,X_0,X_1,Y}}\right],$$

$$L(U;Y) = \mathbb{E}_{P_U P_Y}\left[\log \frac{P_U P_Y}{P_{U,Y}}\right],$$

$$L(X_0, X_1; Y|U) = \mathbb{E}_{P_{X_0,X_1}}\mathbb{E}_{P_{Y,U}}\left[\log \frac{P_{Y,U}P_{X_0,X_1}}{P_{X_0,X_1,Y,U}}\right],$$

$$\begin{aligned} &L(U, X_0, X_1; Y) - L(U;Y) \\ &= \mathbb{E}_{P_{X_0,X_1}}\mathbb{E}_{P_Y P_U}\left[\log \frac{P_{X_0,X_1}P_{Y,U}}{P_{U,X_0,X_1,Y}}\right] \neq L(X_0, X_1; Y|U). \end{aligned}$$

This discrepancy shows that the concavity argument based on the chain rule is incorrect. Therefore, Lautum information is not concave with respect to the input distribution P_X.

REFERENCES

[1] G. Aminian, H. Arjmandi, A. Gohari, M. Nasiri-Kenari, and U. Mitra, "Capacity of diffusion-based molecular communication networks over lti-poisson channels," *IEEE Transactions on Molecular, Biological and Multi-Scale Communications*, vol. 1, no. 2, pp. 188–201, 2015.

[2] G. Aminian, Y. Bu, L. Toni, M. R. Rodrigues, and G. W. Wornell, "Information-theoretic characterizations of generalization error for the gibbs algorithm," *IEEE Transactions on Information Theory*, 2023.

[3] E. T. Jaynes, "Information theory and statistical mechanics," *Physical review*, vol. 106, no. 4, p. 620, 1957.

[4] R. Pereira, X. Mestre, and D. Gregoratti, "Asymptotics of distances between sample covariance matrices," *IEEE Transactions on Signal Processing*, 2024.

[5] G. Aminian, Y. Bu, L. Toni, M. Rodrigues, and G. Wornell, "An exact characterization of the generalization error for the gibbs algorithm," *Advances in Neural Information Processing Systems*, vol. 34, pp. 8106–8118, 2021.

[6] D. P. Palomar and S. Verdú, "Lautum information," *IEEE transactions on information theory*, vol. 54, no. 3, pp. 964–975, 2008.

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

[8] I. Sason and S. Verdú, "f-divergence inequalities," *IEEE Transactions on Information Theory*, vol. 62, no. 11, pp. 5973–6006, 2016.

[9] Y. Yu, "Squeezing the arimoto–blahut algorithm for faster convergence," *IEEE Transactions on Information Theory*, vol. 56, p. 3149–3157, July 2010.

[10] M. Frank and P. Wolfe, "An algorithm for quadratic programming," *Naval Research Logistics Quarterly*, vol. 3, no. 1-2, pp. 95–110, 1956.

[11] Z. Li, R. She, P. Fan, C. Peng, and K. B. Letaief, "Learning channel capacity with neural mutual information estimator based on message importance measure," *IEEE Transactions on Communications*, vol. 72, pp. 1370–1384, 2023.

[12] D. Arnold, H.-A. Loeliger, P. Vontobel, A. Kavcic, and W. Zeng, "Simulation-based computation of information rates for channels with memory," *IEEE Transactions on Information Theory*, vol. 52, no. 8, pp. 3498–3508, 2006.

[13] H. Pfister, J. Soriaga, and P. Siegel, "On the achievable information rates of finite state isi channels," in *GLOBECOM'01. IEEE Global Telecommunications Conference (Cat. No.01CH37270)*, vol. 5, pp. 2992–2996 vol.5, 2001.

[14] A. Hacine-Gharbi and P. Ravier, "A binning formula of bi-histogram for joint entropy estimation using mean square error minimization," *Pattern Recognition Letters*, vol. 101, pp. 21–28, 2018.

[15] N. Bi, J. Tan, J.-H. Lai, and C. Y. Suen, "High-dimensional supervised feature selection via optimized kernel mutual information," *Expert Systems with Applications*, vol. 108, pp. 81–95, 2018.

[16] A. Gretton, R. Herbrich, and A. J. Smola, "The kernel mutual information," in *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03).*, vol. 4, pp. IV–880, IEEE, 2003.

[17] M. M. V. Hulle, "Edgeworth approximation of multivariate differential entropy," *Neural computation*, vol. 17, no. 9, pp. 1903–1910, 2005.

[18] M. I. Belghazi, A. Baratin, S. Rajeswar, S. Ozair, Y. Bengio, A. Courville, and R. D. Hjelm, "Mine: Mutual information neural estimation," 2021.

[19] R. Fritschek, R. F. Schaefer, and G. Wunder, "Deep learning for channel coding via neural mutual information estimation," 2019.

[20] D. Tsur, Z. Aharoni, Z. Goldfeld, and H. Permuter, "Neural estimation and optimization of directed information over continuous spaces," 2022.

[21] F. Mirkarimi, S. Rini, and N. Farsad, "Benchmarking neural capacity estimation: Viability and reliability," *IEEE Transactions on Communications*, vol. 71, no. 5, pp. 2654–2669, 2023.

[22] Z. Aharoni, D. Tsur, Z. Goldfeld, and H. H. Permuter, "Capacity of continuous channels with memory via directed information neural estimator," 2020.

[23] F. Mirkarimi and N. Farsad, "Neural computation of capacity region of memoryless multiple access channels," in *2021 IEEE International Symposium on Information Theory (ISIT)*, IEEE, July 2021.

[24] S. Sreekumar and Z. Goldfeld, "Neural estimation of statistical divergences," 2022.

[25] N. A. Letizia and A. M. Tonello, "Capacity-driven autoencoders for communications," *IEEE Open Journal of the Communications Society*, vol. 2, pp. 1366–1378, 2021.

[26] N. A. Letizia and A. M. Tonello, "Discriminative mutual information estimators for channel capacity learning," *arXiv preprint arXiv:2107.03084*, 2021.

[27] M. I. Belghazi, A. Baratin, S. Rajeswar, S. Ozair, Y. Bengio, A. Courville, and R. D. Hjelm, "Mine: mutual information neural estimation," *arXiv preprint arXiv:1801.04062*, 2018.

[28] J. Song and S. Ermon, "Understanding the limitations of variational mutual information estimators," *arXiv preprint arXiv:1910.06222*, 2019.

[29] R. M. Corless, G. H. Gonnet, D. E. G. Hare, D. J. Jeffrey, and D. E. Knuth, "On the lambert w function," *Advances in Computational Mathematics*, vol. 5, no. 1, pp. 329–359, 1996.

[30] H. Chen, G. Aminian, and Y. Bu, "An algorithm for computing the capacity of symmetrized KL information for discrete channels," *arXiv preprint arXiv:2407.13436*, 2024.

[31] R. A. Horn and C. R. Johnson, *Matrix analysis*. Cambridge university press, 2012.

[32] G. H. Golub and C. F. Van Loan, *Matrix Computations*. The Johns Hopkins University Press, third ed., 1996.

[33] P. A. G. Scheuer and S. P. H. Mandel, "An inequality in population genetics," *Heredity*, vol. 13, pp. 519–524, Nov 1959.

[34] F. Wenzel, K. Roth, B. S. Veeling, J. Świątkowski, L. Tran, S. Mandt, J. Snoek, T. Salimans, R. Jenatton, and S. Nowozin, "How good is the bayes posterior in deep neural networks really?," *arXiv preprint arXiv:2002.02405*, 2020.

[35] B. Adlam, J. Snoek, and S. L. Smith, "Cold posteriors and aleatoric uncertainty," *arXiv preprint arXiv:2008.00029*, 2020.