ON THE INFORMATION THEORY OF CLUSTERING, REGISTRATION, AND BLOCKCHAINS

BY

RAVI KIRAN RAMAN

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Electrical and Computer Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2019

Urbana, Illinois

Doctoral Committee:

Assistant Professor Lav R. Varshney, Chair
Professor Pierre Moulin
Professor Venugopal V. Veeravalli
Assistant Professor Andrew Miller

ABSTRACT

Progress in data science depends on the collection and storage of large volumes of reliable data, efficient and consistent inference based on this data, and trusting such computations made by untrusted peers. Information theory provides the means to analyze statistical inference algorithms, inspires the design of statistically consistent learning algorithms, and informs the design of large-scale systems for information storage and sharing. In this thesis, we focus on the problems of reliability, universality, integrity, trust, and provenance in data storage, distributed computing, and information processing algorithms and develop technical solutions and mathematical insights using information-theoretic tools.

In unsupervised information processing we consider the problems of data clustering and image registration. In particular, we evaluate the performance of the max mutual information method for image registration by studying its error exponent and prove its universal asymptotic optimality. We further extend this to design the max multiinformation method for universal multi-image registration and prove its universal asymptotic optimality. We then evaluate the non-asymptotic performance of image registration to understand the effects of the properties of the image transformations and the channel noise on the algorithms.

In data clustering we study the problem of independence clustering of sources using multivariate information functionals. In particular, we define consistent image clustering algorithms using the cluster information, and define a new multivariate information functional called illum information that inspires other independence clustering methods. We also consider the problem of clustering objects based on labels provided by temporary and long-term workers in a crowdsourcing platform. Here we define budget-optimal universal clustering algorithms using distributional identicality and temporal dependence in the responses of workers.

For the problem of reliable data storage, we consider the use of blockchain systems, and design secure distributed storage codes to reduce the cost of cold storage of blockchain ledgers. Additionally, we use dynamic zone allocation strategies to enhance the integrity and confidentiality of these systems, and frame optimization problems for designing codes applicable for cloud storage and data insurance.

Finally, for the problem of establishing trust in computations over untrusting peer-to-peer networks, we develop a large-scale blockchain system by defining the validation protocols and compression scheme to facilitate an efficient audit of computations that can be shared in a trusted manner across peers over the immutable blockchain ledger. We evaluate the system over some simple synthetic computational experiments and highlights its capacity in identifying anomalous computations and enhancing computational integrity.

To appa, amma, and Suraj for their love and support.
In memory of my late grandfather, Dr. K.S. Narayanan.

ACKNOWLEDGMENTS

As I look back at the last 5 years in hopes of thanking everyone who has been an integral part of this journey, I realize I'm more confident of doing justice to a few more theorems than I am in sufficiently acknowledging all the support I have received. I suppose that shows how lucky I have been through this arduous journey.

It has been my pleasure to have worked with Prof. Lav Varshney through these five years. I will always be thankful for the welcoming, friendly, and supportive environment that Lav created for me at UIUC. Not only did Lav introduce me to a world of wonderful problems that I enjoyed working on, but also encouraged me to build other dimensions as a researcher, be it through organizing the Coordinated Science Lab Student Conference (CSLSC), or through mentoring undergrads. Lav has been an undying source of encouragement, support, and wisdom, even when I made blunders in research. I am immensely grateful for the freedom that Lav gave me and for his support in working on a plethora of topics. Through his support and guidance, Lav made sure I had nothing to worry about other than the research and I could not have asked for a better advisor or collaborator. I truly hope to continue our discussions and research well into the future.

My thesis has been improved a great deal through the suggestions and recommendations of my committee members. I would like to express my gratitude to Prof. Pierre Moulin for his guidance and suggestions regarding Chapters 2 and 3 and to Prof. Venugopal Veeravalli for his instruction and directions in building on my work in Chapter 4. I would also like to thank Prof. Andrew Miller for giving me better insight into blockchains and cryptography that guided me through the work in Chapters 6 and 7.

The work in Chapter 7 was done as part of the Social Good Fellowship at IBM Research and I'm grateful for the collaboration with Kush Varshney, Roman Vaculin, Michael Hind, Sekou Remy, Nelson Bore, and Eleftheria Pissdaki. I would also like to thank my collaborators Julius Kusuma, Andriy Gelman, and Arnaud Jarrot at Schlumberger Doll Research who gave me the chance to create practical applications based on my work in Chapter 2. I am also grateful for the collaboration and discussions with Haizi Yu and Rebecca Chen on parts of Chapters 4 and 2. I also greatly appreciate the opportunity to interact at depth in research

and to work on social learning with Srilakshmi Pattabiraman, Daewon Seo, and Wenxian Zhang, and on connectomics with Malhar Jere and Suraj Kiran Raman.

Coordinated Science Lab (CSL) was a wonderful place to spend 5 years mainly due to my friends and faculty. I will fondly remember room 312 that was always a lively place, especially because of Linjia Chang, Jonathan Ligo, Avhishek Chatterjee, Yuheng Bu, Meghana Bande, and Aditya Deshmukh, in addition to the friends already mentioned. Some of my fondest memories at CSL have been participating in the organizing committees of CSLSC, and I would like to thank all three committees I was part of, especially my co-chair Philip Pare and Prof. Klara Nahrstedt.

My PhD journey is as much about the work at CSL as it is about everything I did outside research. Urbana-Champaign became home away from home because of my friends and I am especially grateful for the several lunches, dinners, and conversations with Sivasakthya Mohan, the late night discussions at home with Sujan Gonugondla, all the games of cricket with Vegnesh Jayaraman, and all the trips around the country with Sivaramakrishnan and Navneeth Nair.

Finally, and most importantly, earning a PhD would have stayed a distant dream if it weren't the never ending support and constant encouragement of my parents and my brothers, and all the assistance of my uncle, K.N. Mohan, who made the transition to the United States smooth. I would like to dedicate this thesis to my grandfather, the late Dr. K.S. Narayanan, who instilled the interest in mathematics by showing its wonders to an earnest kid, and set me on this path.

- Financial support was provided by the Joan and Lalit Bahl Fellowship, IBM Science for Social Good Fellowship, NSF Grants IIS-1550145 and CCF-1623821, and Air Force STTR Contract FA8650-16-M-1819.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

We live in an age of explosive data acquisition rates, in a variety of forms such as pictures, videos, text, and voice. By 2023, it is projected that the per capita amount of data stored in the world will exceed the entire Library of Congress (10^{14} bits) based on when Shannon first estimated it in 1949 [1, 2]. Availability and access to so much information has also paved way for the creation of an environment of large-scale information processing. We constantly use such information processing systems on a daily basis, such as mobile assistants and recommendation systems. This big data era has also introduced us to a new set of problems on reliability, universality, integrity, trust, and provenance in data and computations.

To be precise, the growth in machine learning has resulted in a plethora of increasingly efficient information processing algorithms to better process complex, increasing volumes of data better. Information processing algorithms are in fact increasingly used in fields such as medicine [3], law [4], scientific discovery [5], automation [6], agriculture [7], and a variety of day-to-day applications. However, AI often lacks concrete performance guarantees for the task that they are applied for. With applications such as medicine and law that have far-reaching consequences, it is critical that we have a strong characterization of the reliability of the methods employed. In particular, it is important to develop clear theoretical guarantees on the performance of machine learning algorithms in tasks such as classification, recommendation, and prediction. Such guarantees go a long way in not only establishing confidence in their adoption into practice, but also fuel the development of new and novel information processing algorithms.

Data collected is also drawn from an increasingly diverse set of sources, varying both in the type of data such as text, speech, images, and videos, but also in the statistical properties governing the data [2]. This large diversity poses a problem in that it is increasingly difficult to develop information processing algorithms designed to address specific data types and inference tasks. That is, it is important to be able to develop universal information processing algorithms that can work on large classes of data sources. Further, most of the learning carried out by people and animals is unsupervised—we largely learn how to think by observing the unlabeled world. However machines have been far from universally capable

of reliable unsupervised learning. A central problem of interest in data science has been the design of efficient and reliable unsupervised learning algorithms.

Additionally, given the importance of data in current society, the veracity and provenance of data, though often ignored, are crucial to reliable data analysis. Data collection and sharing forms an important first step to learning, and the secure storage of these large datasets, in a manner that the empirical samples are not corrupted is important. Further, considering the interconnected and interdependent nature of data collection, management, and processing, it is important to design efficient data storage and sharing mechanisms that guarantee the integrity of the data. Not only do we require the data to be shared among several entities, we also need to ensure that the cost of storage is minimized, and the system protects against loss of data from adversarial and technical corruptions.

Another problem area of growing significance involves the issues of fairness, accountability, and transparency in information processing systems [8, 9]. Individual agents do not process the information on their own. Data is often communicated to cloud systems that perform the inference task, rather than relying on on-device computations. Thus information processing is very much a distributed task, in that several agents are involved in performing the computation with data. In an adversarial environment, such distributed computing ends up being untrusted. Guaranteeing validity of the computations performed by untrusting agents can go a long way in the adoption of new and novel information processing systems. This is particularly observed in scientific discovery and policy design where, owing to the participation of a large number of untrusted agents, one is unclear of what computational results are valid and can be trusted. Establishing an environment of distributed computational trust can improve scientific discovery and policy design.

The accountability requirement for computations also extends to the data. It is indeed important that we have reliable knowledge of the data sources and can hold them accountable for any possible misinformation or ethical violations in data shared [10]. We live in an age of automated information generation systems that are capable of creating and sharing misinformation [11], and such misinformation is having significant consequences on everyday decision making. It is thus imperative that data storage and sharing pipelines establish data provenance and a mechanism to trace data back to their source reliably, so as to hold them accountable.

Thus, we are not only in need of efficient solutions for information storage, sharing, and processing, but also require a clearer understanding and theoretical analysis of information processing algorithms. This thesis is composed to two components that respectively focus on addressing some of these issues in unsupervised learning, and through the use of blockchain systems.

1.1 Thesis Composition

The thesis is composed of two main parts. First, we explore the problems of reliability and universality in information processing through information-theoretic analysis and design of novel algorithms for clustering and registration. Next, we consider the problems of integrity and reliability in data storage, sharing, and distributed computing through the use of scalable blockchain systems. A brief outline of the thesis is given here.

1.1.1 Unsupervised, Reliable Information Processing Using Universal Information Theory

Unsupervised information processing algorithms are critical for efficient data management and statistical inference. Whether grouping behavioral traces into categories to understand their neural correlates [12], or aligning medical images [13], or extracting independent features from sensor measurements [14], the importance of unsupervised learning to data science cannot be overstated. Making sense of such large volumes of data, e.g. for statistical inference, requires extensive, efficient preprocessing to transform the data into meaningful, readable forms.

Much research effort has been expended in finding natural mathematical notions that are useful in developing/evaluating unsupervised learning algorithms in these terms, such as clustering [15]. Recently the "strengths" of supervised learning have been preferred over "subjectivity" of unsupervised learning [16, p. 159]. Deep learning has largely driven this growth in interest in supervised learning. However, results for unsupervised learning pale in comparison [17]. Is it possible to make sense of data without people to provide labels? This question forms the backbone of artificial general intelligence [18]. To move beyond data-intensive supervised learning methods, there is growing interest in unsupervised learning problems such as density/support estimation, clustering, and independent component analysis. A careful choice and application of unsupervised methods often reveal structure in the data that is otherwise not evident.

In this thesis, we draw on insights from universal information theory, to design and analyze unsupervised learning algorithms from the perspective of image registration and data clustering problems. By universality, we mean that the system does not have prior access to the statistical properties of the data to be clustered, nor does it have a strong sense of the appropriate notion of similarity to measure which objects are close to one another. A wide variety of universal encoders and decoders have been designed for the problems of compression, both lossless [19–22] and lossy [23–25]. Likewise, several studies in information theory have also

considered the problem of universal communication for a variety of channels [26–29].

In his Shannon lecture, Jorma Rissanen discussed the philosophical undertones of similarity between data compression and estimation theory with a disclaimer [30]:

> in data compression the shortest code length cannot be achieved without taking advantage of the regular features in data, while in estimation it is these regular features, the underlying mechanism, that we want to learn . . . like a jigsaw puzzle where the pieces almost fit but not quite, and, moreover, vital pieces were missing.

This observation is significant not only in reaffirming the connection between the two fields, but also in highlighting the fact that the unification/translation of ideas often requires closer inspection and some additional effort. Drawing inspiration from this connection, we build on results in universal information theory to design and analyze unsupervised information processing algorithms.

First, in **Chapter 2**, we consider the task of image registration (aligning copies of an image). We consider the max mutual information (MMI) method for image registration, and use the method of types to prove asymptotic optimality of the method. We then extend the method to design the max multiinformation method for multi-image alignment. We also consider large-scale registration with limited image resolution and identify sample complexity of consistent registration of images.

Then, in **Chapter 3** we study the fundamental limits of channel-aware two-image registration in the finite-sample context. In particular, we use the Berry-Esseen central limit theorem and strong large deviations analysis respectively to obtain achievable arguments on the tradeoff between the sample size and the information content in the channel. We also study the universal delay estimation problem, which is a simplified version of the image registration problem, and use a tighter analysis of the method of types to understand the performance of the MMI method with respect to the properties of the transformations of the sequences.

Next, in **Chapter 4**, we explore unsupervised clustering of random variables. In particular, we study joint registration and clustering of images using multivariate information functionals and study their consistency under various clustering criteria. We highlight the role of multivariate information functionals in independence clustering. We also define and introduce new multivariate information functionals, illum and sum information, and study their functional and operational properties that inspire new information-based algorithms for independence clustering.

We continue our explorations of the clustering problem in **Chapter 5** where we study the task of clustering objects based on crowdsourced responses. In particular we study responses

generated by temporary and long-term workers, and design budget-optimal universal algorithms that are unaware of the reliabilities of the workers. In particular, we highlight the cost of clustering under the Hamming loss as a function of the number of objects to be clustered and the difficulty of the clustering problem for both classes of responses that are independent and dependent across tasks using distance and information functionals respectively.

1.1.2 Integrity and Provenance in Data and Computation Using Blockchain Systems

As highlighted earlier, the veracity and provenance of data, though often ignored, are crucial to reliable data analysis. Data collection forms an important first step to learning, and the secure storage and sharing of these large datasets, in a manner that the samples are not corrupted is important. Additionally we are also keen on establishing provenance of data to have a reliable and trusted record of the data source, so as to hold agents accountable for their data.

Blockchain systems have emerged as viable candidates for secure storage of information, owing to the immutability property of the hash chains. Separate from their role in the creation of cryptocurrencies and a decentralized economy, blockchains hold significant potential for establishing trust in information-sharing systems. This is precisely quoted in [31]:

> blockchain ... is a new organizing paradigm for the discovery, valuation, and transfer of all quanta (discrete units) of anything for the coordination of all human activity at a much larger scale than has been possible before.

Beside offering a secure storage mechanism, they also provide an avenue for data sharing and thus an avenue for open data science and reproducibility. New mechanisms have been developed for cloud storage [32], electronic heath records [33, 34], personal data [35], and financial accounting [36].

However, blockchain systems also result in significant storage costs owing to the replicated storage of all data at all peers in the network. This is particularly tedious for systems that collect massive volumes of data [37–39]. Thus, any system developed for storing and sharing large datasets also requires solutions to address the scaling problem such that the peers participating in the network share the information securely.

Analogously, the digital trust that is created by blockchains in a distributed network of untrusting peers can be leveraged in creating a platform for trusted collaboration and computations. In particular, blockchain systems allow for provenance, accountability, and trust [40, 41]. This is established by the consensus on the transactions, the availability of

local copies of the set of all transactions, and cryptographic security enabled by the hash chains. Creating scalable blockchain-based platforms and interaction protocols that limit the cost of communication between the peers, and the storage on the blockchain ledger.

In **Chapter 6** we address this issue using distributed storage codes in conjunction with blockchain systems to reduce this storage cost. We use a novel combination of secret sharing, private key encryption, and distributed storage codes to provide a secure storage scheme, wherein each peer stores a fraction of the data. In addition, we study the tradeoffs between integrity, confidentiality, recovery, repair associated with the choice of the parameters of the coding scheme. We also highlight some practical applications of the codes in the form of cloud storage and data insurance.

Finally, we also aim to establish transparency, accountability, and trust in distributed computational systems. This problem is addressed in **Chapter 7** where we study a blockchain system that records frequent audits of the computation, after performing recomputation-based validation using endorsers. This sequence of audits provides a secure, and simple verification mechanism for the computational pipeline. In particular, we design a novel compression schema and validation protocol to establish trust in iterative and enumerative computational experiments over a large peer-to-peer network.

1.2 Bibliographical Note

We now provide a chapter-wise list of the publications that include the work presented in this thesis for the ease of the readers.

Parts of Chapter 2 appear in the following papers:

- R. K. Raman and L. R. Varshney, "Universal joint image clustering and registration using partition information," in *Proc. 2017 IEEE Int. Symp. Inf. Theory*, pp. 2168–2172, June 2017.

- R. K. Raman and L. R. Varshney, "Universal joint image clustering and registration using multivariate information measures," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 5, pp. 928–943, Oct. 2018.

Parts of Chapter 3 are to appear in the manuscript:

- R. K. Raman, L. R. Varshney, "Finite-Sample Analysis of Image Registration", under preparation.

Parts of Chapter 4 appear in the following papers:

- R. K. Raman and L. R. Varshney, "Universal joint image clustering and registration using partition information," in *Proc. 2017 IEEE Int. Symp. Inf. Theory*, pp. 2168–2172, June 2017.

- R. K. Raman and L. R. Varshney, "Universal joint image clustering and registration using multivariate information measures," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 5, pp. 928–943, Oct. 2018.

- R. K. Raman, H. Yu, and L. R. Varshney, "Illum information," in *Proc. 2017 Inf. Theory Appl. Workshop*, Feb. 2017.

Parts of Chapter 5 appear in the paper:

- R. K. Raman and L. R. Varshney, "Budget-optimal clustering via crowdsourcing," in *Proc. 2017 IEEE Int. Symp. Inf. Theory*, pp. 2163–2167, June 2017.

Parts of Chapter 6 appear in the following papers:

- R. K. Raman and L. R. Varshney, "Distributed storage meets secret sharing on the blockchain," in *Proc. 2018 Inf. Theory Appl. Workshop*, Feb. 2018.

- R. K. Raman and L. R. Varshney, "Dynamic distributed storage for blockchains," in *Proc. 2018 IEEE Int. Symp. Inf. Theory*, July 2018.

- Y. Kim, R. K. Raman, Y. Kim, L. R. Varshney, and N. R. Shanbhag, "Efficient local secret sharing for distributed blockchain systems," *IEEE Commun. Lett.*, vol. 23, no. 2, pp. 282–285, Feb. 2019.

Parts of Chapter 7 appear in the following papers:

- R. K. Raman, R. Vaculin, M. Hind, S. L. Remy, E. K. Pissadaki, N. K. Bore, R. Daneshvar, B. Srivastava, and K. R. Varshney, "Trusted multi-party computation and verifiable simulations: A scalable blockchain approach," to appear in *Proc. 2019 IEEE Int. Conf. Blockchains Cryptocurrency*, May 2019.

- N. K. Bore, R. K. Raman, I. M. Markus, S. L. Remy, O. Bent, M. Hind, E. K. Pissadaki, B. Srivastava, R. Vaculin, K. R. Varshney, and K. Weldemariam, "Promoting distributed trust in machine learning and computational simulation via a blockchain network," to appear in *Proc. 2019 IEEE Int. Conf. Blockchains Cryptocurrency*, May 2019.

- R. K. Raman, K. R. Varshney, R. Vaculin, N. K. Bore, S. L. Remy, E. K. Pissadaki, M. Hind, "Constructing and Compressing Frames in Blockchain-based Verifiable Multi-party Computation," to appear in *Proc. 2019 IEEE Int. Conf. Acoust., Speech, Signal Process.*, May 2019.

CHAPTER 2

UNIVERSAL IMAGE REGISTRATION: ASYMPTOTIC OPTIMALITY

Suppose you have an unlabeled repository of MRI, CT, and PET scans of a brain region from different stages of the diagnosis. You wish to align these images to the right orientation. In this section, we address this problem using novel multivariate information functionals.

Image registration is the task of geometrically aligning two or more images of the same scene taken at different points in time, from different viewpoints, or by different imaging devices. It is a crucial step in tasks such as medical diagnosis [42], target detection [43], cryo-electron microscopy [44], image fusion, remote sensing [45] and multimodal image restoration.

Different digital images of the same scene can differ significantly from each other, e.g., images that are negatives of each other. Such factors make image registration harder. Further, such meta-data about the digital images is often not available *a priori*. This emphasizes the need for *universality*, the design of reliable registration algorithms that work without specific knowledge of priors or channel models that govern image generation.

Supervised learning for computer vision has gained prominence through deep convolutional neural networks and other machine learning algorithms. However, these methods require vast amounts of costly labeled training data. Thus, unsupervised methods are always of interest.

Multi-image registration has been studied extensively [46]. Prominent region-based registration methods include maximum likelihood (ML) [47], minimum KL divergence [48], correlation detection [49], and maximum mutual information (MMI) [13,50]. Several feature-based techniques have also been considered [51, 52].

Lower bounds on mean squared error for image registration in the presence of additive noise using Ziv-Zakai and Cramer-Rao bounds have been explored recently [53], [54]. The MMI decoder was originally developed in universal communication [26]. Correctness of the method in image registration using both deterministic reasons [55] and information-theoretic arguments [56] have been identified.

A problem closely related to image registration is multireference alignment. There the aim is to denoise a signal from noisy, circularly translated versions of itself, usually under Gaussian or binary noise models [57–59]. Unlike denoising, we consider only registration, but for a wider class of noise models in a universal setting.

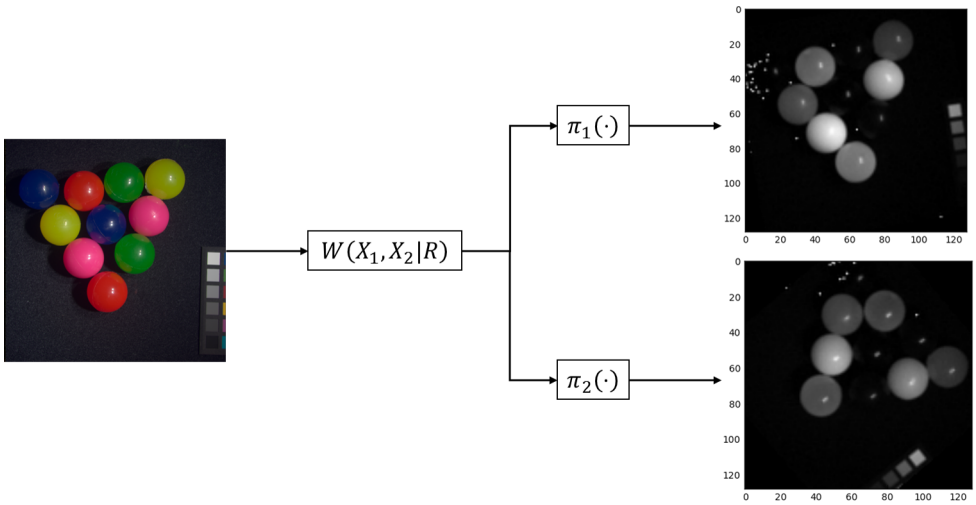


Figure 2.1: Model of the image registration problem: Pixels of the underlying scene are jointly corrupted by DMC W and images are transformed by rigid body transformations π_1, π_2.

Although MMI performs well in numerous empirical studies, concrete theoretical guarantees are still lacking. In this work, we extend the framework of universal delay estimation [60] to derive universal asymptotic optimality guarantees for MMI in registering two images under the Hamming loss, under mild assumptions on the image models [61, 62]. Further, we show that MMI is strictly suboptimal for universal multi-image registration. We define the max multiinformation (MM) algorithm instead for multi-image registration, and prove that it is universally asymptotically optimal. Finally, we show the results of some experiments of image registration using the algorithms on the CAVE multispectral image database [63].

2.1 System Model

We consider a simple image model, wherein each image is a noisy version of a collection of n pixels drawn independently and identically from an unknown prior defined on a finite set of pixel values $[r] = \{1, \ldots, r\}$, as depicted in Fig. 2.1.

Let the scene captured by an image be an n-dimensional random vector, $\mathbf{R} \sim P_R^{\otimes n}$. Consider a collection of m images, each of which is a noisy depiction (channel output) of the scene (source), i.e., outputs of a discrete memoryless channel (DMC) whose input is the scene:

$$\mathbb{P}\left[\tilde{\mathbf{X}}^{[m]} \middle| \mathbf{R}\right] = \prod_{i=1}^{n} W\left(\tilde{X}_i^{[m]} \middle| R_i\right), \tag{2.1}$$

where for any set S, $X^S = \{X_i : i \in S\}$, and $[m] = \{1, \ldots, m\}$. That is, images are jointly corrupted by a DMC, and the pixels of the images are independent of each other. Without loss of generality, we assume $\tilde{\mathbf{X}} \in [r]^n$.

Remark 1. *The model of the image considered here is simpler than real images as it excludes the inter-pixel correlations. Correctness of the forthcoming registration and clustering schemes holds for ergodic sources, but the error exponents are much harder to analyze.*

In this work we consider registration done at the pixel level for ease of description. However, the same methods have been adapted to work with image patches by using independence component analysis (ICA) [52]. That work considers registration of images at the level of small patches, each of which are represented in terms of their ICA basis. For the DMC model, we can define a common ICA basis for the image set. Thus, each image patch is represented by the coefficients of the linear combination. In that model, the i.i.d. assumption translates to one on these coefficients. Thus, even if pixels are not i.i.d., the assumption remains reasonable at the feature level, and the results of the analysis are broadly retained.

Since images are modeled with i.i.d. pixels, distinguishing between alignments is to perform independence testing. The Type-1 and Type-2 error exponents of independence testing are given by the mutual information and the lautum information [64] of the channel. Thus, we exclude the trivial case of channels with unbounded lautum information.

Corrupted images are also subject to independent rigid-body transformations of rotation and translation on the discrete $\mathbb{Z}^2$-lattice. Conventional methods consider transforming the pixels by a rotation of angle $\theta \in [0, 2\pi)$ followed by a translation of $[t_x, t_y]' \in \mathbb{R}^2$. Then, the discrete equivalent of these transformations for a pixel at location $[x, y]'$ is obtained as

$$\pi([x, y]') = \mathcal{D}\left(\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix} \right),$$

where $\mathcal{D} : \mathbb{R}^2 \to \mathbb{Z}^2$ is the rounding function mapping the real space to the discrete pixel lattice, $\mathbb{Z}^2$. In general such mappings are not bijective, owing to the rounding operation. In practice however, they are considered to be invertible through the use of a standard backwards mapping given by $\pi^{-1} = \mathcal{D} \circ \pi^{-1}$ [65].

However, we work with the class of discrete rigid body transformations, Π, for ease of mathematical exposition. This set of transformations has been well studied, and the number of such transformations of an image with n pixels on the $\mathbb{Z}^2$-lattice is polynomial in n, i.e., $|\Pi| = O(n^\alpha)$ for some $\alpha \leq 5$ [65]. We assume Π is known.

Remark 2. *In this work we do not consider a continuous transformation and resampling model for the image registration task. The permutation model of transformations however*

not only generalizes the set of all rigid-body transformations, but is also a better model for the algorithms that work with the pixels of the images. This model gives strong universal, theoretical insights into the problem, but the modeling assumptions can certainly be generalized in future studies.

Since images are vectors of length n, we represent the transformations by permutations of $[n]$. Let $\pi_j \sim \text{Unif}(\Pi)$ be the transformation of image j. Then, the final image is $\mathbf{X}_i^{(j)} = \tilde{\mathbf{X}}_{\pi_j(i)}^{(j)}$, for all $i \in [n]$. Image $\mathbf{X}$ transformed by π is depicted interchangeably as $\pi(\mathbf{X}) = \mathbf{X}_\pi$.

We assume Π forms a *commutative algebra* over the composition operator $\circ$. More specifically,

- for $\pi_1, \pi_2 \in \Pi$, $\pi_1 \circ \pi_2 = \pi_2 \circ \pi_1 \in \Pi$;
- there exists unique $\pi_0 \in \Pi$ s.t. $\pi_0(i) = i$, for all $i \in [n]$;
- for any $\pi \in \Pi$, there exists a unique inverse $\pi^{-1} \in \Pi$, s.t. $\pi^{-1} \circ \pi = \pi \circ \pi^{-1} = \pi_0$.

Even though this is not exactly true of rigid-body transformations, this assumption again helps simplify the mathematical results. Note that the assumption does not remove anything from the problem as conventionally it is presumed that the images undergo a rotation phase followed by a translation phase.

Definition 1. *The* correct registration *of an image* $\mathbf{X}$ *transformed by* $\pi \in \Pi$ *is* $\hat{\pi} = \pi^{-1}$.

Definition 2. *We now define a set of terms related to the set of permutations.*

- *A* permutation cycle *of* $\pi \in \Pi$ *is a subset* $\{i_1, \ldots, i_k\}$ *of* $[n]$*, such that* $\pi(i_j) = i_{j+1}$*, for all* $j < k$ *and* $\pi(i_k) = i_1$*. Let the number of permutation cycles of* π *be* κ_π*.*
- Identity block *of* $\pi \in \Pi$ *is the inclusion-wise maximal subset* $\mathcal{I}_\pi$ *of* $[n]$ *such that* $\pi(i) = i$*, for all* $i \in \mathcal{I}_\pi$*.*
- *A permutation* π *is* simple *if* $\kappa_\pi = 1$*,* $\mathcal{I}_\pi = \emptyset$*.*
- *Any two permutations* $\pi, \pi' \in \Pi$ *are said to be* non-overlapping *if* $\pi(i) \neq \pi'(i)$ *for all* $i \in [n]$*.*

By the pigeonhole principle, $\kappa_\pi \geq 1$ for any $\pi \neq \pi_0$.

Lemma 1. *Let* π *be chosen uniformly at random from the set of all permutations of* $[n]$*. Then for any constants* $c \in (0, 1], C > 0$*,*

$$\mathbb{P}\left[|\mathcal{I}_\pi| > cn\right] \lesssim \exp\left(-cn\right), \quad \mathbb{P}\left[\kappa_\pi > C\tfrac{n}{\log n}\right] = o(1). \tag{2.2}$$

Proof. First, we observe that the number of permutations that have an identity block of size at least cn is given by

$$\nu_c \leq \binom{n}{cn}((1-c)n)! = \frac{n!}{(cn)!}.$$

Thus, from Stirling's approximation,

$$\mathbb{P}\left[|\mathcal{I}_\pi| \geq cn\right] \leq \tfrac{1}{\sqrt{2\pi}} \exp\left(-(cn+\tfrac{1}{2})\log(cn) + cn\right).$$

Lengths and number of cycles in a random permutation may be analyzed as detailed in [66]. In particular, we note that for a random permutation π, $\mathbb{E}\left[\kappa_\pi\right] = \log n + O(1)$. Using Markov's inequality, the result follows. □

Following Lem. 1, we assume that for any $\pi \in \Pi$, $\kappa_\pi = o\left(n/\log(n)\right)$, i.e., the number of permutation cycles does not grow very fast. Further, let $|\mathcal{I}_\pi| = o(n)$ for any $\pi \in \Pi$.

We now introduce formal metrics to quantify algorithm performance.

Definition 3. *A* universal image registration algorithm *is a sequence of functions,* $\Phi^{(n)} : \left\{\mathbf{X}^{(1)}, \ldots, \mathbf{X}^{(m)}\right\} \to \Pi^m$*, designed in the absence of knowledge of* W *and* P_R*. Here,* n *corresponds to the number of pixels in each image.*

We focus on the 0-1 loss function to quantify performance.

Definition 4. *The* error probability *of an algorithm* $\Phi^{(n)}$ *that outputs* $(\hat{\pi}_1, \ldots, \hat{\pi}_m) \in \Pi^m$ *is*

$$P_e(\Phi^{(n)}) = \mathbb{P}\left[\cup_{i\in[m]}\{\hat{\pi}_i \neq \pi_i^{-1}\}\right]. \tag{2.3}$$

Definition 5. *Alg.* $\Phi^{(n)}$ *is* asymptotically consistent *if* $\lim_{n\to\infty} P_e(\Phi^{(n)}) = 0$*, and is* exponentially consistent *if* $\lim_{n\to\infty} -\log P_e(\Phi^{(n)}) > 0$.

Definition 6. *The* error exponent *of an algorithm* $\Phi^{(n)}$ *is*

$$\mathcal{E}(\Phi^{(n)}) = \lim_{n\to\infty} -\tfrac{1}{n}\log P_e(\Phi^{(n)}). \tag{2.4}$$

We use Φ to denote $\Phi^{(n)}$ when clear from context.

2.2 Registration of Two Images

We first consider the problem of registering two images, i.e., $m = 2$. Thus the problem reduces to registering an image $\mathbf{Y}$ obtained as a result of transforming the output of an

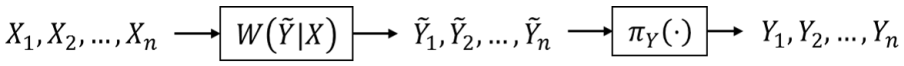


Figure 2.2: Two image registration model: We register image $\mathbf{Y}$ to reference $\mathbf{X}$.

equivalent discrete memoryless channel W, given input image (reference) $\mathbf{X}$. The channel model is depicted in Fig. 2.2.

This problem has been well-studied in practice using the MMI method defined as

$$\hat{\pi}_{\mathrm{MMI}} = \arg\max_{\pi \in \Pi} \hat{I}(X; Y_\pi), \tag{2.5}$$

where $\hat{I}(X;Y)$ is the mutual information of empirical distribution (plug-in estimate) $\hat{P}(x,y) = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}\{X_i = x, Y_i = y\}$, where $\mathbf{1}\{\cdot\}$ is the indicator function. Note that MMI is universal.

Since transformations are chosen uniformly at random, the maximum likelihood (ML) estimate is Bayes optimal:

$$\hat{\pi}_{\mathrm{ML}} = \arg\max_{\pi \in \Pi} \prod_{i=1}^{n} W\left(Y_{\pi(i)}|X_i\right). \tag{2.6}$$

We first show that the MMI method, and consequently the ML method, are exponentially consistent. We then show that the error exponent of MMI matches that of ML.

2.2.1 Exponential Consistency

The plug-in estimates of mutual information, given i.i.d. samples, are exponentially consistent [67, Lem. 7].

Theorem 1. *MMI and ML are exponentially consistent.*

Proof. Let $\Phi_{\mathrm{MMI}}(\mathbf{X}, \mathbf{Y}) = \hat{\pi}_{\mathrm{MMI}}$ and let the correct registration be π^*. Then,

$$\begin{aligned} P_e(\Phi_{\mathrm{MMI}}) &\leq \sum_{\pi \in \Pi} \mathbb{P}\left[\hat{I}(X; Y_\pi) > \hat{I}(X; \tilde{Y})\right] && (2.7)\\ &\leq \sum_{\pi \in \Pi} \mathbb{P}\left[\hat{I}(X; Y_\pi) + |\hat{I}(X; \tilde{Y}) - I(X; \tilde{Y})| > I(X; \tilde{Y})\right] && (2.8)\\ &\leq 2|\Pi| \exp\left\{-Cn(I(X; \tilde{Y}))^4\right\}, && (2.9) \end{aligned}$$

where (2.7) and (2.8) follow from the union bound, and the triangle inequality respectively. Here, we know that $I(X;Y_\pi) = cI(X;Y)$, for some constant $c < 1$, as the identity block size is $o(n)$. So using union bound and [67, Lem. 7], for a constant $C \asymp r^{-2}$ we get (2.9). Since

$|\Pi| = O(n^\alpha)$, MMI is exponentially consistent. Finally, $P_e(\Phi_{\mathrm{ML}}) \leq P_e(\Phi_{\mathrm{MMI}})$ and thus, the ML estimate is also exponentially consistent. □

Theorem 1 implies there exists $\epsilon > 0$ such that

$$\mathcal{E}(\Phi_{\mathrm{ML}}) \geq \mathcal{E}(\Phi_{\mathrm{MMI}}) \geq \epsilon.$$

2.2.2 Whittle's Law and Markov Types

We now summarize a few results on the number of types and Markov types which are eventually used to analyze the error exponent of image registration.

Consider a sequence $\mathbf{x} \in \mathcal{X}^n$. The empirical distribution q_X of $\mathbf{x}$ is the *type* of the sequence. Let $X \sim q_X$ be a dummy random variable. Let T_X^n be the set of all sequences of length n, of type q_X. The number of possible types of sequences of length n is polynomial in n, i.e., $O(n^{|\mathcal{X}|})$ [68].

The number of sequences of length n, of type q_X, is

$$|T_X^n| = \frac{n!}{\prod_{a\in\mathcal{X}}(nq_X(a))!}.$$

From bounds on multinomial coefficients, the number of sequences of length n and type q is bounded as [68]

$$(n+1)^{-|\mathcal{X}|}2^{nH(X)} \leq |T_X^n| \leq 2^{nH(X)}. \tag{2.10}$$

Consider a Markov chain defined on the space $[k]$. Given a sequence of $n+1$ samples from $[k]$ we can compute the matrix $\mathbf{F}$ of transition counts, where F_{ij} corresponds to the number of transitions from state i to state j. By Whittle's formula [69], the number of sequences $(a_1, \ldots, a_{n+1})$ with $a_i \in [k], i \in [n+1]$, with $a_1 = u$ and $a_{n+1} = v$ is

$$N_{uv}^{(n)}(F) = \prod_{i\in[k]} \frac{(\sum_{j\in[k]} F_{ij})!}{\prod_{j\in[k]} F_{ij}!} G_{vu}^*, \tag{2.11}$$

where G_{vu}^* corresponds to the (v,u)th cofactor of the matrix $G = \{g_{ij}\}_{i,j\in[k]}$ with

$$g_{ij} = \mathbf{1}\{i=j\} - \frac{F_{ij}}{\sum_{j\in[k]} F_i j}.$$

The first-order Markov type of a sequence $\mathbf{x} \in \mathcal{X}^n$ is defined as the empirical distribution

q_{X_0,X_1}, given by

$$q_{X_0,X_1}(a_0,a_1) = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}\left\{(x_i,x_{i+1}) = (a_0,a_1)\right\}.$$

Here we assume that the sequence is cyclic with period n, i.e., for any $i > 0, x_{n+i} = x_i$. Let $(X_0,X_1) \sim q_{X_0,X_1}$. Then, from (2.11), the set of sequences of type q_{X_0,X_1}, $T^n_{X_0,X_1}$, satisfies

$$|T^n_{X_0,X_1}| = \left(\sum_{a\in\mathcal{X}} G^*_{a,a}\right) \prod_{a_0\in\mathcal{X}} \frac{(nq_0(a_0))!}{\prod_{a_1\in\mathcal{X}}(nq_{\mathbf{x}_0,\mathbf{x}_1}(a_0,a_1))!}.$$

From the definition of G, we can bound the trace of the cofactor matrix of G as

$$\frac{|\mathcal{X}|}{(n+1)^{|\mathcal{X}|}} \le \sum_{a\in\mathcal{X}} G^*_{a,a} \le |\mathcal{X}|.$$

Again using the bounds on multinomial coefficients, we have

$$|\mathcal{X}|(n+1)^{-(|\mathcal{X}|^2+|\mathcal{X}|)}2^{n(H(X_0,X_1)-H(X_0))} \le |T^n_{X_0,X_1}| \le |\mathcal{X}|2^{n(H(X_0,X_1)-H(X_0))}. \tag{2.12}$$

The joint first-order Markov type of a pair of sequences $\mathbf{x}\in\mathcal{X}^n$, $\mathbf{y}\in\mathcal{Y}^n$ is the empirical distribution

$$q_{X_0,X_1,Y}(a_0,a_1,b) = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}\left\{(x_i,x_{i+1},y_i) = (a_0,a_1,b)\right\}.$$

Then given $\mathbf{x}$, the set of conditional first-order Markov type sequences, $T^n_{Y|X_0,X_1}(\mathbf{x})$ satisfies [60]

$$(n+1)^{-|\mathcal{X}|^2|\mathcal{Y}|}2^{n(H(X_0,X_1,Y)-H(X_0,X_1))} \le |T^n_{Y|X_0,X_1}(\mathbf{x})| \le 2^{n(H(X_0,X_1,Y)-H(X_0,X_1))}. \tag{2.13}$$

Definition 7 (Permutation Type)**.** *For any permutations π_1,π_2, and sequences $\mathbf{x},\mathbf{y}$, the* permutation type*, $q_{X_{\pi_1},X_{\pi_2}}$, and the* joint permutation type *$q_{X_{\pi_1},X_{\pi_2},Y}$ are defined as*

$$q_{X_{\pi_1},X_{\pi_2}}(a_0,a_1) = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}\left\{(x_{\pi_1(i)},x_{\pi_2(i)}) = (a_0,a_1)\right\},$$

$$q_{X_{\pi_1},X_{\pi_2},Y}(a_0,a_1,b) = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}\left\{(x_{\pi_1(i)},x_{\pi_2(i)},y_i) = (a_0,a_1,b)\right\}.$$

Lemma 2. *Let $\pi_1,\pi_2\in\Pi$. For any $\mathbf{x}$,*

$$|T^n_{X_{\pi_1},X_{\pi_2}}| = 2^{n\left(H(q_{X_{\pi_1},X_{\pi_2}})-H(q_X)+o(1)\right)}.$$

The proof is given in the Appendix A.1. Similar decomposition follows for conditional types as well.

2.2.3 Error Exponent Analysis

We are interested in the error exponent of MMI-based image registration, in comparison to ML. We first note that the error exponent of the problem is characterized by the pair of transformations that are the hardest to compare.

Define $\Psi_{\pi,\pi'}$ as the binary hypothesis testing problem corresponding to image registration when the allowed transformations are only $\{\pi, \pi'\}$. Let $P_{\pi,\pi'}(\Phi), \mathcal{E}_{\pi,\pi'}(\Phi)$ be the corresponding error probability and error exponent.

Lemma 3. *If Φ is exponentially consistent, then,*

$$\mathcal{E}(\Phi) = \min_{\pi,\pi' \in \Pi} \mathcal{E}_{\pi,\pi'}(\Phi). \tag{2.14}$$

Proof. The result follows from the fact the union bound and the fact that $|\Pi| = O(n^{\alpha})$. □

Thus, it suffices to consider the binary hypothesis tests to study the error exponent of image registration. We now bound the error exponent using the method of types.

Theorem 2. *The error exponent of MMI satisfies*

$$\mathcal{E}(\Phi_{MMI}) \geq \mathcal{E}_{LB} = \min_{q \in \mathcal{P}_{\mathcal{X},\mathcal{Y}}} I(X;Y) + D(q \| P_{XY}). \tag{2.15}$$

Proof. Consider any $\pi_1 \neq \pi_2$. From symmetry, it follows that Bayes error in the binary hypothesis tests is the same as the conditional error probabilities. Further, probabilities of i.i.d. sequences are defined by their joint type,

$$\begin{aligned} P_{\pi_1,\pi_2}(\Phi_{\mathrm{MMI}}) &= \mathbb{P}\left[\hat{\pi}_{\mathrm{MMI}} = \pi_2 | \pi^* = \pi_1\right] \\ &= \sum_{\mathbf{x} \in \mathcal{X}^n} \sum_{\mathbf{y} \in \mathcal{Y}^n} \mathbb{P}\left[\mathbf{x}, \mathbf{y}\right] \mathbf{1}\left\{\hat{\pi}_{\mathrm{MMI}} \neq \pi^*\right\} \\ &= \sum_{q} \left(\prod_{a \in \mathcal{X}} \prod_{b \in \mathcal{Y}} (\mathbb{P}\left[a, b\right])^{nq(a,b)} \right) \nu_{\mathrm{MMI}}(q), \end{aligned} \tag{2.16}$$

where the summation in (2.16) is over the set of all joint types of sequences of length n and $\nu_{\mathrm{MMI}}(q)$ is the number of sequences $(\mathbf{x}, \mathbf{y})$ of length n such that the joint type of $(\mathbf{x}_{\pi_1}, \mathbf{y})$ is q, and MMI makes an error in the binary hypothesis for $\pi^* = \pi_1$.

If a sequence $\mathbf{y} \in T^n_{Y|X_{\pi_1},X_{\pi_2}}(\mathbf{x})$ is in error, then all sequences in $T^n_{Y|X_{\pi_1},X_{\pi_2}}(\mathbf{x})$ are in error as $\mathbf{x}$ is drawn from an i.i.d. source. Thus,

$$\begin{aligned}\nu(q) &= \sum_{\mathbf{x}\in\mathcal{X}^n} \sum_{T^n_{Y|X_{\pi_1},X_{\pi_2}} \subseteq T^n_{Y|X}:\text{error}} |T^n_{Y|X_{\pi_1},X_{\pi_2}}(\mathbf{x})| \\ &= \sum_{T^n_{X_{\pi_1},X_{\pi_2}} \subseteq T^n_X} |T_{X_{\pi_1},X_{\pi_2}}| \sum |T^n_{Y|X_{\pi_1},X_{\pi_2}}|,\end{aligned}$$

where the sum is taken over $T^n_{Y|X_{\pi_1},X_{\pi_2}} \subseteq T^n_{Y|X}$ such that there is a decision error, i.e., $I(X_{\pi_1};Y) \leq I(X_{\pi_2};Y)$. The final line follows from the fact that given the joint first-order Markov type, the size of the conditional type is independent of the exact sequence $\mathbf{x}$.

From Lem. 2, we then have

$$\nu_{\text{MMI}}(q) \asymp \sum 2^{n\left(H(X_{\pi_1},X_{\pi_2},Y)-H(X_{\pi_1})\right)}, \tag{2.17}$$

where the sum in (2.17) is the same as in the definition of ν_{MMI}. Further, the probability of a pair of sequences $(\mathbf{x}_{\pi_1}, \mathbf{y})$ of joint type q is [68]

$$\mathbb{P}\left[\mathbf{x}, \mathbf{y}\right] = 2^{-n\left(H(X_{\pi_1},Y)+D(q\|P_{XY})\right)}. \tag{2.18}$$

From (2.17) and (2.18), we have

$$\begin{aligned}P_{\pi_1,\pi_2}(\Phi_{\text{MMI}}) &\asymp \sum_q \sum 2^{-n\left[D(q\|P_{XY})+H(X_{\pi_1},Y)+H(X\pi_2)-H(X_{\pi_1},X_{\pi_2},Y)\right]} \\ &= \sum_q \sum 2^{-n\left[D(q\|P_{XY})+I(X_{\pi_2};X_{\pi_1},Y)\right]}.\end{aligned} \tag{2.19}$$

First, from the chain rule and non-negativity of mutual information, note that

$$\begin{aligned}I(X_{\pi_2};X_{\pi_1},Y) &= I(X_{\pi_2};Y) + I(X_{\pi_2};X_{\pi_1}|Y) \\ &\geq I(X_{\pi_2};Y) \geq I(X_{\pi_1};Y),\end{aligned}$$

which follows from the error criterion. Now, from (2.19), and the fact that the number of types scales polynomially in the length of the sequence, the result follows. □

Remark 3. *The error exponent lower bound in Thm. 2 is in principle the same as the random coding error exponent for universal communication using constant composition codes [68], since registration effectively tries to differentiate between a set of possible permutations of the same sequence. However, unlike in communication, we have no control over the design*

of a "codebook" of desired type. So our bound is smaller than in communication.

On the other hand, image registration is simpler since we only have a polynomial number of possible hypotheses rather than the exponential number of messages in communication. The balance between these two tradeoffs manifests in the form of the lower bound established here.

Remark 4. *To establish a universal bound on the error exponent, note that we have significantly loosened the bound from* (2.19)*. Further information on the relationship between the candidate permutations could help establish tighter bounds on the error exponent of the MMI method.*

Remark 5. *The dominant error event is highlighted by the error exponent lower bound in the form of the joint type that minimizes the sum of its distance from the true joint distribution, and the information content in this type. We can directly observe that*

$$\mathcal{E}_{LB} \leq \min\left\{I(X;Y), L(X;Y)\right\}, \tag{2.20}$$

where $I(X;Y), L(X;Y)$ are the mutual information and lautum information [64] as governed by the underlying true channel and source distributions.

In particular, from (2.19)*, we can note that the worst error event is the one that minimizes the sum of the distance and the information that the pixels of the erroneous permutation hold on those of the true permutation and the copy. That is, if the permutations are such that this information content is naturally high when the sequences are close to the true joint distribution, then the error exponent achieved is high as well.*

Theorem 2 establishes a lower bound on the error exponent but its comparison to the Bayes' optimal ML estimate is not evident. We again use the method of types to show that MMI is asymptotically optimal.

Theorem 3. *For any $\pi_1, \pi_2 \subseteq \Pi$, $\pi_1 \neq \pi_2$,*

$$\lim_{n\to\infty} \frac{P_{\pi_1,\pi_2}(\Phi_{MMI})}{P_{\pi_1,\pi_2}(\Phi_{ML})} = 1. \tag{2.21}$$

Proof. From (2.16),

$$\begin{aligned} P_{\pi_1,\pi_2}(\Phi_{\mathrm{MMI}}) &= \sum_{q} \prod_{a\in\mathcal{X}} \prod_{b\in\mathcal{Y}} \left(\mathbb{P}\left[a,b\right]\right)^{nq(a,b)} \nu_{\mathrm{ML}}(q) \left[\frac{\nu_{\mathrm{MMI}}(q)}{\nu_{\mathrm{ML}}(q)}\right] \\ &\leq P_{\pi_1,\pi_2}(\Phi_{\mathrm{ML}}) \max_{q} \left\{\frac{\nu_{\mathrm{MMI}}(q)}{\nu_{\mathrm{ML}}(q)}\right\}. \end{aligned} \tag{2.22}$$

The result then follows from the forthcoming Lem. 4. □

Lemma 4.

$$\lim_{n\to\infty} \max_{q} \left\{ \frac{\nu_{MMI}(q)}{\nu_{ML}(q)} \right\} = 1. \tag{2.23}$$

Proof. Observe that for images with i.i.d. pixels, MMI is the same as minimizing the joint entropy:

$$\begin{aligned} \max_{\pi\in\Pi} \hat{I}(X;\pi(Y)) &= \max_{\pi\in\Pi} \hat{H}(X) + \hat{H}(\pi(Y)) - \hat{H}(X,\pi(Y)) \\ &= \hat{H}(X) + \hat{H}(Y) - \min_{\pi\in\Pi} \hat{H}(X,\pi(Y)). \end{aligned}$$

Further we know there is a bijective mapping between permuted sequences and those of the corresponding first-order Markov type from Lem. 2. Thus, the result follows from [60, Lem. 1]. □

Theorem 4. $\mathcal{E}(\Phi_{MMI}) = \mathcal{E}(\Phi_{ML})$.

Proof. This follows from Thm. 3 and Lem. 3. □

Thus, we can see that using MMI for image registration is not only universal, but also asymptotically optimal. We now obtain a simple upper bound on the error exponent of ML.

Theorem 5.

$$\mathcal{E}(\Phi_{ML}) \leq \mathcal{E}_{UB} = I(X;Y) + L(X;Y). \tag{2.24}$$

Proof. From Lem. 3 we know that the error exponent is the same as that of the worst binary hypothesis test. Further, the Bayes error corresponding to binary hypothesis tests can be lower bounded using the Kailath lower bound [70]. That is, for any $\pi_1 \neq \pi_2$

$$P_{\pi_1,\pi_2}(\Phi_{\mathrm{ML}}) \geq \exp\left(-D(p_1\|p_0)\right), \tag{2.25}$$

where p_0, p_1 are the conditional distributions of $\mathbf{X}, \mathbf{Y}$, given the underlying hypotheses π_1, π_2 respectively, i.e.,

$$\begin{aligned} p_0(\mathbf{X},\mathbf{Y}) &= \prod_{i=1}^{n} \mathbb{P}\left[X_{\pi_1(i)}\right] W(Y_i|X_{\pi_1(i)}), \\ p_1(\mathbf{X},\mathbf{Y}) &= \prod_{i=1}^{n} \mathbb{P}\left[X_{\pi_2(i)}\right] W(Y_i|X_{\pi_2(i)}). \end{aligned}$$

Then the KL divergence is given by

$$
\begin{aligned}
D(p_1\|p_0) &= \mathbb{E}_{p_1}\left[\log\frac{p_1(\mathbf{X},\mathbf{Y})}{p_0(\mathbf{X},\mathbf{Y})}\right] && (2.26)\\
&\asymp n\mathbb{E}_{X,\tilde{X}\sim P_X^{\otimes 2}}\left[D\left(W(Y|X)\|W(Y|\tilde{X})|X,\tilde{X}\right)\right] && (2.27)\\
&= n\left[I(X;Y)+\mathbb{E}_{X,\tilde{X}\sim P_X^{\otimes 2}}\left[D(P_Y\|W(Y|\tilde{X})|X,\tilde{X}\right]\right]\\
&= n\left[I(X;Y)+L(X;Y)\right], && (2.28)
\end{aligned}
$$

where the mutual and lautum information are defined by the source and channel distributions. Here (2.27) follows from the fact that $|\mathcal{I}_\pi| = o(n)$. Thus the result follows. □

Remark 6. *The upper bound on the error exponent in Thm. 5 is the sum information [71] in the channel defining the image pair. The information terms are the Type-1 and Type-2 error exponents corresponding to the independence testing binary hypothesis test. Image registration corresponds to identifying which of any pair of permutations results in a more dependent configuration of the images. This problem can be equivalently viewed as the result of two independence tests. Thus the probability of an error in both these tests results in an error exponent of $I(X;Y)+L(X;Y)$. Since this event is a subset of the error events, it results in the upper bound.*

To get a sense of the error exponent, in particular the lower bound, let us consider a simple context. Let the source $\mathbf{X} \overset{i.i.d.}{\sim} \text{Bern}(\rho)$ and consider a binary symmetric channel with crossover probability $\delta \in [0, 0.5]$. First, we can solve the optimization problem in (2.15) for BSC(δ) to obtain the error exponent. Let $\mathbf{q}, \mathbf{q}_x, \mathbf{q}_y$ be the vectors corresponding to the joint distribution and the marginals. Then the optimization problem is

$$
\begin{aligned}
&\min_{\mathbf{q},\mathbf{q}_x,\mathbf{q}_y} I(X;Y)+D(q\|p)\\
&\text{s.t.}\sum_{x,y} q(x,y)=1,\\
&\qquad\sum_{y} q(x,y)=q_x(x),\ \text{for all } x\\
&\qquad\sum_{x} q(x,y)=q_y(y),\ \text{for all } y.
\end{aligned}
$$

Then, solving the optimization problem using the Lagrangian

$$\mathcal{L}(\mathbf{q}, \mathbf{q}_x, \mathbf{q}_y) = \sum_{x,y} q(x,y) \left[\log \left(\frac{q(x,y)^2}{q(x)q(y)p(x,y)} \right) - \lambda \right] - \sum_{x \in \mathcal{X}} \mu_x \left(\sum_y q(x,y) - q_x(x) \right) - \sum_{y \in \mathcal{Y}} \nu_y \left(\sum_x q(x,y) - q_y(y) \right).$$

where $\lambda, \{\mu_x\}_{x \in \mathcal{X}}, \{\nu_y\}_{y \in \mathcal{Y}}$ are Lagrange multipliers. Then, solving the optimization problem, we obtain

$$\mathcal{E}_{\text{BSC}}(\pi, \delta) = 1 - \log_2 \left(\sqrt{\pi} + \sqrt{1-\pi} \right) - \log_2 \left(\sqrt{\delta} + \sqrt{1-\delta} \right) \text{ bits.} \tag{2.29}$$

On the other hand, the mutual information in the channel is

$$I(X;Y) = H_2(\pi * \delta) - H_2(\delta),$$

where $H_2(\cdot)$ is the binary entropy function and $x * y = x(1-y) + (1-x)y$.

We plot $\mathcal{E}_{lb}$ and the mutual information in the channel in Fig. 2.3. Note that we plot the mutual information here instead of the sum information, as the mutual information is a tighter upper bound on the error exponent, as discussed in more detail in Chapter 3.

As expected, the noisier the channel, the worse the error exponent. These error exponents can be used to get a first-order sense of the necessary and sufficient sample complexities for two image registration. Also, the more deterministic the source, the less information the pixel pairs carry and in that sense reduce the error exponent.

Although the bounds on the error exponent are not tight, they provide insight into the dominating error event and the fundamental connection to independence testing. Further information on the nature of the transformation can lead to stronger characterization of the error exponent. We next study the problem of registering multiple images.

2.3 Multi-Image Registration

Having universally registered two images, we now consider aligning multiple copies of the same image. For simplicity, let us consider aligning three images; results can directly be extended to any finite number of images. Let $\mathbf{X}$ be the source image and $\mathbf{Y}, \mathbf{Z}$ be the noisy,

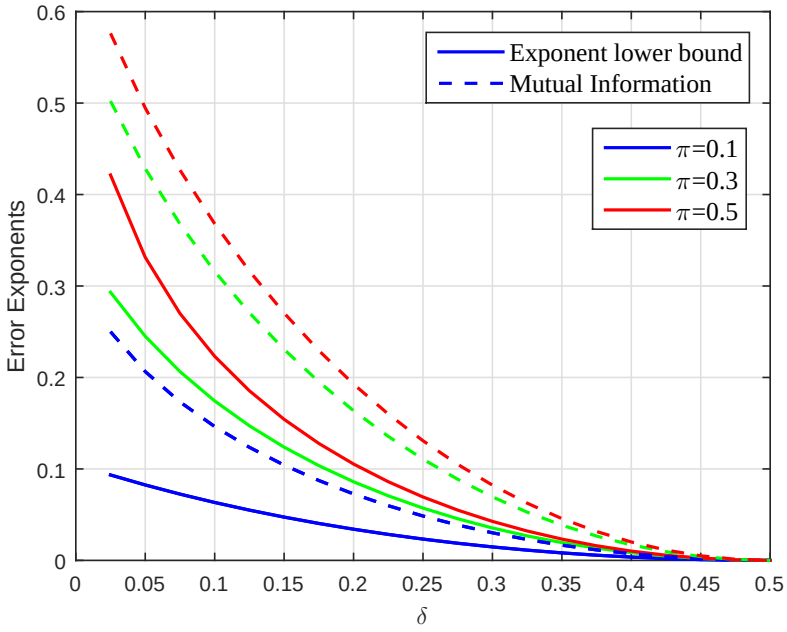


Figure 2.3: Error exponent bounds for BSC: Variation in error exponent lower bound and mutual information for varying crossover and source probability δ, π. Exponent and information decay with increasing crossover probability, and decrease as source gets more deterministic as the images have lesser information.

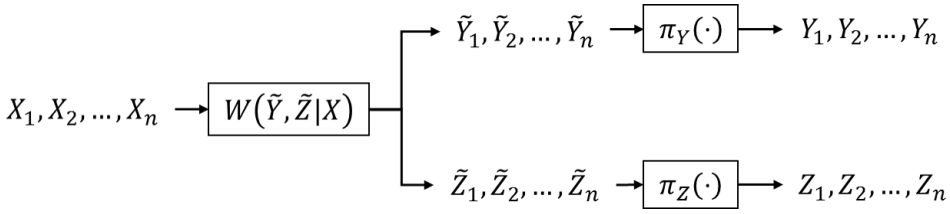


Figure 2.4: Model of three-image registration problem: Images $\mathbf{Y}, \mathbf{Z}$ are to be registered with source image $\mathbf{X}$.

transformed versions to be aligned as shown in Fig. 2.4. Here, the ML estimates are

$$(\hat{\pi}_{Y,\mathrm{ML}}, \hat{\pi}_{Z,\mathrm{ML}}) = \arg\max_{\pi_1,\pi_2} \prod_{i=1}^{n} W\left(Y_{\pi_1(i)}, Z_{\pi_2(i)}|X_i\right). \tag{2.30}$$

2.3.1 MMI Is Not Optimal

We know MMI is asymptotically optimal at aligning two images. Is pairwise MMI, i.e.

$$\hat{\pi}_Y = \arg\max_{\pi\in\Pi} \hat{I}(X; Y_\pi), \;\; \hat{\pi}_Z = \arg\max_{\pi\in\Pi} \hat{I}(X; Z_\pi), \tag{2.31}$$

optimal for multi-image registration? We show pairwise MMI is suboptimal even though individual transformations are chosen independently and uniformly from Π.

Theorem 6. *There exists channel W and prior, such that pairwise MMI is suboptimal for multi-image registration.*

Proof. Let $X_i \overset{\text{i.i.d.}}{\sim} \text{Bern}(1/2), i \in [n]$. Consider physically degraded images $\mathbf{Y}, \mathbf{Z}$ obtained as outputs of the channel $W(y, z|x) = W_1(y|x)\, W_2(z|y)$. This is depicted in Fig. 2.5. Naturally, the ML estimate is obtained by registering $\mathbf{Z}$ to $\mathbf{Y}$ and $\mathbf{X}$ to $\mathbf{Y}$, instead of registering each pairwise to $\mathbf{X}$ as evident from

$$(\hat{\pi}_{Y,\text{ML}}, \hat{\pi}_{Z,\text{ML}}) = \arg \max_{(\pi_1,\pi_2)\in\Pi^2} \prod_{i=1}^{n} W_1(Y_{\pi_1(i)}|X_i) W_2(Z_{\pi_2(i)}|Y_{\pi_1(i)}).$$

Since ML registers image $\mathbf{Y}$ to $\mathbf{X}$ and image $\mathbf{Z}$ to $\mathbf{Y}$, the error exponent of ML is $\mathcal{E}_W(\Phi_{\text{ML}}) = \min\{\mathcal{E}_{W_1}(\Phi_{\text{ML}}), \mathcal{E}_{W_2}(\Phi_{\text{ML}})\}$.

Let $\mathcal{E}_Q(\Phi_{\text{MMI}})$ be the error exponent of MMI for this channel. Then, error exponent of pairwise MMI is $\mathcal{E}(\Phi_{\text{MMI}}) = \min\{\mathcal{E}_{W_1}(\Phi_{\text{MMI}}), \mathcal{E}_{W_1 * W_2}(\Phi_{\text{MMI}})\}$. We know MMI is asymptotically optimal for two image registration and so $\mathcal{E}_{W_1}(\Phi_{\text{MMI}}) = \mathcal{E}_{W_1}(\Phi_{\text{ML}})$, $\mathcal{E}_{W_1 * W_2}(\Phi_{\text{MMI}}) = \mathcal{E}_{W_1 * W_2}(\Phi_{\text{ML}})$.

More specifically, let $W_1 = BSC(\alpha)$ and $W_2 = BSC(\beta)$ for some $\alpha, \beta \in (0, 1/2)$. Let, $W_1 * W_2 = BSC(\gamma)$, where $\gamma = \alpha(1-\beta) + (1-\alpha)\beta > \max\{\alpha, \beta\}$. Then,

$$\begin{aligned} \mathcal{E}(\Phi_{\text{MMI}}) &\le \mathcal{E}_{W_1 * W_2}(\Phi_{\text{MMI}}) = \mathcal{E}_{W_1 * W_2}(\Phi_{\text{ML}}) \\ &< \min\{\mathcal{E}_{W_1}(\Phi_{\text{ML}}), \mathcal{E}_{W_2}(\Phi_{\text{ML}})\} = \mathcal{E}_W(\Phi_{\text{ML}}). \end{aligned} \tag{2.32}$$

□

Suboptimality of pairwise MMI is due to the rigidity of the scheme in not considering that images $\mathbf{Y}$ and $\mathbf{Z}$ are dependent conditioned on $\mathbf{X}$. Thus, it behooves us to design universal algorithms that account for such dependencies.

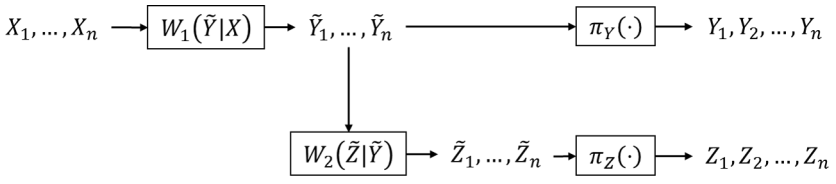


Figure 2.5: Three-image registration problem with image $\mathbf{Z}$ a physically degraded version of $\mathbf{Y}$ and to be registered with source image $\mathbf{X}$.

2.3.2 Max Multiinformation for Multi-Image Registration

To consider all correlations across images, we use the multiinformation functional [72] which is a more inclusive formulation of the underlying information in the system.

Definition 8. *The* multiinformation *of* $\{X_1, \ldots, X_n\}$ *is*

$$I_M(X_1; \ldots; X_n) = \sum_{i=1}^{n} H(X_i) - H(X_1, \ldots, X_n). \tag{2.33}$$

The chain rule for multiinformation is given by

$$I_M(X_1; \ldots; X_n) = \sum_{i=2}^{n} I(X_i; X^{[i-1]}). \tag{2.34}$$

Let us define the max multiinformation (MM) estimate, defined for three images as

$$(\hat{\pi}_{Y,\mathrm{MM}}, \hat{\pi}_{Z,\mathrm{MM}}) = \arg \max_{(\pi_1,\pi_2)\in\Pi^2} \hat{I}_M(X; Y_{\pi_1}; Z_{\pi_2}), \tag{2.35}$$

where $\hat{I}_M(\cdot)$ is the empirical estimate of multiinformation. This generalizes directly to any finite number of images. When $\mathbf{Y}$ and $\mathbf{Z}$ are conditionally independent (empirically) given $\mathbf{X}$, this is the same as pairwise MMI.

Lemma 5. *MM estimates are exponentially consistent for multi-image registration.*

Proof. The proof is analogous to that of Thm. 1. □

Again, this holds for ergodic sources. We restrict to the i.i.d. case to show asymptotic optimality using type counting.

2.3.3 Error Exponent Analysis

We compare the error exponent of MM to that of ML. We again show the error exponent is characterized by the pair of transformations of $\mathbf{Y}, \mathbf{Z}$ that are the hardest to differentiate.

Let $\bar{\pi}, \bar{\pi}' \in \Pi^2$ and let $\psi(\bar{\pi}, \bar{\pi}')$ be the binary hypothesis test of three-image registration when the set of transformations is $\{\bar{\pi}, \bar{\pi}'\}$. Let $P_{\bar{\pi},\bar{\pi}'}(\Phi), \mathcal{E}_{\bar{\pi},\bar{\pi}'}(\Phi)$ be the error probability and the error exponent of Φ respectively.

Lemma 6. *Let* Φ *be an asymptotically exponentially consistent estimator of three-image registration. Then,*

$$\mathcal{E}(\Phi) = \min_{\bar{\pi},\bar{\pi}'\in\Pi^2} \mathcal{E}_{\bar{\pi},\bar{\pi}'}(\Phi). \tag{2.36}$$

Proof. The proof is analogous to that of Lem. 3. □

Lemma 7. *Let $\bar{\pi}_0 = (\pi_0, \pi_0)$. For any $\bar{\pi}_1, \bar{\pi}_2 \in \Pi^2$, with $\bar{\pi}_i = (\pi_i, \pi_i')$, there exists $\bar{\pi} = (\pi, \pi)$ such that $\mathcal{E}_{\bar{\pi}_1,\bar{\pi}_2}(\Phi) \geq \mathcal{E}_{\bar{\pi}_0,\bar{\pi}}(\Phi)$, for $\Phi \in \{\Phi_{ML}, \Phi_{MM}\}$.*

Proof. We prove the result for ML; it extends directly for MM. Let $\tilde{\pi}_1$ be the permutation such that $\tilde{\pi}_1 \circ \pi_1' = \pi_1$. That is, the application of the transformation $\tilde{\pi}_1$ to an image that has been transformed by π_1' results in an image that is effectively transformed by π_1. Let $\tilde{\pi}_2 = \tilde{\pi}_1 \circ \pi_2'$.

To obtain the ML decision for $\psi(\bar{\pi}_1, \bar{\pi}_2)$, let

$$(\hat{\pi}_Y, \hat{\pi}_Z) = \arg \max_{(\pi_Y,\pi_Z)\in\{(\pi_1,\pi_1),(\pi_2,\tilde{\pi}_2)\}} W(\mathbf{Y}_{\pi_Y}, \mathbf{Z}'_{\pi_Z}|\mathbf{X}),$$

where $\mathbf{Z}' = \mathbf{Z}_{\tilde{\pi}_1}$, is the received image, transformed by $\tilde{\pi}_1$. Then, $(\hat{\pi}_{Y,\mathrm{ML}}, \hat{\pi}_{Z,\mathrm{ML}}) = (\hat{\pi}_Y, \tilde{\pi}_1^{-1} \circ \hat{\pi}_Z)$.

Since the source is i.i.d. and the channel is memoryless, $\mathcal{E}_{\bar{\pi}_1,\bar{\pi}_2}(\Phi_{\mathrm{ML}}) = \mathcal{E}_{(\pi_1,\pi_1),(\pi_2,\tilde{\pi}_2)}(\Phi_{\mathrm{ML}})$. Finally, if π_1^{-1} is the inverse π_1, let $\pi = \pi_1^{-1} \circ \pi_2, \pi' = \pi_1^{-1} \circ \pi_2'$. Then,

$$(\hat{\pi}_Y, \hat{\pi}_Z) = \arg \max_{(\pi_Y,\pi_Z)\in\{(\pi_0,\pi_0),(\pi,\pi')\}} W(\mathbf{Y}'_{\pi_Y}, \mathbf{Z}'_{\pi_Z}|\mathbf{X}),$$

where $\mathbf{Y}' = \mathbf{Y}_{\pi_1^{-1}}$ and $\mathbf{Z}' = \mathbf{Z}_{\pi_1^{-1}\circ\tilde{\pi}_1}$, then

$$(\hat{\pi}_{Y,\mathrm{ML}}, \hat{\pi}_{Z,\mathrm{ML}}) = (\pi_1 \circ \hat{\pi}_Y, (\pi_1^{-1} \circ \tilde{\pi}_1)^{-1} \circ \hat{\pi}_Z).$$

We note now that

$$D(P_{(\pi_0,\pi_0)}(X,Y,Z)\|P_{(\pi,\pi')}(X,Y,Z)) = I_M(X;Y;Z). \tag{2.37}$$

Alternately,

$$D(P_{(\pi_0,\pi_0)}(X,Y,Z)\|P_{(\pi,\pi)}(X,Y,Z)) = I(X;Y,Z). \tag{2.38}$$

From (2.34), it is evident that the binary hypothesis test in the second scenario is harder than the first. That is, identifying if sequences are scrambled is easier if they are scrambled by different transformations than by the same.

As the sequence of transformations is deterministic, the source i.i.d. and the channel memoryless, the result follows. The same arguments hold for the MM decoder. □

Lemma 7 implies that to study the error exponent of the multi-image registration, it suffices to study those of binary hypothesis tests of the form $\psi(\bar{\pi}_0, \bar{\pi})$, for all $\pi \in \Pi$. Now we analyze error exponents of $\psi(\bar{\pi}_0, \bar{\pi})$.

Theorem 7. *Let $\bar{\pi} = (\pi, \pi) \in \Pi^2$. Then,*

$$\lim_{n\to\infty} \frac{P_{\bar{\pi}_0,\bar{\pi}}(\Phi_{MM})}{P_{\bar{\pi}_0,\bar{\pi}}(\Phi_{ML})} = 1. \tag{2.39}$$

Proof. The analysis is similar to that of Thm. 3.

$$P_{\bar{\pi}_0,\bar{\pi}}(\Phi_{\mathrm{MM}}) \leq P_{\bar{\pi}_1,\bar{\pi}_2}(\Phi_{\mathrm{ML}}) \max_q \left\{ \frac{\nu_{\mathrm{MM}}(q)}{\nu_{\mathrm{ML}}(q)} \right\}.$$

Here

$$\begin{aligned}\nu_{\mathrm{MM}}(q) &= \sum_{\mathbf{y},\mathbf{z}} \sum |T^n_{X|Y_{\pi_0},Z_{\pi_0},Y_\pi,Z_\pi}(\mathbf{y},\mathbf{z})| \\ &= \sum |T^n_{Y_{\pi_0},Z_{\pi_0},Y_\pi,Z_\pi}| \sum |T^n_{X|Y_{\pi_0},Z_{\pi_0},Y_\pi,Z_\pi}|,\end{aligned}$$

where the first sum is taken over the set of all types $T^n_{Y_{\pi_0},Z_{\pi_0},Y_\pi,Z_\pi} \subseteq T^n_{Y,Z}$ and the second sum is over the set of all conditional types $T^n_{X|Y_{\pi_0},Z_{\pi_0},Y_\pi,Z_\pi} \subseteq T^n_{X|Y,Z}$ such that Φ_{MM} gives a decision error. Similarly, $\nu_{\mathrm{ML}}(q)$ is defined analogously for ML decision errors. The result follows from the forthcoming Lem. 8. □

Lemma 8. *There exists a non-negative sequence $\{\epsilon_n\}_{n\geq 1}$ with $\lim_{n\to\infty} n\epsilon_n = 0$, such that*

$$\max_q \left\{ \frac{\nu_{MM}(q)}{\nu_{ML}(q)} \right\} \leq 2^{n\epsilon_n}.$$

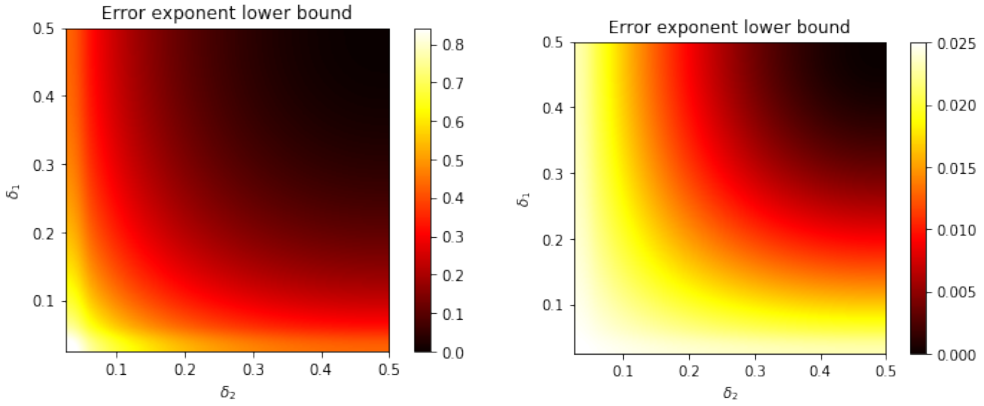
Proof. Note that maximizing multiinformation is the same as minimizing joint entropy. Thus, it is the same as Lem. 4 with a larger output alphabet and so the result follows. □

Theorem 8. $\mathcal{E}(\Phi_{MM}) = \mathcal{E}(\Phi_{ML})$.

Proof. The result follows from Lem. 6 and 7, and Thm. 7. □

These results directly extend to registering any finite number of images, indicating that the MM method for multi-image registration is asymptotically optimal and universal.

Remark 7. *The results for image models with i.i.d. pixels extend directly to memoryless or exchangeable sources. That is, we can generalize the proofs of asymptotic optimality of the MM method for image registration to image models where pixels are drawn from a memoryless or an exchangeable distribution. This follows as the probabilities of sequences with the same type are of equal probability for such sources.*



(a) High Prior: When the source is more random, the error exponents are better as there is more information in the image triplet.

(b) Low Prior: When the source is more deterministic, the error exponents are worse as the pixels are harder to differentiate.

Figure 2.6: Error exponent lower bounds for independent BSCs.

Corollary 1. *The error exponents of multi-image registration satisfy*

$$\min_{q\in\mathcal{P}_{\mathcal{X},\mathcal{Y},\mathcal{Z}}} I(X;Y;Z) + D(q\|P_{XYZ}) = \mathcal{E}_{LB}^{(m)} \leq \mathcal{E}(\Phi_{MMI})$$
$$= \mathcal{E}(\Phi_{ML}) \leq \mathcal{E}_{UB}^{(m)} = I(X;Y;Z) + L(X;Y;Z).$$

Proof. Proof is analogous to those of Thms. 2 and 5. □

Again, although not tight, the bounds on the error exponent provide basic insight into the nature of the error exponent, and the worst error event.

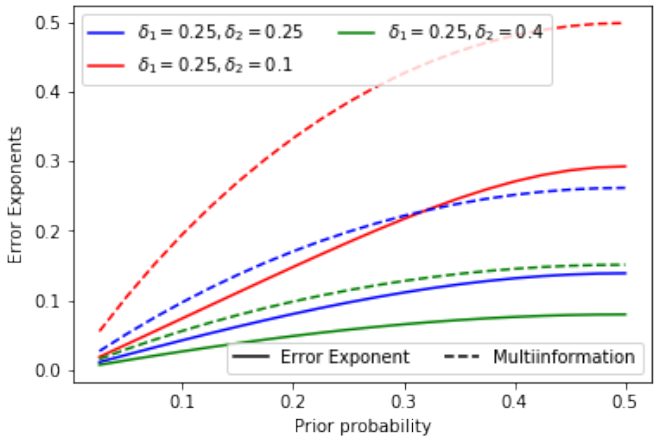
To understand the error exponents better let us consider a binary source $\mathbf{X} \overset{i.i.d.}{\sim} \text{Bern}(\rho)$. First let us consider the independent noise channel

$$W(Y,Z|X) = W_1(Y|X)W_2(Z|X),$$

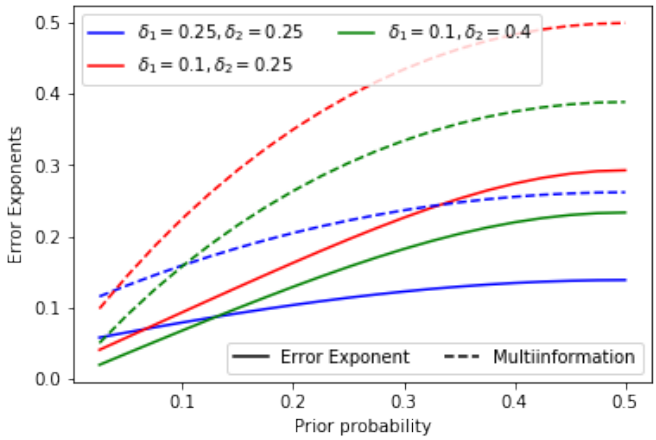
where W_1, W_2 are binary symmetric channels with crossover probabilities δ_1, δ_2 respectively. Then, the error exponent lower bound as a function of the crossover probabilities is shown in for prior probabilities $\rho = 0.5, \rho = 0.05$ respectively in Figs. 2.6a and 2.6b.

Since the channels are conditionally independent, the registration problem translates to one of pairwise image registration with image $\mathbf{X}$. The independence also results in the symmetry in the error exponents as observed in the heatmaps. That is, $\mathcal{E}_{lb}(\rho, \delta_1, \delta_2) = \mathcal{E}_{lb}(\rho, \delta_2, \delta_1)$.

Further, note that the error exponents achieved in the case of $\rho = 0.5$ is far more than that for $\rho = 0.05$. This is owing to the fact that more randomness in the source also



(a) Independent BSC: Exponents and multiinformation increase with prior and when noise is less, as image triplets are more informative.



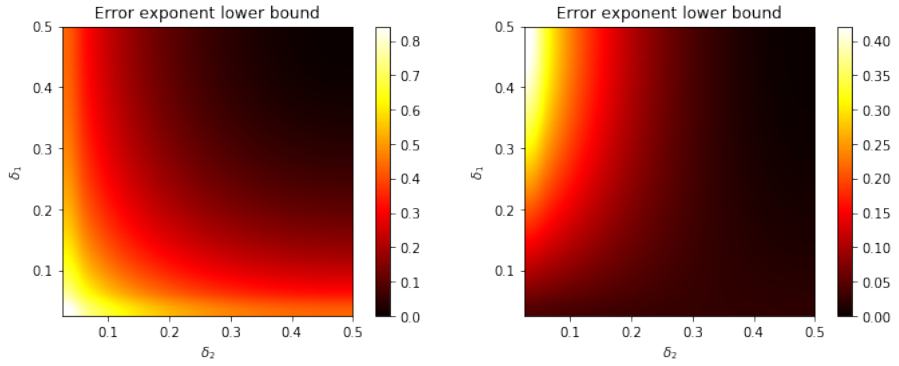
(b) Degraded BSC: Exponents and multiinformation are improved with the use of better cascades, especially when the prior of $\mathbf{Y}$ is close to uniform.

Figure 2.7: Error exponent variation with prior.

translates to more information in the information triplet. Thus it is easier to register when the source generates on average an equal number of both kinds of pixels. This is further evident in Fig. 2.7a which shows the variation of the error exponent lower bound and the multiinformation with the prior probability.

Let us now consider the degraded channel model from Fig. 2.5 where W_1 and W_2 are binary symmetric channels with crossover probability δ_1, δ_2 respectively. First, the error exponent lower bounds for prior probabilities $\rho = 0.5, \rho = 0.05$ respectively are shown in Figs. 2.8a and 2.8b respectively.

We first observe that in the case of $\rho = 0.5$, the error exponents are symmetric. This follows since this context corresponds precisely to Fig. 2.6a as image $\mathbf{Y}$ can be considered as the



(a) High Prior: When the source is uniformly distributed the error exponents are symmetric with respect to the crossover probabilities.

(b) Low Prior: When source is more deterministic, exponents are not symmetric and the most informative context is the one in which the marginal of $\mathbf{Y}$ is close to uniform.

Figure 2.8: Error exponent lower bounds for degraded BSC.

uniform source with independent BSCs generating images $\mathbf{X}, \mathbf{Z}$. Thus the error exponents are not only symmetric, but also the same as in the previous example.

On the other hand, when $\rho = 0.05$, the source pixels are more deterministic therein generating more pixels of one kind than the other. However, we have a degraded channel and thus when the marginal corresponding to $\mathbf{Y}$ is uniform, registration becomes easier. This is owing to the fact that the information in the triplet is higher, especially between $\mathbf{Y}$ and $\mathbf{Z}$. Thus the channels close to $\delta_1 = 0.5$ result in higher error exponents since it is easier to register $\mathbf{Y}$ with $\mathbf{Z}$. This is reflected in the maximum error exponent achieved by the pair $(\delta_1, \delta_2) = (0.5, 0)$.

The relationship of the error exponent to the prior probability of the source is similar to the previous example and results in the variation shown in Fig. 2.7b. We can again note that larger priors and less channel noise in the degraded model result in better error exponents. Note, however, there is a non-trivial gap between the error exponent lower bound and the multiinformation in the system that can be improved with a stronger understanding of the set of transformations and type counting.

2.4 Large-Scale Registration of Images

Practical applications such as cryo-EM often have to work with massive datasets of images, each having limited resolution. Especially, if the number of pixels in each image is sub-exponential in the number of images, i.e., $n = o\left(\frac{r^m}{m}\right)$, then, accurate computation of the multiinformation of all images is not universally feasible [73].

In such a context, it is important to understand the sample complexity of the problem and design algorithms accordingly. We consider a divide and conquer approach to registration here. Consider a set of corrupted and transformed images $\mathbf{Y}^{(1)}, \ldots, \mathbf{Y}^{(m)}$. Even though MM is asymptotically optimal for image registration, it is not feasible due to limited availability of pixels. We first establish necessary conditions on the sample complexity, before defining an achievable scheme.

2.4.1 Necessary Conditions

Note that registering images is equivalent to clustering them by the transformations. We thus use ideas from [74] to lower bound registration cost.

Theorem 9. *Given a set of m copies of an image, channel-aware registration is feasible only if $n = \Omega(\log m)$.*

Proof. Consider the simpler case where only two transformations exist, $\Pi = \{\pi_0, \pi\}$. Group the set of images into pairs at random. For ease, assume that m is divisible by 2. Define the binary hypothesis tests given by

$$\psi_{ij} : \begin{cases} H_0 : \pi_i = \pi_j, \\ H_1 : \pi_i \neq \pi_j. \end{cases}$$

Then, from the Kailath lower bound [70], we have

$$\mathbb{P}\left[\text{Error in } \psi_{ij}\right] \geq \exp\left(-\tfrac{1}{2}(D(p_0\|p_1) + D(p_1\|p_0))\right), \tag{2.40}$$

where p_i is the conditional distribution of the observation, given hypothesis H_i. Let $\Delta_{i,j} = \frac{1}{2}(D(p_0\|p_1) + D(p_1\|p_0))$ for each $(i,j) \in \psi$, and let $\Delta_{\max} = \max_{(i,j)\in\psi} \Delta_{i,j}$.

Here, observations are transformed images; under H_0,

$$p_0(\mathbf{Y}^{(i)}, \mathbf{Y}^{(j)}) = \prod_{k=1}^{n} \mathbb{P}\left[Y_k^{(i)}\right] \tilde{W}(Y_k^{(j)} | Y_k^{(i)}),$$

where $\tilde{W}$ is the equivalent channel relating image $\mathbf{Y}^{(i)}$ to $\mathbf{Y}^{(j)}$; $\tilde{W}$ is specific to the pair (i,j)

Algorithm 1 Blockwise Registration

$k \leftarrow \arg\min_{2\le\ell\le(\log n)/(\log r)} \ell \log|\Pi| - \frac{nc}{r^\ell}\ell^4(\log r)^4 - \log(\ell-1)$, where c is a sufficiently small constant
for $i = 1$ to $\lceil \frac{m-1}{k-1} \rceil$ **do**
  $T_i \leftarrow \{(i-1)*(k-1)+1, \ldots, i*(k-1)\} \cup \{m\}$
  Determine $\{\hat{\pi}_j : j \in T_i\}$ using MM method
end for

and we have ignored the indices for simplicity. Similarly, under hypothesis H_1

$$p_1(\mathbf{Y}^{(i)}, \mathbf{Y}^{(j)}) = \frac{1}{2}\prod_{k=1}^{n} \mathbb{P}\left[Y^{(i)}_{\pi(k)}\right] \tilde{W}\left(Y^{(j)}_k | Y^{(i)}_{\pi(k)}\right) + \frac{1}{2}\prod_{k=1}^{n} \mathbb{P}\left[Y^{(i)}_{\pi^{-1}(k)}\right] \tilde{W}\left(Y^{(j)}_k | Y^{(i)}_{\pi^{-1}(k)}\right).$$

Let $q(\mathbf{Y}^{(i)}, \mathbf{Y}^{(j)}) = \prod_{k=1}^{n} \mathbb{P}\left[Y^{(i)}_{\pi(k)}\right] \tilde{W}\left(Y^{(j)}_k | Y^{(i)}_{\pi(k)}\right)$. Then,

$$D(p_0 \| q) \asymp n\left[I(Y^{(i)}; Y^{(j)}) + L(Y^{(i)}; Y^{(j)})\right] = n\Delta_{\max}, \tag{2.41}$$

which follows in the same way as in (2.28).

Correct registration implies correct inference in all the hypothesis tests ψ_{ij}, and so

$$\begin{aligned} P_e(\Phi) &\ge 1 - \prod_{(i,j)\in\psi} \left(1 - \mathbb{P}\left[\text{Error in } \psi_{ij}\right]\right) \\ &\gtrsim 1 - \left(1 - \exp\left(-n\Delta_{\max}\right)\right)^{m/2}, \end{aligned} \tag{2.42}$$

where (2.42) follows from (2.40). Finally, from (2.42), Φ is consistent only if $n = \Omega(\log m)$. □

2.4.2 Achievable Scheme

Having obtained necessary conditions for large-scale image registration, we now provide an information-based registration algorithm that is order-optimal in the number of pixels.

We know MM is asymptotically optimal for a fixed and finite number of images. Thus, we can register the images by splitting them into subsets of appropriate size, keeping one reference common in all sets, Alg. 1.

The size of the subsets of images is essentially chosen to be a constant and for $k = 2$ reduces to pairwise MMI.

Theorem 10. *For any channel W, and sufficiently large number of pixels $n = O(\log m)$, Alg. 1 is consistent.*

Proof. From Lem. 5, we know MM is exponentially consistent. In particular, for registering k images with multiinformation γ_k, there exists a universal constant c such that

$$P_e(\Phi^k) \leq 2|\Pi|^k \exp\left(-n\frac{c}{r^k}\gamma_k^4\right).$$

From union bound and Lem. 5, error probability of Alg. 1 is

$$P_e(\Phi) \leq 2\frac{m-1}{k-1}|\Pi|^k \exp\left(-n\frac{c}{r^k}\gamma_k^4\right). \tag{2.43}$$

For any fixed k, we note that the sufficient condition for consistency is $n = O(\log m)$. □

From (2.43), we observe that a viable choice of k can be obtained by minimizing the upper bound. Since γ_k is proportional to the minimum multiinformation of k images, and since we have no additional knowledge regarding the scaling of such information, one choice is to use the scaling of its trivial upper bound given by $\gamma_k \leq k \log r$ and minimize the resulting expression as in Alg. 1.

Thus, we note that using a blockwise MM method, we can perform order-optimal registration of a large number of images with the number of pixels scaling logarithmically with the number of images. This scaling also explains the efficiency of simple pairwise MMI methods in practice.

2.5 Experiments

We now present the results of some experiments of the pairwise MMI and MM image registration techniques applied to a set of multispectral images from the CAVE multispectral image dataset [63, 75]. We choose multispectral images as they constitute a collection of images of the same scene varying in terms of the constituent frequency patterns, resulting in signals whose pixel distributions are quite divergent. However, the multispectral database retains all the structural information in the images, thereby retaining a high mutual information among aligned orientations of different images. Thus, the MMI and MM methods are appropriate for such a context.

We consider the three images in Fig. 2.9, and add i.i.d. Gaussian noise to the image pixels, reducing the SNR and making the registration harder. We then rotate the images at random, and test both pairwise MMI and MM.

We consider a simple set of rotations in the range of $[-60°, 60°]$, uniformly quantized into 25 levels. We restrict the set of transformations owing to the computational complexity of the registration algorithms. We also restrict to only three images to cater to the sample complexity of the information estimates, beside the computational complexity. In practice, efficient heuristics can be developed to solve the optimization problem without exhaustive search over the space of permutations.

Further, evaluate the joint distributions of the corresponding pixels, we use Gaussian kernel density estimates. This method significantly speeds up the process over the empirical estimate, and when the number of pixels is large enough, proves to be a good approximation.

First consider the case where the SNR of the reference image (first image) is set to 40 dB. As can be observed in Fig. 2.10a, the structural features of the images are well-retained in the noisy images. Thus pairwise registration using MMI suffices in this case.

We saw in Sec. 2.3.1 that one setting where MM is better than MMI is when the noise model sets up a Markovian relation in the images. So, we also consider a low SNR version of 20 dB for the reference image, with a noise model such that the three images constitute a Markov chain. This is shown in Fig. 2.10b, and it can be noted that the first image has very few feature points that can be directly matched with the third. In particular, note that the first and second images are closer to each other, while the second and third are closer to each other. This is reflected in the results of the two methods. Since pairwise MMI attempts aligning all images to the first, it fails. On the other hand, as MM exploits all existing dependencies among the images, aligns the images better, as expected.

Registering low SNR images that may have such correlation is important especially for applications such as cryo-electron microscopy where images are typically very noisy, and of molecules in various orientations [44]. The efficiency of MM for image registration inspires the design of various information-based efficient heuristics for multi-image registration.

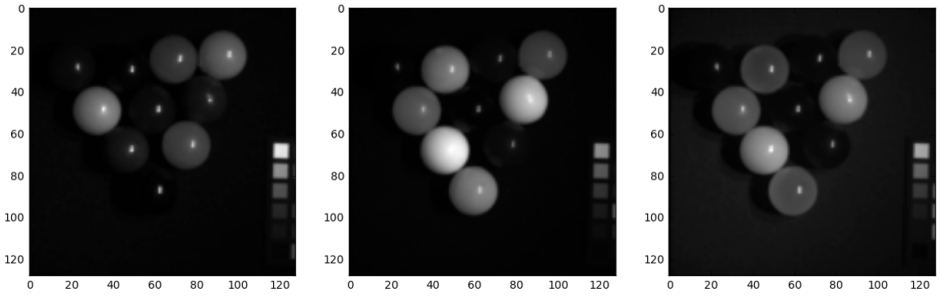
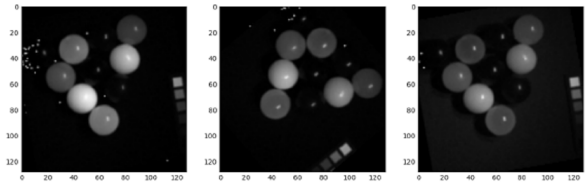
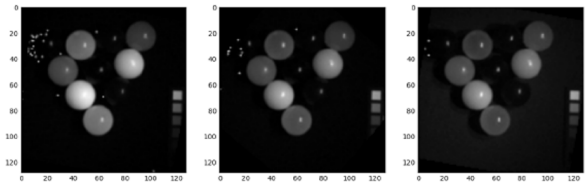


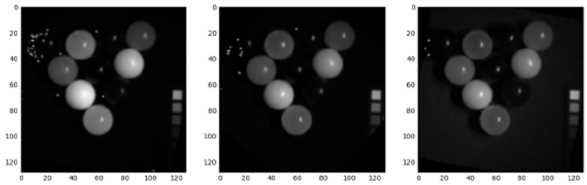
Figure 2.9: Pre-registered multispectral image set.



(a) Noisy, misaligned images

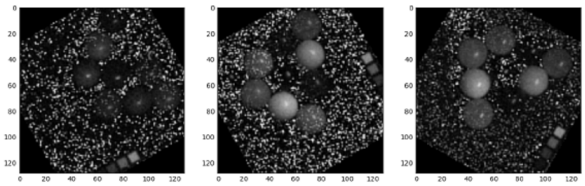


(b) Image registration with pairwise MMI

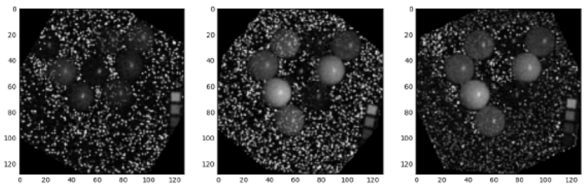


(c) Image registration with MM

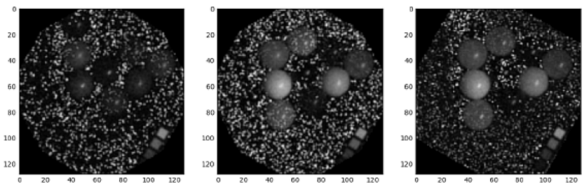
(a) High SNR: both pairwise MMI and MM work well as the information structure in the images is well-retained.



(a) Noisy, misaligned images



(b) Image registration with pairwise MMI



(c) Image registration with MM

(b) Low SNR and Markov relation: Here pairwise MMI fails to perfectly compensate for the rotation, whereas MM aligns accurately.

Figure 2.10: Image Registration using pairwise MMI and MM estimates.

2.6 Discussion

This chapter studies the problem of universal joint image registration. In particular we considered the class of rigid-body transformations of 2D images and studied the performance of algorithms in terms of the error probability of perfect alignment. This chapter performs a first-order analysis of universal algorithms in terms of their error exponents.

In Sec. 2.2 we studied the problem of registering a reference image to its misaligned noisy copy corrupted by an unknown discrete memoryless channel W. We benchmark the Bayes' optimal maximum likelihood decoder which is channel aware, to the max mutual information decoder for image registration which is universal. We studied the error exponents of the decoder by studying the class of permutation types and relating them to first order Markov types using Whittle's law. We showed the universal asymptotic optimality of the MMI decoder and established lower and upper bounds on the error exponent. We also gave a characterization of these exponents for the simplified context of a binary symmetric channel (BSC).

In Sec. 2.3 we considered the problem of registering multiple images and observed that the pairwise application of MMI is suboptimal. We then defined the max multiinformation decoder for the registration of multiple images and extend the same type counting argument to prove its universal asymptotic optimality. We also obtain a characterization of the error exponent through the lower and upper bounds, which are evaluated for the independent and degraded BSCs.

Finally we considered the problem of large-scale image registration and observed necessary and sufficient conditions on the sample complexity as a function of the number of images to be registered. The section essentially provides an asymptotic analysis for the image registration by studying the error exponents and the sample complexity in the order sense. In Chapter 3 we consider the finite sample analysis of the image registration problem to get a stronger understanding of algorithm performance for image registration using the Berry-Esseen style results for the central limit theorem.

CHAPTER 3

FINITE-SAMPLE ANALYSIS OF IMAGE REGISTRATION

Conventional information-theoretic investigations of communications study the channel capacity (mean) and error exponent (large deviations) of communication over channels. Similarly, our explorations in Secs. 2.2 and 2.3 focused on asymptotic optimality of universal algorithms in terms of the error exponent. However, we noted that the analysis does not provide us insight into the non-asymptotic regime where the image resolution is limited. The subsequent analysis in Sec. 2.4 considered the other asymptotic, and obtained some insight into the necessary and sufficient sample complexity for large-scale registration.

However, it would help to understand the tradeoff between the moments of information density of the channel and the sample complexity using finite-sample performance analysis for a finite number of images. Additionally, the performance in the image registration problem is defined by the properties of the rigid-body transformations. Under our consideration of such transformations as permutations, the sizes of the permutation cycles and the number of fixed points define the ease and/or difficulty of aligning the images. On the one hand, the more the fraction of fixed points between two permutations, the fewer the informative pixels to differentiate between them. On the other side, the more the permutation cycles between two permutations, the less informative each such cycle is. Thus it is important to understand the tradeoff between the properties of such transformations and the fundamental performance of algorithms in the image registration problem.

Going beyond the asymptotic analyses, information-theoretic studies have also been dedicated to understanding second-order terms such as the bivariate information-theoretic quantity called channel dispersion (variance) to allow better understanding of non-asymptotic performance limits of communication [76–79]. The basic idea is to not just study the mean and large deviations, but also to use the central limit theorem to study the second- and higher-order terms of the performance analysis.

In particular, the Berry-Esseen version of the central limit theorem (CLT) [80,81] highlights the fact that for a collection of n independent zero mean random variables $X_1, \ldots, X_n$, with

variances σ_i^2 respectively,

$$\sup_{a\in\mathbb{R}}\left|\mathbb{P}\left[\frac{1}{\sigma\sqrt{n}}\sum_{i=1}^{n}X_i > a\right] - Q(a)\right| \leq \frac{6T}{\sigma^3\sqrt{n}},$$

where $Q(\cdot)$ is the tail probability (complementary cumulative distribution function) of the standard normal distribution, and

$$\sigma^2 = \frac{1}{n}\sum_{i=1}^{n}\sigma_i^2, \quad \text{and } T = \frac{1}{n}\sum_{i=1}^{n}T_i = \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}\left[|\imath(X_i;Y_i) - I|^3\right].$$

Using such results, it has been observed that Gaussian approximations are good proxies to the true non-asymptotic fundamental limits at moderate blocklengths and moderate error probabilities for some channels and sources.

A large recent body of work has focused on the study of the higher-order error terms, especially in the contexts of point-to-point compression and communication. In both lossless and lossy compression, in the context of known source models, tight results on the higher-order terms have been obtained [77,82–86]. Separate from compression rates given the source, universal codes for compression are of interest as well. The set of results in this domain, though limited, have recently established second- and third-order error terms for universal lossless compression under fixed blocklength coding [87, 88].

In the communication framework as well, there exists a large body of work that characterize the non-asymptotic fundamental limits of communication [78, 89–91]. Second-order methods with sharp non-asymptotics have also been derived in designing constrained encoding and decoding for communication [91, 92]. Of particular interest to us is the set of results on non-asymptotic fundamental limits for constant composition coding [93–95].

Beside source and channel coding, a variety of finite-sample performance analyses have also been performed for statistical inference [96–98]. In particular, the conditional error probabilities for Neyman-Pearson (NP) binary hypothesis test in the finite-sample setting have been studied in detail using the Berry-Esseen theorem [77, 90]. Strong large deviations analyses informed by the Cramer-Esseen theorem have also been used in deriving strong converses in the finite-sample regime [94, 99].

In this chapter we first consider the problem of universal delay estimation and use a tighter analysis of the type counting argument to obtain a stronger characterization of the upper bound on the MMI method in terms of the size of the permutation cycles. We then build on the results in finite-sample analyses to study the fundamental performance limits of image registration for the channel-aware context. In particular we establish achievability

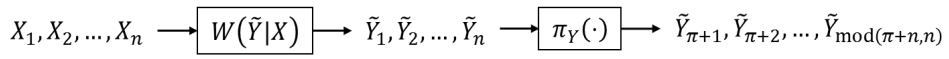


Figure 3.1: Delay Estimation Model: We estimate the cyclic delay π to align the sequence $\mathbf{Y}$ to the source $\mathbf{X}$.

arguments on the tradeoff between the moments of the information density in the channel and the sample size.

3.1 Universal Delay Estimation: Relationship to Cycles

The universal version of the image registration problem is complicated by two aspects—the growing number of hypotheses with sample size, and the complicated transformations (permutations) applied to the images. In particular, understanding the functioning of the MMI decoder is affected by the nature of these permutations and it is important to understand the effect of the number of permutation cycles on the performance of MMI.

In order to understand this, here we consider the problem of universal delay estimation [60] which is a simplified version of the image registration problem. It has been established that the MMI estimator is asymptotically universally optimal in the error exponent in detecting finite cyclic delays of memoryless sources. Here we perform a stricter analysis of the type counting argument to better emphasize the effect of the number of permutation cycles.

3.1.1 System Model

Consider a source sequence of length n, $\mathbf{X} \overset{i.i.d.}{\sim} P_X$, and the output, $\tilde{\mathbf{Y}}$, of the sequence generated by a discrete memoryless channel W. This output sequence undergoes a cyclic shift of length $\pi \in [\kappa]$, chosen uniformly at random. The objective is to estimate this delay in the absence of knowledge of the statistics of the channel. The source and output are presumed to be drawn from the discrete finite alphabet $[r]$.

As shown by the system model in Fig. 3.1, the delay estimation problem is a simplified version of the two-image registration problem. Consequently, note that $\gamma = 0$, and all permutation cycles are of equal length. For ease, let us presume that n is divisible by κ.

We again consider the MMI estimate of the delay which is given by

$$\hat{\pi}_{\mathrm{MMI}} = \arg\max_{\pi \in [\kappa]} \hat{I}(X; Y_\pi), \tag{3.1}$$

where $\hat{I}(X;Y)$ is the plug-in estimate of the mutual information. And again, the Bayes

optimal estimator is the maximum likelihood (ML) estimate:

$$\hat{\pi}_{\mathrm{ML}} = \arg\max_{\pi\in[\kappa]} \prod_{i=1}^{n} W\left(Y_{\pi(i)}|X_i\right). \tag{3.2}$$

We now evaluate and observe the gap in performance of the two decoders using the type counting argument.

3.1.2 Size of Delay Types

Just as we defined permutation types, we now consider delay types given by the empirical joint distribution of $\mathbf{x}, \mathbf{x}_\pi$. Corresponding conditional types can also be defined. We now characterize the size of the delay types. Let κ_π be the number of cycles created by the shift. Then the length of each cycle is $\frac{n}{\kappa_\pi}$. Here we limit the delays just as in the image registration problem, such that $\kappa_\pi = o\left(\frac{n}{\log n}\right)$.

Lemma 9. *For cyclic shift π, and sequence $\mathbf{x}$, we have*

$$\left|\log_2 |T^n_{X_0,X_\pi}| - n\left(H(X_0,X_\pi) - H(X)\right) - \kappa_\pi \log r\right| \leq \kappa_\pi r^2 \log_2\left(1+\frac{n}{\kappa_\pi}\right),$$

where κ_π is the number of permutation cycles created by the cyclic shift π.

Proof. Recall the bounds of Whittle's theorem from (2.12). Then, similar to Lem. 2, we derive the upper bound by studying the first-order Markov types over the cycles of the delay type. Let q be the joint permutation type of $\mathbf{X}$. Let the first-order Markov type over cycle i of the cyclic shift be q_i. Then,

$$q(a_0,a_1) = \frac{1}{\kappa_\pi}\sum_{i=1}^{\kappa_\pi} q_i(a_0,a_1). \tag{3.3}$$

Let q', q'_i be the marginals corresponding to the joint types q, q_i respectively.

Then, the size of the type class is obtained by summing over all valid decompositions in

(3.3) as

$$|T^n_{X_0,X_\pi}| = \sum \prod_{i=1}^{\kappa_\pi} |T^{n/\kappa_\pi}_{q_i}| \tag{3.4}$$

$$\leq \prod_{i=1}^{\kappa_\pi} r \left(1 + \frac{n}{\kappa_\pi}\right)^{r^2} 2^{\frac{n}{\kappa_\pi}\left(H(q_i) - H(q_i')\right)} \tag{3.5}$$

$$\leq 2^{\left[n\left(\frac{1}{\kappa_\pi}\sum_{i=1}^{\kappa_\pi}(H(q_i)-H(q_i'))\right)+\kappa_\pi\left(\log r + r^2 \log_2\left(1+\frac{n}{\kappa_\pi}\right)\right)\right]}, \tag{3.6}$$

where (3.4) follows from the fact that the joint type is composed of a sequence from each of the first-order Markov types that compose the joint type in (3.3). Then, (3.5) follows from the upper bound on the size of the first-order Markov types by Whittle's law and from the fact that the number of types is polynomial. Finally, we bound the average of the entropy in (3.6) as in Lem. 2 to obtain the upper bound.

The lower bound is obtained by the observation that the number of sequences from (3.4) is lower bounded by any one viable decomposition in (3.3). Thus, the lower bound is obtained in the same manner as in the proof of Lem. 2. □

Note that the result obtained here is essentially similar to Lem. 2, except that the lemma makes the dependence on the number of cycles more explicit. Similar results can be obtained for the conditional joint types as well, as shown below.

Lemma 10. *For any delay π, and any $\mathbf{x}, \mathbf{y}$, we have*

$$0 \leq \log_2 |T^n_{Y|X_0,X_\pi}| - n\left(H(X_0, X_\pi, Y) - H(X_0, X_\pi)\right) \leq \kappa_\pi r^3 \log_2\left(1 + \frac{n}{\kappa_\pi}\right). \tag{3.7}$$

3.1.3 Performance Analysis

The error probability of the MMI decoder can now be bounded more precisely as follows.

Theorem 11. *Let the maximum number of permutation cycles, given the set of possible delays, be κ. Then,*

$$-\frac{1}{n}\log_2 P_e(\Phi_{MMI}) \tag{3.8}$$

$$\geq \mathcal{E}^* - \frac{\kappa}{n}\left(r^2(r+1)\right)\log_2\left(1+\tfrac{n}{\kappa}\right) - r^3\frac{\log_2(1+n)}{n} - \frac{\kappa}{n}\log r - \frac{1}{n}\log\kappa, \tag{3.9}$$

where $\mathcal{E}^*$ is the error exponent.

Proof. The result for the error in the binary hypothesis test with respect to the null hypothesis and a delay of π is analogous to Thm. 2. The result can be obtained by substituting the tighter bounds derived in Lems. 9 and 10, and the observation that the number of joint types over X_0, X_π, Y are bounded by $(n+1)^{r^3}$. Finally, using union bound results in the upper bound shown here. □

Similarly we can obtain the converse by studying the performance of the maximum likelihood decoder.

Theorem 12. *Let the maximum number of permutation cycles, given the set of possible delays, be κ. Then,*

$$-\frac{1}{n}\log_2 P_e(\Phi_{ML}) \leq \mathcal{E}^* + \frac{\kappa}{n}\left(r^2\right)\log_2\left(1+\tfrac{n}{\kappa}\right) - \frac{\kappa}{n}\log r. \tag{3.10}$$

Proof. The proof is analogous to that of Thm. 11 and follows from Lems. 9 and 10. □

Corollary 2.

$$\begin{aligned} 0 &\leq \frac{1}{n}\left[\log_2 P_e(\Phi_{MMI}) - \log_2 P_e(\Phi_{ML})\right] \\ &\leq \frac{\kappa}{n}(r^2(r+2))\log_2\left(1+\frac{n}{\kappa}\right) + r^3\frac{\log_2(1+n)}{n} + \frac{1}{n}\log_2\kappa. \end{aligned} \tag{3.11}$$

Proof. The result is a direct consequence of Thms. 11 and 12. □

Remark 8. *We observe that the dominant higher-order term is $O(\kappa\log_2 n)$ reinforcing the fact that for $\kappa = o\left(\frac{n}{\log n}\right)$, the exponents match that of the maximum likelihood decoder. Further we observe that the error probability is worsened when the number of cycles generated by the delay increases.*

The result characterizes the loss in performance from the lack of knowledge of the statistics of the channel, i.e., a bound on the cost of universality. Through a stricter analysis of the type counting argument in order to characterize the effect of the number and size of permutation cycles on the performance of the MMI method in universal delay estimation. Whereas the finite-sample analysis for universal methods proves to be hard, such tighter type counting arguments provide stronger insight into the relationship between the nature of the transformations and algorithm performance.

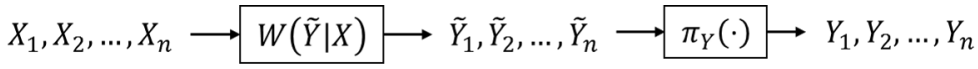


Figure 3.2: Two-image registration model: We register image $\mathbf{Y}$ to reference $\mathbf{X}$.

3.2 Channel-Aware Image Registration

To establish the fundamental performance limits for image registration in the finite-sample context we now consider the simplified context of two-image registration when the channel is given. In particular, we define and study a likelihood ratio test for image registration based on the Feinstein decoder.

3.2.1 Model

We adopt an image model similar to that of Chapter 2. We model the reference and copy images $\mathbf{X}, \mathbf{Y}$ by a collection of n pixels, and we treat the images as sequences. In particular, we presume that the reference image is drawn as $\mathbf{X} \overset{i.i.d.}{\sim} P_X$ and let $\tilde{\mathbf{Y}}$ be the output corresponding to the discrete memoryless channel W. We presume the set of pixel values is $[r] = \{1, \ldots, r\}$. The final copy image $\mathbf{Y}$ is generated as a rigid-body transformation of $\tilde{\mathbf{Y}}$. The image model is the same as in Chapter 2 and is depicted in Fig. 3.2. We consider the same set of rigid-body transformations whose properties were introduced in Sec. 2.1.

Definition 9. *The* correct registration *of an image* $\mathbf{X}$ *transformed by* $\pi \in \Pi$ *is* $\hat{\pi} = \pi^{-1}$.

We focus on the 0-1 loss function to quantify performance.

Definition 10. *The* error probability *of an algorithm* $\Phi^{(n)}$ *that outputs* $\hat{\pi}$ *is*

$$P_e(\Phi^{(n)}) = \mathbb{P}\left[\hat{\pi} \neq \pi_i^{-1}\right]. \tag{3.12}$$

We use Φ to denote $\Phi^{(n)}$ when clear from context.

3.2.2 Moments of Information Density

For this chapter we will characterize the performance in terms of the moments of information density, and we introduce the moments here for reference.

Definition 11. *Given X, Y sampled according to the joint distribution p, and the corresponding marginals p_x, p_y, the information density is defined as*

$$\imath(x;y) = \log \frac{p(x,y)}{p_x(x)p_y(y)}.$$

The information density is also alternatively called the *information spectrum*. However, we clarify that the quantity is different from the marginal information density (also called the Bayesian surprise) [100, 101] which is defined as

$$s(x) = D(p(Y|X=x)\|p(Y)).$$

That is,

$$s(x) = \mathbb{E}\left[\imath(x;Y)|X=x\right].$$

Both the information density and the Bayesian surprise have the same expected value, which is the mutual information. In fact, the moments of the information density are as follows:

1. Mutual information, $I(X;Y) = \mathbb{E}\left[\imath(X;Y)\right]$.
2. Dispersion, $V(X;Y) = \mathbb{E}\left[(\imath(X;Y) - I(X;Y))^2\right]$.
3. Third absolute moment, $T(X;Y) = \mathbb{E}\left[|\imath(X;Y) - I(X;Y)|^3\right]$.

Several properties of these moments have been studied. In particular, we note that the information density, mutual information, dispersion, and third absolute moments are continuous over the probability simplex. Further, the dispersion and the third absolute moment are bounded above as shown in [90].

3.2.3 An Achievable Method: Feinstein Decoder

The two-image registration problem has been well-studied and a variety of registration algorithms have been defined, including the maximum mutual information decoder studied in the last chapter. Since we consider the Hamming loss, the optimal algorithm is the maximum likelihood estimate. Whereas the ML decoder is Bayes optimal, to assist with the analysis, we consider the Feinstein version of a likelihood ratio test to perform the registration [102]. To define this decoder, let us presume that the possible transformations are ordered as $\Pi = \{\pi_i : i \in [M]\}$. Then the transformation is estimated as

$$\hat{\pi}_{\mathrm{F}} = \pi_{i^*}, \text{ where } i^* = \min\left\{i \in [M] : \imath(\mathbf{X}; \mathbf{Y}_{\pi_i}) \geq \delta\right\}. \tag{3.13}$$

We now derive upper bounds on the error probability of the Feinstein decoder for image registration in the channel-aware, finite-sample context. In particular we characterize the tradeoff between the sample size (image resolution) and channel properties (moments of information density) under which the Feinstein decoder achieves an error probability of at most ϵ.

Lemma 11. *For any pair of images $\mathbf{X}, \mathbf{Y}$, and transformation π,*

$$\imath(\mathbf{X}, \mathbf{Y}_\pi) = L_\pi(\mathbf{X}, \mathbf{Y}) + C(\mathbf{X}, \mathbf{Y}),$$

where $L_\pi(\cdot)$ is the log likelihood ratio given the transformation $\pi \in \Pi$, and $C(\cdot)$ is a function independent of the transformation.

Proof. From the definition of the information density and the memorylessness of the channel and source,

$$\imath(\mathbf{X}; \mathbf{Y}_\pi) = \sum_{i=1}^{n} \log \frac{p(x_i, y_{\pi(i)})}{p(x_i)p(y_{\pi(i)})} = \log L_\pi \left[\mathbf{X}, \mathbf{Y}\right] - \log \left(p(\mathbf{X})p(\mathbf{Y})\right).$$

Here $C(\mathbf{X}, \mathbf{Y}) = -\log \left(p(\mathbf{X})p(\mathbf{Y})\right)$ is independent of the permutation π owing to the memorylessness of the source and channel. □

Thus the Feinstein likelihood decoder in (3.13) is a version of the likelihood ratio test. The performance of the decoder can be analyzed as follows.

Theorem 13. *The error probability of the Feinstein decoder is bounded as*

$$P_e(\Phi_F) \leq \mathbb{P}_{\pi_0} \left[\imath(\mathbf{X}; \mathbf{Y}) \leq \delta\right] + \frac{M-1}{2} \mathbb{P}_\pi \left[\imath(\mathbf{X}; \mathbf{Y}_\pi) > \delta\right], \tag{3.14}$$

where π is the transformation with the maximum number of fixed points.

Proof. First, we bound the conditional error probability, given the true transformation is $\pi^* = \pi_j$. In this case, the decoder declares the wrong transformation if $\imath(\mathbf{X}; \mathbf{Y}_{\pi_j}) \leq \delta$ or

there exists $i < j$ such that $\imath(\mathbf{X}; \mathbf{Y}_{\pi_i}) > \delta$, i.e.,

$$\begin{aligned}
P_{e,j} &= \mathbb{P}\left[\hat{\pi}_F \neq \pi_j | \pi^* = \pi_j\right] \\
&= \mathbb{P}\left[\left\{\imath(\mathbf{X}; \mathbf{Y}_{\pi_j}) \leq \delta\right\} \cup_{i<j} \left\{\imath(\mathbf{X}; \mathbf{Y}_{\pi_i}) > \delta\right\} | \pi^* = \pi_j\right] \\
&\leq \mathbb{P}\left[\imath(\mathbf{X}; \mathbf{Y}_{\pi_j}) \leq \delta | \pi^* = \pi_j\right] + \sum_{i<j} \mathbb{P}\left[\imath(\mathbf{X}; \mathbf{Y}_{\pi_i}) > \delta | \pi^* = \pi_j\right] \qquad (3.15) \\
&= \mathbb{P}_{\pi_0}\left[\imath(\mathbf{X}; \mathbf{Y}) \leq \delta\right] + \sum_{i<j} \mathbb{P}\left[\imath(\mathbf{X}; \mathbf{Y}_{\pi_i}) > \delta | \pi^* = \pi_j\right] \qquad (3.16) \\
&\leq \mathbb{P}_{\pi_0}\left[\imath(\mathbf{X}; \mathbf{Y}) \leq \delta\right] + \sum_{i<j} \mathbb{P}_{\pi_0}\left[\imath(\mathbf{X}; \mathbf{Y}_{\pi})\right] \qquad (3.17) \\
&= \mathbb{P}_{\pi_0}\left[\imath(\mathbf{X}; \mathbf{Y}) \leq \delta\right] + (j-1)\mathbb{P}_{\pi_0}\left[\imath(\mathbf{X}; \mathbf{Y}_{\pi}) > \delta\right], \qquad (3.18)
\end{aligned}$$

where (3.15) follows from the union bound, (3.16) follows from the fact that the information density between the correctly transformed pairs conditioned on the true transformation is the same as that between the given image pairs under the null hypothesis. Finally, (3.17) is obtained by bounding the conditional probabilities by the transformation with the most fixed points with respect to the null hypothesis. This is since it has the least informative samples, and thus serves as a bound for the probability.

Finally, (3.14) follows from (3.18) as

$$P_e(\Phi_F) = \sum_{j \in [M]} \frac{1}{M} P_{e,j} \leq \mathbb{P}_{\pi_0}\left[\imath(\mathbf{X}; \mathbf{Y}) \leq \delta\right] + \frac{1}{M}\binom{M}{2}\mathbb{P}_{\pi_0}\left[\imath(\mathbf{X}; \mathbf{Y}_{\pi}) > \delta\right].$$

□

Next, we bound the two probabilities in (3.14) in the following lemmas.

Lemma 12. *Given n i.i.d. pairs $(\mathbf{X}, \mathbf{Y}) \overset{i.i.d.}{\sim} p_{X,Y}$,*

$$\mathbb{P}\left[\imath(\mathbf{X}; \mathbf{Y}) \leq \delta\right] \leq Q(\tau) + \frac{B}{\sqrt{n}}, \qquad (3.19)$$

where

$$\tau = \frac{nI(X;Y) - \delta}{\sqrt{nV(X;Y)}}, \text{ and } B = \frac{6T(X;Y)}{V(X;Y)^{3/2}}.$$

Proof. Let $Z_i = \imath(X_i; Y_i)$. Then, $\imath(\mathbf{X}; \mathbf{Y}) = \sum_{i=1}^n Z_i$, and for any $i \in [n]$,

$$\begin{aligned}
\mu_Z &= \mathbb{E}\left[Z_i\right] = I(X;Y), \\
V_Z &= \mathrm{var}(Z_i) = \mathbb{E}\left[(Z_i - \mu_Z)^2\right] = V(X;Y), \\
T_Z &= \mathbb{E}\left[|Z_i - \mu_Z|^3\right] = T(X;Y).
\end{aligned}$$

Then, from the Berry-Esseen theorem, we have

$$\mathbb{P}\left[\imath(\mathbf{X};\mathbf{Y}) \le \delta\right] = \mathbb{P}\left[\sum_{i=1}^{n} Z_i \le \delta\right] \le Q\left(\frac{n\mu_Z - \delta}{\sqrt{nV_Z}}\right) + \frac{6T_Z}{\sqrt{n}(V_Z)^{3/2}},$$

and the result is obtained by substituting the values of the computed moments. □

On the other hand, Sanov's theorem results in the following upper bound on the tail probability of the information density.

Lemma 13. *Given, n i.i.d. samples $(X_i, Y_i) \overset{i.i.d.}{\sim} p$, then for any constant $\lambda > 0$,*

$$\mathbb{P}\left[\imath(\mathbf{X};\mathbf{Y}) \ge n\delta\right] \le (n+1)^{|\mathcal{X}||\mathcal{Y}|} \exp\left(-n\left[\lambda I(X;Y) - \log \mathbb{E}\left[\exp\left(\lambda \imath(X;Y)\right)\right]\right]\right). \tag{3.20}$$

In particular, the tightest upper bound is obtained by using the constant

$$\lambda^* = \arg\max_{\lambda > 0} \lambda I(X;Y) - \log \mathbb{E}\left[\exp\left(\lambda \imath(X;Y)\right)\right].$$

Proof. The proof uses Sanov's theorem. For simplicity, let $Z_i = \imath(X_i;Y_i)$. Then, from Sanov's theorem, we have

$$\mathbb{P}\left[\imath(\mathbf{X};\mathbf{Y}) \ge n\delta\right] \le (n+1)^{|\mathcal{X}||\mathcal{Y}|} \exp\left(-nD(q^*\|p)\right),$$

where

$$q^* = \arg \max_{q:\mathbb{E}_q[\imath(X;Y)] \ge \delta} D(q\|p).$$

Using Lagrange multiplier $\lambda > 0$, consider the Lagrangian

$$\begin{aligned}
\mathcal{L}(q) &= D(q\|p) - \lambda \mathbb{E}_q\left[\imath(X;Y)\right] \\
&= \sum_{(x,y)\in\mathcal{X}\times\mathcal{Y}} q(x,y)\left[\log\frac{q(x,y)}{p(x,y)} - \lambda \log\frac{p(x,y)}{p(x)p(y)}\right].
\end{aligned}$$

To maximize the Lagrangian, we set the partial derivatives to 0, and have

$$\begin{aligned}
&\frac{\partial}{\partial q(x,y)} = 0 \Leftrightarrow \log\frac{q(x,y)}{p(x,y)} - \lambda\frac{p(x,y)}{p(x)p(y)} + 1 = 0 \\
&\Leftrightarrow q(x,y) = \frac{1}{Z}p(x,y)\left[\frac{p(x,y)}{p(x)p(y)}\right]^{\lambda},
\end{aligned}$$

where

$$Z = \mathbb{E}\left[\left(\frac{p(x,y)}{p(x)p(y)}\right)^{\lambda}\right],$$

is the normalization constant. Since the optimization is over a convex objective with linear inequality constraints, from KKT conditions, it is evident that

$$\delta = \mathbb{E}_{q^*}\left[\imath(X;Y)\right] = \frac{1}{Z}\mathbb{E}_p\left[\left(\frac{p(X,Y)}{p(X)p(Y)}\right)^{\lambda}\log\left(\frac{p(X,Y)}{p(X)p(Y)}\right)\right].$$

Thus, we can compute the maximum KL divergence as

$$\begin{aligned} D(q^*\|p) &= \mathbb{E}_{q^*}\left[\log\frac{q^*(x,y)}{p(x,y)}\right] \\ &= \frac{1}{Z}\mathbb{E}_p\left[\left(\frac{p(X,Y)}{p(X)p(Y)}\right)^{\lambda}\log\left(\frac{1}{Z}\left(\frac{p(X,Y)}{p(X)p(Y)}\right)^{\lambda}\right)\right] \\ &= \lambda\delta - \log Z \\ &= \lambda\delta - \log\mathbb{E}\left[\exp\left(\lambda\imath(X;Y)\right)\right]. \end{aligned}$$

Now, let $\tilde{X} = \exp\left(\lambda\imath(X;Y)\right)$. Then, we have

$$\lambda\delta = \frac{\mathbb{E}\left[\tilde{X}\log\tilde{X}\right]}{\mathbb{E}\left[\tilde{X}\right]}.$$

Since the function $f(x) = x\log x$ is convex, using Jensen's inequality, we have

$$\lambda\delta \geq \frac{\mathbb{E}\left[\tilde{X}\right]\log\mathbb{E}\left[\tilde{X}\right]}{\mathbb{E}\left[\tilde{X}\right]} \geq \log\left(\exp\left(\lambda\mathbb{E}\left[\imath(X;Y)\right]\right)\right) = \lambda I(X;Y).$$

Thus, for any $\lambda > 0$, $\delta \geq I(X;Y)$ and so the result follows. □

Finally, let us consider the information density generated by pairs of pixels in a derangement by using the Berry-Esseen theorem.

Lemma 14. *Consider n i.i.d. pairs $(X_i, Y_i) \overset{i.i.d.}{\sim} p$ and let π be a derangement of $[n]$. That is, for all $i \in [n]$, $\pi(i) \neq i$. Then, for $\imath(\mathbf{X}, \mathbf{Y}_\pi) = \sum_{i=1}^n \imath(X_i; Y_{\pi(i)})$, the tail probability is bounded as*

$$\mathbb{P}\left[\imath(\mathbf{X}, \mathbf{Y}_\pi) \geq \delta\right] \leq 6\sqrt{3}\left(\frac{\log 2}{\sqrt{2\pi}} + 2B(X;Y)\right)\frac{1}{\sqrt{nV(X;Y)}}\exp\left(-\frac{\delta}{3}\right). \tag{3.21}$$

Proof. Note that for any i, $(X_i, Y_{\pi(i)}) \sim p_X p_Y$. However, note that the samples themselves are dependent as we are considering permutations of the sequence. So we first split the samples into sets of independent pairs. First construct a graph based on the permutation π on the set of vertices $V = [n]$ with edges $(i, \pi(i))$, for all $i \in [n]$. Since the permutation is a derangement, the resulting graph is composed of a set of disjoint cycles, each of length at least two. Thus the vertices are 3-colorable. By uniformly distributing the three colors among the nodes, divide the set as $[n] = \mathcal{V}_1 \cup \mathcal{V}_2 \cup \mathcal{V}_3$, according to the colors of the corresponding nodes in the graph. It is easy to see that there exists a coloring such that $|\mathcal{V}_i| \geq \left\lfloor \frac{n}{3} \right\rfloor$ for all $i \in [3]$. For simplicity, we assume that n is a multiple of 3 and that $|\mathcal{V}_i| = n/3$. The results generalize trivially.

Since $\mathcal{V}_i$ includes nodes of the same color, for any $j, k \in \mathcal{V}_i$, it is evident that $\pi(j), \pi(k) \notin \mathcal{V}_i$. Consequently, $(X_j, Y_{\pi(j)})$ and $(X_k, Y_{\pi(k)})$ are independent. More generally, the pairs corresponding to the indices in any $\mathcal{V}_i$ are mutually independent and $(X_j, Y_{\pi(j)}) \overset{i.i.d.}{\sim} p_X p_Y$, for any $j \in \mathcal{V}_i$.

We first note that

$$
\begin{aligned}
&\mathbb{P}\left[\imath(\mathbf{X}, \mathbf{Y}_\pi) \geq \delta\right] && (3.22)\\
&= p_{X,Y}^{\otimes n}\left[\sum_{i\in[n]} \imath(X_i, Y_{\pi(i)}) \geq \delta\right] \\
&\leq \sum_{i\in[3]} p_{X,Y}^{\otimes n}\left[\sum_{j\in\mathcal{V}_i} \imath(X_j, Y_{\pi(j)}) \geq \frac{\delta}{3}\right] && (3.23)\\
&= 3 p_X p_Y^{\otimes \frac{n}{3}}\left[\sum_{i\in[n/3]} \imath(X_i; Y_i) \geq \frac{\delta}{3}\right] && (3.24)\\
&= 3\mathbb{E}_{p_{X,Y}^{\otimes \frac{n}{3}}}\left[\frac{p_X(\mathbf{X}) p_Y(\mathbf{Y})}{p_{X,Y}(\mathbf{X}, \mathbf{Y})} \mathbf{1}\left\{\sum_{i\in[n/3]} \imath(X_i; Y_i) \geq \frac{\delta}{3}\right\}\right] && (3.25)\\
&= 3\mathbb{E}_{p_{X,Y}^{\otimes \frac{n}{3}}}\left[\exp\left(-\sum_{i\in[n/3]} \imath(X_i; Y_i)\right) \mathbf{1}\left\{\sum_{i\in[n/3]} \imath(X_i; Y_i) \geq \frac{\delta}{3}\right\}\right], && (3.26)
\end{aligned}
$$

where (3.23) follows from the union bound, (3.24) follows from the fact that the probabilities are the same across the three color sets and that samples from each set are sampled independently according to the product distribution. Then, we change the distribution over which the expectation is computed to the joint distribution by appropriately scaling the indicator random variable, and finally, (3.26) follows from the definition of the information

density. Thus we have upper bounded the tail probability of the information density of the derangement by an indicator-weighted moment of the information density of samples drawn according to the joint distribution $p_{X,Y}$.

From [90, Lemma 47], given n independent random variables $Z_1, \ldots, Z_n$, and if $V_Z = \sum_{j\in[n]} \mathrm{Var}(Z_j) \neq 0$, and $T_Z = \sum_{j\in[n]} \mathbb{E}\left[|Z_j - \mathbb{E}\left[Z_j\right]|^3\right] < \infty$, then for any δ,

$$\mathbb{E}\left[\exp\left(-\sum_{j\in[n]} Z_j\right)\mathbf{1}\left\{\sum_{j\in[n]} Z_j > \delta\right\}\right] \leq 2\left(\frac{\log 2}{\sqrt{2\pi}} + \frac{12T_Z}{V_Z}\right)\frac{1}{\sqrt{V_Z}}\exp\left(-\delta\right).$$

Thus, if $Z_i = \imath(X_i;Y_i)$, the term in the upper bound in (3.26) is bounded as

$$\begin{aligned}\mathbb{E}_{p_{X,Y}^{\otimes\frac{n}{3}}}&\left[\exp\left(-\sum_{i\in[\frac{n}{3}]}\imath(X_i;Y_i)\right)\mathbf{1}\left\{\sum_{i\in[\frac{n}{3}]}\imath(X_i;Y_i) \geq \frac{\delta}{3}\right\}\right]\\ &\leq \left(\frac{\log 2}{\sqrt{2\pi}} + 2B(X;Y)\right)\frac{2\sqrt{3}}{\sqrt{nV(X;Y)}}\exp\left(-\frac{\delta}{3}\right),\end{aligned}$$

which proves the lemma. □

Remark 9. *Note that the constant in Lem. 14 can be reduced to $4\sqrt{2}$ from $6\sqrt{3}$ if we knew the permutation cycles generated by π were all of even length. This is because even length cycles can be vertex colored using two colors, resulting in two sets of size $n/2$.*

Theorem 14. *Let $M = 2cn^\alpha + 1$ and $\delta \geq 3\alpha\log n + \gamma_n nI(X;Y) + 3\log c$. Then, the probability of error achieved by the Feinstein decoder is upper bounded as*

$$P_e(\Phi_F) \leq Q\left(\frac{nI-\delta}{\sqrt{nV}}\right) + \frac{B}{\sqrt{n}} + \frac{6\sqrt{3}}{\sqrt{(1-\gamma_n)n}}\left(\frac{\log 2}{\sqrt{2\pi}} + 2B\right) \tag{3.27}$$

$$+ \frac{M-1}{2}(\gamma_n n + 1)^{|\mathcal{X}||\mathcal{Y}|}\exp\left(-\gamma_n nD^*\right), \tag{3.28}$$

where

$$\begin{aligned} I &= I(X;Y) = \mathbb{E}\left[\imath(X;Y)\right],\\ V &= V(X;Y) = Var\left(\imath(X;Y)\right),\\ T &= T(X;Y) = \mathbb{E}\left[|\imath(X;Y) - I|^3\right],\\ B &= B(X;Y) = \frac{6T}{V},\\ D^* &= \lambda^* I(X;Y) - \log\mathbb{E}\left[\exp\left(\lambda^*\imath(X;Y)\right)\right].\end{aligned}$$

Proof. From Thm. 13, we know that

$$P_e(\Phi_F) \leq \mathbb{P}_{\pi_0}\left[\imath(\mathbf{X};\mathbf{Y}) \leq \delta\right] + \frac{M-1}{2}\mathbb{P}_{\pi}\left[\imath(\mathbf{X};\mathbf{Y}_{\pi}) > \delta\right].$$

From Lem. 12, we have

$$\mathbb{P}_{\pi_0}\left[\imath(\mathbf{X};\mathbf{Y}) \leq \delta\right] \leq Q\left(\frac{nI-\delta}{\sqrt{nV}}\right) + \frac{B}{\sqrt{n}}.$$

Next, from the union bound, we have

$$\begin{aligned}
\mathbb{P}_{\pi}\left[\imath(\mathbf{X};\mathbf{Y}_{\pi}) > \delta\right] &= \mathbb{P}_{\pi}\left[\sum_{j\in[n]} \imath(X_j;Y_{\pi(j)}) > \delta\right] \\
&= \mathbb{P}_{\pi}\left[\sum_{j\in\mathcal{I}_{\pi}} \imath(X_j;Y_j) + \sum_{j\in\mathcal{I}_{\pi}^c} \imath(X_j;Y_{\pi(j)}) > \delta\right] \\
&\leq \mathbb{P}_{\pi}\left[\sum_{j\in\mathcal{I}_{\pi}} \imath(X_j;Y_j) \geq \delta_1\right] + \mathbb{P}_{\pi}\left[\sum_{j\in\mathcal{I}_{\pi}^c} \imath(X_j;Y_j) > \delta_2\right], \qquad (3.29)
\end{aligned}$$

where $\delta_1 + \delta_2 = \delta$, and $\delta_1 \geq \gamma_n nI(X;Y)$, $\delta_2 \geq 3\alpha\log n + 3\log c$.

From Lem. 13, for $\delta_1 \geq \gamma_n nI(X;Y)$,

$$\mathbb{P}_{\pi}\left[\sum_{j\in\mathcal{I}_{\pi}} \imath(X_j;Y_j) \geq \delta_1\right] \leq (\gamma_n n+1)^{|\mathcal{X}||\mathcal{Y}|}\exp\left(-\gamma_n nD^*\right).$$

Next, from Lem. 14, for $\delta_2 \geq 3\alpha\log n + 3\log c$,

$$\mathbb{P}_{\pi}\left[\sum_{j\in\mathcal{I}_{\pi}^c} \imath(X_j;Y_j) > \delta_2\right] \leq \frac{2}{M-1}\frac{6\sqrt{3}}{\sqrt{(1-\gamma_n)n}}\left(\frac{\log 2}{\sqrt{2\pi}} + 2B\right).$$

Substituting the probabilities, the result follows. □

Corollary 3. *If* $\frac{\log(1+\gamma_n n)}{\gamma_n n} \leq \frac{D^*}{2|\mathcal{X}||\mathcal{Y}|}$, *and*

$$(1-\gamma_n)nI(X;Y) \geq \sqrt{nV(X;Y)}Q^{-1}(\epsilon) + 3\alpha\log n + \Delta, \qquad (3.30)$$

where Δ *is a constant independent of the sample size* n*, but is dependent on the dispersion of the channel* $V(X;Y)$*, then, there exists a threshold* δ *such that the Feinstein decoder achieves an average error probability less than* ϵ*.*

Proof. First, from Thm. 14 we know that if $\delta_1 \geq \gamma_n n I(X;Y)$, $\delta_2 \geq 3\alpha \log n + 3 \log c$, then the probability of error of the Feinstein decoder is upper bounded as in (3.27). Thus,

$$\delta \geq \gamma_n n I(X;Y) + 3\alpha \log n + 3 \log c. \tag{3.31}$$

Next, we note that if

$$\gamma_n n \left(D^* - O\left(\tfrac{\log(\gamma_n n)}{\gamma_n n} \right) \right) \geq \left(\alpha + \tfrac{1}{2} \right) \log n, \tag{3.32}$$

then

$$\frac{M-1}{2} (\gamma_n n + 1)^{|\mathcal{X}||\mathcal{Y}|} \exp\left(-\gamma_n n D^*\right) \leq \frac{c}{\sqrt{n}}.$$

Since an achievability criterion for a larger γ_n is also one for a smaller worst-case number of fixed points, it suffices to consider the case where $\gamma_n n \geq \frac{2\left(\alpha+\frac{1}{2}\right)}{D^*} \log n$. Since $\frac{\log(1+\gamma_n n)}{\gamma_n n} \leq \frac{D^*}{2|\mathcal{X}||\mathcal{Y}|}$, (3.32) is satisfied. Thus,

$$P_e(\Phi_F) \leq Q\left(\frac{nI - \delta}{\sqrt{nV}} \right) + \frac{B}{\sqrt{n}} + \frac{6\sqrt{3}}{\sqrt{(1-\gamma_n)n}} \left(\frac{\log 2}{\sqrt{2\pi}} + 2B \right) + \frac{c}{\sqrt{n}}.$$

Thus, $P_e(\Phi_F) \leq \epsilon$, if

$$\begin{aligned} \delta &\leq nI - \sqrt{nV} Q^{-1}\left(\epsilon - \left[\frac{B}{\sqrt{n}} + \frac{6\sqrt{3}}{\sqrt{(1-\gamma_n)n}} \left(\frac{\log 2}{\sqrt{2\pi}} + 2B \right) + \frac{c}{\sqrt{n}} \right] \right) \\ &= nI(X;Y) - \sqrt{nV(X;Y)} Q^{-1}(\epsilon) + \Delta, \end{aligned} \tag{3.33}$$

where

$$\eta\left(\epsilon - \left[\frac{B}{\sqrt{n}} + \frac{6\sqrt{3}}{\sqrt{(1-\gamma_n)n}} \left(\frac{\log 2}{\sqrt{2\pi}} + 2B \right) + \frac{c}{\sqrt{n}} \right] \right) \leq \frac{\Delta}{B\sqrt{V} + \frac{6\sqrt{3}}{\sqrt{1-\gamma_n}} \left(\frac{\log 2}{\sqrt{2\pi}} + 2B \right) + c} \leq \eta(\epsilon),$$

where $\eta(\cdot)$ is the derivative of the inverse Q-function. Here (3.33) follows from the differentiability of the Q^{-1} function.

Thus, from (3.31) and (3.33), it is evident that an optimal threshold δ can be chosen provided (3.30) holds. □

Corollary 3 characterizes the tradeoff between the channel properties, in terms of the moments of information density, and the sample size. For instance, let us consider the

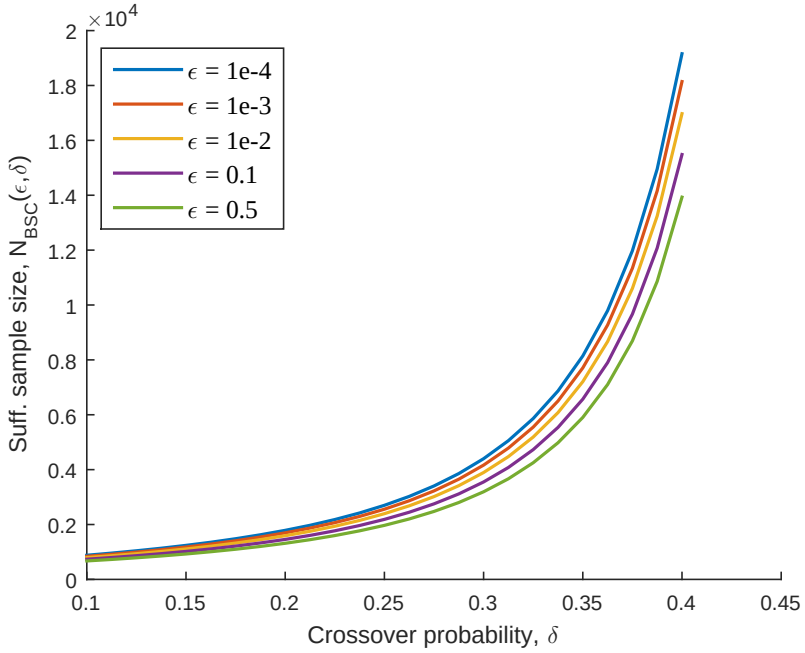


Figure 3.3: Sufficient sample size $N(\epsilon, \delta)$ for BSC(δ) and error probability ϵ. The sufficient sample size increases with a decrease in target error probability, and with increasing crossover probability.

simple case of a binary symmetric source $\mathbf{X} \overset{i.i.d.}{\sim}$ Bern(0.5) and binary symmetric channel with crossover probability δ. For simplicity let us presume that $\gamma_n = n^{-1/2}$ and let $M = n^5$.

The minimum sample size, $N_{BSC}(\epsilon, \delta)$, that satisfies (3.30) as a function of the error probability ϵ and the crossover probability δ is shown in Fig. 3.3. As expected, with decreasing target error probability, the sufficient sample size increases. Similarly as the channel gets noisier, that is, as the crossover probability increases, the sufficient sample size increases.

3.3 Discussion

In this chapter we studied the image registration problem with a focus on tighter performance analyses from two angles. First, we considered the simplified problem of universal delay estimation using the max mutual information decoder to understand the effect of the number of permutation cycles on the error probability. Having observed the asymptotic optimality for finite delays in [60], this analysis provides an insight into delays that can scale with n and therein characterizing the sample complexity of universal delay estimation better. Considering the similarity of the image registration problem to the delay estimation problem, such studies inform us better regarding the problem difficulty given a set of transformations.

We then considered the problem of channel-aware image registration using likelihood ratio

tests to study their finite-sample performance. Using the Feinstein decoder we obtained achievable sample sizes given the moments of the channel information density and the target error probability ϵ. This study helps us establish some preliminary results on the fundamental achievable finite-sample performance of image registration. Future work can build on strong large deviations and central limit theorems to obtain converse criteria for image registration.

Improved analyses through strong large deviations informed by a better characterization of the set of transformations help obtain tight sample complexity bounds. On the other hand, we believe the universal image registration problem can be better understood by obtaining strong finite-sample performance characterization of plug-in estimates of information functionals. To this end, studying the convergence of functions of types to the function of the true distribution would prove handy [103].

CHAPTER 4

MULTIVARIATE INFORMATION FUNCTIONALS FOR CLUSTERING

Chapters 2 and 3 considered the unsupervised information processing task of image registration in considerable detail with an emphasis on information-based methods for aligning images. In this chapter we explore another problem in unsupervised information processing—clustering. Unsupervised clustering forms a significant component of machine learning, data analytics, and information processing at large. At its core, unsupervised clustering seeks to work with minimal or no contextual information, much in the same spirit of how humans learn without a teacher. For this reason, design of good unsupervised clustering algorithms is also crucial to our search for artificial general intelligence.

In this chapter we specifically consider the problem of clustering using information-based algorithms, particularly from the standpoint of joint image clustering and registration. We then highlight the role that multivariate information functionals can play with clustering in particular, and dependence structure recovery in general by studying their functional and operational properties, with an emphasis on illum information, multivariate information functional defined later in this chapter.

Unsupervised clustering has been studied under numerous optimality and similarity criteria [104, 105]. Popular techniques for unsupervised image clustering include affinity propagation [106], expectation maximization [107], independent component analysis [108], and orthogonal subspace projection [109]. The focus here is on information-based clustering algorithms [71,110–112], as the information functionals are ubiquitous in universal information processing.

We now consider the task of clustering a set of n objects $\{X_1, \ldots, X_n\}$. The term object here is defined broadly to encompass a variety of data types such as images, object labels, and more generally random variables. Let $\mathcal{P}$ be the set of all partitions of the index set $[n]$. Given a set of constraints to be followed by the clusters, as defined by the problem, let $\mathcal{P}_c \subseteq \mathcal{P}$ be the subset of viable partitions satisfying the system constraints. We presume a framework wherein each object has a true label associated with it.

Definition 12 (Clustering)**.** *A* clustering *of a set of n objects is a partition $P \in \mathcal{P}_c$ of the set $[n]$ that satisfies the set of all constraints. A* cluster *is a set in the partition and objects*

X_i and X_j are in the same cluster when $i, j \in C$ for some cluster $C \in P$.

A clustering is said to be *correct* if all objects with the same label are clustered together, when such a unique solution, P^*, exists. Otherwise, we might be interested in a hierarchical clustering of the objects using relative similarity structures. The performance of a clustering algorithm is characterized by distance metrics such as the symbol or blockwise error probabilities.

Here we focus on universal clustering algorithms that are defined as follows.

Definition 13 (Universal Clustering)**.** *A clustering algorithm Φ is* universal *if it performs clustering in the absence of knowledge of the source and/or channel distributions and parameters defining the data samples.*

The specific source or channel models that define the data samples are application-specific.

The space of partitions includes a natural ordering to compare the partitions, and is defined as follows.

Definition 14 (Partition Ordering)**.** *For $P, P' \in \mathcal{P}$, P is* finer *than P', if the following ordering holds*

$$P \preceq P' \Leftrightarrow \text{ for all } C \in P, \text{ there exists } C' \in P' : C \subseteq C'.$$

Similarly, P is denser *than P', $P \succeq P' \Leftrightarrow P' \preceq P$. If $P' \not\preceq P \not\preceq P'$, then the partitions are* comparable*, $P \sim P'$.*

We aim to identify the finest clustering maximizing the similarity criteria. Alternatively, in the context of designing hierarchical clustering structures (also known as taxonomies), algorithms identify sequentially finer partitions to construct the phylogenetic tree.

We now study the problem of joint image clustering and registration using mutivariate information functionals.

4.1 Joint Image Clustering and Registration

Consider the problem of joint clustering and registration of multiple images. That is, grouping images according to the underlying scene and aligning images within each cluster. Having identified asymptotically optimal registration algorithms using multiinformation, we now design universal algorithms to also perform clustering.

We extend the image and transformation models defined in Sec. 2.1. Consider a finite collection of ℓ distinct scenes (drawn i.i.d. according to the prior) $\mathcal{R} = \left\{\mathbf{R}^{(1)}, \ldots, \mathbf{R}^{(\ell)}\right\}$.

This is the collection of different scenes and each image is a noisy depiction of an underlying scene.

Now, given the correct clustering $P^* = \{C_1^*, \ldots, C_K^*\}$, let $j_1, \ldots, j_K$ be the labels of the scenes corresponding to the clusters. That is, the scene corresponding to cluster C_k^* is $\mathbf{R}^{(j_k)}$. The images are outputs of a discrete memoryless channel (DMC) whose inputs are the underlying scenes:

$$\mathbb{P}\left[\tilde{\mathbf{X}}^{[m]} \middle| \mathcal{R}, P^*\right] = \prod_{k=1}^{K} \prod_{i=1}^{n} W\left(\tilde{X}_i^{C_k^*} \middle| R_i^{j_k}\right), \tag{4.1}$$

where for any set S, $X^S = \{X_i : i \in S\}$. That is, images of the same scene are jointly corrupted by a DMC, while the images corresponding to different scenes are independent of each other. Here we assume $\tilde{\mathbf{X}} \in [r]^n$.

Finally, image i is transformed by the rigid body transformation π_i, for each $i \in [m]$ resulting in the image $\mathbf{X}_i$. Thus, we know that images corresponding to different scenes are independent, and we use this as the notion of similarity to perform clustering.

In this case it is worth noting that ML estimates are not Bayes optimal unless the partitions are uniformly likely. Thus, the Bayes optimal test is the likelihood ratio test, i.e., maximum a posteriori (MAP) estimate Φ_{MAP} given by

$$\left(\hat{P}_{\text{MAP}}, \hat{\pi}_{\text{MAP}}\right) = \arg\max_{P, \bar{\pi}} \mathbb{P}\left[P^* = P\right] \prod_{C \in P} \prod_{i=1}^{n} \mathbb{P}\left[\mathbf{X}_{\pi^{(C)}(i)}^{(C)}\right], \tag{4.2}$$

where the probability is computed by averaging over scene configurations and the corresponding channel model. Both ML and MAP require knowledge of the channel and the prior, and are also hard to compute. Hence we design computationally efficient, exponentially consistent algorithms.

4.1.1 Multivariate Information Functionals for Clustering

Clustering random variables using information functionals has been well studied [111, 112]. Here we adopt an approach similar to the minimum partition information (MPI) framework.

For the problem at hand, let $X^{(i)}$ be the random variable representing a pixel of image i. Let us define a multivariate information functional called *cluster information* to quantify intra-cluster information.

Definition 15. *The* cluster information *(CI) of* $Z^{[n]}$ *for partition* $P = \{C_1, \ldots, C_k\}$ *of* $[n]$

is

$$I_C^{(P)}(Z_1; \ldots; Z_n) = \sum_{C \in P} I_M(Z^C). \tag{4.3}$$

Lemma 15. *If the images are self-aligned, i.e., $\bar{\pi}^* = \bar{\pi}_0$, then the correct clustering is the finest partition that maximizes the cluster information.*

Proof. Since images in different clusters are independent,

$$P^* \in \arg\max_{P \in \mathcal{P}} I_C^{(P)}(X^{[m]}),$$

from the chain rule and non-negativity of multiinformation.

For any $P \not\succeq P^*$, there exist clusters $C_1, C_2 \in P$ such that there exist images in each cluster corresponding to the same scene. This indicates that $I(X^{C_1}; X^{C_2}) > 0$, i.e.,

$$I_C^{(P^*)}(X^{[m]}) > I_C^{(P)}(X^{[m]}).$$

This indicates that the set of all partitions that minimize the partition information is the set $\{P : P \succeq P^*\}$. Hence the finest such partition is the correct clustering. □

We refer to the finest partition maximizing the cluster information as the *fundamental partition*.

Lemma 16. *Let $\bar{\pi} = (\pi_1, \ldots, \pi_m)$ be the estimated transformations and let $\bar{\pi}^*$ be the correct registration. Then, if $\bar{\pi} \neq \bar{\pi}^*$, then the fundamental partition $\hat{P}$ of $\{X_{\pi_1}^{(1)}, \ldots, X_{\pi_m}^{(m)}\}$ satisfies $\hat{P} \prec P^*$, where P^* is the correct clustering.*

Proof. We first note that images that correspond to different scenes are independent of each other, irrespective of the transformation. Second, an image that is incorrectly registered appears less dependent of any other image corresponding to the same scene. This in turn yields the result. □

These properties provide a natural estimator for joint clustering and registration, provided the information values can be computed accurately.

Corollary 4. *Let $\hat{P}_{\bar{\pi}}$ be the fundamental partition corresponding to the estimated transformation vector $\bar{\pi}$. Then $(P^*, \bar{\pi}^*)$ is the densest partition in $\{\hat{P}_{\bar{\pi}} : \bar{\pi} \in \Pi^m\}$ and the corresponding transformation vector.*

Proof. This follows directly from Lems. 15 and 16. □

We now consider the computational complexity of identifying the partition that maximizes the cluster information.

Lemma 17. *The clustering information of a set of random variables $\{Z_1, \ldots, Z_n\}$ is supermodular.*

Proof. The cluster information may be decomposed as

$$I_C^{(P)}(Z_1; \ldots; Z_n) = \sum_{i=1}^{n} H(Z_i) - \sum_{C \in P} H(Z^C).$$

Since entropy is submodular, the result follows. □

Supermodular function maximization can be done efficiently in polynomial time using the Dilworth truncation lattice [62, 112]. From Lem. 17 we see that given the distribution, we can obtain the fundamental partition efficiently.

Lemma 18. *The plug-in estimates of cluster information are exponentially consistent.*

Proof. The result for the cluster information estimate follows from (4.3), the union bound, and the exponential consistency of plug-in estimates of multiinformation as

$$\mathbb{P}\left[\left|\hat{I}_C^{(P)}\left(X^{[m]}\right) - I_C^{(P)}\left(X^{[m]}\right)\right| > \epsilon\right] \leq C \exp\left(-n\delta_\epsilon\right),$$

where $C = 2m$, $\delta_\epsilon = \frac{\epsilon^4}{32m^4|\mathcal{X}|^{2m}\log 2} + o(1) = \theta\epsilon^4 + o(1)$. □

We now use these observations to define clustering algorithms using plug-in estimates, under various criteria.

4.1.2 Clustering Criteria

Designing any unsupervised clustering algorithm requires a similarity criterion. Here, we know that the similarity criterion is dependence of pixel values among images of the same cluster. Thus we consider the following criteria.

(B1) *ϵ-likeness:* A given source and channel model for images, $\mathbf{X}^{[m]}$, is said to satisfy *ϵ-likeness* criterion if

$$\min_{P^* \in \mathcal{P}} \min_{P \not\preceq P^*, P \not\succeq P^*} I_C^{P^*}(X^{[m]}) - I_C^{P}(X^{[m]}) \geq \epsilon.$$

(B2) *Given number of clusters:* Given the number of clusters k in the set of images, we can define an exponentially consistent universal clustering algorithm.

Algorithm 2 ϵ-like Clustering, $\Phi_{\mathrm{C}}\left(\mathbf{X}^{(1)},\ldots,\mathbf{X}^{(m)},\epsilon\right)$

for all $\bar{\pi}\in\Pi^m$ **do**
Compute empirical pmf $\hat{p}\left(X_{\pi_1}^{(1)},\ldots,X_{\pi_m}^{(m)}\right)$
$\tilde{I}=\max_{P\in\mathcal{P}}\hat{I}_C^{(P)}(X_{\pi_1}^{(1)},\ldots,X_{\pi_m}^{(m)})$
P_π = Finest $\left\{P:\hat{I}_C^{(P)}(X_{\pi_1}^{(1)},\ldots,X_{\pi_m}^{(m)})\geq\tilde{I}-\frac{\epsilon}{2}\right\}$, where Finest $\{\cdot\}$ refers to the finest partition in the set.
end for
$(\hat{P},\hat{\pi})=\left\{(P,\bar{\pi}):P=\hat{P}_{\bar{\pi}}\succeq\hat{P}_{\bar{\pi}'},\ \text{for all }\bar{\pi}'\neq\bar{\pi}\right\}$

(B3) *Non-exponentially consistent:* Any two images $\mathbf{X},\mathbf{Y}$ that belong to different clusters are independent, i.e., $I(X;Y)=0$. Hence, using a threshold γ_n, decreasing with n, we can define a consistent clustering algorithm which however lacks exponential consistency, cf. [74].

(B4) *Hierarchical clustering:* If it suffices to create the tree of similarity among images through hierarchical clustering, then we can define a consistent algorithm that determines such topological relation among images.

Criterion (B1) restricts the priors and channels, and may be interpreted as a capacity. On the other hand (B2) restricts the space of partitions much like k-means clustering. Criterion (B3) focuses on the design of fully universal clustering algorithms albeit with sub-exponential consistency. Finally, criterion (B4) aims to develop a topology of independence-based similarity relations among images. We address the clustering problem for each of these criteria.

4.1.3 ϵ-Like Clustering

We define ϵ-like clustering, Alg. 2, using the fact that CI is maximized by the correct clustering.

Lemma 19. *Let the source and channel be ϵ-like. Then, Φ_C is exponentially consistent.*

Proof. First, let us assume that the estimated transformation is correct, $\hat{\pi}=\pi^*$. Let $\tilde{I}=\max_{P\in\mathcal{P}}\hat{I}_C^{(P)}(X_{\pi_1}^{(1)},\ldots,X_{\pi_m}^{(m)})$, $\tilde{P}$ the maximizing partition. Then, there are constants c,δ_ϵ such that

$$\begin{aligned}\mathbb{P}\left[\hat{I}_C^{(P^*)}(X_{\pi_1}^{(1)},\ldots,X_{\pi_1}^{(m)})\leq\tilde{I}-\tfrac{\epsilon}{2}\right]&\leq\mathbb{P}\left[|\tilde{I}-\hat{I}_C^{(P^*)}(X_{\pi_1}^{(1)},\ldots,X_{\pi_m}^{(m)})|\geq\tfrac{\epsilon}{2}\right]\\&\leq\mathbb{P}\left[|\hat{I}_C^{(\tilde{P})}-I_C^{(\tilde{P})}|+|\hat{I}_C^{(P^*)}-I_C^{(P^*)}|\geq\tfrac{\epsilon}{2}\right]\qquad(4.4)\\&\leq 2c\exp\left(-n\delta_{\epsilon/4}\right),\qquad(4.5)\end{aligned}$$

where (4.4) follows from the triangle inequality and the fact that the multiinformation is maximized by the correct clustering, and (4.5) follows from the union bound and Lem. 18. Here in (4.4) the information measures are computed for the random variables $X_{\bar{\pi}}^{[[m]} \triangleq (X_{\pi_1}^{(1)}, \ldots, X_{\pi_m}^{(m)})$.

Further, for any $P \in \mathcal{P}$ such that $P \not\geq P^*$ and $P \not\leq P^*$

$$\mathbb{P}\left[\hat{I}_C^{(P)}(X_{\pi_1}^{(1)}, \ldots, X_{\pi_1}^{(m)}) \geq \tilde{I} - \tfrac{\epsilon}{2}\right] \tag{4.6}$$

$$\leq \mathbb{P}\left[\hat{I}_C^{(P^*)}(X_{\pi_1}^{(1)}, \ldots, X_{\pi_m}^{(m)}) - \hat{I}_C^{(P)}(X_{\pi_1}^{(1)}, \ldots, X_{\pi_m}^{(m)}) \leq \tfrac{\epsilon}{2}\right] \tag{4.7}$$

$$\leq \mathbb{P}\left[\left|\left(\hat{I}_C^{(P^*)} - I_C^{(P^*)}\right) - \left(\hat{I}_C^{(P)} - I_C^{(P)}\right)\right| \geq \tfrac{\epsilon}{2}\right] \tag{4.8}$$

$$\leq 2c \exp\left(-n\delta_{\epsilon/4}\right), \tag{4.9}$$

where (4.7) follows from the fact that $\tilde{I}$ is the maximum empirical cluster information and (4.8) follows from the ϵ-likeness criterion. Finally, (4.9) follows from the triangle inequality, union bound and Lem. 18. Here again, in (4.8) the information measures are computed for $X_{\bar{\pi}}^{[m]}$.

From (4.5), (4.9), and the union bound we know that

$$\mathbb{P}\left[\hat{P}_{\bar{\pi}} \neq P^*\right] \leq 4c \exp\left(-n\delta_{\epsilon/4}\right). \tag{4.10}$$

Now, invoking Lem. 16, we know from similar analysis, that the densest fundamental partition is exactly P^*. More specifically, for any $\bar{\pi} \neq \bar{\pi}^*$, the equivalent fundamental partition is finer than P^*. This in turn indicates that

$$P_e(\Phi_C) \leq 4c|\Pi| \exp\left(-n\delta_{\epsilon/4}\right), \tag{4.11}$$

owing to the union bound and (4.10). □

4.1.4 K-Info Clustering

Under (B2), i.e., given number of clusters K in the set of images, let $\mathcal{P}_K \subset \mathcal{P}$ be the set of all partitions consisting K clusters. Then, much in the spirit of K-means clustering, we define the K-info clustering estimate as

$$(\hat{P}, \hat{\pi}) = \arg \max_{P \in \mathcal{P}_K, \bar{\pi} \in \Pi^m} \hat{I}_C^{(P)}(X_{\pi_1}^{(1)}, \ldots, X_{\pi_1}^{(m)}). \tag{4.12}$$

Lemma 20. *Given the number of clusters K in the set, the K-info clustering estimates are*

Algorithm 3 Thresholded Clustering, $\Phi_{\mathrm{T}}\left(\mathbf{X}^{(1)},\ldots,\mathbf{X}^{(m)},\alpha\right)$

$\gamma_n \leftarrow c_1 n^{-\alpha}$, for some constant $c_1 > 0$
for all $\bar{\pi} \in \Pi^m$ **do**
  Compute empirical pmf $\hat{p}\left(X_{\pi_1}^{(1)},\ldots,X_{\pi_m}^{(m)}\right)$
  $\tilde{I} = \max_{P\in\mathcal{P}} \hat{I}_C^{(P)}(X_{\pi_1}^{(1)},\ldots,X_{\pi_m}^{(m)})$
  $P_\pi = \text{Finest}\left\{P : \hat{I}_C^{(P)}(X_{\pi_1}^{(1)},\ldots,X_{\pi_m}^{(m)}) \geq \tilde{I} - \gamma_n\right\}$
end for
$(\hat{P},\hat{\pi}) = \left\{(P,\bar{\pi}) : P = \hat{P}_{\bar{\pi}} \succeq \hat{P}_{\bar{\pi}'},\ \text{for all } \bar{\pi}' \neq \bar{\pi}\right\}$

exponentially consistent.

Proof. Let $P_{\bar{\pi}} = \arg\max_{P\in\mathcal{P}_K} I_C^{(P)}(X_{\bar{\pi}}^{[m]})$ and $\hat{P}_{\bar{\pi}} = \arg\max_{P\in\mathcal{P}_K} \hat{I}_C^{(P)}(X_{\bar{\pi}}^{[m]})$. Then, for any $\bar{\pi} \in \Pi^m$,

$$\mathbb{P}\left[\hat{P}_{\bar{\pi}} \neq P_{\bar{\pi}}\right] = \mathbb{P}\left[\hat{I}_C^{(\hat{P}_{\bar{\pi}})}(X_{\bar{\pi}}^{[m]}) > I_C^{(P_{\bar{\pi}})}(X_{\bar{\pi}}^{[m]})\right] \leq 2(|\mathcal{P}_K|-1)c\exp\left(-n\delta_{\epsilon_{\bar{\pi}}}\right), \tag{4.13}$$

where, for $\epsilon_{\bar{\pi}} = I_C^{(P_{\bar{\pi}})}(X_{\bar{\pi}}^{[m]}) - \max_{P\neq P_{\bar{\pi}}} I_C^{(P)}(X_{\bar{\pi}}^{[m]})$, (4.13) follows from the union bound and Lem. 18.

Next, for any $\bar{\pi} \neq \bar{\pi}^*$, we have

$$\begin{aligned}
&\mathbb{P}\left[\hat{I}_C^{(P_{\bar{\pi}})}(X_{\bar{\pi}}^{[m]}) > \hat{I}_C^{(P_{\bar{\pi}^*})}(X_{\bar{\pi}^*}^{[m]})\right] && (4.14)\\
&\quad\leq \mathbb{P}\left[\left|\hat{I}_C^{(P_{\bar{\pi}})} - I_C^{(P_{\bar{\pi}})} - \hat{I}_C^{(P_{\bar{\pi}^*})} + I_C^{(P_{\bar{\pi}^*})}\right| \geq I_C^{(P_{\bar{\pi}^*})} - I_C^{(P_{\bar{\pi}})}\right] && (4.15)\\
&\quad\leq 2c\exp\left(-n\delta_{\tilde{\epsilon}_{\bar{\pi}}}\right), && (4.16)
\end{aligned}$$

where $\tilde{\epsilon}_{\bar{\pi}} = I_C^{(P_{\bar{\pi}^*})}(X_{\bar{\pi}^*}^{[m]}) - I_C^{(P_{\bar{\pi}})}(X_{\bar{\pi}}^{[m]})$. In (4.15), $I_C^{(P_{\bar{\pi}^*})}, \hat{I}_C^{(P_{\bar{\pi}^*})}$ are the information measures computed for $X_{\bar{\pi}^*}^{[m]}$, while $I_C^{(P_{\bar{\pi}})}, \hat{I}_C^{(P_{\bar{\pi}})}$ are computed for $X_{\bar{\pi}}^{[m]}$. Finally, the result follows from (4.13), (4.16), and the union bound. □

4.1.5 Clustering with Sub-Exponential Consistency

Under criterion (B3) we seek a universal clustering algorithm that is unaware of parameters such as ϵ and K. Since dependence characterizes similarity, given a source and channel, there exists $\epsilon > 0$ such that the images are ϵ-like. However, the decoder does not know ϵ. Since plug-in estimates are exponentially consistent, we adapt Alg. 2 to work with a dynamic threshold dependent on n. The threshold-based clustering algorithm is given in Alg. 3.

Lemma 21. *Let $\alpha \in (0, \frac{1}{4})$. Then, the thresholded clustering algorithm, Φ_T is asymptotically*

Algorithm 4 Hierarchical Clustering, $\Phi_{\mathrm{H}}\left(\mathbf{X}^{(1)}, \ldots, \mathbf{X}^{(m)}\right)$

$\hat{\pi} = \arg\max_{\bar{\pi} \in \Pi^m} \hat{I}_M(X_{\bar{\pi}}^{[m]})$
$\mathbf{Y}^{[m]} \leftarrow X_{\hat{\pi}}^{[m]}$
$P \leftarrow \{[m]\}$, $\hat{P}(1) \leftarrow P$
for $k = 2$ **to** m **do**
  for all $C \in P$ **do**
    $J_C \leftarrow \max_{\tilde{P} \in \mathcal{P}_2(C)} I_C^{(\tilde{P})}(Y^C)$
    $\tilde{P}_C \leftarrow \arg\max_{\tilde{P} \in \mathcal{P}_2(C)} I_C^{(\tilde{P})}(Y^C)$
  end for
  $\tilde{C} \leftarrow \arg\max_{C \in P} J_C$
  $P \leftarrow P \backslash \tilde{C}$
  $P \leftarrow P \cup \tilde{P}_{\tilde{C}}$
  $\hat{P}(k) \leftarrow P$
end for

consistent.

Proof. Let the given source and channel be ϵ-like. Since γ_n decreases with n, there exists $N_\epsilon < \infty$ such that for all $n > N_\epsilon$, $\gamma_n < \epsilon$. The proof now follows analogous to that of Lem. 19, as $n\delta_{\gamma_n} \to \infty$ as $n \to \infty$. □

At the expense of exponential consistency, we define a consistent universal algorithm only using independence.

4.1.6 Hierarchical Clustering

Finally we consider clustering according to criterion (B4). Here, we aim to establish the hierarchical clustering relation among images to establish a dendrogram for the images. From the nature of independence among dissimilar images, we design a natural algorithm for hierarchical clustering, Alg. 4. Fundamentally, at each stage, the algorithm splits a cluster into two such that the resulting clusters are most independent.

Lemma 22. *Consider a set of images with K clusters. Then, the partition at iteration $k = K$ of Φ_H is almost surely the correct clustering. Further, the transformation estimates are exponentially consistent.*

Proof. The proof follows directly from exponential consistency of the other algorithms. □

4.1.7 Computational and Sample Complexity

Although not the principal focus, we briefly note the computational and sample complexity of the algorithms. We have established that the fundamental partitions of CI may be obtained efficiently for any chosen transformation vector, given the joint distribution. However, exploring the neighborhood of the maximizing partition, for empirical estimates is a harder problem to address and it is not clear if there is an efficient algorithm, other than exhaustive search, to do this.

Further, identifying the transformation involves an exhaustive search in the absence of additional information on transformations, dependency across images, or underlying inter-pixel correlations. Since $|\Pi| = O(n^{\alpha})$, the complexity is $O(n^{\alpha m})$, i.e., exponential in the number of images. For a constant number of images to be clustered and registered, this method is efficient.

With regard to sample complexity, as we deal with the computation of information functionals of m random variables, we require $n = O(r^m)$ samples [73] highlighting the need for high-resolution images. Additional information on image model or transformations could reduce this.

4.1.8 Large-Scale Joint Registration and Clustering

One can directly extend Alg. 1 to incorporate clustering as well. In particular, we partition the images into subsets of a fixed size k and depending on the context of clustering, we perform the appropriate joint registration and clustering algorithm defined in Sec. 4.1. Then, representative images of each cluster in each subset are collected and the process is repeated to merge clusters across the subsets. Direct analysis similar to Thms. 9 and 10 shows that consistent clustering and registration requires $n = \Theta(\log m)$.

The reduction to blockwise registration and clustering also implies that the exhaustive searches are limited. For any $k = O(1)$, the exhaustive search for registration in each subset costs $O(n^{\alpha k}) = O\left((\log m)^{\alpha k}\right)$ computations. Similarly, clustering involves searching over $O(\exp(ck)) = O\left((\log m)^{c}\right)$ partitions for some constant c. Aggregation of subsets requires $O(m)$ block computations. Thus the overall computational cost scales as $O(m(\log m)^{\beta})$, for some $\beta > 0$, i.e. polynomially with number of images. However this is computationally expensive in practice and needs appropriate heuristic adaptations.

4.2 Multivariate Information Functionals and Clustering

As highlighted in Sec. 4.1, multivariate information functionals form a potent tool for independence-based clustering of random variables. A variety of algorithms that determine the dependence structure and perform clustering accordingly.

As observed with multiinformation in Sec. 4.1, the multiinformation quantifies the dependence among a set of random variables. The multiinformation satisfies several operational and informational properties, as shown in diverse application areas such as psychology [113, 114], machine learning [72, 115], image processing [116–118], cybernetics [119, 120], neuroscience [121], and multiterminal communication [122–124].

The purpose of this section is to provide some operational characterizations and useful properties for an alternative measure of dependence where the roles of the joint and product-of-marginal distributions are reversed. In this section we study the Csiszár conjugate of the multiinformation, what we call illum information[1] [71].

Definition 16. *Given a set of random variables $(X_1, \ldots, X_n) \sim p$, their Illum information is defined as*

$$L(X_1; \ldots; X_n) := D\left(P_{X_1} \ldots, P_{X_n} \| P_{X_1, \ldots, X_n}\right). \tag{4.17}$$

It is the multivariate extension of the lautum information [64] and has similar functional and operational characteristics. Let us define the sum of multiinformation and illum information, which we refer to as *sum information*:

$$S(X_1; \ldots; X_n) = I(X_1; \ldots; X_n) + L(X_1; \ldots; X_n). \tag{4.18}$$

4.2.1 Basic Properties

Since illum information is the multivariate extension of lautum information and the Csiszár conjugate of multiinformation, several informational features extend naturally. Some such properties are the following.

1. *Non-negativity:* Illum information, $L(X_1; \ldots; X_n) \geq 0$ with equality if and only if $\mathbb{P}[X_1, \ldots, X_n] = \prod_{i=1}^{n} \mathbb{P}[X_i]$. This follows directly from the fact that illum information is a relative entropy.

2. *Monotonicity:* For any $n > m \geq 2$, $L(X_1; \ldots; X_n) \geq L(X_1; \ldots; X_m)$. This follows from the chain rule and non-negativity of relative entropy as discussed later (4.23).

[1] *Illum* ("that" in Latin) is the reverse spelling of *multi*, if we do not cross our *t*s. We leave that, and working through the appropriate Radon-Nikodym derivatives to the interested reader, especially since this section is largely restricted to discrete alphabets.

3. *Data processing inequality:* If $X_1 \leftrightarrow X_2 \leftrightarrow \cdots \leftrightarrow X_n$ forms a Markov chain, then the data processing inequality of lautum information [64] extends to illum information as $L(X_1;\ldots;X_{n-1}) \geq L(X_1;\ldots;X_{n-2};X_n)$. The data processing inequality also extends to tree-structured Bayesian networks.

4. *Convexity:* Directly extending the results from [64], the illum information is

   (a) a concave function of $\mathbb{P}[X_i]$ for any $i \in [n]$, for a given $\mathbb{P}\left[X_{\setminus i}|X_i\right]$,

   (b) a convex function of $\mathbb{P}\left[X_i|X_{\setminus i}\right]$ for any $i \in [n]$, for a given $\mathbb{P}\left[X_{\setminus i}\right]$.

5. *Invariance under bijection:* Let $f : \mathcal{X}^n \to \mathcal{Y}^n$ be a bijective mapping and let $(Y_1,\ldots,Y_n) = f(X_1,\ldots,X_n)$. The illum information is invariant to such bijective transformations, i.e.,
$$L(X_1,\ldots,X_n) = L(Y_1,\ldots,Y_n).$$

6. *Lower and upper bounds:* Let the variational information for random variables $X_1,\ldots,X_n$ be defined as
$$V(X_1,\ldots,X_n) = D_f(p_{X_1}\ldots p_{X_n}\|p_{X_1,\ldots,X_n}),$$
for the convex function $f(x) = \frac{1}{2}|x-1|$. Using Pinsker's [125] and reverse Pinsker's [126] inequalities, the illum information can be bounded in terms of the variational information as
$$L(X_1;\ldots;X_n) \geq \frac{\log_2 e}{2} V^2(X_1,\ldots,X_n), \tag{4.19}$$
$$L(X_1;\ldots;X_n) \leq \frac{\log_2 e}{p_{\min}} V^2(X_1,\ldots,X_n), \tag{4.20}$$
where $p_{\min} = \min_{x_1,\ldots,x_n} p(x_1,\ldots,x_n)$, if $p_{\min} > 0$.

4.2.2 Chain Rules

Not all informational characterizations of multiinformation extend to illum information. In particular, multiinformation satisfies the chain rule given by
$$I(X_1;\ldots;X_n) = \sum_{i=2}^{n} I(X^{i-1};X_i). \tag{4.21}$$

Table 4.1: Joint distribution p of (X, Y, Z) for which clustering the random variables does not reduce illum information.

(X,Y,Z)	$p(X,Y,Z)$	(X,Y,Z)	$p(X,Y,Z)$
$(0,0,0)$	0.04	$(1,0,0)$	0.34
$(0,0,1)$	0.29	$(1,0,1)$	0.12
$(0,1,0)$	0.01	$(1,1,0)$	0.06
$(0,1,1)$	0.11	$(1,1,1)$	0.03

Let $\mathbf{X}$ be a random vector drawn according to a Bayesian network where A_i is the set of parents of node i. Then, the multiinformation decomposes as

$$I(X_1; \ldots; X_n) = \sum_{i=1}^{n} I(X_i; X_{A_i}).$$

In particular, if $P = \{C_1, \ldots, C_k\}$ is a partition of $[n]$, then

$$I(X_1; \ldots; X_n) \geq I(X_{C_1}; \ldots; X_{C_k}).$$

However, such decompositions do not necessarily hold for the case of illum information. For instance, consider $(X, Y, Z) \in \{0,1\}^3$ drawn as follows. Let $X \sim \text{Bern}(1/2)$,

$$Y \sim \begin{cases} \text{Bern}(\epsilon), \text{ if } X = 0 \\ \text{Bern}(\bar{\epsilon}), \text{ if } X = 1, \end{cases} \quad \text{and } Z \sim \begin{cases} \text{Bern}(\epsilon), \text{ if } X = Y \\ \text{Bern}(\bar{\epsilon}), \text{ if } X \neq Y, \end{cases}$$

for $\bar{\epsilon} = 1 - \epsilon$. Then, for $\epsilon < 1/2$,

$$L(X; Y; Z) > L(X; Y) + L(X, Y; Z).$$

On the other hand, for the distribution p given in Table 4.1,

$$L(X; Y; Z) < L(X; Y) + L(X, Y; Z).$$

In fact, $L(X; Y; Z) < L(X, Y; Z)$, i.e., clustering random variables does not necessarily decrease illum information as it does for multiinformation. Hence in general, the illum information does not satisfy a chain rule of the form of (4.21).

However, the chain rule does hold for tree-structured Bayesian networks. In particular, if the n-dimensional random vector, $\mathbf{X}$ is distributed according to a tree-structured Bayesian

network such that the parent of node i is A_i, then

$$L(X_1;\ldots;X_n) = \sum_{i=1}^{n} L(X_i;X_{A_i}). \tag{4.22}$$

Let $\mathbf{X} \sim p$ and let p_i be the marginal distribution of X_i. Let $q(\mathbf{X}) = \prod_{i=1}^{n} p_i(X_i)$. Using the chain rule of relative entropy, we have

$$L(X_1;\ldots;X_n) = \sum_{i=2}^{n} D\left(p_i(X_i)\middle\|p(X_i|X^{i-1})\middle|q(X^{i-1})\right). \tag{4.23}$$

4.2.3 Operational Characterizations: Independence Testing

Here, we identify the operational significance of multiinformation and illum information in multivariate independence testing problem defined as

$$\begin{cases} H_0: & (X_1,\ldots,X_m) \sim p \\ H_1: & (X_1,\ldots,X_m) \sim p_1 \times \cdots \times p_m, \end{cases} \tag{4.24}$$

where p_i is the marginal distribution of X_i corresponding to p. That is, the null and alternate hypothesis correspond to the cases of mutual independence and dependence among the random variables. For ease, let $Y = (X_1,\ldots,X_m)$, and let $Y_1,\ldots,Y_n$ be drawn independently and identically according to the underlying hypothesis. Let

$$\alpha = \mathbb{P}\left[\text{Decide } \{Y_1,\ldots,Y_n\} \in H_1 | H_0\right],$$
$$\beta = \mathbb{P}\left[\text{Decide } \{Y_1,\ldots,Y_n\} \in H_0 | H_1\right].$$

From [127], we have

$$d(\alpha\|1-\beta) \leq nI(X_1;\ldots;X_m), \tag{4.25}$$
$$d(\beta\|1-\alpha) \leq nL(X_1;\ldots;X_m), \tag{4.26}$$

where $d(a\|b) = a\log\left(\frac{a}{b}\right) + (1-a)\log\left(\frac{1-a}{1-b}\right)$, $a, b \in (0,1)$. It may be noted that (4.25) and (4.26) yield upper bounds on the receiver operating characteristic (ROC) for the independence testing problem.

Let $L(X_1;\ldots;X_m)$ and $I(X_1;\ldots;X_m) < \infty$. In the asymptotic setting, Stein's lemma [128] gives an estimate of the maximum conditional error exponents in the Neyman Pearson

formulation. In particular, for the best hypothesis test such that $\alpha < \delta$,

$$\lim_{n\to\infty} \frac{1}{n} \log(\beta) = -I(X_1; \ldots; X_m), \tag{4.27}$$

and similarly, for the best hypothesis test such that $\beta < \delta$,

$$\lim_{n\to\infty} \frac{1}{n} \log(\alpha) = -L(X_1; \ldots; X_m). \tag{4.28}$$

That is, we note that the Type I and Type II error exponents of independence testing are given by the illum information and multiinformation respectively.

In the Bayesian setting, let π_0 and $\pi_1 = 1 - \pi_0$ be the prior probabilities of hypotheses H_0 and H_1 respectively. Then, the log-likelihood ratio of samples $Y_1, \ldots, Y_n$ is

$$\ell_n(Y_1, \ldots, Y_n) = \log\left(\frac{\pi_1}{\pi_0}\right) + \sum_{i=1}^{n} \log\left(\frac{\mathbb{P}\left[Y_i|H_1\right]}{\mathbb{P}\left[Y_i|H_0\right]}\right).$$

Then, from the law of large numbers, we have

$$\frac{1}{n}\ell_n(Y_1, \ldots, Y_n) \xrightarrow{a.s.} \begin{cases} I(X_1; \ldots; X_m), & \text{if } H_0 \\ -L(X_1; \ldots; X_m), & \text{if } H_1 \end{cases} \tag{4.29}$$

if the information values are finite.

Under the sequential testing framework, the sequential probability ratio test (SPRT) tracks the log-likelihood ratio which is given by

$$S_n = \sum_{i=1}^{n} \log \frac{\mathbb{P}\left[Y_i|H_1\right]}{\mathbb{P}\left[Y_i|H_0\right]},$$

and declares one of $\{H_0, H_1\}$ once the sum crosses an appropriately chosen threshold. Let D be the mean drift of S_n. Then,

$$D = \begin{cases} -I(X_1; \ldots; X_m), & \text{if } H_0 \\ L(X_1; \ldots; X_m), & \text{if } H_1. \end{cases}$$

Consequently, Wald's approximation indicates that the expected sample size, N, required

for independence testing with Type I and Type II error levels α and β, is given by

$$N \approx \begin{cases} d(\alpha \| 1-\beta)/I(X_1;\ldots;X_m), & \text{if } H_0 \\ d(\beta \| 1-\alpha)/L(X_1;\ldots;X_m), & \text{if } H_1. \end{cases}$$

Observing the role played by the multivariate information functionals, we can observe that both multiinformation and illum information serve as two sides of the same coin with regard to the independence testing problem. As highlighted in this chapter, a variety of information-based clustering mechanisms have been defined recently to separate random variables into clusters which have minimal inter-cluster dependence [62, 111, 112]. These formulations use multivariate information functionals such as partition information and multiinformation to perform clustering, especially in universal settings. However, one practical issue in implementing such algorithms is that these information functionals are upper-bounded by entropy terms. In practice, these quantities could potentially be arbitrarily small, thereby making the clustering process very difficult.

However, we note that the illum information has no such generic upper bound in terms of entropy. Let $\mathbf{X} \sim p$, where the joint distribution p is a product over clusters in a partition P of $[n]$. That is, $p(\mathbf{X}) = \prod_{C \in P} p_C(X_C)$. Then for any partition $P' = \{C'_1, \ldots, C'_{|P'|}\}$,

$$L(X_{C'_1};\ldots;X_{C'_{|P'|}}) \geq 0, \tag{4.30}$$

with equality if and only if $P' \succeq P$.

For any partition $P = \{C_1, \ldots, C_{|P|}\}$, define $I^P(\mathbf{X}) = I(X_{C_1};\ldots;X_{C_{|P|}})$ and $L^P(\mathbf{X}) = L(X_{C_1};\ldots;X_{C_{|P|}})$. Then, the correct clustering of the given set of random variables, P^*, minimizes $I^P(\mathbf{X}) + L^P(\mathbf{X})$ over all partitions P. Additionally, due to non-negativity of the information functionals, it is easier to resolve between two possible partitions. Hence we claim that clustering using the sum information functional may be more robust for universal clustering than multiinformation or partition information.

However, at the same time, illum information is sensitive to simple corner cases as it can diverge to infinity. This in turn also calls for the need for stable and efficient estimators of illum information. A potential approach to computation could be through importance sampling, kernel, and nearest neighbor methods. It has been observed that the empirical estimation of information functionals for samples from non-parametric distributions is complicated by its computational and sample complexities [73, 129]. However, the estimation of the functionals is easier and more practical when the samples are drawn from restricted families of distributions. To exploit such distributional restrictions on the data, we now study the information functionals for some common distributional classes.

4.2.4 Information for Exponential Family

Consider an n-dimensional exponential family of distributions, $\{p_{\boldsymbol{\theta}}, \boldsymbol{\theta} \in \mathbb{R}^n\}$, of the form

$$p_{\boldsymbol{\theta}}(\mathbf{X}) = h(\mathbf{X}) \exp\left\{\boldsymbol{\theta}^T T(\mathbf{X}) - A(\boldsymbol{\theta})\right\}, \ \mathbf{X} \in \mathcal{X}^n, \tag{4.31}$$

where $\boldsymbol{\theta}$ is the vector of parameters, $T : \mathcal{X}^n \to \mathbb{R}^n$ is the function of sufficient statistics, $h : \mathcal{X}^n \to \mathbb{R}$, and $A(\boldsymbol{\theta})$ is the log-partition function. Let $\mathbf{X} \sim p_{\boldsymbol{\theta}}$ be the n-dimensional random vector $\mathbf{X} = (X_1, \ldots, X_n)$. Let $p_i(\cdot)$ be the marginal distribution of X_i and $q(\mathbf{X}) = \prod_{i \in [n]} p_i(X_i)$.

The multiinformation of $p_{\boldsymbol{\theta}}$ is given by

$$I(X_1; \ldots; X_n) = \sum_{i=1}^{n} H(X_i) - A(\boldsymbol{\theta}) - \mathbb{E}_{p_{\boldsymbol{\theta}}}\left[\log h(\mathbf{X})\right] + \boldsymbol{\theta}^T \nabla A(\boldsymbol{\theta}), \tag{4.32}$$

where $\nabla A(\boldsymbol{\theta})$ is the gradient of the log-partition function. This follows from the fact that $\mathbb{E}_{p_{\boldsymbol{\theta}}}\left[T(\mathbf{X})\right] = \nabla A(\boldsymbol{\theta})$.

On the other hand, the illum information is given by

$$L(X_1; \ldots; X_n) = A(\boldsymbol{\theta}) - \sum_{i=1}^{n} H(X_i) + \mathbb{E}_q\left[\log h(\mathbf{X})\right] - \boldsymbol{\theta}^T \mathbb{E}_q\left[T(\mathbf{X})\right]. \tag{4.33}$$

Consequently, we note that the sum information

$$I(\mathbf{X}) + L(\mathbf{X}) = \boldsymbol{\theta}^T \left[\nabla A(\boldsymbol{\theta}) - \mathbb{E}_q\left[T(\mathbf{X})\right]\right] + \left[\mathbb{E}_q\left[\log h(\mathbf{X})\right] - \mathbb{E}_{p_{\boldsymbol{\theta}}}\left[\log h(\mathbf{X})\right]\right]. \tag{4.34}$$

In particular, consider an n-dimensional jointly Gaussian random vector, $\mathbf{X} \sim \mathcal{N}(0, \Sigma)$. Consider the eigendecomposition of the covariance matrix as

$$\Sigma = \sum_{i=1}^{n} \lambda_i u_i u_i^T,$$

where $\{\lambda_i, i \in [n]\}$ are the eigenvalues and $\{u_i, i \in [n]\}$ are the corresponding orthonormal eigenvectors. Without loss of generality, let $\lambda_i > 0$ for all i. Then,

$$I(X_1; \ldots; X_n) = \frac{1}{2} \sum_{i=1}^{n} \log\left(\frac{\sigma_i^2}{\lambda_i}\right), \tag{4.35}$$

$$L(X_1; \ldots; X_n) = \frac{1}{2} \sum_{i=1}^{n} \left[\frac{u_i^T \hat{\Sigma} u_i}{\lambda_i} - \log\left(\frac{\sigma_i^2}{\lambda_i}\right) - 1\right], \tag{4.36}$$

where $\hat{\Sigma}$ is the diagonal matrix of variance values.

Let $Y_i = X_i/\sigma_i$ and let $\tilde{\Sigma}$ be the covariance matrix of the normalized Gaussian random variables such that $\tilde{\Sigma} = \sum_{i=1}^{n} \tilde{\lambda}_i \tilde{u}_i \tilde{u}_i^T$ is the orthonormal eigendecomposition. Since information is invariant to bijective transformations, we have $L(\mathbf{X}) = L(\mathbf{Y})$ and $I(\mathbf{X}) = I(\mathbf{Y})$, where

$$I(Y_1; \ldots; Y_n) = \frac{1}{2} \sum_{i=1}^{n} \log\left(\frac{1}{\tilde{\lambda}_i}\right), \tag{4.37}$$

$$L(Y_1; \ldots; Y_n) = \frac{1}{2} \sum_{i=1}^{n} \left[\frac{1}{\tilde{\lambda}_i} - \log\left(\frac{1}{\tilde{\lambda}_i}\right) - 1\right], \tag{4.38}$$

$$I(\mathbf{Y}) + L(\mathbf{Y}) = \frac{1}{2} \sum_{i=1}^{n} \left[\frac{1}{\tilde{\lambda}_i} - 1\right], \tag{4.39}$$

$$L(\mathbf{Y}) - I(\mathbf{Y}) = \sum_{i=1}^{n} \frac{1}{2\tilde{\lambda}_i} - \frac{1}{2} - \log\left(\frac{1}{\tilde{\lambda}_i}\right). \tag{4.40}$$

For $n = 2$, $L(\mathbf{X}) \geq I(\mathbf{X})$ [64]. However the result does not extend for $n > 2$. For instance, consider the 3-dimensional jointly Gaussian vector with covariance matrix

$$\Sigma = \begin{bmatrix} 1 & 0.25 & 0.25 \\ 0.25 & 1 & -0.25 \\ 0.25 & -0.25 & 1 \end{bmatrix}. \tag{4.41}$$

Then, $L(\mathbf{X}) - I(\mathbf{X}) = -0.0032$.

4.2.5 Information for Pairwise Markov Random Fields

Consider an undirected graph $G = (V, E)$ and the pairwise Markov random field (MRF) defined on G, parametrized by the node potential functions $\{\psi_i(\cdot), i \in V\}$ and edge potential functions $\{\psi_{ij}(\cdot), (i,j) \in E\}$, given by

$$p_G(\mathbf{X}) = \exp\left(\sum_{i \in V} \psi_i(X_i) + \sum_{(i,j) \in E} \psi_{ij}(X_i, X_j) - A(\boldsymbol{\psi})\right), \tag{4.42}$$

where $A(\boldsymbol{\psi})$ is the log-partition function. Again, let p_i be the marginal distribution of X_i and $q(\mathbf{X}) = \prod_{i \in V} p_i(X_i)$.

Then, we have

$$L(\mathbf{X}) = A(\boldsymbol{\psi}) - \sum_{i \in V} H(X_i) - \sum_{i \in V} \mathbb{E}\left[\psi_i(X_i)\right] - \sum_{(i,j) \in E} \mathbb{E}_q\left[\psi_{ij}(X_i, X_j)\right], \tag{4.43}$$

$$I(\mathbf{X}) = \sum_{i \in V} H(X_i) - A(\boldsymbol{\psi}) + \sum_{i \in V} \mathbb{E}\left[\psi_i(X_i)\right] + \sum_{(i,j) \in E} \mathbb{E}_{p_G}\left[\psi_{ij}(X_i, X_j)\right]. \tag{4.44}$$

This in turn indicates that the sum information

$$L(\mathbf{X}) + I(\mathbf{X}) = \sum_{(i,j) \in E} \mathbb{E}_{p_G}\left[\psi_{ij}(X_i, X_j)\right] - \mathbb{E}_q\left[\psi_{ij}(X_i, X_j)\right], \tag{4.45}$$

which is equivalent to the cumulative potential difference across edges owing to independence. Note that the sum information is independent of node potentials and the partition function. This indicates that the "effective information" or the symmetric distance from independence is quantified entirely by the edge effects of the MRF. Additionally it may be noted that this sum information may be estimated easily from data, given the edge potentials.

In particular, let us consider the Ising model defined on a graph $G = (V, E)$, with parameter set $\{\theta_i, i \in V\} \cup \{\theta_{ij}, (i,j) \in E\}$. For an Ising model, $\mathbf{X} \in \{-1, +1\}^{|V|}$ and the potentials are defined as

$$\psi_i(X_i) = \theta_i X_i, \text{ and } \psi_{ij}(X_i, X_j) = \theta_{ij} X_i X_j.$$

Let the log-partition function be $A(\boldsymbol{\theta})$.

Then,

$$L(\mathbf{X}) = A(\boldsymbol{\theta}) - \sum_{i \in V} \left(H(X_i) + \theta_i \bar{X}_i\right) - \sum_{(i,j) \in E} \theta_{ij} \bar{X}_i \bar{X}_j, \tag{4.46}$$

$$I(\mathbf{X}) = \sum_{i \in V} \left(H(X_i) + \theta_i \bar{X}_i\right) - A(\boldsymbol{\theta}) + \sum_{(i,j) \in E} \theta_{ij} \mathbb{E}\left[X_i X_j\right]. \tag{4.47}$$

It may be noted here that the information functionals depend only on the mean and entropy of the nodes, and the correlation across the edges. In particular,

$$L(\mathbf{X}) + I(\mathbf{X}) = \sum_{(i,j) \in E} \theta_{ij} C_{ij}, \tag{4.48}$$

where $C_{ij} = \mathbb{E}\left[(X_i - \bar{X}_i)(X_j - \bar{X}_j)\right]$ is the covariance corresponding to edge (i,j). This indicates that the sum information is effectively the sum of edge covariances weighted by the edge potential parameters. Using the Cauchy-Schwarz inequality and the non-negativity of

multiiinformation, we note that for generic Ising models,

$$L(\mathbf{X}) \leq \sum_{(i,j)\in E} \theta_{ij}\sigma_i\sigma_j \leq \sum_{(i,j)\in E} \theta_{ij}, \tag{4.49}$$

where σ_i^2 is the variance of X_i. That is, the illum information is bounded in terms of the variance values for Ising models, and more loosely by just the edge potential weights. Note that this upper bound holds for the multiinformation as well.

Remark 10. *Result* (4.45)*, suggests a simple thought experiment. Let the graph* $G = (V, E)$ *represent a network of friends who are out to vote for a dinner restaurant from a list* $\mathcal{X}$*. Let us additionally assume that the self choice is reflected in node potentials* $\{\psi_i(\cdot), i \in V\}$*, and that homophily additionally increases the likelihood of similar responses among friends through edge potentials* $\{\psi_{ij}(\cdot), (i,j) \in E\}$ *of the form*

$$\psi_{ij}(X_i, X_j) = \mathbf{1}\{X_i = X_j\}, \quad \textit{for all } (i,j) \in E,$$

where $\mathbf{1}\{\cdot\}$ *is the indicator function.*

Friendships are strained when any two friends vote for different restaurants. To this end, a fair cost function to consider would be the cumulative strain reflected by

$$C(\mathbf{X}) = \sum_{(i,j)\in E} \mathbf{1}\{X_i \neq X_j\}.$$

From (4.45) *and the non-negativity of information, we observe that*

$$\mathbb{E}_{p_G}[C(\mathbf{X})] \leq \mathbb{E}_q[C(\mathbf{X})], \tag{4.50}$$

which indicates that the effective strain on a social group that colludes in taking a decision is less than the strain on a group with matching marginals, but that answers independently. This indicates the need for discussion in a social group to achieve cohesive decision making.

4.2.6 Distribution Approximation Problem

Analytically approximating (as opposed to sampling) a target distribution is common in variational inference, where the true yet intractable posterior is to be approximated by a distribution that is easy to handle. This is typically done by restricting the distributions under consideration to a class that has certain properties, e.g. assuming that the class of distributions factorize in a particular way or that they all have a specific parametric form such

as Gaussian. As a consequence, the approximation problem reduces to finding a distribution in the restricted class that best approximates the target distribution, i.e., a *projection* of the target distribution onto the restricted class.

Consider the problem of projecting a probability distribution p onto a set S of probability distributions. We define the projection of p onto the set S as the "closest" distribution to p among all distributions in S, where "closeness" between two distributions is measured in the following two ways:

$$\mathcal{P}_S(p) = \arg\min_{p' \in S} D(p\|p') \tag{4.51}$$

$$\mathcal{P}'_S(p) = \arg\min_{p' \in S} D(p'\|p) \tag{4.52}$$

where we have taken the two forms of relative entropy. In general, $\mathcal{P}_S(p) \neq \mathcal{P}'_S(p)$, due to the asymmetry of relative entropy. However it is trivial to see that for any $p \in S$, we have $\mathcal{P}_S(p) = \mathcal{P}'_S(p) = p$, which shows that both $\mathcal{P}_S(\cdot)$ and $\mathcal{P}'_S(\cdot)$ are indeed projections (idempotent).

Now we consider a special class of distributions, which all factorize over a given directed acyclic graph (Bayesian network). Formally, let G be a Bayesian network over random variables $X_1, \ldots, X_n$, and S_G be the set of distributions that factorize over G, then one can show that

$$\mathcal{P}_{S_G}(p) = \prod_i p_{i|A_i},$$

where inferred from p, $p_{i|A_i}$ is the conditional distribution of X_i given its parents in G. In an extreme case, let G_0 be the Bayesian network containing no edges (every node is independent of the others), then

$$\mathcal{P}_{S_{G_0}}(p) = \prod_i p_i,$$

i.e. the product of marginals.

In addition, one can show that

$$I(X_1; \ldots; X_n) = D\left(p\|\mathcal{P}_{S_{G_0}}(p)\right) \tag{4.53}$$

$$= D\left(p\|\mathcal{P}_{S_G}(p)\right) + \sum_{i=1}^{n} I(X_i; X_{A_i}), \tag{4.54}$$

where A_i denotes the parents of X_i in G. Note that both terms in (4.54) are nonnegative, which implies that $I(X_1; \ldots; X_n) \geq D\left(p\|\mathcal{P}_{S_G}(p)\right)$, where equality holds if and only if $G = G_0$; and $I(X_1; \ldots; X_n) \geq \sum_{i=1}^{n} I(X_i; X_{A_i})$, where equality holds if and only if $p \in S_G$. This

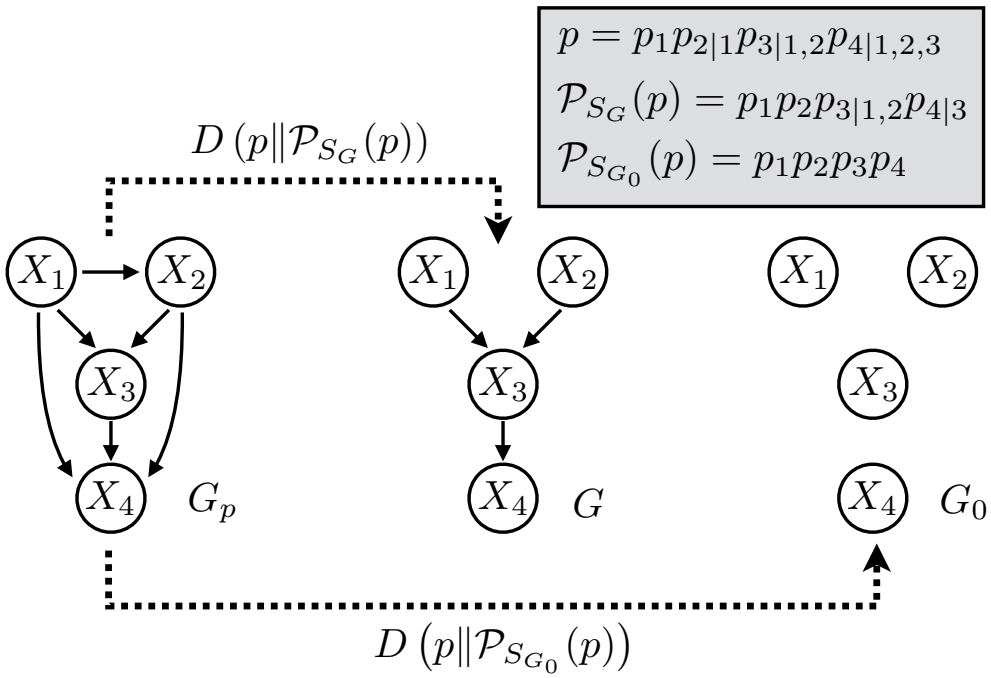


Figure 4.1: Project a distribution p onto S_G and S_{G_0}, respectively. Note that: $G_p \supset G \supset G_0$, in which the numbers of independence conditions are getting larger and larger.

additive projection property of multiinformation is depicted in Fig. 4.1.

On the other hand, such projection properties do not hold for illum information, when the other projection operator $\mathcal{P}'_S(\cdot)$ is adopted. To see this, let us consider the specific example of projecting a distribution onto the set of distributions that are product-of-marginals, i.e. the mean-field approximation of a distribution into the product distribution. Specifically, let S_{G_0} be the set of all distributions of the form $q(\mathbf{X}) = \prod_{i=1}^n q_i(X_i)$. Then, the mean-field approximation,

$$q^\star = \mathcal{P}'_{S_{G_0}}(p) = \arg\min_{q \in S_{G_0}} D(q\|p), \tag{4.55}$$

is given by the recursive formulation:

$$q_i^\star(x_i) = \exp\left(\mathbb{E}_{q^\star}\left[\log p(X^{i-1}, x_i, X_{i+1}^n)\right] - \lambda_i\right), \tag{4.56}$$

where λ_i is the log-partition function.

Such mean field approximation characterizes an analytical version of the independence clustering problem for distributions of random variables. In particular, the clustering obtained from multiinformation and illum information cost functions characterize different approximations for the joint distribution.

4.3 Discussion

In this chapter, we considered the problem of clustering using multivariate information functionals from two viewpoints. In the first part we studied the role of multivariate information functionals in clustering by studying the problem of joint clustering and registration of images. In the second part, we explored new multivariate information functionals of illum and sum information, and explored their operational and functional properties that inspire its use in independence clustering.

In particular, the first part of the chapter explores a variety of joint registration and clustering algorithms using clustering information. We highlighted the fact that the two tasks are not separate, and that one can invoke novel multivariate information functionals in order to perform joint image clustering and registration universally. Further, we also studied the consistency of the algorithms in the respective criteria, and the computational and statistical complexities involved. Finally, we explored the setting of large-scale clustering and registration, and briefly studied the computational complexity of the problem.

In the second half of the chapter we defined the illum information functional and studied its operational and functional properties. We also studied the information function for common distributional families such as the exponential family and pairwise Markov random fields. In these forms we observe the potentially efficiency of the illum and sum information functionals in performing independence clustering over data. We also considered the problem of mean field approximation with respect to the illum information, therein exploring an analytical version of the independence clustering problem.

The algorithms and the properties of multivariate information functionals highlight the potency of information-based methods in unsupervised clustering. The ubiquitous nature of information inspires the design of a variety of heuristic algorithms for the independence clustering problem. Future work on independence clustering would thus benefit from consistent information estimators and efficient clustering algorithms to use them.

CHAPTER 5

BUDGET-OPTIMAL CLUSTERING FOR CROWDSOURCING

In Chapter 4 we considered the problem of clustering using multivariate information functionals. In this chapter we consider the problem of clustering objects based on crowdsourced responses in a universal setting [74]. Crowdsourcing has grown in recent times as a potent tool for performing complex tasks using human skill and knowledge. It is increasingly being used to collect training data for novel machine learning problems. Almost *a fortiori*, there is no prior knowledge on the nature of the task and so the use of general human intelligence has been needed [130]. As such, this setting requires processing human-generated signals in the absence of prior knowledge about their properties; this setting requires universality.

Because crowdsourcing often employs unreliable workers, the signals they generate are inherently noisy [131]. Hence, responses of crowd workers are modeled as outputs of a noisy channel. Although these channels are unknown to the employer, crowdsourcing techniques have thus far made assumptions on either the channel distribution or structure to design appropriate decoders. We define an alternative approach—*universal crowdsourcing*—that designs decoders without channel knowledge, and develop achievability and converse arguments that demonstrate order-optimality.

The emergence of diverse online crowdsourcing platforms such as Amazon Mechanical Turk and Upwork has created the option of choosing between temporary workers and long-term workers. That is, tasks can be completed by either soliciting responses from a large number of workers performing small parts of a large task, or a specialized group of employees who work long-term on the task at hand.

The trade-off between reliability and cost of each type of worker pool warrants systematic study. Whereas temporary workers are inexpensive and easily available, some labor economists argue the excess cost of long-term employment is worthwhile due to the reliability and quality of work it ensures [132]. However, no quantitative characterization for this conjecture exists. As we will see, the results of this work allow such comparisons.

Since workers are human, they are subject to several factors that affect human decision making, as identified in behavioral economics [133]. A standard assumption in crowdsourcing and in universal clustering [134, 135] has been independent and identically distributed (i.i.d.)

worker responses across time/tasks. However, empirical evidence argues against temporal independence for individual worker responses [136].

Due to the *availability heuristic*, worker responses may rely on the immediate examples that come to a person's mind, indicating memory in responses across tasks/time. Further, this influence may be more due to salient (vivid) information rather than full statistical history. Due to the *anchoring and adjustment heuristic*, people tend to excessively rely on a specific trait of an object in decision making and further due to the *representativeness heuristic*, people tend to assume commonality among objects. These traits indicate there is a notion of distance among the response distributions corresponding to different objects.

To capture memory and distance, we define a unified model of worker responses, and then study two limiting cases [74]. First we consider responses of temporary workers who respond independently across tasks and across workers. We then consider responses of long-term workers with object-specific memory. Specifically, we consider a Markov memory model wherein the response to an object is dependent on the most recent response and the response to the most recent occurrence of an object of the same class; generalizations to other Markov models follows readily. In both cases, we address questions of universality and sample complexity for reliable clustering, providing benchmarks for worst-case performance.

The presence of memory in responses, however, demands an approach that differs from past crowd algorithms defined for i.i.d. models [134]. Notwithstanding [137], in the crowdsourcing framework herein, encoding is not feasible. This calls for a treatment different from past work in universal clustering [135].

There is a vast and rich literature on crowdsourcing and clustering; we describe a non-exhaustive listing of particularly relevant prior work.

Algorithm design for crowdsourcing typically focuses on minimizing the cost of reliability. In particular, algorithms with order-optimal budget-reliability trade-offs have been designed for binary classification with unknown (but i.i.d.) crowd reliabilities [134]. Efficient algorithms for multi-class labeling have been proposed in [138] albeit without cost optimality guarantees. More recently, non-parametric permutation models of crowd workers were considered for binary classification [139]. Classification using crowdsourced responses in a clustering framework, followed by a labeling phase performed by a domain expert has been studied experimentally [140].

Separate from crowdsourcing, the problem of clustering has been widely studied. Algorithms such as k-means clustering and its generalization to other Bregman divergence similarity measures [105] are popular methods that incorporate distance-based clustering. The problem of universal clustering was considered in a communication setting [135, 141], such that messages communicated across an unknown channel, after encoding using a ran-

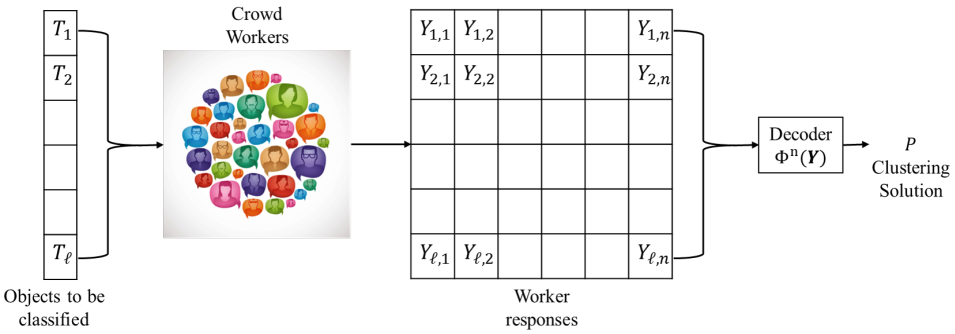


Figure 5.1: Model of the crowdsourcing system.

dom codebook, are clustered by exploiting dependency among outputs of similar messages. Particularly the decoder uses the minimum partition information functional [112] to perform optimal clustering. Similar information-based agglomerative clustering schemes have also been explored [110].

Some other studies on crowdsourcing have also considered clustering using comparative evaluations of objects. Varied types of comparative queries yield different cost-reliability tradeoffs [142]. Information-theoretic bounds on query complexity under this model have been established recently [143]. Multi-class labeling by decomposing the task into simpler subtasks, and introduce redundancy through error control codes have also been considered [137, 144].

5.1 System Model

We consider the problem of crowd workers employed to perform classification of objects. For instance, consider the task of classifying images of dogs according to their breeds. The workers observe images and respond with the breed of the dog in the image. Since worker responses are noisy, in the absence of knowledge of worker channels it is not feasible to identify the labels (breeds of dogs) accurately. Thus, we aim to cluster the dogs according to their breeds and determine the labels of each cluster by using a domain expert. The crowdsourcing system model is depicted in Fig. 5.1.

Let $\mathcal{T} = [\tau] = \{1, \ldots, \tau\}$, where $\tau < \infty$ is a constant, be the set of labels. Let $X \in \mathcal{X}$ be the object viewed by crowd workers and let its label be T. Let the set of objects to be clustered be $\{X_1, \ldots, X_\ell\}$ with T_i, the label of object X_i. That is, the objects are manifestations of the labels and we seek to cluster according to the labels. Let us assume that these labels are drawn according to an unknown prior $P_T(\cdot)$.

For each object X_i, the crowdsourcing system solicits n worker responses, Y_i^n, with $\mathbf{Y}$

being the matrix of responses. Let $W \in \mathcal{W}$ be the index of a worker and let $S_W \subset [\ell] \times [n]$ be the index set of the responses offered by W.

Let $\mathcal{Q}$ be the set of all conditional probability mass functions (PMFs) that characterize worker responses. Then, $Y_{S_W} \sim Q^{(W)}(Y_{S_W}|T^\ell)$ for each worker $w \in \mathcal{W}$. Models of these distributions are detailed later.

In practice, it is not essential for a worker to answer each assigned query. Thus we assume that the workers either respond with an answer in $\mathcal{T}$ or offer a "null" response ξ [145] i.e., $Y_{ij} \in \mathcal{Y} = \mathcal{T} \cup \{\xi\}$, for all $i \in [\ell], j \in [n]$.

5.1.1 Universal Clustering Performance

Definition 17 (Correct Clustering)**.** *A* clustering *of a set of objects $X_1, \ldots, X_\ell$ is a partition P of $[\ell]$. The sets of a partition are referred to as* clusters*. The clustering is said to be* correct *if*

$$T_i, T_j \in C \Leftrightarrow T_i = T_j,$$

for all $i, j \in [\ell]$, $C \in P$. For a given set of object labels, T^ℓ, let $P^*\left(T^\ell\right)$ be the correct clustering.

Let $\mathcal{P}$ be the set of all partitions of $[\ell]$.

Definition 18 (Partition Ordering)**.** *A partition P is* finer *than P', if the following ordering holds*

$$P \preceq P' \Leftrightarrow \text{ for all } C \in P, \text{ there exists } C' \in P' : C \subseteq C'.$$

Similarly, a partition P is said to be denser *than P' if $P \succeq P' \Leftrightarrow P' \preceq P$.*

Definition 19 (Universal Clustering Decoder)**.** *A* universal clustering decoder *is a sequence of functions $\Phi^{(n)} : \mathbf{Y} \to \mathcal{P}$ that are designed in the absence of knowledge of $\mathcal{Q}$ and P_T. Here the index n corresponds to the number of crowd responses collected per object.*

We now define how to characterize decoder performance.

Definition 20 (Error Probability)**.** *Let $\Phi^{(n)}(\cdot)$ be a universal decoder. Then, the* error probability *is given by*

$$\begin{aligned} P_e(\Phi^{(n)}) &= \mathbb{P}\left[\Phi^{(n)}(\mathbf{Y}) \neq P^*\left(T^\ell\right)\right] \\ &= \mathbb{E}_{P_T^{\otimes \ell}}\left[\mathbb{E}\left[\mathbf{1}\left\{\Phi^{(n)}(\mathbf{Y}) \neq T^\ell\right\} | T^\ell\right]\right], \end{aligned} \tag{5.1}$$

where $P_T^{\otimes \ell}(t^\ell) = \prod_{i=1}^{\ell} P_T(t_i)$ and $\mathbf{1}\{\cdot\}$ is the indicator function.

Definition 21 (Asymptotic Consistency)**.** *A sequence of decoders* $\Phi^{(n)}$ *is said to be* universally asymptotically consistent *if*

$$\lim_{n\to\infty} P_e(\Phi^{(n)}) = 0, \text{ for all } P_T \in \mathcal{M}(\mathcal{T}),$$

where $\mathcal{M}(\cdot)$ *is the space of all prior distributions on the set of objects,* $\mathcal{T}$*.*

Definition 22 (Sample Complexity)**.** *Let* $\epsilon > 0$ *be the permissible error margin. Then the* sample complexity *of the universal clustering problem is*

$$N^*(\epsilon) = \min\left\{n \in \mathbb{N} : \max_{P_T \in \mathcal{M}(\mathcal{T})} P_e(\Phi^{(n)}) < \epsilon\right\},$$

where the minimum is taken over the set of all sequences of universal decoders $\Phi^{(n)}$*.*

For simplicity, we will use Φ to denote $\Phi^{(n)}$ when it is clear from context.

5.1.2 Workers

We now define the unified worker model. Let us assume the crowdsourcing system employs n crowd workers, $[n] \subseteq \mathcal{W}$, chosen at random. We assume that each worker responds to every object $X_i, i \in [\ell]$ and $S_j = \{(1,j), \ldots, (\ell,j)\}, j \in [n]$.

Incorporating the availability and representativeness heuristics, we assume responses of each worker depend on prior responses in a Markov sense. In particular, define the neighbor set as $\mathcal{N}_i = \{i-1\} \cup \{\max\{k \in [i-1] : T_k = T_i\}\}$, for all $i \in [\ell]$ i.e., the most recent object and the most recent occurrence of a similar object. Then for any $j \in [n]$, $k \leq \ell$

$$\mathbb{P}\left[(Y_{1,j}, \ldots, Y_{k,j}) = y^k | T^k = t^k\right] = \prod_{i=1}^{k} Q^{(j)}(y_i | y_{\mathcal{N}_i}, t_i). \tag{5.2}$$

Additionally we assume that for any $j \in [n]$, $i \leq \ell$, $t \in \mathcal{T}$,

$$\mathbb{P}\left[Y_{i,j} = y | T_i = t\right] = Q^{(j)}\left(y|t\right). \tag{5.3}$$

That is, marginals are invariant across permutations of objects (anchoring and adjustment heuristic). We also assume that, given the object, responses are independent across workers:

$$\mathbb{P}\left[(Y_{i,1}, \ldots, Y_{i,k}) = y^k | T_i = t\right] = \prod_{j=1}^{k} Q^{(j)}\left(y_j|t\right). \tag{5.4}$$

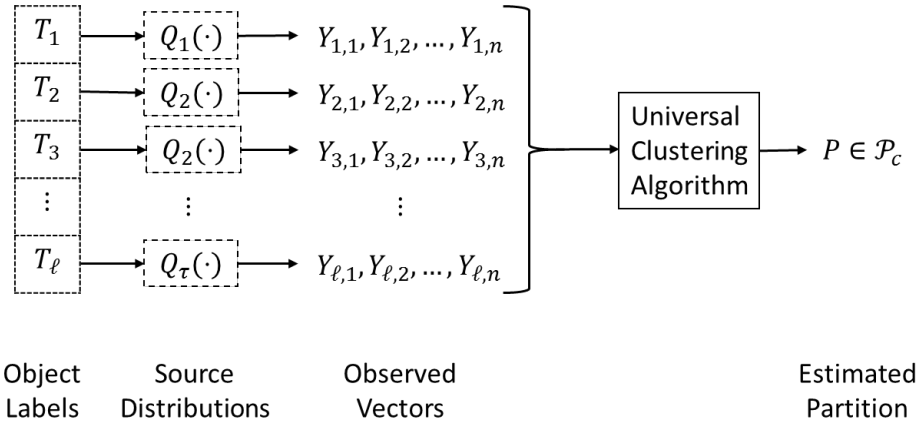


Figure 5.2: Universal distance-based clustering model: Here we cluster ℓ sources with labels $(T_1, \ldots, T_\ell) = (1, 2, 2, \ldots, \tau)$, as highlighted by the indices of their source distributions. True labels define the source distributions and for each source, n i.i.d. samples are drawn to generate the observation vectors.

We assume temporary workers do not retain memory of prior responses and so are independent across time as well. In order to solicit such responses, we may assume that the platform delegates each task to a sequence of n workers selected at random from a sufficiently large pool to ensure temporal independence. That is,

$$\mathbb{P}\left[\mathbf{Y}_{S_w} = y_{S_w} | T^\ell = t^\ell\right] = \prod_{(i,j)\in S_w} Q^{(w)}(y_{i,j}|t_i). \tag{5.5}$$

In addition, the responses also satisfy (5.4).

The second special class of workers we consider is long-term workers with Markov memory whose responses are characterized by (5.2) and (5.4). This model is inspired by the fact that long-term workers in the system are typically influenced by their prior responses as they tend to learn or get bored of the task. Thus the responses offered are dependent not only on the object label, but also the most recent instance of its sighting by the worker.

5.2 Temporary Workers

Let us first consider the case of temporary workers. The universal clustering problem for this context is depicted in Fig. 5.2. As depicted, the objects are equivalently defined by the sources that characterize the responses and thus clustering them translates to grouping identical distributions.

We test this distributional identicality in terms of an appropriately chosen f-divergence

between response distributions. In this section we elaborate the algorithm design and the results first with respect to total variational distance, and then show the extension to a much larger class of f-divergence functionals as detailed in [67].

Let $Q_1, \ldots, Q_\tau$ be defined as

$$Q_i(y) \triangleq \mathbb{P}\left[Y = y | T = i\right] = \mathbb{E}\left[Q^{(W)}(Y = y | T = i)\right], \tag{5.6}$$

where the expectation is taken over $W \in \mathcal{W}$. Since the workers assigned to tasks are chosen i.i.d. from the worker pool, the empirical distribution of the responses for an object with label i converges to the distribution Q_i almost surely. The quality of the given worker pool and assignment policy is characterized by the distance between the response distributions for objects with different labels. Let us thus define the difficulty of the clustering problem, and equivalently the quality of the crowd, in terms of the distance quality in the problem.

Definition 23 (Distance Quality)**.** *For a given pool of temporary workers, the difficulty of the tasks is quantified by the* distance quality*,*

$$\theta_d \triangleq \min_{\{i,j \in \mathcal{T}, i \neq j\}} d_{\mathrm{TV}}(Q_i, \|Q_j). \tag{5.7}$$

The operational significance of this informational definition of distance (Definition 23) will emerge in coding theorems Lem. 24 and Thm. 17.

Clustering is performed using the maximum likelihood estimates of the f-divergence between distributions corresponding to responses to objects. Let us first provide a brief introduction to f-divergences and the family of divergences we employ here for clustering.

5.2.1 f-Divergence

To measure the separation among the conditional distributions of crowd responses to different object classes, we use the Csiszár f-divergence [146, 147].

Definition 24 (f-divergence)**.** *Let p, q be discrete probability distributions defined on a space of m alphabets. Given a convex function $f : [0, \infty) \to \mathbb{R}$, the* f-divergence *is defined as:*

$$D_f(p\|q) \triangleq \sum_{i=1}^{m} q_i f\left(\frac{p_i}{q_i}\right). \tag{5.8}$$

The function f is said to be normalized *if $f(1) = 0$.*

Some specific f-divergences are the KL divergence $D(p\|q)$ and the total variational distance $d_{\mathrm{TV}}(p\|q)$. Specifically, the KL divergence and the total variational distance are the f-divergences corresponding to the functions $f(x) = x\log x$ and $f(x) = |x-1|$ respectively. We now state some bounds for f-divergences.

Theorem 15 ([148, Chapter II.1]). *Let p, q be discrete probability distributions on a space of m alphabets such that there exist r, R satisfying $0 \le r \le \frac{p_i}{q_i} \le R \le \infty$ for all $i \in [n]$. Let $f : [0,\infty) \to \mathbb{R}$ be a convex and normalized function satisfying the following criteria:*

1. *f is twice differentiable on $[r, R]$, and*
2. *there exist real constants $c, C < \infty$ such that*

$$c \le xf''(x) \le C, \text{ for all } x \in (r, R).$$

Then, we have

$$cD(p\|q) \le D_f(p\|q) \le CD(p\|q). \tag{5.9}$$

For ease, we refer to the constraints on f in Thm. 15 as *smoothness* constraints. If f is twice differentiable in $[r, R]$, then we know that there exists a constant L such that f is L-Lipschitz.

Theorem 16 ([148, Chapter II.3]). *Let $f : [0,\infty) \to \mathbb{R}$ be convex, normalized, and L-Lipschitz on $[r, R]$. Then,*

$$0 \le D_f(p\|q) \le Ld_{\mathrm{TV}}(p\|q). \tag{5.10}$$

Further, Pinsker's inequality lower bounds the KL divergence $D(p\|q)$ with respect to the total variational distance as

$$D(p\|q) \ge (2\log_2 e)\delta^2(p\|q). \tag{5.11}$$

Corollary 5. *For any convex and normalized function f that satisfies the smoothness constraints and is L-Lipschitz,*

$$\kappa d_{\mathrm{TV}}^2(p\|q) \le D_f(p\|q) \le Ld_{\mathrm{TV}}(p\|q), \tag{5.12}$$

where $\kappa = 2c\log_2 e$.

Proof. The result follows from Thm. 15 and 16, and (5.11). □

The maximum likelihood (plug-in) estimates of the divergences are asymptotically consistent.

Algorithm 5 Clustering with temporary workers, $\Phi_{\text{temp}}(\mathbf{Y})$

$\gamma_n \leftarrow c_1 n^{-\alpha}$, where c_1 is a constant and $\alpha \in [0, 1/2]$
Determine empirical distributions q_j, $j \in [\ell]$
Create $G = ([\ell], E)$, s.t., $(i, j) \in E$ if $d_{\text{TV}}(q_i \| q_j) \leq \gamma_n$
$\mathcal{C} = \{C : C \text{ is a maximal clique in } G\}$
Select minimal weight partition of $[\ell]$ from $\mathcal{C}$

Lemma 23. *If p and $\hat{p}$ are the true and empirical distributions respectively, then*

$$\mathbb{P}\left[d_{\text{TV}}(\hat{p}\|p) \geq \epsilon\right] \leq (n+1)^{|\mathcal{Z}|} \exp\left(-c_0 n \epsilon^2\right), \tag{5.13}$$

where $c_0 = 2 \log_2 e$. Further, for any convex function f satisfying the smoothness constraints,

$$\mathbb{P}\left[D_f(\hat{p}\|p) \geq \epsilon\right] \leq (n+1)^{|\mathcal{Z}|} \exp\left(-n\epsilon/C\right), \tag{5.14}$$

where $C < \infty$ is a constant such that $x f''(x) < C$.

Proof. The results follow from Pinsker's inequality (5.11), Thm. 15, and Sanov's theorem. □

These inequalities and relationships among the f-divergences are used to design universal clustering algorithms and prove their consistency.

5.2.2 Universal Distance Clustering Algorithm

Responses to objects of the same class are identical in distribution. Thus, we perform universal clustering, $\Phi_{\text{temp}}(\mathbf{Y})$, according to Alg. 5.

The algorithm first computes the empirical distributions corresponding to the responses for each object. Then, the plug-in estimates of the f-divergences between these empirical distributions are used to generate a weighted graph G where the nodes are the objects and the edges are weighed by the distance between them. Then the clusters are identified as the maximal cliques in the graph obtained by thresholding the f-divergence between the corresponding empirical distributions. The functioning of the algorithm is depicted in Fig. 5.3.

When the empirical distributions are sufficiently close to the true distributions, Φ_{temp} outputs P^* as highlighted by the following lemma.

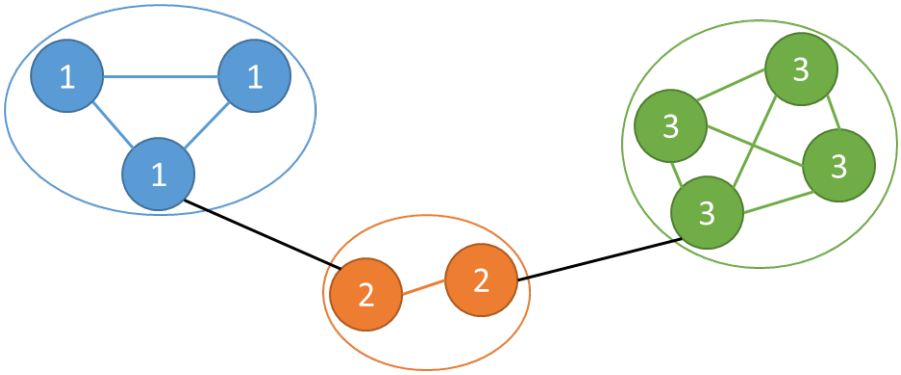


Figure 5.3: Distance-based clustering: The graph is obtained by thresholding the f-divergences of empirical distributions of responses to objects. The clustering is then done by identifying the maximal clusters in the thresholded graph.

Lemma 24. *For $f(x) = |x-1|$ and $\gamma_n = c_1 n^{-\alpha}$, $\alpha \in [0, 1/2]$ let $\gamma_n < \theta_d/2$. For any other convex function f satisfying the smoothness constraints let*

$$\gamma_n = c_1 n^{-\beta} < \frac{\kappa}{2L}\theta_d^2 + 2\frac{\kappa}{L^2}\left(1 - \sqrt{1 + \frac{L\theta_d^2}{2}}\right),$$

where $\kappa = 2c \log_2 e$ and the function f is L-Lipschitz.

Define the ball of radius ρ centered at p as

$$B_f(p, \rho) = \{q : D_f(p\|q) \le \rho\}.$$

If for all $i \in [\ell]$, the empirical distribution of responses $q_i \in B_f(Q_{t_i}, \gamma_n/2)$ then, $\Phi_{temp}(\mathbf{Y}) = P^*(t^\ell)$, the correct clustering of the set of objects.

Proof. Let us first consider $f(x) = |x-1|$. Since $q_i \in B_f(Q_{t_i}, \gamma_n/2)$ for all $i \in [\ell]$, we have

$$\max_{\{i,j \in [\tau]: T_i = T_j\}} d_{\mathrm{TV}}(q_i\|q_j) \le \gamma_n \le \frac{\theta_d}{2} \le \min_{\{i,j \in [\tau]: T_i \neq T_j\}} d_{\mathrm{TV}}(q_i\|q_j).$$

Let $C_i = \{j \in [\ell] : T_j = i\}$, for any $i \in [\tau]$. Then, for $j, k \in C_i$, $D_f(q_j, q_k) < \gamma_n$ and so $(j,k) \in E$. Thus, C_i is a clique of G.

Further, this observation also implies that for any $i \in C_t, j \in C_{t'}, t \neq t'$, $D_f(q_i, q_j) > \gamma_n$ and so $(i,j) \notin E$. Thus, any set $C \subseteq [\ell]$ such that there exist $i, j \in C$, with $T_i \neq T_j$, is not a clique in G. Thus C_i is a maximal clique in G for all $i \in [\tau]$. Thus $\Phi_{\mathrm{temp}}(\mathbf{Y}) = \{C_i : i \in [\tau]\} = P^*(t^\ell)$, the correct partition.

For the second part of the lemma, from (5.12), we note that the condition on γ_n guarantees

$$\max_{\{i,j\in[\tau]:T_i=T_j\}} D_f(q_i\|q_j) \leq \gamma_n < \min_{\{i,j\in[\tau]:T_i\neq T_j\}} D_f(q_i\|q_j).$$

Thus the result follows from a very similar argument. □

Using this result, we derive the sample complexity of the algorithm.

Theorem 17. *If $f(x) = |x-1|$, then for any $\alpha \in (0, 1/2)$ and constant c_1, for*

$$n \gtrsim \max\left\{\left(\frac{2c_1}{\theta_d}\right)^{1/\alpha}, \left(\frac{4\log \ell}{c_0 c_1^2}\right)^{1/(1-2\alpha)}\right\}, \tag{5.15}$$

sufficiently large, $\Phi_{temp}(\cdot)$ achieves arbitrarily low clustering error probability. For fixed ℓ and θ_d, it is universally asymptotically consistent.

For any other convex, normalized function f satisfying the smoothness constraints, for any $\beta \in (0,1)$ and constant c_1, for

$$n \gtrsim \max\left\{\left(\frac{2c_1 L^2}{\kappa}\mu_{\theta_d}\right)^{1/\beta}, \left(\frac{C\log l}{c_1}\right)^{1/(1-\beta)}\right\}, \tag{5.16}$$

where

$$\mu_{\theta_d} = \left(L\theta_d^2 + 4\kappa\left(1 - \sqrt{1 + \frac{L\theta_d^2}{2}}\right)^{-1}\right),$$

with sufficiently large constant c_1, $\Phi_{temp}(\cdot)$ achieves arbitrarily low clustering error probability. For fixed ℓ and θ_d, it is universally asymptotically consistent.

Proof. For $f(x) = |x-1|$ and $n \geq \left(\frac{2c_1}{\theta_d}\right)^{1/\alpha}$, $\gamma_n \leq \theta_d/4$. Thus, when $f(x) = |x-1|$, we can bound the error probability as follows

$$\begin{aligned} P_e(\Phi_{\text{temp}}) &\leq \mathbb{E}_{P_T^{\otimes l}}\left[\mathbb{P}\left[\exists i \in [l] : q_i \notin B(Q_{t_i}, \gamma_n/2)\right]\right] \\ &\leq \ell\left[(n+1)^{|\mathcal{Y}|}\exp\left(-c_0 n\frac{\gamma_n^2}{4}\right)\right] \qquad (5.17) \\ &= \exp\left(\left(\log\ell + (\tau+1)\log(n+1) - \frac{c_0 c_1^2}{4}n^{1-2\alpha}\right)\right), \end{aligned}$$

where (5.17) follows from the union bound and Lem. 23. Thus, the cost conditions given in (5.15) and asymptotic consistency follow.

Using a similar argument and Lem. 24, we obtain (5.16). □

Corollary 6. *Given $\mathcal{T} = [\tau]$ with $\tau < \infty$ a constant:*

1. *for a constant $\theta_d > 0$, $N^*_{temp}(\epsilon) = O\left((\log \ell)^{1/(1-\beta)}\right)$ and taking $\beta \to 0$, we observe that $N^*_{temp}(\epsilon) = O(\log \ell)$ for any of the similarity metrics;*

2. *for a constant ℓ, for $f(x) = |x-1|$, $N^*_{temp}(\epsilon) = O\left(\theta_d^{-1/\alpha}\right)$ and taking $\alpha \to 1/2$, $N^*_{temp}(\epsilon) = O(\theta_d^{-2})$. On the other hand, for other convex functions f satisfying the smoothness constraints, $N^*_{temp}(\epsilon) = O(\theta_d^{-1/\beta})$ and specifically, taking $\beta \to 1$, $N^*_{temp}(\epsilon) = O(\theta_d^{-1})$.*

Proof. The results follow directly from Thm. 17. □

Note that this result, analogous to Thms. 9 and 10, indicates that $N^*_{\text{temp}} = O\left(\frac{\log \ell}{\theta_d^2}\right)$. Having obtained sufficient conditions on the sample complexity by defining a simple distance-based clustering algorithm, we now study the necessary conditions on the sample complexity of the clustering problem.

5.2.3 Lower Bound on Sample Complexity

Here we will establish lower bounds on the sample complexity for universal clustering.

Theorem 18. *Let $\min_{i,j\in[\tau]} d_{\text{TV}}(Q_i \| Q_j) = \theta_d$. Then the sample complexity of universal clustering satisfies*

$$N^*_{temp} = \Omega\left(\frac{\log \ell - \log \epsilon}{\theta_d^2}\right).$$

Proof. Consider the prior distribution P_T such that $P_T(1) = P_T(2) = 0.5$. Let ψ_{ij} be the binary hypothesis testing problem given by:

$$\psi_{ij} : \begin{cases} H_0 : T_i = T_j \\ H_1 : T_i \neq T_j \end{cases} . \tag{5.18}$$

There exists $\binom{\ell}{2}$ such binary hypothesis tests. Choose a set of tests $\tilde{\psi} \subset \{\psi_{ij} : i, j \in [\ell], i \neq j\}$ of cardinality $\ell/2$ such that no two tests in the set share a common object. This indicates that the binary hypothesis tests are independent of each other owing to the independence across objects.

Let Φ be a decoder for the clustering problem. Then a correct solution to the clustering problem implies a correct solution to ψ_{ij}, for all $i, j \in [\ell], i \neq j$. This implies that an instance

of correct clustering translates to correct decisions in all tests in $\tilde{\psi}$. Thus,

$$P_e(\Phi) \geq 1 - \prod_{\{i,j\in[\ell]:\psi_{ij}\in\tilde{\psi}\}} (1 - \mathbb{P}\left[\text{error in } \psi_{ij}\right]) \tag{5.19}$$

$$\geq 1 - \prod_{\{i,j\in[\ell]:\psi_{ij}\in\tilde{\psi}\}} \left(1 - \frac{1}{4}\exp\left(-2nB_{ij}\right)\right) \tag{5.20}$$

$$\geq 1 - \left(1 - \frac{1}{4}\exp\left(-2nB_{\max}\right)\right)^{\lfloor \ell/2 \rfloor} \tag{5.21}$$

$$= \frac{\lfloor \ell/2 \rfloor}{4}\exp\left(-2nB_{\max}\right) + o\left(\exp\left(-4nB_{\max}\right)\right), \tag{5.22}$$

where (5.19) follows from the independence of the binary tests and (5.20) follows from the Kailath lower bound [70]. Here B_{ij} is the Bhattacharyya distance corresponding to the hypotheses of the test ψ_{ij} and considering non-triviality, there exists a test such that $B_{ij} > 0$. Thus bounding from below by the test with maximum distance (5.21), $B_{\max} = \max_{\{i,j\in[\ell]:\psi_{ij}\in\tilde{\psi}\}} B_{ij} > 0$ and using the binomial expansion, we obtain (5.22).

Now, using Jensen's inequality, we have $B_{ij} \leq \frac{1}{2}\left(D(Q_1\|Q_2) + D(Q_2\|Q_1)\right)$. This follows from the definition of the binary hypotheses tests and the independence of samples.

From Pinsker's inequality and reverse Pinsker's inequality [149], we have

$$(2\log_2 e)d_{\mathrm{TV}}^2(P\|Q) \leq D(P\|Q) \leq \left(\frac{4\log_2 e}{Q_{\min}}\right) d_{\mathrm{TV}}^2(P\|Q),$$

where $Q_{\min} = \min_{x\in\mathrm{supp}(P)} Q(x)$ and $\mathrm{supp}(\cdot)$ is the support of the distribution. Since we are concerned with the sample complexity in the worst case when $\delta(P,Q) \to 0$, it suffices to consider $Q_{\min} > 0$. Thus, the above bounds indicate that

$$D(Q_1\|Q_2) \asymp D(Q_2\|Q_1) \asymp \theta_d^2,$$

where it is said $g \asymp h$, if there exists constants $a, b > 0$ such that $ah \leq g \leq bh$.

Thus, we have

$$P_e(\Phi) \geq \frac{1}{8}\exp\left(\log\ell - n(c\theta_d^2)\right),$$

where c is the constant scaling based on Pinsker's and reverse Pinsker's inequalities.

From this we observe that $N_{\mathrm{temp}}^* = \Omega\left(\frac{\log\ell - \log\epsilon}{\theta_d^2}\right)$. □

Corollary 7. *Let $f(\cdot)$ be a convex function satisfying the smoothness constraints. Further, let $\min_{i,j\in[\tau]} D_f(Q_i\|Q_j) = \theta_d$. Then for constant ℓ,*

$$N_{temp}^* = \Omega(\theta_d^{-1}).$$

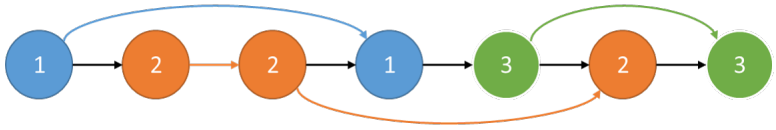


Figure 5.4: Bayesian network model of responses to a set of 7 objects chosen from a set of 3 types. We observe that the most recent response and the response to the most recent object of the same type influence every response.

Proof. The result follows directly from (5.12) since

$$\min_{i,j\in[\tau]} d_{\mathrm{TV}}(Q_i \| Q_j) \lesssim \sqrt{\theta_d}.$$

□

Remark 11. *It is worth noting that the quantity θ_d^2 is equivalent in definition to the crowd quality defined in [134] and matches the lower bound obtained on the cost for binary classification using crowd workers biased toward giving the right label. The factor of* $\log \ell$ *in the cost per object arises since error probability studied here is the block (blocklength ℓ) error probability whereas [134] studies the average symbol (classification) error probability.*

Note that Φ_{temp} achieves the lower bound up to constant factor in sample complexity, i.e., our simple clustering algorithm is asymptotically order optimal in the number of objects to be clustered and the distance quality. We will now study the context of clustering using responses of long-term workers.

5.3 Workers with Memory

Now consider long-term workers with Markov memory. Recall that in Sec. 5.1 we defined the structure of the stochastic kernel that determines the responses of workers with memory. In particular, we considered a Markov memory structure (5.2). This structure is represented in the Bayesian network depicted in Fig. 5.4.

Specifically, we assume that the response $Y_{i,j}$ to an object X_i by worker j is dependent on the response to the most recent object, X_{i-1}, and the response to the most recent object of the same class. This set of indices for any object X_i is given by $\mathcal{N}_i$.

Let $\mathcal{Q}$ be the set of such Markov-structured distributions representing the worker pool. Then, the conditional distribution of the response vector for a random worker is

$$\mathbb{P}\left[Y^\ell = y^\ell \middle| T^\ell = t^\ell\right] = \mathbb{E}\left[Q^W\left(y^\ell \middle| t^\ell\right)\right] = \bar{Q}\left(y^\ell \middle| t^\ell\right),$$

where the expectation is taken over $W \in \mathcal{W}$.

We know that a sufficient statistic of the worker responses is the empirical distribution. Asymptotically, we know that the empirical pmf converges to $\bar{Q}$ by the strong law of large numbers. It thus suffices to study the decoder with regard to this characteristic worker response pmf that retains the assumed memory properties.

Throughout the section, for any $i \in [\ell]$, denote by $\tilde{\imath}$ the index such that $\tilde{\imath} \in \mathcal{N}_i, T_i = T_{\tilde{\imath}}$.

Definition 25 (Memory Quality)**.** The memory quality *in a given pool of long-term workers is*

$$\theta_m = \frac{1}{2}\left(\min_{i\in[\ell]} I(Y_i; Y_{\mathcal{N}_i}) - \max_{i\in[\ell], j<i, j\notin\mathcal{N}_i} I(Y_i; Y_{i-1}, Y_j)\right). \tag{5.23}$$

That is, the memory quality is defined by the difference between the information provided by the neighbors of the object and that provided by two other objects.

Since the crowd responses are defined by conditional independence across objects, we quantify the quality of the crowd in terms of the amount of memory retained by the workers. The above definition can be equivalently viewed as the *task difficulty* for a given pool of long-term crowd workers.

This informational definition finds operational significance in the coding theorems, Thm. 20 and 21.

5.3.1 Information Clustering Using Neighbors

From the model of worker responses, we know that identifying the parents of each node is critical to cluster.

Lemma 25. *Let $G = (V, E)$ be the Bayesian network representation of the worker responses. Let $\nu_i = \{j \in V : (j, i) \in E\}$. If $|\nu_i| = 1$, then either $T_{i-1} = T_i$ or $T_j \neq T_i$ for all $j < i$. If $|\nu_i| > 1$, $\nu_i \backslash \{i-1\}$ is in the same cluster as i.*

Proof. The results follow directly from the model definition. □

We use the data processing inequality to obtain the following property that motivates the algorithm definition.

Lemma 26. *Let $C = \{c_1, \ldots, c_k\}$ and without loss of generality, let $c_1 < c_2 < \cdots < c_k$. Given (5.2), $C \in P^*$ if and only if for all $j \in [k]$,*

$$c_j = \arg\max_{i<c_{j+1}} I(Y_{c_{j+1}}; Y_i, Y_{c_{j+1}-1}).$$

Proof. Let $i, j \in [\ell]$ such that $j < i$. Then, the result follows from the data processing inequality:

$$\begin{aligned} I(Y_i; Y_{i-1}, Y_{\tilde{\imath}}, Y_j) &= I(Y_i; Y_{\mathcal{N}_i}) + I(Y_i; Y_j | Y_{\mathcal{N}_i}) \\ &= I(Y_i; Y_{i-1}, Y_j) + I(Y_i; Y_{\tilde{\imath}} | Y_{i-1}, Y_j). \end{aligned}$$

For the given model, $I(Y_i; Y_j | Y_{\mathcal{N}_i}) = 0$. In the non-trivial problem it is natural that $I(Y_i; Y_{\tilde{\imath}} | Y_{i-1}, Y_j) > 0$. That is, the most recent object of the same class has residual information, given any other pair from the past. This in turn implies that for all $j < i, j \neq \tilde{\imath}$,

$$I(Y_i; Y_{i-1}, Y_j) < I(Y_i; Y_{\mathcal{N}_i}).$$

The result thus follows. □

It is evident that the partition can be obtained through a careful elimination process using mutual information values. Maximum likelihood estimates of the mutual information can be obtained from the samples using and asymptotically consistent estimators. Note that such estimators converge exponentially; convergence rates are detailed in Appendix B.1.

5.3.2 Information Clustering Algorithm

We now describe the clustering algorithm in two stages. First we describe an algorithm that, given the set of objects and mutual information values, outputs a partition that is possibly denser than the correct partition. We then describe an algorithm that overcomes this shortcoming by identifying sub-clusters within the identified clusters recursively. We then show correctness of the algorithm and prove it is asymptotically consistent when the ML estimates of mutual information are used.

From the directed acyclic graph (Bayesian network) corresponding to the given set of objects, we know that identifying the parents of each node is sufficient to identify clusters such that objects of the same type are in the same cluster. From Lem. 26, for any $i \in [\ell]$, $I(Y_i; Y_{i-1}, Y_j) < I(Y_i; Y_{\mathcal{N}_i})$. Thus identifying the parents of node i is equivalent to solving

$$\eta_i = \arg \max_{j \leq i-1} I(Y_i; Y_{i-1}, Y_j).$$

Using this feature we design Algorithm 6, $\Phi_{\text{info}}(I)$.

The algorithm outputs the partition of a set of objects when the corresponding mutual information values are given as input. The algorithm starts from the last object and iterates

Algorithm 6 Clustering given MI, $(P) = \Phi_{\text{info}}(I)$

$F(i) \leftarrow 0$ for all $i \in [m]$
$P \leftarrow \emptyset$
$M_c \leftarrow 0$
$\gamma_n = c_1 n^{-\alpha}$, where c_1 is a constant and $\alpha \in (0, 1/2)$
for $i = m$ **to** 1 **do**
  if $F(i) = 0$ **then**
    $F(i) \to 1$
    $M_c \leftarrow M_c + 1$, $C(i) \leftarrow M_c$
  end if
  $I_{\max,i} \leftarrow \max_{k \le i-1} I(Y_i; Y_{i-1}, Y_k)$
  $\eta_i = \max\{j < i : I(Y_i; Y_{i-1}, Y_j) \ge I_{\max,i} - \gamma_n\}$
  if $F(\eta_i) = 0$ **then**
    $C(\eta_i) = C(i), F(\eta_i) = 1$
  end if
end for
$P = \{\{i : C(i) = k\} : k \in [M_c]\}$

backward while finding the parents of each node. Upon identification, the parent is added to the same cluster as the object.

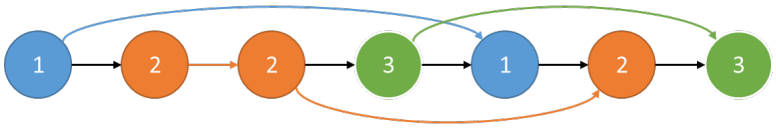
Theorem 19. *Given a set of objects T^ℓ and the corresponding set of mutual informations $\{I(Y_i; Y_{i-1}, Y_j) : i \in [\ell], j < i\}$, the output of Alg. 6 satisfies $P = \Phi_{info}(I) \succeq P^*$.*

Proof. From Lem. 26, $I(Y_i; Y_{i-1}, Y_j) \le I(Y_i; Y_{\mathcal{N}_i})$ for all $j < i$ with equality if and only if $j \in \mathcal{N}_i$. Thus, the parents of every node $i \in [\ell]$ in the Bayesian network can be determined, given the mutual information values.

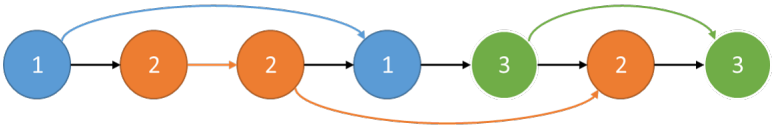
For $P = \Phi_{\text{info}}(I)$, for every object $t \in \mathcal{T}$, there exists $C \in P$ such that $\{i \in [\ell] : T_i = t\} \subseteq C$. Hence the result follows. □

Theorem 19 indicates that, given the mutual information values, the objects of the same type are clustered together. The maximizer η_i in Alg. 6 is clustered only if it is not assigned a cluster before the iteration. Thus object i is not paired with $i - 1$ unless it has not been assigned a cluster. This particular scenario is depicted in the Bayesian network in Fig. 5.5a.

However, the algorithm fails in a specific scenario. When there exists clusters C_1 and C_2 such that $i < j$ for every $i \in C_1, j \in C_2$, and, $\max\{i \in C_1\} = \min\{j \in C_2\} - 1$, then the resulting partition consists of the single cluster $C_1 \cup C_2$ rather than the two individual clusters. This is because objects of C_1 have not yet been encountered and thus the immediate neighbor of the first object of C_2 is clustered along with C_2 due to the Markov memory structure. This particular scenario is depicted in Fig. 5.5b.



(a) Alg. 6 outputs $P = \{\{1,5\},\{2,3,6\},\{4,7\}\} = P^*$. Here, object 4 of type 3 is not clustered with object 3 of type 2 as it is already assigned a cluster.



(b) Alg. 6 outputs $P = \{\{1,4,5,7\},\{2,3,6\}\} \succeq P^*$. Here object 5 is clustered with 4 as objects of Type 1 are not encountered before.

Figure 5.5: Functioning and shortcoming of Alg. 6.

Algorithm 7 Clustering with memory, $P = \Phi_{\text{mem}}(T^\ell)$

Choose constant $k = \lceil \frac{-\log \epsilon}{(\log \ell - 2\log \tau)} \rceil$
for $i = 1$ **to** k **do**
  Choose a permutation $\xi([\ell])$ uniformly at random
  Collect responses $\mathbf{Y}^{(i)}$ for the sequence $T^{\xi([\ell])}$
  $I \leftarrow \{\hat{I}(Y_j; Y_{j-1}, Y_k) : k < j, j \in [\ell]\}, P_i = \Phi_{\text{info}}(I)$
end for
Choose finest partition P such that $P \preceq P_i$ for all $i \in [k]$

Such shortcomings of the algorithm however happen with low probability when $\ell \gg \tau$. Thus, if the finest partition in a collection of permutations of a given set of objects, chosen uniformly at random, is obtained using Φ_{info}, then with high probability, the correct partition is obtained. Thus the algorithm can be refined by repeating the clustering process as in Alg. 7 for $P_e(\Phi_{\text{mem}}) \leq 2\epsilon$.

That is Φ_{mem} repeats Φ_{info} for different permutations of the objects and selects the finest partition. For each permutation of the objects, n responses are obtained for each object from the workers. Thus, the overall number of samples per object obtained is kn.

Theorem 20. *Let T^ℓ be the set of objects, $\ell \geq \tau^2$, and, let $\mathbf{Y}$ be the set of responses. Let $k \geq \lceil \frac{-\log \epsilon}{(\log \ell - 2\log \tau)} \rceil$ be the number of permutations chosen in Alg. 7. Then, for*

$$n \gtrsim \max\left\{ (\log \ell - \log \epsilon)^{\frac{1}{(1-2\alpha-\beta)}}, (\log \ell - \log \epsilon)^{\frac{1}{(1-4\alpha)}}, \theta_m^{\frac{-1}{\alpha}} \right\}, \tag{5.24}$$

for $0 < \alpha < 1/2$ and $0 < \beta < 1$ such that $\log_2^2 n \leq n^\beta$, $P_e(\Phi_{mem}) \leq 2\epsilon$, for any $\epsilon > 0$. Further, for constant ℓ and θ_m, the algorithm is asymptotically consistent.

Proof. We first observe that when $|\hat{I}(Y_i; Y_{i-1}, Y_j) - I(Y_i; Y_{i-1}, Y_j)| < \theta_m$ for all $i \in [\ell], j < i$,

$\Phi_{\text{info}}(\hat{I}) = \Phi_{\text{info}}(I)$. That is, when the empirical mutual information values do not deviate from the actual values significantly, the clustering algorithm works without error.

For $n \geq (c_1/\theta_m)^{1/\alpha}$, $\gamma_n < \theta_m$. Let $I_{ij} = I(Y_i; Y_{i-1}, Y_j)$. Then,

$$\begin{aligned}
\mathbb{P}\left[\Phi_{\text{info}}(\hat{I}) \neq \Phi_{\text{info}}(I)\right] &\leq \sum_{i\in\ell, j<i} \mathbb{P}\left[|\hat{I}_{ij} - I_{ij}| > \gamma_n\right] && (5.25)\\
&\leq \binom{\ell}{2} \exp\left(\frac{-n\gamma_n^2}{18\log_2^2 n} + o(1)\right) && (5.26)\\
&\leq \exp\left(2\log\ell - \frac{c_1^2}{18} n^{(1-\nu)} + o(1)\right),
\end{aligned}$$

implying asymptotic consistency. Here $\nu = 2\alpha + \beta$, (5.25) follows from the union bound, and (5.26) follows from Lem. 39.

Additionally, we note that

$$\begin{aligned}
\mathbb{P}\left[\Phi_{\text{info}}(\hat{I}) \neq \Phi_{\text{info}}(I)\right] &\leq \sum_{i\in\ell, j<i} \mathbb{P}\left[|\hat{I}_{ij} - I_{ij}| > \gamma_n\right]\\
&\leq 3\binom{\ell}{2} (n+1)^{\tau^2} \exp\left(-\tilde{c} n \gamma_n^4\right) && (5.27)\\
&\leq \exp\left(2\log\ell - \tilde{c} c_1^4 n^{(1-4\alpha)} + o(1)\right),
\end{aligned}$$

where (5.27) follows from (B.6).

To obtain the correct partition, we use the responses generated for several uniformly random permutations of the given set of objects and select the finest partition. The correct partition may not be recovered when there exists $t, t' \in \mathcal{T}$, such that $\max\{i \in [\ell] : T_i = t\} = \min\{i \in [\ell] : T_i = t'\} - 1$ in every chosen partition.

Let $K_t = |\{i \in [\ell] : T_i = t\}|$, $t \in [\tau]$ and let $M_t = \max\{i \in [\ell] : T_i = t\}$ and $m_t = \min\{i \in [\ell] : T_i = t\}$. Thus, the probability that $P = \Phi_{\text{mem}}(T^\ell) \succ P^*$ is bounded as:

$$\begin{aligned}
\mathbb{P}\left[P \succ P^*\right] &= \mathbb{P}\left[\exists t \neq t' : M_t = m_{t'} - 1\right]^k\\
&\leq \left(\sum_{t,t'\in\mathcal{T}, t\neq t'} \mathbb{P}\left[M_t = m_{t'} - 1\right]\right)^k.
\end{aligned}$$

The total number of possible permutations of the given set of objects is given by

$$\kappa_{\text{permut}} = \frac{\ell!}{\prod_{\tilde{t}\in\mathcal{T}} K_{\tilde{t}}!}.$$

The number of sequences such that $M_t = m_{t'} - 1$ can be determined by choosing $K_t + K_{t'} - 1$

locations out of $\ell - 1$ locations to fill the objects of type t and t' and permute over the other objects. Thus,

$$\kappa_{\mathrm{t,t'}} \leq \binom{\ell - 1}{K_t + K_{t'} - 1} \frac{(\ell - K_t - K_{t'})!}{\prod_{\tilde{t} \in \mathcal{T} \setminus \{t,t'\}} K_{\tilde{t}}!}.$$

Since the permutations are chosen uniformly at random,

$$\begin{aligned} \mathbb{P}\left[m_{t'} - M_t = 1\right] &\leq \frac{\kappa_{t,t'}}{\kappa_{\text{permut}}} \\ &= \frac{1}{\ell} \frac{K_t! K_{t'}!}{(K_t + K_{t'} - 1)!} \leq \frac{1}{\ell}, \end{aligned} \tag{5.28}$$

where (5.28) follows from the fact that

$$\frac{K_t! K_{t'}!}{(K_t + K_{t'} - 1)!} \leq 1, \tag{5.29}$$

as $K_t, K_{t'} \geq 1$.

Thus, we have

$$\mathbb{P}\left[P \succ P^*\right] \leq \left(\binom{\tau}{2} \frac{2}{\ell}\right)^k \leq \exp\left(-k(\log \ell - 2 \log \tau)\right). \tag{5.30}$$

Thus, for $k \geq \lceil \frac{-\log \epsilon}{(\log \ell - 2 \log \tau)} \rceil$, $\mathbb{P}\left[P \succ P^*\right] \leq \epsilon$.

Now we prove consistency of Alg. 7. Using the union bound, we observe that the probability of error is bounded as

$$\begin{aligned} P_e &\leq \mathbb{P}\left[P \succ P^*\right] + k\mathbb{P}\left[\Phi_{\text{info}}(\hat{I}) \neq \Phi_{\text{info}}(I)\right] \\ &\leq \exp\left(-k(\log \ell - 2 \log \tau + \log 2)\right) \\ &\quad + \exp\left(2 \log \ell - \max\left\{\frac{c_1^2}{18} n^{(1-\nu)}, \tilde{c} c_1^4 n^{(1-4\alpha)}\right\} + o(1)\right) \qquad (5.31) \\ &\leq 2\epsilon \end{aligned}$$

for a large enough n. Here (5.31) follows from the two concentration bounds on empirical mutual information described in Appendix B.1.

Thus, for any $\epsilon > 0$, there exists n, k sufficiently large, such that $P_e < \epsilon$. Hence the consistency result follows. The sample complexity is obtained from the error exponent in (5.31). □

We observe from the proof that there is a trade-off between the values of k and n needed to achieve a certain level of accuracy. In particular, we observe that when ℓ is large, it suffices

to consider a small number of permutations of the set of objects, while each permutation requires a larger number of samples. On the other hand, when ℓ is relatively small, one needs a large number of permutations while each permutation requires far fewer samples.

We restrict focus to the case where $\ell \gg \tau^2$ and under this scenario find the following result on sample complexity.

Corollary 8. *Given $\mathcal{T} = [\tau]$ with $\tau < \infty$ a constant and $\ell \gg \tau^2$,*

$$N^*_{mem}(\epsilon) = O\left(\frac{(\log \ell - \log \epsilon)^{\min\{1/(1-2\alpha-\beta),1/(1-4\alpha)\}}}{\theta_m^{(2/(1-\beta))}}\right).$$

Proof. Using Thm. 20 and the fact that the total number of samples used is kn (since clustering with n samples is done k times) per object, we obtain the result. □

For large ℓ, we can thus observe that $N^*_{\text{mem}}(\epsilon) = O\left(\frac{\log \ell}{\theta_m^2}\right)$. Note that under the Markov memory model for long-time workers, the sufficient number of samples per object is the same in order as for temporary workers.

5.3.3 Lower Bound on Sample Complexity

We now provide matching lower bounds by studying the probability of error of a problem which is a reduction of the universal clustering problem.

Theorem 21. *The sample complexity of universal clustering using workers with memory satisfies*

1. *for a fixed $\theta_m > 0$, $N^*_{mem} = \Omega(\log \ell)$, and*
2. *for a fixed $\ell < \infty$, $N^*_{mem} = \Omega(\theta_m^{-1})$.*

Proof. Choose a prior, parametrized by the size of the problem ℓ as $P_T(1) = 1 - \frac{1}{\ell}, P_T(2) = \frac{1}{\ell}$ such that $\frac{1}{\ell} = \frac{1}{\ell}$. Let $\mathcal{E}$ be the set of all vectors of objects with at most one object of type 2. Then,

$$\mathbb{P}\left[T^\ell \in \mathcal{E}\right] = \left(2 - \frac{1}{\ell}\right)\left(1 - \frac{1}{\ell}\right)^{\ell-1} \overset{\ell\to\infty}{\longrightarrow} 1.$$

In particular, we note that $\mathbb{P}\left[T^\ell \in \mathcal{E}\right]$ is an increasing function of ℓ and is at least $\frac{1}{2}$ for any $\ell > 3$.

For a given constant θ_m, consider the special case of the problem where $\mathcal{N}_i = \{\tilde{i}\}$. That is, consider the problem where any two objects are dependent if and only if they are of the

same type. Clearly, any algorithm that solves the universal clustering with memory problem solves this simplified problem as well. Thus, following the convention established, we have

$$\begin{cases} I(Y_i; Y_{\tilde{i}}) \geq 2\theta_m, & \text{for all } i \in [\ell] \\ I(Y_i; Y_j) = 0, & \text{for all } i, j \in [\ell], T_i \neq T_j \end{cases}.$$

Define

$$W = [W_{ij}]_{1 \leq i,j \leq 2} = \begin{bmatrix} \frac{1}{2} + \epsilon & \frac{1}{2} - \epsilon \\ \frac{1}{2} - \epsilon & \frac{1}{2} + \epsilon \end{bmatrix}.$$

Consider the scenario where worker responses are inertial over time and characterized as:

$$\mathbb{P}\left[Y_i = k | Y_{\tilde{i}} = j\right] = W_{kj},$$

for any $k, j \in \{0, 1\}$ and $i \in [\ell]$. Additionally, assume that the marginals of the responses are uniform (that is, the response to the first object of each type is distributed as Bern(1/2)).

The information constraint implies

$$\frac{1}{2} - h^{-1}(1 - 2\theta_m) \leq \epsilon < \frac{1}{2},$$

where $h(\cdot)$ is the binary entropy function and $h^{-1}(\cdot)$ is its inverse. Let $\epsilon = \frac{1}{2} - h^{-1}(1 - 2\theta_m)$.

From the definition of the error probability,

$$P_e \geq \frac{1}{2}\mathbb{P}\left[\hat{P} \neq P^* | T^\ell \in \mathcal{E}\right]. \tag{5.32}$$

Now consider the set $\mathcal{E}$ of vectors. Identifying the correct partition for a vector of objects from this space is equivalent to identifying the objects. Thus consider the $(\ell + 1)$-ary hypothesis testing problem defined by

$$\begin{cases} H_0 : T_j = 1, \text{ for all } j \in [\ell] \\ H_i : T_i = 2, T_j = 1, \text{ for all } j \neq i \end{cases}. \tag{5.33}$$

We seek to compute the average error probability of (5.33) corresponding to the prior P_T. Due to symmetry, note that the optimal decoder accrues the same probability of error under

H_i for any $i > 0$. Thus

$$\mathbb{P}\left[\text{error in (5.33)}\right] = \mathbb{P}\left[H_0\right]\mathbb{P}\left[\text{error in (5.33)}|H_0\right] + \mathbb{P}\left[\text{error in (5.33)}|H_1\right]\left(\sum_{i\in[\ell]}\mathbb{P}\left[H_i\right]\right).$$

Now, note that

$$\sum_{i\in[\ell]}\mathbb{P}\left[H_i\right] = \left(1-\frac{1}{\ell}\right)^{\ell-1}, \quad \mathbb{P}\left[H_0\right] = \left(1-\frac{1}{\ell}\right)^{\ell}.$$

Thus, for $\ell > 1$,

$$\frac{1}{2}\mathbb{P}\left[\{H_i : i > 0\}\right] \le \mathbb{P}\left[H_0\right] \le \mathbb{P}\left[\{H_i : i > 0\}\right].$$

Thus, $\mathbb{P}\left[\{H_i : i > 0\}\right] \asymp \mathbb{P}\left[H_0\right]$. This indicates that the average error probability is lower-bounded by a constant factor of the minimax error probability lower bound for (5.33).

Let Q_i be the distribution of the set of responses corresponding to the hypotheses defined in (5.33):

$$Q_i(Y^\ell) = \begin{cases} \frac{1}{2}\prod_{k=2}^{\ell} W_{Y_kY_{k-1}}, & i = 0 \\ \frac{1}{4}\prod_{j=2}^{i-1} W_{Y_jY_{j-1}}\prod_{k=i+1}^{\ell} W_{Y_kY_{k-1}}, & i \in [\ell] \end{cases}. \tag{5.34}$$

Lemma 27. *For all ℓ, $D\left(Q_i\|Q_j\right) = O(1)$.*

Proof. See Appendix B.2. □

Having bounded the KL divergences between the hypotheses, we obtain a lower bound on the error probability of (5.33) using the generalized Fano inequality [150].

Let $\beta = \max_{i,j\in[\ell]\cup\{0\}, i\neq j} D(Q_i\|Q_j)$. The loss function considered here is the 0-1 loss. Hence,

$$\begin{aligned}\mathbb{P}\left[\hat{P} \neq P^*|T^\ell \in \mathcal{E}\right] &= \mathbb{P}\left[\text{error in problem (5.33)}\right] \\ &\asymp \max_{0\le i\le \ell}\mathbb{P}\left[\text{error in problem (5.33)}|H_i\right] \\ &\ge \frac{1}{2}\left(1-\frac{n\beta+\log 2}{\log(\ell+1)}\right).\end{aligned} \tag{5.35}$$

Hence, for a constant $\theta_m > 0$, the sample complexity of universal clustering satisfies

$$N^*_{\text{mem}} = \Omega(\log \ell).$$

Now, when ℓ is fixed, we seek to understand the sample complexity with respect to the memory quality of the crowd. To this end, we note that any consistent clustering algorithm is also consistent for the binary hypothesis test

$$\psi : \begin{cases} H_0 : I(Y_1; Y_2) = 0 \\ H_1 : I(Y_1; Y_2) \geq 2\theta_m \end{cases} . \tag{5.36}$$

That is, if Φ is a decoder for the universal clustering problem, then Φ also solves ψ.

Since the sufficient statistics for detection of the binary hypothesis testing above are the responses to T_1, T_2, it suffices to consider Y_1^n, Y_2^n. Let the prior here be $P_T(1) = P_T(2) = 1/2$. Let the corresponding distributions of worker responses be $p(Y_1^n, Y_2^n)$ and $q(Y_1^n, Y_2^n)$ under H_0 and H_1 respectively. Here,

$$p(y_1^n, y_2^n) = \frac{1}{2} \prod_{i=1}^{n} \mathbb{P}\left[y_{1,i}, y_{2,i} | T_1 = T_2 = 1\right] + \frac{1}{2} \prod_{i=1}^{n} \mathbb{P}\left[Y_{1,i}, Y_{2,i} | T_1 = T_2 = 2\right],$$

and

$$q(Y_1^n, Y_2^n) = \frac{1}{2} \prod_{i=1}^{n} \mathbb{P}\left[Y_{1,i} | T_1 = 1\right] \mathbb{P}\left[Y_{2,i} | T_1 = 2\right] + \frac{1}{2} \prod_{i=1}^{n} \mathbb{P}\left[Y_{1,i} | T_1 = 2\right] \mathbb{P}\left[Y_{2,i} | T_1 = 1\right].$$

Let $p_j^{(i)}(y) = \mathbb{P}\left[Y_i = y | T_i = j\right]$, $q_j^{(i)}(y) = \mathbb{P}\left[Y_i = y | T_i = j\right]$, under H_0 and H_1 respectively. Without loss of generality, we assume $I_p(Y_1, Y_2) = 2\theta_m$.

Since the distributions satisfy the information constraints, we have,

$$\begin{aligned} D(p\|q) &\leq \frac{n}{4} \left(4 I_p(Y_1; Y_2) + \sum_{i,j,k \in [2]} D(p_j^{(i)} \| q_k^{(i)}) \right) \\ &= 2n\theta_m, \end{aligned} \tag{5.37}$$

when the marginals under the two hypotheses are equal as was the case in the inertial worker response channel. Here (5.37) follows from convexity. Thus the minimum upper bound on the KL divergence between the hypotheses is $2\theta_m$. Since we consider the worst case with respect to θ_m, it suffices to consider this upper bound.

Thus,

$$P_e(\Phi) \geq \mathbb{P}\left[\text{error in } \psi\right] \geq \frac{1}{4}\exp\left(-2B(p,q)\right) \tag{5.38}$$

$$\geq \frac{1}{4}\exp\left(-2D(p\|q)\right) \geq \frac{1}{4}\exp\left(-4n\theta_m\right), \tag{5.39}$$

where (5.38) follows from the Kailath lower bound [70]. Then, using Jensen's inequality, we obtain (5.39). Thus $N^*_{\text{mem}} = \Omega\left(\theta_m^{-1}\right)$. □

We observe from Thm. 21 that the universal clustering algorithm is order optimal in terms of the number of objects, ℓ. However, there is a gap between the lower bound and the achievable cost in terms of the crowd quality θ_m. This gap is exactly the well-known gap for entropy estimation [73, Cor. 2].

5.3.4 Reductions to Other Clustering Algorithms

There exist several clustering paradigms based on mutual information. Here we describe two such algorithms and reductions of our model under which they are the same as Φ_{info}.

Recall the minimum partition information [112] based clustering described in Sec. 4.2. Consider the Markov memory model such that $\mathcal{N}_i = \{\tilde{\imath}\}$, that is, if Y_i is conditionally dependent on an object only if it is of the same type. Then, if $|P^*| > 1$, then

$$I(Y^\ell) = I_{P^*}(Y^\ell) = 0.$$

That is, the correct partition is the finest partition that minimizes the partition information. The following reduction indicates that minimizing the partition information is the same as our algorithm, and so [112] is equivalent to our approach.

First, we have

$$H(Y^\ell) = \sum_{i=1}^{\ell} H(Y_i|Y_{\tilde{\imath}}).$$

Similarly,

$$H(Y_C) = \sum_{i\in C} H(Y_i|Y_{j(i)}),$$

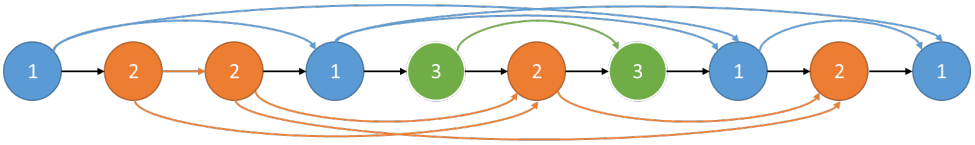


Figure 5.6: Bayesian network model of responses of to a set of 10 objects chosen from a set of 3 types with $\zeta = 2$. We observe that the most recent response and the response to the two most recent object of the same type influence every response.

where $j(i) = \max\{k < i : k \in C\}$. This implies that

$$I_P(Y^\ell) = \frac{1}{|P| - 1} \left(\sum_{i=1}^{\ell} (I(Y_i; Y_{\tilde{i}}) - I(Y_i; Y_{j(i)})) \right),$$

where $j(i) = \max\{k < i : i, k \in C\}$. This indicates that minimizing the partition information is equivalent to $\Phi_{\text{info}}(I)$.

Another information-based clustering strategy is the mutual information relevance network (MIRN) clustering [111]. Here, for a given threshold γ, the clustering strategy determines the connected components of $G = ([\ell], E)$ such that $(i, j) \in E \Leftrightarrow I(Z_i; Z_j) > \gamma$. For the Markov memory model, under the restriction that

$$\min_{\{i,j \in [\ell] : T_i = T_j\}} I(Y_i; Y_j) > \gamma \geq \max_{\{i,j \in [\ell] : T_i \neq T_j\}} I(Y_i; Y_j),$$

MIRN outputs the correct partition.

However, in the universal clustering scenario, the decoder is not aware of γ and thus, it may not be feasible to implement MIRN clustering optimally. Nevertheless. under the restriction that $\mathcal{N}_i = \{\tilde{i}\}$ and $\gamma = 0$, the MIRN clustering algorithm is equivalent to $\Phi_{\text{info}}(I)$.

5.3.5 Extended Worker Memory

While the model defined above considers the dependence of responses on just the most recent object of the same kind, our results hold for any fixed, finite-order Markov memory as well. In particular, consider the scenario where worker responses are dependent on the set $\mathcal{N}_i = \{i-1\} \cup \mathcal{M}_i$, where $\mathcal{M}_i \subseteq \{j < i : T_j = T_i\}$ such that it contains at most the most recent ζ indices of the same type of object. That is, the response to an object is dependent not only on the most recent response, but also a number of prior responses to objects of the same type as in Fig. 5.6.

Then the algorithm and sample complexity can be extended to this scenario. More specif-

Algorithm 8 Clustering under unified worker model, $\Phi_{\mathrm{u}}(T^{\ell})$

$P_{\mathrm{info}} \leftarrow \Phi_{\mathrm{mem}}(T^{\ell})$
for $C \in P_{\mathrm{info}}$ **do**
  $P_C \leftarrow \Phi_{\mathrm{temp}}(T^C)$
end for
$P \leftarrow \cup_{C \in P_{\mathrm{info}}} P_C$

ically, the parents of node i can be determined using

$$I(Y_i; Y_{\mathcal{N}_i}) > I(Y_i; Y_S),$$

for any index set $S \subseteq [i-1]$ such that $|S| \leq \zeta$. Then for any constant ζ and $n \gtrsim (\tau+1)^{2\zeta}$, the consistency of the algorithm holds. That is, as long as $\zeta = O(1)$, the sample complexity results follow.

5.4 Unified Worker Model

While we studied two distinct classes of worker models in temporary workers and workers with memory, these two scenarios are limiting cases of a unified worker model described here. After all, it is reasonable to characterize practical crowd worker decisions as influenced by both aspects—memory of individual responses and task difficulty with respect to objects.

For the unified model, we provide an achievable scheme that makes use of the algorithms defined earlier. Further, we prove consistency and order optimality of the scheme. As in Sec. 5.1, consider worker model (5.3), where each worker is characterized by a Markov memory model subject to fixed conditional marginal distributions. Worker quality in the unified worker model is a combination of the distance and memory quality parameters θ_d, θ_m.

5.4.1 Unified Clustering Algorithm

We now provide the universal clustering strategy for the unified worker model, Alg. 8, drawing on achievable schemes from before. First perform the memory-based clustering defined in Alg. 7; then for every cluster in the partition output by the algorithm, perform distance-based clustering. We now show the consistency of the algorithm.

Theorem 22. *Let T^ℓ be the set of objects and $\ell > \tau^2$. Then, for*

$$n \gtrsim \max\left\{(\log \ell - \log \epsilon)^{\frac{1}{(1-2\alpha-\beta)}}, \tag{5.40}$$

$$(\log \ell - \log \epsilon)^{\frac{1}{(1-4\alpha)}}, (\theta_m + \theta_d)^{\frac{-1}{\alpha}}\right\}, \tag{5.41}$$

for $0 < \alpha < 1/2$ and $0 < \beta < 1$, $P_e(\Phi_u) \leq 2\epsilon$, for any $\epsilon > 0$.

Proof. First we note that for $n \geq (c_1/(\theta_m + \theta_d/4))^{1/\alpha}$, $\gamma_n \leq \theta_d/4 + \theta_m$. Thus, at least one of $\gamma_n \leq \theta_d/4$ or $\gamma_n \leq \theta_m$ is true. This in turn indicates that at least one of Φ_{mem} or Φ_{temp} is consistent.

Next, from Theorem 19, we note that the output $P_{\text{info}} \succeq P^*$. Thus, subsequent clustering of the individual clusters is sufficient. This in turn indicates the correctness and asymptotic consistency of Alg. 8. □

We now observe that the sample complexity with respect to the number of objects to be clustered is still $O(\log \ell)$ while that with respect to the quality parameters is $O((\theta_m + \theta_d)^{-2})$.

Corollary 9. *Given $\mathcal{T} = [\tau]$ with $\tau < \infty$ a constant,*

1. *for a constant $\theta_m, \theta_d > 0$, $N_u^*(\epsilon) = O(\log \ell)$;*
2. *for a constant ℓ, $N_u^*(\epsilon) = O((\theta_d + \theta_m)^{-2})$.*

It is worth noting the limiting cases of the unified worker model. In particular, when $\theta_m \to 0$, the problem reduces to clustering with temporary workers as do the achievable scheme and sample complexity requirements. On the other hand, $\theta_d \to 0$ corresponds to a particular case of clustering using workers with memory.

5.4.2 Lower Bound on Sample Complexity

We now derive the lower bound on sample complexity by extending the proof of the converse for workers with memory.

Theorem 23. *Sample complexity of universal clustering under the unified worker model satisfies*

1. *for a fixed $\theta > 0$, $N_u^* = \Omega(\log \ell)$, and*
2. *for a fixed $\ell < \infty$, $N_u^* = \Omega((\theta_m + \theta_d^2)^{-1})$.*

Proof. We proceed in similar fashion to the proof for the case of workers with memory. Again, consider the prior parametrized by the size of the problem ℓ as $P_T(1) = 1 - \pi, P_T(2) = \pi$ such that $\pi = \frac{1}{\ell}$. Again, we will use the generalized Fano's inequality over the space $\mathcal{E}$ of vectors of objects. We again consider the case of $\mathcal{N}_i = \{\tilde{i}\}$.

Consider workers such that marginals of responses to an object satisfy

$$\mathbb{P}\left[Y = i | T = j\right] = \begin{cases} p, & i = j \\ 1 - p, & i \neq j \end{cases}$$

irrespective of the order of occurrence. Define the matrices

$$W^{(1)} = [W_{ij}^{(1)}]_{1 \leq i,j \leq 2} = \begin{bmatrix} a & 1-a \\ 1-b & b \end{bmatrix},$$

and

$$W^{(2)} = [W_{ij}^{(2)}]_{1 \leq i,j \leq 2} = \begin{bmatrix} b & 1-b \\ 1-a & a \end{bmatrix}.$$

Let the worker responses be characterized by

$$\mathbb{P}\left[Y_i = k \middle| Y_{\tilde{i}} = \tilde{k}, T_i = T_{\tilde{i}} = j\right] = W_{k\tilde{k}}^{(j)}.$$

From the constraint on distance quality, we have $2p - 1 \geq \theta_d$. The constraint on the nature of the marginals establishes $ap - b(1-p) = p$. The restriction on the information quality implies:

$$h(p) - ph(a) - (1-p)h(b) \geq 2\theta_m.$$

Let us consider the case when both inequalities hold with equality. This yields a specific worker channel that satisfies the memory and distance quality requirements. We analyze the error probability on this worker channel.

Again, using analysis similar to the proof of Lem. 27, we observe the KL divergences between the hypotheses in the $(\ell + 1)$-ary hypothesis testing problem are $O(1)$. Hence there exists a constant β such that (5.35) holds. Thus, for constant θ_m and θ_d, the sample complexity of universal clustering satisfies:

$$N_u^* = O(\log \ell).$$

Now, when ℓ is fixed, we study the necessary sample complexity of universal clustering with respect to θ_m, θ_d. We know that a consistent universal clustering algorithm also solves

the binary hypothesis test

$$\psi : \begin{cases} H_0 : I(Y_1; Y_2) = 0 \\ H_1 : I(Y_1; Y_2) \geq 2\theta_m. \end{cases}$$

Following the analysis from the proof of Theorem 21, from (5.37), we have

$$D(p\|q) \leq n \left(2\theta_m + \theta_d \log \left(\frac{1+\theta_d}{1-\theta_d} \right) \right) \lesssim n(\theta_m + \theta_d^2).$$

Finally, using the Kailath lower bound, we obtain

$$P_e(\Phi) \geq \frac{1}{4} \exp\left(-4cn(\theta_m + \theta_d^2)\right). \tag{5.42}$$

Thus, for constant ℓ,

$$N_u^* = \Omega((\theta_m + \theta_d^2)^{-1}).$$

□

From the theorem, we note the universal clustering algorithm is order optimal in sample complexity in terms of the number of objects, for a crowd of given quality. However, for a given number of objects, there exists an order gap between achievable sample complexity and the converse. As expected, the gap follows from the gap in the case of workers with memory, which in turn is from the gap in estimating entropy [73].

In particular, we observe that as $\theta_d \to 0$, the problem reduces to the case of workers with memory, and on the other hand as $\theta_m \to 0$, it reduces to the problem of clustering using temporary workers without memory.

A finer point in the analysis to be noted is that the worst-case channels considered in the converse proofs is the inertial channel considered in the proof of Thm. 21, which is also the solution to the set of constraints for the channel in the unified scenario under the limit of $\theta_d \to 0$.

Thus, temporary and long-term workers with memory are indeed closely related through the unified worker model, and are limiting cases.

5.5 Discussion

This chapter establishes an information-theoretic framework to study the universal crowdsourcing problem. We designed universal clustering algorithms using distributional identicality for temporary workers without memory, and response dependence for workers with

memory. We also derived necessary and sufficient conditions on the sample complexity, proving budget-optimality in terms of the number of objects. We then integrated the limiting cases to develop an budget optimal universal clustering algorithm for the unified worker model. Further behavioral experiments using crowd workers can be performed to gain insight into the performance of the algorithms in practice and to validate the unified worker models.

Our results provide a way to compare costs between crowdsourcing platforms, allowing us to choose the right task-dependent worker pool. Further, they provide a window into more general studies of the computational capabilities and complexities of human-based information systems. In particular, the work sheds light on the influence of various attributes of crowd workers such as object-specific memory. In essence, the work studies a space-time tradeoff for human computation systems and to the best of our knowledge is the first of its kind.

CHAPTER 6

STORAGE ON BLOCKCHAINS

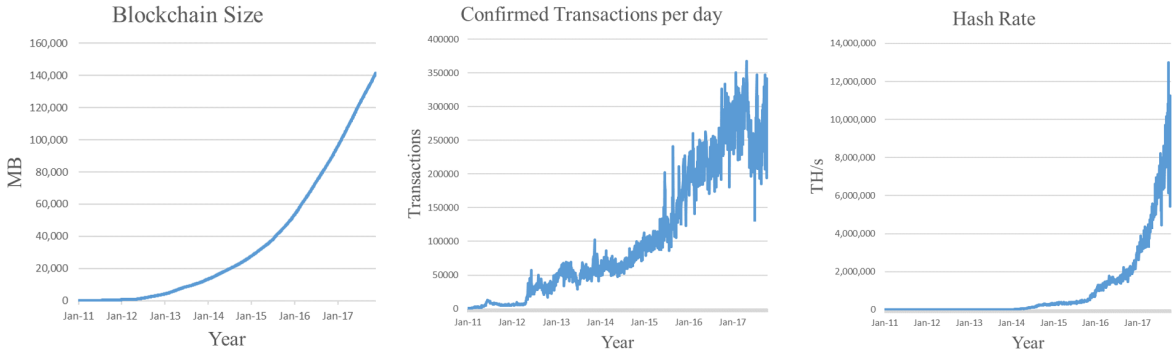
The invention of bitcoin [151] almost a decade ago brought blockchains into prominence in the business world. Blockchains maintain a shared version of a transaction ledger with each peer in the network storing a copy, reducing the friction in financial networks caused by intermediaries using different technology infrastructures. The technology has created a new environment of business transactions and self-regulated cryptocurrencies [152, 153].

Owing to their favorable properties, blockchains are being adopted extensively outside cryptocurrencies in a variety of novel application domains such as medicine [154], supply chain management, global trade [155], and government services [31, 156]. Blockchains are expected to revolutionize the way financial/business transactions are done, such as through smart contracts [157, 158]. More recently, cloud storage systems such as STORJ and SIA have been designed using blockchain, offering heightened security guarantees and a new approach to decentralized storage.

However, blockchain works on the premise that every peer stores the entire transaction ledger as a hash chain, even though the data is meaningless to peers that are not party to the transaction. Consequently, individual nodes incur a significant, ever-increasing storage cost [37, 159] as shown in Figs. 6.1a and 6.1b. Note that *secure* storage may be much more costly than raw hard drives, e.g. due to infrastructure and staffing costs. With storage costs expected to saturate due to the ending of Moore's law, storage is a pressing concern for the large-scale adoption of blockchain.

In current practice, the most common technique to reduce the storage overload is to prune old transactions in the chain. However, this mechanism is not sustainable for blockchains that have to support a high arrival rate of transaction data. Hence the blockchain architecture is not scalable owing to the heavy storage requirements.

For instance, consider the bitcoin network. Bitcoin currently serves an average of just under 3.5 transactions per second [37]. This number is low for a variety of reasons including the economics involved in maintaining a high value for the bitcoin. However, even at this rate, storage requirements average 160 MB per day [159], i.e., about 60 GB per year. SETL, an institutional payment and settlement infrastructure based on blockchain, claims to support



(a) Blockchain size

(b) Average number of transactions per day

(c) Hash-rate

Figure 6.1: Increase in transactions, storage, and hash-rate in bitcoin. Data obtained from [159].

1 billion transactions per day. This is dwarfed by the Federal Reserve, which processes 14 *trillion* financial transactions per day. If cryptocurrencies are to become financial mainstays, they would need to be scaled by several orders. Specifically, they will eventually need to support about 2000 transactions per second on average [37]. Note that this translates to an average storage cost of over 90 GB per day. This is to say nothing about uses for blockchain in global trade and commerce, healthcare, food and agriculture, and a wide variety of other industries. With storage cost expected to saturate soon due to the end of Moore's law, storage is emerging as a pressing concern for the large-scale adoption of blockchain.

This impending end to Moore's law also results in a saturation of computational speeds. Notwithstanding new efforts [160], block validation (mining) in bitcoin-like networks involves an expensive hash computation stage that requires high-end hardware and much energy—a demand that has only grown with the increasing hash rates as shown in Fig. 6.1c. Recent studies have estimated that global energy consumption of bitcoin is of the order of 700 MW [161] – enough to power over 325,000 homes, and over 5000 times the energy per transaction on a credit card.

Distributed storage schemes have been considered in the past in the form of information dispersal algorithms (IDA) [162, 163] and in the form of distributed storage codes [164, 165]. In particular [163] considers an information dispersal scheme that is secure from adaptive adversaries. We note that the coding scheme we define here is stronger than such methods as it handles active adversaries. Secure distributed storage codes with repair capabilities to protect against colluding eavesdroppers [166] and active adversaries [167] have also been considered. The difference in the nature of attacks by adversaries calls for the new coding scheme described here.

To address rising storage costs and increasing transaction volumes [37], we proposed secure,

distributed storage [168]. In particular, we design a distributed storage scheme for the cold storage of the blockchain ledger in the presence of active adversaries. This work uses a novel combination of distributed storage codes [164], private key encryption, and secret key sharing [169], inspired by [170], to distribute data among peers. Alternative coding approaches through Lagrange coded computing have been explored for hot distributed storage of the transactions [171, 172]. More recently new protocols and blockchain systems have also been designed to address the concern of scalability of blockchain systems [173].

The construction of such a code results in tradeoffs not only between storage and recovery costs, but also among the associated integrity and confidentiality guarantees of the system. Here we describe the coding scheme, highlighting the flexibility of the code choice and the resulting applications of blockchain toward data insurance and cloud storage. We also study the effects of denial of service and targeted corruptions on data loss and compute the probabilities of such corruptions. We also study the confidentiality of the data stored by the system, determining the minimum extent of collusion under which a data leak is feasible. We elaborate how the storage and recovery costs depend on the coding parameters.

6.1 System Model

We now abstract blockchain systems by a mathematical model for the peer network and hash chain. The model described here is mainly based on the Hyperledger Fabric [174].

6.1.1 Ledger Construction

The blockchain comprises a connected peer-to-peer network of nodes, where nodes are placed into three primary categories based on functionality:

1. **Clients:** nodes that invoke or are involved in a transaction, have the blocks validated by endorsers, and communicate them to the orderers.

2. **Peers:** nodes that commit transactions and maintain a current version of the ledger. Peers may also adopt endorser roles.

3. **Orderer:** nodes that communicate the transactions to the peers in chronological order to ensure consistency of the hash chain.

Note that the classification highlighted here is only based on function, and individual nodes in the network can serve multiple roles.

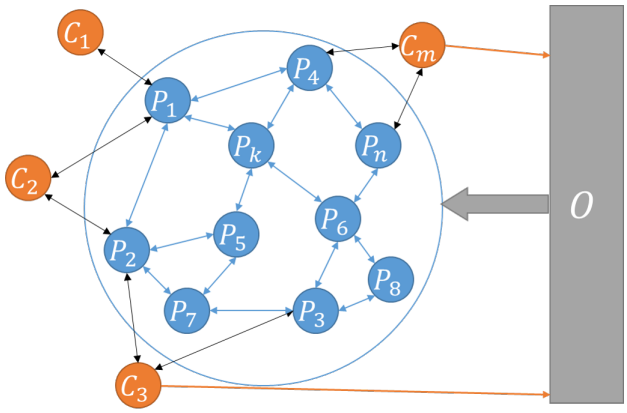


Figure 6.2: Architecture of a blockchain network: Here the network is categorized by functional role into clients C_i, peers P_i, and orderers O. As mentioned earlier, the clients initialize transactions. Upon validation, the transactions are communicated to peers by orderers. The peers maintain an ordered copy of the ledger of transactions.

The distributed ledger of the blockchain maintains a current copy of the sequence of transactions. A transaction is initiated by the participating clients and is verified by endorsers (select peers). Subsequently, the verified transaction is communicated to the orderer. The orderer then broadcasts the transaction blocks to the peers to store in the ledger. The nodes in the blockchain are as depicted in Fig. 6.2. Here nodes C_i are clients, P_i are peers, and O is the set of orderers in the system, categorized by function.

A transaction and the nature of the data associated with it is application-specific such as proof of fund transfer across clients in bitcoin-like cryptocurrency networks, smart contracts in business applications, patient diagnoses/records in medical record storage, and raw data in cloud storage. We use the term *transaction* broadly to represent all such categories.

A transaction is initiated by participating clients, verified by endorsers (select peers), and broadcast to peers through orderers. The ledger is stored as a (cryptographic) hash chain.

Definition 26. *Let $\mathcal{M}$ be a set of messages of arbitrary lengths, $\mathcal{H}$ the set of (fixed-length)* hash values. *A* cryptographic hash function *family is a function $h : \mathcal{I} \times \mathcal{M} \to \mathcal{H}$, where $\mathcal{I}$ is the set of parameters that dictate the deterministic map that is employed.*

Good hash functions hold several salient properties [175] such as

1. **Computational ease:** Hash values are easy to compute.

2. **Pre-image resistance:** Given $H \in \mathcal{H}$, it is computationally infeasible to find $M \in \mathcal{M}$ such that $h(\mathcal{M}) = H$. To be precise, given a randomized and computationally limited adversary who samples the message $M' = A(H, I)$, we consider the pre-image resistance

in terms of the hitting probability

$$P_{pre-image} = \mathbb{P}\left[h(I, A(H, I)) = H\right]. \tag{6.1}$$

3. **Collision resistance:** It is computationally infeasible to find $M_1, M_2 \in \mathcal{M}$ such that $h(M_1) = h(M_2)$. Again, to be precise, given a randomized and computationally limited adversary who sample messages $(M, M') = A(I)$, we consider the always collision resistance in terms of the hitting probability

$$P_{collision} = \mathbb{P}\left[\{M \neq M'\} \cap \{h(I, M) = h(I, M')\}\right]. \tag{6.2}$$

A hash chain is a sequence of data blocks such that each block includes a header, which is the hash value of the previous (header included) block.

To be precise, the clients can invoke the following operations on the blockchain:

- WRITE(B): initiate a block B of data that include transaction data which are verified and appended to the blockchain ledger.
- READ(t): call with index t to recover block B_t from the blockchain ledger.

The endorsers on the other hand perform the following operations:

- VERIFY(B): check the details of the transaction and verify authenticity.
- MINE(B, t): recover hash value H_{t-1} and use B to compute hash H_t and report to orderer to include block in blockchain ledger.

The operations of the orderer are:

- VALIDATE(B, H, t): validate block and hash value reported by endorser.
- APPEND(B, H, t): encode data and hash blocks and communicate to peers, to append block at index t in the ledger.

Bitcoin-type blockchains use the hash chain structure but store the individual transaction data as a Merkle tree, with the hash chain constructed using the Merkle root as the data in the block [37]. We consider a simple form of this as shown in Fig. 6.3. Let $\mathbf{B}_t$ be the data block corresponding to the tth transaction. Let g, h be two hash functions. Let $W_t = (H_{t-1}, g(\mathbf{B}_t))$ be the concatenation of the previous hash and a hash of the current data. Then, $H_t = h(I_t, W_t)$ is the hash value stored with the $(t+1)$th block, where the index I_t is sampled uniformly.

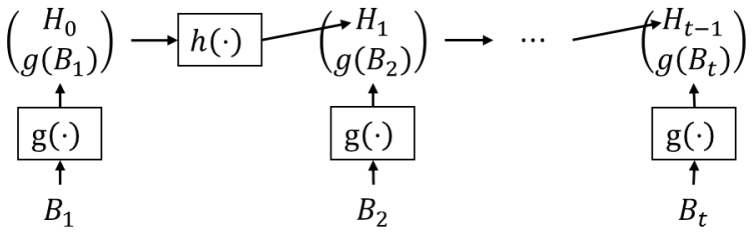


Figure 6.3: Hash chain structure for the ledger. The chain is constructed by hashing a hash value of the data for easier recovery and consistency verification.

Using such a hashed form to construct the chain simplifies consistency verification and reduces recovery costs, while retaining all the salient features of the hash chain that directly hashed block values would have. In more general forms, the data block can be replaced by a Merkle tree structure with the results extending directly.

For all t, let $\mathbf{B}_t \sim \text{Unif}(\mathbb{F}_q)$ and $g(B_t), H_t \in \mathbb{F}_p$, where $q, p \in \mathbb{N}$ and $\mathbb{F}_q, \mathbb{F}_p$ are finite fields of order q and p respectively. Thus, the cost of storage per peer per transaction in conventional implementation is

$$\tilde{R}_s = \log_2 q + 2\log_2 p \text{ bits.} \tag{6.3}$$

In practice, data blocks can be of varying sizes and the results follow *mutatis mutandis*. For the uniform random oracle model, the pre-image and collision resistance characteristics of the hash family in use is

$$P_{pre-image} \approx \frac{1}{p}, \quad P_{collision} \approx \frac{1}{p}.$$

We perform the analysis with respect to the random oracle model [176].

Transactions stored in the ledger may at a later point be recovered in order to validate claims or verify details of the past transaction by nodes that have read access to the data. Different implementations of the blockchain invoke different recovery mechanisms depending on the application. One such method is to use an authentication mechanism wherein select peers return the data stored in the ledger and the other peers validate (sign) the content. Depending on the application, one can envision varying the number of authorization checks necessary to validate the content. For convenience, we restrict this work to one form of retrieval which broadly encompasses a wide class of recovery schemes. Specifically, we assume that in order to recover the tth transaction, each peer returns its copy of the transaction and the majority rule is applied to recover the block.

6.1.2 Blockchain Security

Two aspects of security are of interest in blockchains—*integrity* and *confidentiality*. Whereas an integrity property guarantees that the stored data cannot be corrupted unless most of the peers are corrupted, a confidentiality property ensures that local information from individual peers does not reveal sensitive transaction information.

Corruption of data in the blockchain requires corrupting a majority of the peers in the network to alter the data stored in the distributed, duplicated copy of the transaction ledger. Thus, the blockchain system automatically ensures a level of integrity in the transaction data.

Conventional blockchain systems such as bitcoin enforce additional constraints on the hash values to enhance data integrity. For instance, in the bitcoin network, each transaction block is appended with a nonce which is typically a string of zeros, such that the corresponding hash value satisfies a difficulty target i.e., is in a specified constraint set. The establishment of such difficulty targets in turn implies that computing a nonce to satisfy the hash constraints is computationally expensive. Thus data integrity can be tested by ensuring the hash values are consistent as it is computationally infeasible to alter the data.

The hash chain, even without the difficulty targets, offers a mechanism to ensure data integrity in the blockchain. Specifically, note that the pre-image resistance of the cryptographic hash function ensures that any change to H_{t-1} or B_t would require recomputing H_t with high probability as it is computationally infeasible for an adversary to determine W_t such that $h(I_t, W_t) = H_t$. To be precise, owing to collision resistance, any alteration of data in W_t results in an alteration of H_t with probability at least $1 - \frac{1}{p}$.

Thus storing the ledger in the form of a hash chain ensures that corrupting a past transaction not only requires the client to corrupt at least half the set of peers to change the majority, but also maintain a consistent hash chain following the corrupted transaction. That is, say a participating client wishes to alter transaction B_1 to B_1'. Let there be T transactions in the ledger. Then, corrupting B_1 implies that the client would also have to replace H_1 with $H_1' = h(I_1, W_1')$ at the corrupted nodes. This creates a domino effect, in that all subsequent hashes must in turn be altered with high probability as well, to maintain integrity of the chain. This strengthens the integrity of the transaction data in blockchain systems.

Confidentiality of information is typically guaranteed in these systems through the use of private key encryption methods, where the key is shared with a select set of peers are authorized to view the contents of the transactions. Note however that in such implementations, a leak at a single node could lead to a complete disambiguation of the information. One approach to enhance data confidentiality would be to distribute the data so that a single point of leakage does not divulge all the information.

In this chapter we design distributed storage codes with these two aspects of security in

mind. We first establish the adversary model considered.

6.1.3 Active Adversary Model

In this work, we explore the construction of a distributed storage coding scheme for the *cold storage* of the blockchain ledger that ensures heightened confidentiality and integrity of the data, even when the mining process is computationally inexpensive. Let us assume that each transaction B_t also has a corresponding *access list*, which is the set of nodes that have permission to read and edit the content in B_t. Note that this is equivalent to holding a private key to decrypt the encrypted transaction data stored in the ledger.

In this work, we primarily focus on *active adversaries* who alter a transaction content B_t to a desired value B'_t. Let us explicitly define the semantic rules of a valid corruption for such an adversary. If a client corrupts a peer, then the client can

1. learn the contents stored in the peer;
2. alter block content only if it is in the access list of the corresponding block; and
3. alter hash values as long as chain integrity is preserved, i.e., an attacker cannot invalidate the transaction of another node in the process.

The active adversary in our work is assumed to be aware of the contents of the hash chain and the block that it wishes to corrupt. We elaborate on the integrity of our coding scheme against such active adversaries. We also briefly elaborate on the data confidentiality guaranteed by our system against local information leaks.

Another typical attack of interest in such blockchain systems is the denial of service attack where an adversary corrupts a peer in the network to deny the requested service, which in this case is returning the data stored in the ledger. We also briefly describe the vulnerability of the system to denial of service attacks owing to the distributed storage.

Before we describe the code construction, we first give a preliminary introduction to coding and encryption schemes that we use as the basis to build our coding scheme.

6.2 Preliminaries

This work uses a private key encryption scheme with a novel combination of secret key sharing and distributed storage codes to store the transaction data and hash values. We now provide a brief introduction to these elements.

6.2.1 Shamir's Secret Sharing

Consider a secret $S \in \mathbb{F}_q$ that is to be shared with $n < q$ nodes such that any subset of size less than k get no information regarding the secret upon collusion, while any subset of size at least k get complete information. Shamir's (k, n) secret sharing scheme [169] describes a method to explicitly construct such a code. All the arithmetic performed here is finite field arithmetic on $\mathbb{F}_q$.

Draw $a_i \overset{i.i.d}{\sim} \text{Unif}(\mathbb{F}_q)$, for $i \in [k-1]$ and set $a_0 = S$. Then, compute

$$y_i = a_0 + a_1 x_i + a_2 x_i^2 + \cdots + a_{k-1} x_i^{k-1}, \text{ for all } i \in [n],$$

where $x_i = i$. Node $i \in [n]$ receives the share y_i.

Since the values are computed according to a polynomial of order $k-1$, the coefficients of the polynomial can be uniquely determined only when we have access to at least k points. Recovering the secret key involves polynomial interpolation of the k shares to obtain the secret key (intercept). Thus, the secret can be recovered if and only if at least k nodes collude.

In this work, we presume that each secret share is given by (x_i, y_i), and that the unique abscissa values are chosen uniformly at random from $\mathbb{F}_q \backslash \{0\}$. That is, $\{x_i : i \in [n]\}$ are drawn uniformly at random without replacement from $\mathbb{F}_q \backslash \{0\}$. Then, given any $k-1$ shares and the secret, the final share uniformly likely in a set of size $q - k$.

It is worth noting that Shamir's scheme is minimal in storage as the size of each share is the same as the size of the secret key. Shamir's scheme however is not secure to active adversaries. In particular, by corrupting $n-k+1$ nodes, the secret can be completely altered.

Secret key sharing codes have been widely studied in the past [177–179]. In particular, it is known that Reed-Solomon codes can be adopted to define secret shares. Linear codes for minimal secret sharing have also been considered [180]. In this work however, we restrict to Shamir's secret sharing scheme for simplicity.

6.2.2 Data Encryption

Shannon considered the question of perfect secrecy in cryptosystems from the standpoint of statistical security of encrypted data [181]. There, he concluded that perfect secrecy required the use of keys drawn from a space as large as the message space. This is practically unusable as it is difficult to use and securely store such large key values. Thus practical cryptographic systems leverage computational limitations of an adversary to guarantee security over perfect statistical secrecy.

We define a notion of encryption that is slightly different from that used typically in cryptography. Consider a message $\mathbf{M} = (M_1, \ldots, M_m) \in \mathcal{M}$, drawn uniformly at random. Let $K \in \mathcal{K}$ be a private key drawn uniformly at random.

Definition 27. *Given message, key, and code spaces $\mathcal{M}, \mathcal{K}, \mathcal{C}$ respectively, a* private key encryption scheme *is a pair of functions $\Phi : \mathcal{M} \times \mathcal{K} \to \mathcal{C}$, $\Psi : \mathcal{C} \times \mathcal{K} \to \mathcal{M}$, such that for any $\mathbf{M} \in \mathcal{M}$,*

$$\Phi(\mathbf{M}; K) = \mathbf{C}, \text{ such that, } \Psi(\mathbf{C}, K) = \mathbf{M},$$

and it is ϵ-secure if it is statistically impossible to decrypt the codeword in the absence of the private key K beyond a confidence of ϵ in the posterior probability. That is, if

$$\max_{\mathbf{M} \in \mathcal{M}, \mathbf{C} \in \mathcal{C}} \mathbb{P}\left[\Psi(\mathbf{C}, K) = M\right] \leq \epsilon. \tag{6.4}$$

The definition indicates that the encryption scheme is an invertible process and that it is statistically infeasible to decrypt the plaintext message beyond a degree of certainty. We know that given the codeword $\mathbf{C}$, decrypting the code is equivalent to identifying the chosen private key. In addition, from (6.4), we observe that the uncertainty in the message estimation is at least $\log_2\left(\frac{1}{\epsilon}\right)$. Thus,

$$\log_2\left(\tfrac{1}{\epsilon}\right) \text{ bits} \leq H(\mathbf{M}|\mathbf{C}) \leq \log_2 |\mathcal{K}| \text{ bits}.$$

For convenience, we assume without loss of generality that the encrypted codewords are vectors of the same length as the message from an appropriate alphabet, i.e., $\mathbf{C} = (C_1, \ldots, C_m)$.

Since we want to secure the system from corruption by adversaries who are aware of the plaintext message, we define a stronger notion of secure encryption. In particular, we assume that an attacker who is aware of the message $\mathbf{M}$, and partially aware of the codeword, $\mathbf{C}_{-j} = (C_1, \ldots, C_{j-1}, C_{j+1}, \ldots, C_m)$, is statistically incapable of guessing C_j in the absence of knowledge of the key K. That is, for any $\mathbf{M}, \mathbf{C}_{-j}$,

$$\mathbb{P}\left[\Phi(\mathbf{M}; K) = \mathbf{C}|\mathbf{M}, \mathbf{C}_{-j}\right] \leq \frac{1}{2}, \text{ for any } C_j. \tag{6.5}$$

A more general requirement would bound the probability of recovering the key, given incomplete cipher text, by a parameter δ. In this description, for ease of description let us work with the factor $\frac{1}{2}$. The generalization of the analysis to a parameter δ is straightforward. Note that this criterion indicates that the adversary is unaware of at least 1 bit of

information in the unknown code fragment, despite being aware of the message, i.e.,

$$H(\mathbf{C}|\mathbf{M}, \mathbf{C}_{-j}) \geq 1 \text{ bit.}$$

6.2.3 Distributed Storage Codes

This work aims to reduce the storage cost for blockchains by using distributed storage codes. Distributed storage codes have been widely studied [164,182] in different contexts. In particular, aspects of repair and security, including explicit code constructions have been explored widely [165–167, 183, 184]. In addition, information dispersal algorithms [162, 163] have also considered the question of distributed storage of data. This is a non-exhaustive listing of the existing body of work on distributed storage and most algorithms naturally adapt to the coding scheme defined here. However, we consider the simple form of distributed storage that just divides the data evenly among nodes.

6.3 Coding Scheme

For this section, assume that at any point of time t, there exists a partition $\mathcal{P}_t$ of the set of peers $[n]$ into sets of size m each. In this work we presume that n is divisible by m. Let each set of the partition be referred to as a *zone*. Without loss of generality, the zones are referred to by indices $1, \ldots, \frac{n}{m}$. At each time t, for each peer $i \in [n]$, let $p_t^{(i)} \in [\frac{n}{m}]$ be the index that represents the zone that includes peer i. We describe the zone allocation scheme in detail in Sec. 6.5.

6.3.1 Coding Data Block

In our coding scheme, a single copy of each data block is stored in a distributed fashion across each zone. Consider the data block $\mathbf{B}_t$ corresponding to time t. We use a technique inspired by [170]. First a private key K is generated at each zone and the data block is encrypted using the key. The private key is then stored by the peers in the zone using Shamir's secret key sharing scheme. Finally, the encrypted data block is distributed amongst peers in the zone using a distributed storage scheme. The process involved in storage and recovery of a block, given a zone division is shown in Fig. 6.4.

More generally we can allow the zone sizes at time t to be chosen by the client. For ease however, we describe the coding scheme for constant zone sizes m. To customize the storage,

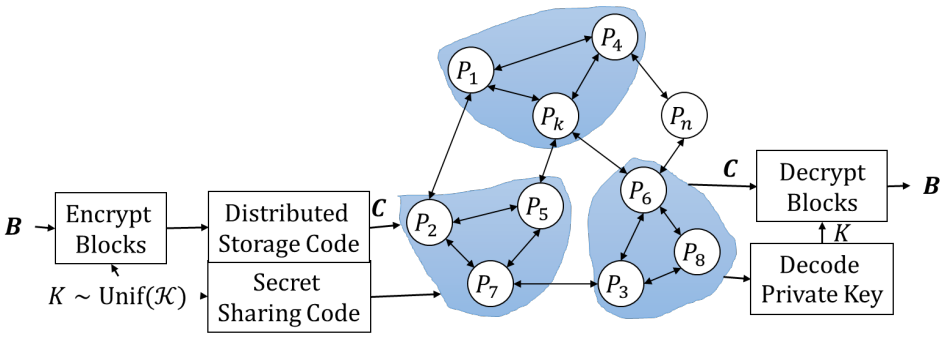


Figure 6.4: Encryption and decryption process for a given zone allocation. The shaded regions represent individual zones in the peer network. The data is distributed among peers in each zone and the data from all peers in a zone are required to recover the transaction data.

Algorithm 9 Coding scheme for data block

for $z = 1$ to $\frac{n}{m}$ **do**
  Generate private key $K_t^{(z)} \sim \text{Unif}(\mathcal{K})$
  Encrypt block with key $K_t^{(z)}$ as $\mathbf{C}_t^{(z)} = \Phi(\mathbf{B}_t; K_t^{(z)})$
  Distribute $\mathbf{C}_t$ and store among peers in $\{i : p_t^{(i)} = z\}$
  Use Shamir's (m, m) secret sharing on $K_t^{(z)}$ and distribute shares $(K_1^{(z)}, \ldots, K_{k_t}^{(z)})$ among peers in the zone
end for

we just need to replace the zone sizes by k_t and use a corresponding key space $\mathcal{K}_{k_t}$. The coding scheme is given by Alg. 9. In this discussion we will assume that the distributed storage scheme just distributes the components of the code vector $\mathbf{C}_t$ among the peers in the zone. The theory extends naturally to other distributed storage schemes.

To preserve the integrity of the data, we use secure storage for the hash values as well. In particular, at time t, each zone $Z \in \mathcal{P}_t$ stores a secret share of the hash value H_{t-1} generated using Shamir's (m, m) secret sharing scheme.

The storage per transaction per peer is thus given by

$$R_s = \frac{1}{m} \log_2 |\mathcal{C}| + 2 \log_2 |\mathcal{K}| + 2 \log_2 p \text{ bits}, \tag{6.6}$$

where $|\mathcal{C}| \geq q$ depending on the encryption scheme. In particular, when the code space of encryption matches the message space, i.e., $|\mathcal{C}| = q$, the gain in storage cost per transaction per peer is given by

$$\text{Gain in storage cost} = \tilde{R}_s - R_s = \frac{m-1}{m} \log_2 q - 2 \log_2 |\mathcal{K}| - \log_2 p \text{ bits}. \tag{6.7}$$

Thus, in the typical setting where the size of the private key space is much smaller than the

Algorithm 10 Recovery scheme for data block

$\mathcal{N} \leftarrow [n]$
Compute $K_t^{(z)}$, for all z, by polynomial interpolation
Decode blocks $B_t^{(z)} \leftarrow \Psi\left(\mathbf{C}_t^{(z)}; K_t^{(z)}\right)$, for all $z \in [\frac{n}{m}]$
if $|\{B_t^{(z)} : z \in [\frac{n}{m}]\}| > 1$ **then**
  for $\tau = t$ to $\min\{t + d_t, T\}$ **do**
    Compute $H_\tau^{(z)}$, for all z, by polynomial interpolation
    Determine $W_\tau^{(i)} = \left(g(B_\tau^i), H_{\tau-1}^i\right)$, for all $i \in [n]$
    $\mathcal{I} \leftarrow \left\{i : h(W_\tau^{(i)}) \neq H_\tau^{(z)}, z = p_{\tau+1}^{(i)}\right\}$
    $\mathcal{N} \leftarrow \mathcal{N} \backslash \mathcal{I}$
    if $|\{B_t^{(p_t^{(i)})} : i \in \mathcal{N}\}| = 1$ **then**
      break
    end if
  end for
end if
return Majority in $\{\{B_t^{(p_t^{(i)})} : i \in \mathcal{N}\}\}$

size of the blocks, we have a reduction in the storage cost.

6.3.2 Recovery Scheme

We now describe the algorithm to retrieve a data block B_t in a blockchain system comprising a total of T transactions. However, instead of exploring the entire length until we identify a unique consistent version, we provide the client the freedom to choose the depth d_t of transactions that follow in the hash chain, and return the majority consistent version. The algorithm to recover block B_t is described in Alg. 10.

According to Alg. 10, each peer first communicates the codeword corresponding to the data block and the secret share of the encryption key. This corresponds to $\frac{1}{k_t}\log_2 q + k_t(2\log_2 k_t + 1)$ bits. Additionally, each peer also communicates the secret shares of hash values corresponding to the next d blocks, each of which contributes $2\log_2 p$ bits, and the corresponding data blocks for consistency check. Thus the total worst-case cost of recovering the tth data block is

$$R_r^{(t)} = C_r\left(\tfrac{1}{m}\log_2 q + \log_2|\mathcal{K}| + 4d_t\log_2 p\right), \tag{6.8}$$

where C_r is the cost per bit of communication.

The recovery algorithm exploits information-theoretic security in the form of the coding

scheme, and also invokes the hash-based computational integrity check established in the chain. First, the data blocks are recovered from the distributed, encrypted storage from each zone. In case of a data mismatch, the system inspects the chain for consistency in the hash chain. The system scans the chain for hash values and eliminates peers that have inconsistent hash values. A hash value is said to be inconsistent if the hash value corresponding to the data stored by a node in the previous instance does not match the current hash value. Through the inconsistency check, the system eliminates some, if not all corrupted peers. Finally, the majority consistent data is returned.

In the implementation, we presume that all computation necessary for the recovery algorithm is done privately by a black box. In particular, we presume that the peers and clients are not made aware of the code stored at other peers or values stored in other blocks. Specifics of practical implementation of such a black box scheme are beyond the scope of this work.

6.3.3 Feasible Encryption Scheme

The security of the coding scheme from corruption by active adversaries depends on the encryption scheme used. We first describe the necessary condition on the size of the key space.

Lemma 28. *A valid encryption scheme satisfying* (6.4) *and* (6.5)*, has*

$$|\mathcal{K}| \geq 2^m.$$

Proof. First, by chain rule of entropy,

$$H(K, \mathbf{C}|\mathbf{M}) = H(K) + H(\mathbf{C}|\mathbf{M}, K) = H(K), \tag{6.9}$$

where (6.9) follows from the fact that the codeword is known given the private key and the message.

Again using the chain rule and (6.9), we have

$$\begin{aligned} H(K) &= H(\mathbf{C}|\mathbf{M}) + H(K|\mathbf{C}, \mathbf{M}) \\ &\geq \sum_{j=1}^{m} H(C_j|\mathbf{C}_{-j}, \mathbf{M}) && (6.10) \\ &\geq m, && (6.11) \end{aligned}$$

Algorithm 11 Encryption scheme

$T \leftarrow \text{Unif}(\mathcal{T})$, $K \leftarrow \text{Key}(T)$; $\mathbf{b} \leftarrow \text{Binom}(n, 1/2)$
Assign peers to vertices, i.e., peer i is assigned to node θ_i
For all $i \neq v_0$, $\tilde{C}_i \leftarrow B_i \oplus B_{\mu_i}$; flip bits if $b_i = 1$.
$\tilde{C}_{v_0} \leftarrow \left(\oplus_{j \neq v_0} \tilde{C}_j\right) \oplus B_{v_0}$
if $b_{v_0} = 1$ **then**
  Flip the bits of $\tilde{C}_{v_0}$
end if
Store $C_i \leftarrow \tilde{C}_{\theta_i}$ at each node i in the zone
Store (K, θ) using Shamir's secret sharing at the peers
Store the peer assignment θ_i locally at each peer i

where (6.10) follows from non-negativity of entropy and the fact that conditioning only reduces entropy. Finally, (6.11) follows from the condition (6.5). Since keys are chosen uniformly at random, the result follows. □

More generally, if $\delta = \frac{1}{2^\ell}$, then $|\mathcal{K}| \geq 2^{\ell m}$.

We now describe an encryption scheme that is order optimal in the size of the private key space upto log factors. Let $\mathcal{T}$ be the set of all rooted, connected trees defined on m nodes. Then, by Cayley's formula [185],

$$|\mathcal{T}| = m^{(m-1)}.$$

Let us define the key space by the entropy-coded form of uniform draws of a tree from $\mathcal{T}$. Hence in the description of the encryption scheme, we presume that given the private key K, we are aware of all edges in the tree. Let $V = [m]$ be the nodes of the tree and v_0 be the root. Let the parent of a node i in the tree be μ_i.

Consider the encryption function given in Alg. 11. The encryption algorithm proceeds by first selecting a rooted, connected tree uniformly at random on m nodes. Then, each peer is assigned to a particular node of the tree. For each node other than the root, the codeword is created as the modulo 2 sum of the corresponding data block and that corresponding to the parent. Finally, the root is encrypted as the modulo 2 sum of all codewords at other nodes and the corresponding data block. The bits stored at the root node are flipped with probability half. The encryption scheme for a sample data block is shown in Fig. 6.5. We refer to Alg. 11 as Φ from here on.

The decryption of the stored code is as given in Alg. 12. That is, we first determine the private key, i.e., the rooted tree structure, the bit, and peer assignments. Then we decrypt the root node by using the codewords at other peers. Then we sequentially recover the other blocks by using the plain text message at the parent node.

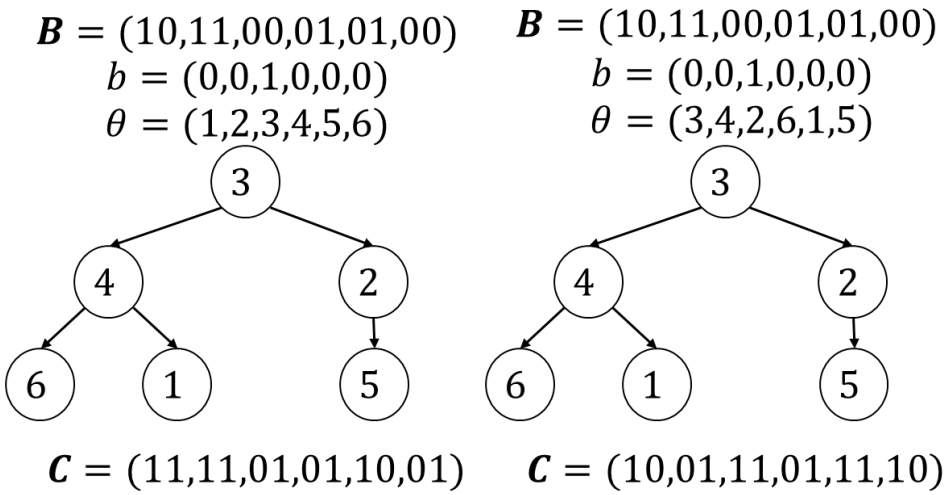


Figure 6.5: Encryption examples for a zone with six peers. The data block, parameters, tree structure, and corresponding codes are shown. The two cases consider the same rooted tree with varying peer assignments. The corresponding change in the code is shown.

Algorithm 12 Decryption scheme

Use polynomial interpolation to recover $(K, \mathbf{b}, \theta)$
Define $\tilde{\theta}_i \leftarrow j$ if $\theta_j = i$
Flip the bits of $C_{\tilde{\theta}_{v_0}}$, if $b_{v_0} = 1$
$B_{v_0} \leftarrow C_{\tilde{\theta}_{v_0}} \oplus_{j \neq \tilde{\theta}_{v_0}} C_j$
For all $i \in [n] \backslash \{v_0\}$, flip bits of $\mathbf{C}_i$ if $b_i = 1$
Iteratively compute $B_i \leftarrow C_{\tilde{\theta}_i} \oplus B_{\mu_i}$ for all $i \neq v_0$
return B

Lemma 29. *The encryption scheme* Φ *satisfies* (6.4) *and* (6.5)*.*

Proof. Validity of (6.4) follows directly from the definition of the encryption scheme as the message is not recoverable from just the encryption.

To check the validity of (6.5), note that given the data at all peers other than one node, the adversary is unaware of the parent of the missing node. Since this is uniformly likely, the probability that the adversary can guess the encrypted data is at most 1/2, with the maximum being if the root is not recovered. □

Lemma 30. *The storage cost per peer per transaction under* (Φ, Ψ) *is*

$$R_s(\Phi, \Psi) = \frac{1}{m} \log_2 q + m \left(2 \log_2 m + 1\right) + 2 \log_2 p \ \textit{bits}. \tag{6.12}$$

Proof. First note that $\mathcal{C} = \mathbb{F}_q$. Next, the number of rooted, connected trees on m nodes is given by Cayley's formula as $m^{(m-1)}$. The peer assignments can be stored locally and so cost only $\log_2 m$ bits per node per transaction. Thus, the result follows. □

From Lem. 30, we can see that the encryption scheme guarantees order-optimal storage cost per peer per transaction up to log factor in the size of the key space. The security of the encrypted data can be enhanced by increasing the inter-data dependency by using directed acyclic graphs (DAGs) with bounded in-degree in place of the rooted tree. Then, the size of storage for the private key increases by a constant multiple.

If we work with the Merkle tree representation of the hash chain, then each peer stores two hash values per block, the Merkle root, and the hash header for the block. Thus, the storage cost in (6.12) is increased by $2\log_2 p$ bits additional bits for the Merkle root which is also stored as a secret using Shamir's secret sharing.

6.4 Performance of Coding Scheme

We now evaluate the data integrity and confidentiality that arise from the coding scheme described here.

6.4.1 Individual Block Corruption

We now establish the security guarantees of individual blocks in each zone from active adversaries. First, consider an adversary who is aware of the hash value H_t and wishes to alter it to H_t'.

Lemma 31. *Say an adversary, aware of the hash value H_t and the peers in a zone z, wishes to alter the value stored in the zone to H_t'. Then, the probability of successful corruption of such a system when at least one peer is honest, is $O(1/p)$.*

Proof. Assume the adversary knows the secret shares of $k-1$ peers in the zone. Since the adversary is also aware of $H_t = a_0$, the adversary is aware of the coding scheme through polynomial interpolation. However, since the final peer is honest, the adversary is unaware of the secret share stored here. Hence the result follows. □

This indicates that in order to corrupt a hash value, the adversary practically needs to corrupt all nodes in the zone.

To understand corruption of data blocks, we first consider the probability of successful corruption of a zone without corrupting all peers of the zone.

Lemma 32. *Consider an adversary, aware of the plain text* **B** *and the peers in a zone. If the adversary corrupts $c < m$ peers of the zone, then the probability that the adversary can*

alter the data to $\mathbf{B}'$ *is at most* $\frac{c^2}{m^2}$*, i.e.,*

$$\mathbb{P}\left[\mathbf{B} \to \mathbf{B}' \text{ in zone } z\right] \leq \exp\left[2\log\tfrac{c}{m} - \left(1-\tfrac{2}{m}\right)\left(1-\tfrac{c}{m}\right)\right], \tag{6.13}$$

for all $\mathbf{B} \neq \mathbf{B}', z \in [\frac{n}{m}]$.

Proof. For ease, let us assume that $\theta_i = i$ for all $i \in [m]$. From the construction of the encryption scheme, we note that if $B_i \to B_i'$ is to be performed, then all child blocks of i are to be altered as well. Further, for any change in the block contents, the root is also to be altered.

Thus, a successful corruption is possible only if all nodes in the next level and the root have been corrupted. However, the adversary can only corrupt the peers at random and has no information regarding the structure of the tree. Thus, the corruption is successful only if the adversary samples the root, the node to be altered, and its set of children. Let the number of children in the random tree be given by the random variable K. Thus, the probability of corruption can be bounded by the probability of corrupting these nodes, when only one block is to be altered in the data, as follows:

$$\begin{aligned}
&\mathbb{P}\left[\mathbf{B} \to \mathbf{B}' \text{ in zone } z\right] \\
&\quad \leq \mathbb{E}_K\left[\mathbb{P}\left[\text{Pick root, node, children in } c \text{ draws w/o replacement from } [m]\right]\right] \\
&\quad = \mathbb{E}_K\left[\binom{m-K-2}{c-k-2} \Big/ \binom{m}{c}\right] \\
&\quad \leq \mathbb{E}_K\left[\frac{c^{K+2}}{m^{K+2}}\right] \qquad (6.14) \\
&\quad = \mathbb{E}_K\left[\exp\left((K+2)\log\tfrac{c}{m}\right)\right]. \qquad (6.15)
\end{aligned}$$

Now the degree of a fixed node in a random labeled rooted tree is given by the binomial distribution, Binomial$(m-2, \frac{1}{m})$ [186]. Then, error probability essentially caters to computing the moment generating function of the binomial distribution. Thus,

$$\begin{aligned}
&\mathbb{P}\left[\mathbf{B} \to \mathbf{B}' \text{ in zone } z\right] \\
&\quad \leq \left(\frac{c}{m}\right)^2 \left(1 - \frac{1}{m}\left(1-\frac{c}{m}\right)\right)^{m-2} \qquad (6.16) \\
&\quad \leq \exp\left[2\log\tfrac{c}{m} - \left(1-\tfrac{2}{m}\right)\left(1-\tfrac{c}{m}\right)\right], \qquad (6.17)
\end{aligned}$$

where (6.17) follows from the fact that $1 - x \leq \exp(-x)$. □

A consistent corruption of a transaction by an active adversary however requires corruption

of at least $\frac{n}{2m}$ zones. This in turn characterizes the probability of successful correction as shown below.

Theorem 24. *Consider an active adversary who corrupts $c_1, \ldots, c_{n/2m}$ peers respectively in $n/2m$ zones. Then, the probability of successful corruption across the set of all peers is*

$$\begin{aligned}&\mathbb{P}\left[\textit{Successful corruption } \mathbf{B} \to \mathbf{B}'\right] \\ &\quad \leq \exp\left(\frac{n}{m}\left[\log\left(\frac{2\sum_{i=1}^{n/2m} c_i}{n}\right) - \left(1-\frac{2}{m}\right)\left(\frac{1}{2} - \frac{\sum_{i=1}^{n/2m} c_i}{n}\right)\right]\right), \end{aligned} \tag{6.18}$$

for all $\mathbf{B} \neq \mathbf{B}'$.

Proof. From Lem. 32 and independence of the encryption across zones, we have

$$\begin{aligned}&\mathbb{P}\left[\text{Successful consistent corruption } \mathbf{B} \to \mathbf{B}'\right] \\ &\leq \prod_{i=1}^{n/2m} \exp\left[2\log\tfrac{c}{m} - \left(1-\tfrac{2}{m}\right)\left(1-\tfrac{c}{m}\right)\right] \\ &= \exp\left(2\sum_{i=1}^{n/2m}\log c_i - 2\frac{n}{2m}\log m - \left(1-\frac{2}{m}\right)\left(\frac{n}{2m} - \frac{\sum_{i=1}^{n/2m} c_i}{m}\right)\right) \\ &\leq \exp\left(\frac{n}{m}\left[\log\left(\tfrac{2\sum_{i=1}^{n/2m} c_i}{n}\right) - \left(1-\frac{2}{m}\right)\left(\frac{1}{2} - \frac{\sum_{i=1}^{n/2m} c_i}{n}\right)\right]\right), \end{aligned} \tag{6.19}$$

where (6.19) follows from the arithmetic-geometric mean inequality. □

Note that

$$\sum_{i=1}^{n/2m} c_i \leq \frac{n}{2},$$

and thus the upper bound on successful corruption decays with the size of the peer network if less than half the network is corrupted. From Thm. 24, we immediately get the following corollary.

Corollary 10. *If an adversary wishes to corrupt a data block with probability at least $1-\epsilon$, for some $\epsilon > 0$, then the necessary condition on the total number of nodes to be corrupted satisfies*

$$\sum_{i=1}^{n/2m} c_i \geq \frac{n}{2}(1-\epsilon)^{\frac{m}{n}}. \tag{6.20}$$

Corollary 10 indicates that when the network size is large, the adversary practically needs to corrupt at least half the network to have the necessary probability of successful corruption.

Thus we observe that for a fixed zone division, the distributed storage system loses an arbitrarily small amount of data integrity as compared to the conventional scheme.

In Sec. 6.5, we introduce a dynamic zone allocation scheme to divide the peer network into zones for different time slots. We show that varying the zone allocation patterns over time appropriately yields even better data integrity.

6.4.2 Alternative Corruption Check

One method to increase the strength of the encryption is to generalize the scheme using directed acyclic graph with constant in-degree. Instead of encrypting the data prior to distribution, in this section we propose a hashing idea to enable a consistency check within each zone.

Given a block B_t to be distributed among the peers of the zone z, sample a nonce $N_z \sim \text{Unif}(\mathcal{N})$, and compute the hash value $G_t = h(I, (B_t, N_z))$, where $I \sim \text{Unif}(\mathcal{I})$. Then, we share the nonce and the hash value as a secret among the peers of the zone. Then, the block B_t is stored according to a distributed storage code among the peers of the zone as plaintext.

When recovering the data, note that each zone can verify the integrity of the data by recovering the secret nonce, padding it with the data, and checking the consistency of the stored hash value. Note that the storage cost of this scheme is characterized by

$$\mathcal{R}'_s = \frac{1}{m} \log_2 q + 2 \left(\log_2 |\mathcal{N}| + 2 \log_2 p \right) \text{bits}.$$

When $|\mathcal{N}| \gg p$, the probability that an active adversary, who corrupts only $c < m$ peers of a zone, can successfully corrupt a data block B is essentially the same as finding a collision in the hash value. This is owing to the fact that the adversary has no way to recover the nonce and hash value stored as a secret in the zone, and so has to essentially guess a transformation to the data that happens to share the same hash value. This is the same as the probability of collision and so for any zone z, and any block B,

$$\mathbb{P}\left[\text{Successful corruption of zone } z\right] \leq P_{\text{collision}}.$$

Thus, the probability of successful data corruption by adversary who corrupts fewer than $n/2$ peers is upper bounded owing to the independence across zones as

$$\mathbb{P}\left[\text{Successful corruption}\right] \leq \exp\left(-\frac{n}{2m} \log \frac{1}{P_{\text{collision}}} \right) \approx \exp\left(-\frac{n}{2m} \log p \right).$$

6.4.3 Data Loss

A data block is lost when some peers undergo a DoS attack and a sufficiently large number incur random data loss. Consider an adversary that wishes to prevent the recovery of a block $\mathbf{B}$ distributed according to the parameter k. Let $r = n/k$ be the number of copies of the data in the peer network. The data is lost when there exists at least one node failure or DoS attack in each zone.

The adversary picks a random number $C \sim P_{\text{dl}}$ and performs a DoS attack on C uniformly random peers. Let the number of peers corrupted in zone i be X_i and Y_i be the number that undergo data loss. Thus, $(X_1, \ldots, X_r)$ are distributed according to the multivariate hypergeometric distribution with n objects, C draws, and k objects of each of the r types.

The probability of data loss given the adversary attacks C peers is

$$\begin{aligned}\mathbb{P}\left[\text{Data Loss}|C\right] &= \mathbb{P}\left[X_i + Y_i > 0, \text{ for all } i \in [r]|C\right] \\ &= \mathbb{P}\left[Y_i > 0, \text{ for all } i \in [r] \text{ s.t. } X_i = 0|C\right] \\ &= \mathbb{E}\left[(1-\bar{\rho}^k)^{(r-\sum_{i=1}^r \mathbf{1}\{X_i>0\})}|C\right], \end{aligned} \tag{6.21}$$

where (6.21) follows from the independence of nodal failure and the expectation is taken over the multivariate hypergeometric distribution described above. Here $\mathbf{1}\{\cdot\}$ is the indicator function. The probability of data loss is then obtained by averaging (6.21) over C.

Then, for a DoS adversary limited by a budget B_{dl} of the expected number of peers it can corrupt, P_{dl} can be determined by solving the following linear program (LP)

$$\begin{aligned} P_{\text{dl}} \in \arg\max_{p} & \sum_{c=0}^{n} p(c)\mathbb{E}\left[(1-\bar{\rho}^k)^{(r-\sum_{i=1}^r \mathbf{1}\{X_i>0\})}|C=c\right] \qquad (6.22) \\ \text{s.t. } & \sum_{c=0}^{n} cp(c) \le B_{\text{dl}}, \ \sum_{c=0}^{n} p(c) = 1, \text{ and } p(c) \ge 0, \text{ for all } c. \end{aligned}$$

Computing the conditional expectation in (6.22) requires knowledge of the probability mass function (pmf) of the number of zones with non-zero corruption, given by

$$\mathbb{P}\left[\sum_{i=1}^{r} \mathbf{1}\{X_i > 0\} = \tilde{r}\right] = \binom{r}{\tilde{r}}\mathbb{P}\left[\sum_{i=1}^{\tilde{r}} X_i = c\right] \tag{6.23}$$

$$= \binom{r}{\tilde{r}}\binom{\tilde{r}k}{c}\Big/\binom{n}{c}, \tag{6.24}$$

where (6.23) follows from the symmetry in the zones. Then, the random variable $\sum_{i=1}^{\tilde{r}} X_i$ follows the hypergeometric distribution with parameters n, c, and $\tilde{r}k$ representing size of

population, number of draws, and number of successes respectively. This results in (6.24).

Solving the LP (6.22) gives us an idea of the budget-limited DoS adversary and so the data loss probability can be subsequently computed from (6.21). The optimal design choice to account for the worst-case DoS adversary would then be to pick k such that it minimizes the worst-case data loss probability, i.e.,

$$k^* = \arg\min_{k} \max_{P_{\mathrm{dl}}} \mathbb{P}\left[\text{Data Loss}\right].$$

However, the design choice is more nuanced and application-specific as it has to also account for other costs.

6.4.4 Data Confidentiality

We earlier stated that the two aspects of security needed in blockchain systems are data integrity and confidentiality. We addressed the question of data integrity in the previous subsection. We now consider confidentiality of transaction data.

Consider the situation where a peer i in a zone is compromised. That is, an external adversary receives the data stored by the peer for one particular slot. This includes the secret share of the private key K_i, the encrypted block data C_i, and secret share corresponding to the hash of the previous block.

From Shamir's secret sharing scheme, we know that knowledge of K_i gives no information regarding the actual private key K. Thus, the adversary has no information on the rooted tree used for encryption.

We know that the transaction data are chosen uniformly at random. Since the adversary is unaware of the relation of the nodes to one another, from the encryption scheme defined, we know that given the entire encrypted data $\mathbf{C}$, the probability of recovering the block $\mathbf{B}$ is uniformly distributed on the set of all possible combinations obtained for all possible tree configurations. That is, each possibly rooted tree yields a potential candidate for the transaction data.

This observation implies that

$$H(\mathbf{B}|\mathbf{C}) \leq H(K, \mathbf{B}|\mathbf{C}) = H(K) + H(\mathbf{B}|K, \mathbf{C}) = m \log_2 m. \tag{6.25}$$

We know that the entropy of the transaction block is actually $H(\mathbf{B}) = \log_2 q > m \log_2 m$. That is, the adversary does learn the transaction data partially and has a smaller set of candidates in comparison to the set of all possible values, given the entire codeword.

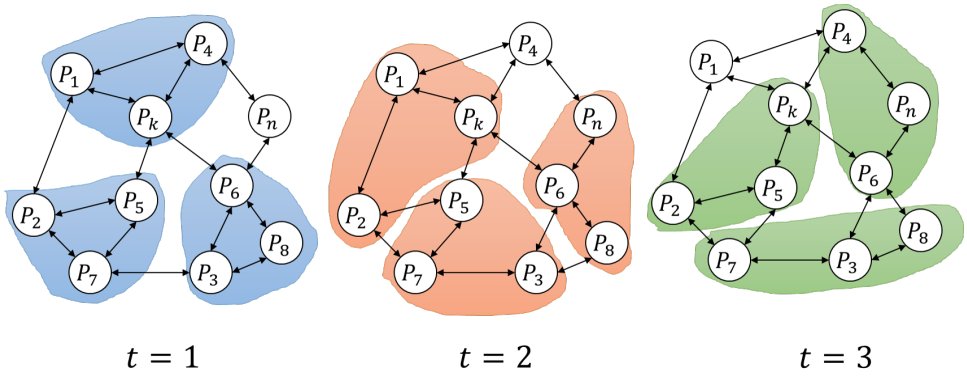


Figure 6.6: Dynamic zone allocation over time: Iterate the zone allocation patterns among the peers so that increasing number of peers need to be corrupted to maintain a consistent chain structure.

However, in the presence of just C_i, the adversary has no way to determine any of the other stored data, nor does it have any information on the position of this part of the code in the underlying transaction block. Thus, local leaks reveal very little information regarding the transaction block. Thus, we observe that the coding scheme also ensures a high degree of confidentiality in case of data leaks from up to $m - 1$ peers in a zone.

6.5 Dynamic Zone Allocation

In the definition of the coding and recovery schemes, we presumed the existence of a zone allocation strategy over time. Here we make it explicit.

Corollary 10 and Lem. 31 highlighted the fact that the distributed secure encoding process ensures that corrupting a transaction block or a hash requires an adversary to corrupt all peers in the zone. This fact can be exploited to ensure that with each transaction following the transaction to be corrupted, the client would need to corrupt an increasing set of peers to maintain a consistent version of the corrupted chain.

In particular, let us assume a blockchain in the following state

$$(H_0, \mathbf{B}_1) - (H_1, \mathbf{B}_2) - \cdots - (H_{t-1}, \mathbf{B}_t).$$

Let us assume without loss of generality that an adversary wishes to corrupt the transaction entry $\mathbf{B}_1$ to $\mathbf{B}'_1$. The validated, consistent version of such a corrupted chain would look like

$$(H_0, \mathbf{B}'_1) - (H'_1, \mathbf{B}_2) - \cdots - (H'_{t-1}, \mathbf{B}_t).$$

If the zone segmentation used for the encoding process is static, then the adversary can easily maintain such a corrupted chain at half the peers to validate its claim. If each peer is paired with varying sets of peers across blocks, then, for sufficiently large t, each corrupted peer eventually pairs with an uncorrupted peer.

Let us assume that this occurs for a set of corrupted peers at slot τ. Then, in order to successfully corrupt the hash $H_{\tau-1}$ to $H'_{\tau-1}$, the adversary would need to corrupt the rest of the uncorrupted peers in the new zone. On the other hand, if the client does not corrupt these nodes, then the hash value remains unaltered indicating the inconsistencies of the corrupted peers.

Thus, it is evident that if the zones are sufficiently well distributed, corrupting a single transaction would eventually require corruption of the entire network, and not just a majority. A sample allocation scheme is shown in Fig. 6.6.

However, the total number of feasible zone allocations is given by

$$\text{No. of zone allocations} = \frac{n!}{(m!)^{\frac{n}{m}}} \approx \frac{\sqrt{2\pi n}}{\left(\sqrt{2\pi m}\right)^{\frac{n}{m}}} \left(\frac{n}{m}\right)^n, \tag{6.26}$$

which increases exponentially with the number of peers and is monotonically decreasing in the zone size m. This indicates that naive deterministic cycling through this set of all possible zone allocations is practically infeasible.

To ensure that every uncorrupted peer is eventually grouped with a corrupted peer, we essentially need to ensure that every peer is eventually grouped with every other peer. Further, the blockchain system needs to ensure uniform security for every transaction and to this end, the allocation process should also be fair.

In order to better understand the zone allocation strategy, we first study a combinatorial problem.

6.5.1 K-Way Handshake Problem

Consider a group of n people. At each slot of time, the people are to be grouped into sets of size m. A peer gets acquainted with all other peers in the group whom they have not met before. The problem can thus be viewed as an m-way handshake between people.

Lemma 33. *The minimum number of slots required for every peer to shake hands with every other peer is* $\frac{n-1}{m-1}$.

Proof. At any slot, a peer meets at most $m-1$ new peers. Thus the lower bound follows. □

Algorithm 13 Dynamic Zone Allocation Strategy

Let $\nu_2 \ldots, \nu_{2n'}$ be the vertices of a $2n'-1$ regular polygon, and ν_1 its center
for $i = 2$ to $2n'$ **do**
  Let L be the line passing through ν_1 and ν_i
  $M \leftarrow \{(\nu_j, \nu_k) :$ line through ν_j, ν_k is perpendicular to $L\}$
  $M \leftarrow M \cup \{(\nu_1, \nu_i)\}$
  Construct zones as $\{\nu_j \cup \nu_k : (\nu_j, \nu_k) \in M\}$
end for
restart for loop

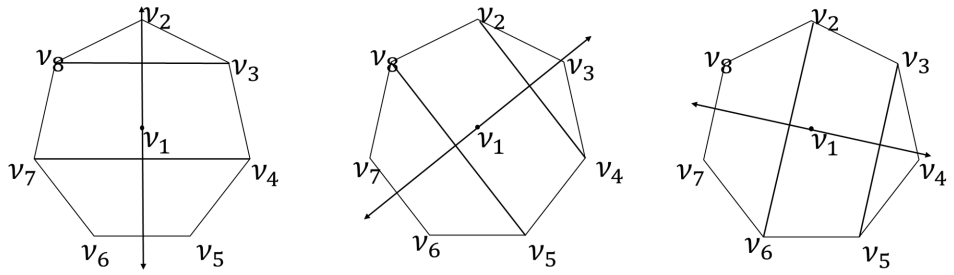


Figure 6.7: Dynamic zone allocation strategy when $n = 4m$. The zone allocation scheme cycles through matchings of the complete graph by viewing them in the form of the regular polygon.

Remark 12. *Note that each such grouping of nodes constitutes a matching (1-factor) of an m-uniform complete hypergraph on n nodes, K_m^n. Baranyai's theorem [187] states that if n is divisible by m, then, there exists a decomposition of K_m^n into $\binom{n-1}{m-1}$ 1-factors. However, we do not require every hyperedge to be covered by the allocation scheme, but only for every node to be grouped with every other node eventually.*

Note that for $m = 2$, it shows that we can decompose a graph into $n-1$ different matchings. In this case, the handshake problem is the same as the decomposition of the graph into matchings. Thus Baranyai's theorem in this case gives us the exact number of slots to solve the problem.

We use the tightness observed for the two-way handshake problem to design a strategy to assign the peers in zones. Let $n' = \frac{n}{m}$. Partition the peers into $2n'$ sets, each containing $m/2$ peers. Let these sets be given by $\nu_1, \ldots, \nu_{2n'}$. Then, we can use matchings of $K_{2n'}$ to perform the zone allocation.

Consider Alg. 13. The algorithm provides a constructive method to create zones such that all peers are grouped with each other over time. The functioning of the algorithm is as in Fig. 6.7.

Lemma 34. *The number of slots required for every peer to be grouped with every other peer is $2n' - 1$.*

Proof. The result follows directly from the cyclic decomposition and Baranyai's theorem for $m = 2$. □

We see that the scheme matches the lower bound on the number of slots for coverage in the order sense. Thus we consider this allocation strategy in the following discussion. In addition to the order optimality, the method is also fair in its implementation to all transactions over time.

6.5.2 Security Enhancement

From Alg. 10, we know that inconsistent peers are removed from consideration for data recovery. While Lem. 34 guarantees coverage in $2\frac{n}{m}$ slots, we are in fact interested in the number of slots for all uncorrupted peers to be paired with corrupt peers. We now give insight into the rate at which this happens.

We know that an adversary who wishes to corrupt a block corrupts at least $n/2$ nodes originally.

Lemma 35. *Consider an adversary who successfully corrupts W_t to W_t'. Further, let us assume the adversary requires successful corruption with probability at least $1 - \epsilon$, where $1 - \epsilon > \frac{1}{p}$. Then under the cyclic zone allocation scheme, the adversary needs to corrupt at least m new nodes with probability at least $1 - \frac{1}{p}$, in order to successfully alter H_t.*

Proof. From the cyclic zone allocation strategy and the pigeonhole principle at least two honest nodes in the graph are paired with corrupt nodes in each slot. These nodes are to be corrupted by the adversary in order to preserve hash consistency if the corresponding hash value changes.

By the collision resistance property of the hash family, the probability that the corruption of a block W_t corrupts H_t is at least $1 - P_{\text{collision}} = 1 - \frac{1}{p}$. Thus the result follows. □

Suppose the adversary corrupts a peer independently with probability $P_{\text{tc}} \in (0, 1)$. This probability represents the ability of the adversary to corrupt other peers in the network.

Theorem 25. *Let the depth explored for consistency to return a data block be d. The probability of successful targeted corruption of such a system is*

$$\mathbb{P}\left[\textit{Targeted Corruption}\right] \leq \binom{r}{r/2} P_{tc}^{\left(\frac{n}{2}+2m\right)} \frac{\left[1 - \left(P_{tc}^{2m}(1 - 1/p)\right)^{d}\right]}{p\left[1 - (1 - 1/p)^{d}\right]}, \tag{6.27}$$

where $r = n/m$.

Proof. Note that the adversary successfully corrupts only if it manages to corrupt as many as $\frac{n}{2} + 2Km$ peers where K is the random variable that represents the depth at which the hash value is not altered by a change to the previous block. From Lem. 35 and the fact that $K \leq d$, it is evident that K is distributed according to the geometric distribution with parameter $P_{\text{collision}}$, and truncated at d.

Thus, the probability of successful targeted corruption is given by

$$\begin{aligned}\mathbb{P}\left[\text{Targeted Corruption}\right] &= \mathbb{E}_K\left[\binom{r}{r/2}(P_{tc})^{\frac{n}{2}+2Km}\right] \\ &= \binom{r}{r/2}(P_{tc})^{\frac{n}{2}}\mathbb{E}_K\left[\exp\left((2m\log P_{tc})K\right)\right]. \end{aligned} \tag{6.28}$$

Finally, the result is obtained by noting that the moment generating function of the geometric random variable with parameter ρ is

$$M(t) = \frac{\rho e^t}{1-(1-\rho)e^t},$$

and that for the truncated random variable is

$$M'(t) = M(t)\left[\frac{1-\left(e^t(1-\rho)\right)^d}{1-(1-\rho)^d}\right].$$

□

Naturally this implies that in the worst-case, with $\frac{n}{2m}$ transactions, the data becomes completely secure in the network. That is, only a corruption of *all peers* (not just a majority) leads to a consistent corruption of the transaction.

6.6 Data Recovery and Repair

6.6.1 Recovery Cost

As highlighted in the scheme, the recovery process of a transaction block or hash value requires the participation of all peers in the zones. Thus the cost of data recovery using the (Φ, Ψ)-encryption scheme is

$$R_r^{(t)} = C_r\left(\tfrac{1}{m}\log_2 q + m(2\log_2 m + 1) + 4d_t\log_2 p\right). \tag{6.29}$$

However, practical systems often have peers that are temporarily inactive or undergo data failure. In such contexts the recovery of the data from the corresponding zone becomes infeasible.

Thus it is of interest to know the probability that it may not be feasible to recover an old transaction at any time slot. Consider a simple model wherein the probability that a peer is inactive in a slot is ρ, and peer activity across slots and peers is independent and identically distributed.

Theorem 26. *For any $\delta > 0$, probability of successful recovery of a data block at any time slot is at least $1 - \delta$ if and only if $m = \Theta(\log n)$.*

Proof. First, the probability that the data stored can be recovered at any time slot is bounded according to the union bound as follows:

$$\begin{aligned}\mathbb{P}\left[\text{Recovery}\right] &= \mathbb{P}\left[\text{there exists a zone with all active peers}\right] \\ &\leq \frac{n}{m}(1-\rho)^m.\end{aligned}$$

Thus, to guarantee a recovery probability of at least $1 - \delta$, we need

$$\frac{n}{m}(1-\rho)^m \geq 1 - \delta \tag{6.30}$$

$$\implies m \leq \frac{1}{\log\left(\frac{1}{1-\rho}\right)}\left(\log n - \log(1-\delta)\right). \tag{6.31}$$

Next, to obtain sufficient conditions on the size of the zones, note that

$$\begin{aligned}\mathbb{P}\left[\text{Failure}\right] &= \mathbb{P}\left[\text{at least one peer in each zone is inactive}\right] \\ &= (1-(1-\rho)^m)^{\frac{n}{m}} \leq \exp\left(-(1-\rho)^m\frac{n}{m}\right),\end{aligned}$$

where the last inequality follows from the fact that $1 - x \leq \exp(-x)$. Hence a sufficient condition for guaranteeing an error probability of less than δ is

$$\exp\left(-(1-\rho)^m\frac{n}{m}\right) \leq \delta$$

$$\implies m \leq \frac{1}{\log\left(\frac{1}{1-\rho}\right)}\left(\log n - \log\log(1/\delta)\right). \tag{6.32}$$

Thus the result follows. □

Theorem 26 indicates that the zone sizes have to be of the order of $\log n$ to guarantee a required probability of recovery in one slot when node failures are possible.

6.6.2 Data Repair

The distributed secure storage ensures that individual entries stored at each peer can not be recovered from the knowledge of other entries in the zone. Thus it is not feasible to repair nodes locally within a zone. However, it suffices to substitute a set of bits such that the code structure is retained.

A node failure indicates that the private key is lost. Thus repairing a node involves recoding the entire zone using data from a neighboring zone. Thus owing to the encryption, it is difficult to repair nodes upon failure.

Thus, the transaction data is completely lost if one peer from every zone undergoes failure. That is, the system can handle up to n/m node failures. This fact emphasizes the need to ensure that m is small in comparison to n, so as to avoid data loss from node repair.

6.7 Blockchain-Based Cloud Storage

As mentioned earlier, parameter design for individual clients is influenced by the variety of tradeoffs established here. Blockchain-based storage systems are of interest owing to the immutability guarantees on stored data. Using the tradeoffs established here, we describe a scheme selection mechanism by which the client can opt for a service that best serves the data being stored, details of which are given in [188]. Additionally it is important to note that the clients can inherently value different data blocks differently by appropriately varying the design choices.

6.7.1 Security-Based Scheme Selection

Consider a cloud storage system that implements our code to store data on the blockchain. Without loss of generality, let us assume that the cost of storing one unit of data at all peers per unit time is one. The communication cost for data recovery, C_r is priced in relation to this. Let the frequency of data retrieval be ν, and let the parameters be k, d. Then, the storage cost per unit time is

$$\text{Service Cost} = R_s + \nu R_r \tag{6.33}$$

$$= \left(\tfrac{1}{k}\log_2 q + k(2\log_2 k + 1)\right)(1 + \nu C_r) + 2\log_2 p(1 + d\nu C_r), \tag{6.34}$$

which is obtained from (6.6) and (6.8).

Naturally, given the set of parameters, the probability of data loss, targeted corruption,

and the fraction of colluding peers for information leak are determined by the maximum value of the LP (6.22), (6.27), and f_{il} respectively.

Thus, the client can choose the design parameters by solving the following integer program

$$(k^*, d^*) \in \arg\min_{k,d} R_s + \nu R_r \tag{6.35}$$

such that

$$\mathbb{P}\left[\text{Data Loss}\right] \leq \delta_{\text{dl}}, \tag{6.36}$$

$$\mathbb{P}\left[\text{Targeted Corruption}\right] \leq \delta_{tc}, \text{ and} \tag{6.37}$$

$$f_{il} \geq \delta_{il}. \tag{6.38}$$

Note that (6.35) is a non-linear integer program and presumes knowledge of the parameters that define the adversary strength.

6.7.2 Data Insurance

Distributed storage on blockchain systems provide us with an interesting opportunity to offer data insurance [189] for saved blocks of data owing to the security guarantees. Here, we briefly describe parameter selection (storage code design) to store data valued at a certain level such that the service provider on average obtains a certain desired profit margin.

Consider storing a data block valued by the client at V. Let $\mu \in [0, 1]$ be the profit margin desired by the service provider. Let $w_1 \in [0, 1]$ be the fraction of DoS adversaries, and $w_2 = 1 - w_1$ be the fraction of active adversaries. Here we do not consider information leak through collusion. Now, in any slot, the probability that the data is lost or successfully corrupted in a slot is given by

$$\theta = w_1 \mathbb{P}\left[\text{Data Loss}\right] + w_2 \mathbb{P}\left[\text{Targeted Corruption}\right].$$

Let $R = R_s + \nu R_r$ be the service cost per unit time. The time T for data loss or successful corruption is distributed geometrically with the parameter θ. Then the expected time for payout of the insured data upon losing it is $\frac{1}{\theta}$. Thus, the service provider can select the storage parameters by solving the following problem

$$(k^*, d^*) \in \arg\min_{k,d} C, \quad \text{such that } C \geq (1 + \mu) V p. \tag{6.39}$$

Again this is a non-linear integer program that is to be solved to obtain the desired profit margin on insured data blocks.

Thus, the code provides the opportunity to design efficient data storage and insurance mechanisms over the blockchain. A variety of other applications may also be developed by studying the corresponding tradeoff in operational costs, required security and privacy, and the requirements of the application.

6.8 Discussion

This chapter introduced a simple mathematical model of blockchain systems and leveraged information theoretic secrecy, distributed storage coding, and a novel grouping mechanism to allow efficient storage of the blockchain ledger among the peers. In the process we addressed the associated cost in recovery, and also the guarantees provided by the system on the security and privacy of the data. In particular we designed secure distributed storage codes for the cold storage of the blockchain ledger in the presence of active adversaries.

Whereas we adopt a system-theoretic framework here to design the coding scheme, a closer analysis of the requirements of the system through coding theory can help develop efficient codes for the dynamic distributed storage scheme. We have recently developed a coding scheme with a view on sharing several local and a global secret among peers [190], that proves to be an efficient code for the dynamic distributed system.

Exploring other coding schemes that might be conducive for implementing the dynamic distributed storage scheme could yield more practical solutions. In particular, incorporating features of error and erasure tolerance, and local repair capabilities, through the use of appropriate locally repairable codes (LRC) [191–193] can enhance the recovery and repair costs in the network, making the storage scheme more robust.

More efficient implementations of the proposed scheme in practice also require a careful exploration of network costs and performing resource allocation in accordance to these network costs. In particular, it is important to group nodes into zones such that they can communicate easily with each other. At the same time however we also need to ensure that such familiar grouping does not facilitate the corruption of peers by adversaries. Accounting for such costs in the zone allocation algorithm can lead to better practical methods.

This work establishes the feasibility of enhancing blockchain performance and reducing associated costs through novel use of coding-theoretic techniques. We believe such methods enhance the ability of blockchain systems to scale to address practical applications in a variety of industries.

CHAPTER 7

TRUSTED MULTI-PARTY COMPUTATIONS USING BLOCKCHAIN SYSTEMS

Machine learning, data science, and large-scale computation in general has created an era of computation-driven inference, applications, and policymaking [194, 195]. Technological solutions and policies with far-reaching consequences are increasingly being derived from computational frameworks and data. Multi-agent sociotechnical systems that are tasked with working collaboratively on such tasks function by interactively sharing data, models, and results of local computation.

However, when such agents are independent and lack trust, they might not collaborate with or trust the validity of reported computations of other agents. Quite often, these computations are expensive and time consuming, and thus recomputation by the doubting peer is infeasible as a general course of action. In such systems, creating an environment of trust, accountability, and transparency in the local computations of individual agents promotes collaborative operation.

For instance, consider training a deep neural network with a given architecture using stochastic gradient descent (SGD). Here, the model and computations are deterministic given the data used for gradient computation. Applications are primarily interested in using the trained model represented by the weights of the trained network. But, if there is lack of trust in the training agent, there is no simpler way to verify the network than to retrain it. This is often impractical since the (re)training process consumes extensive amounts of time and tends to require the use of specialized hardware like GPUs or TPUs. It is thus important to establish trust in the computations involved in the training phase.

To emphasize the importance of trust in multi-agent systems, let us also consider the case of policy design for malaria. OpenMalaria (OM) [196] is an open source simulation environment, collaboratively developed to study malaria epidemiology and the effectiveness of control mechanisms. It is used extensively to design policies to tackle the disease. Here, individual agencies propose hypotheses regarding the disease and/or intervention policies, and study them by simulating them under specific environments [197]. Considering the potential impact of such work in designing disease control policies, it is important to establish accountability and transparency in the process, so as to facilitate trusted adoption of results.

Calls have been made for accountability and transparency in multi-agent computational systems, especially in high impact fields such as health [198]. A framework for decision provenance helps track the source of results, transparent computational trajectories, and a unified, trusted platform for information sharing. In fact, the US Centers for Disease Control and Prevention [199] states that:

> ... public health and scientific advancement are best served when data are released to, or shared with, other public health agencies, academic researchers, and appropriate private researchers in an open, timely, and appropriate way. The interests of the public ... transcends whatever claim scientists may believe they have to ownership of data acquired or generated using federal funds.

This call implicitly assumes an inherent trust in the shared material. However, significant disparity and inconsistency in current information-sharing mechanisms not only hinder access, but also lead to questionable informational integrity [200]. Here, trust and transparency are critical, but absent in current practice.

Establishing trust in computations translates to guaranteeing correctness of *individual steps* of the simulation, and the integrity of the overall computational process leading to the reported results. Importantly, when computational models and parameters along with intermediate results of individual steps are shared, these steps can be validated by other agents who can recompute them, thereby validating the entire computation in a distributed manner.

Blockchain is a distributed ledger (database) technology that enables multiple distributed, potentially untrusted agents to transact and share data in a safe, secure, verifiable and trusted manner through mechanisms providing transparency, distributed validation, and cryptographic immutability [37]. As such, blockchain provides a viable platform to establish this type of trust for complex, long running computations of interest. In fact, blockchain-based platforms for trusted clinical trials have been proposed recently [201]. In this chapter we use blockchain to record, share, and validate frequent audits (model parameters with the intermediate results) of individual steps of the computation. We describe how blockchain-based distributed validation and shared, immutable storage can be used and extended to enable efficient, trusted, and verifiable computations.

A common challenge arising in blockchain-based solutions is its scalability [188, 202, 203]. The technology calls for significant peer-to-peer interaction and local storage of large records that include all the data generated in the network. These fundamental requirements result in significant communication and storage costs respectively. Thus, using the technology for large-scale computational processes over large multi-agent networks may be prohibitively

expensive. Here, we address this scalability challenge by developing a novel compression schema and a distributed, parallelizable validation mechanism that is particularly suitable for such large-scale computation contexts.

Allowing validation and verification of computations not only creates an environment of trust among agents, but also enforces a higher degree of conformation and consistency in experiments. Necessitating validation and verification also implies a shared common mechanism for model and data sharing, enabling scientific reproducibility. The setup also facilitates well-defined processes for distributed and derived computing, wherein the former involves a computational framework performed piecewise at multiple nodes, and the latter concerns deriving new experiments using checkpoints drawn from the intermediate audits of prior computational experiments. The specifics of the system design described in this chapter can be found at [204, 205] and are also presented in [206–208].

7.1 Prior Work

We now provide a brief summary of prior work in related areas.

A variety of applications with widespread impact are being designed with the help of improved computational capabilities, easier access to data, and machine learning algorithms. Taking the context of malaria, as studied through OpenMalaria simulations, new pipelines for integrating AI tools and algorithms have been considered [209]. Regression-based methods for better policy search have also been integrated with the open-source platform [210].

Considering the impact of such simulations, researchers have recently raised alarm over their lack of reproducibility. Reproducing results from research papers in AI have been found to be challenging as a significant fraction of hyperparameters and model considerations are not documented [211]. In another paper focused on reproduction of results in deep learning [212], the authors explore the possible reasons, and cite variability in evaluation metrics and reporting among different algorithms and implementations. More recently efforts have been made at coupling the data with the processes involved to facilitate scientific reproducibility [213].

Accountability and transparency are being increasingly sought after in large-scale computational platforms, with particular focus on establishing tractable, consistent computational pipelines. The problem of establishing provenance in decision-making systems has been considered [214] through the use of an audit mechanism. Distributed learning in a federated setting with security and privacy limitations has also been considered recently [215]. Accountability in peer-to-peer, cloud, and distributed computating systems in the presenc of

Byzantine faults have also been considered [216–219].

The notion of trust has been considered from a variety of standpoints [220] and has contextually varied definitions as described in depth in [221]. A qualitative definition of trust in multi-agent computational systems can be adapted from [222, 223] as:

> *Trust* is the belief an agent has that the other party will execute an agreed upon sequence of actions and reports an accurate representation of computed result (being honest and reliable).

We provide a more specific characterization of trust in Sec. 7.2.

In fact, the problem of trust in multi-agent computational systems was considered at the beginning of the 20th century from the viewpoint of reducing errors in complex calculations performed by human workers [224]. Large-scale computational problems were solved using redundant evaluation of smaller sub-tasks assigned to human workers, and verified using computational checkpoints. We can draw significant insight into reliable distributed computing from these practices.

Blockchain systems have brought forth the means for creating distributed trust in peer-to-peer networks for transactional systems [174]. A variety of applications that invoke interactions and transactions among untrusting agents have benefited from the trust enabled by blockchains [156, 157, 225]. More recently, blockchains have been used in creating secure, distributed, data sharing pipelines in healthcare [226] and genomics [227]. This trust can also be leveraged in creating trusted distributed computing systems, as highlighted in this work.

7.2 Computation and Trust Model

Let us now mathematically formalize the computation model, and validation and verification requirements under consideration. We consider an iterative computational algorithm.

Consider a computational process that updates a system state, $X_t \in \mathbb{R}^d$, over iterations $t \in \{1, 2, \ldots\}$, depending on the current state and an external source of randomness $\theta_t \in \mathbb{R}^{d'}$, according to an atomic operation $f : \mathbb{R}^d \times \mathbb{R}^{d'} \to \mathbb{R}^d$ as follows

$$X_{t+1} = f(X_t, \theta_t). \tag{7.1}$$

For simplicity we assume that θ_t is common randomness that is shared by all agents. This can easily be generalized as elaborated later. We also assume that the function $f(\cdot)$ is L-Lipschitz continuous in the system state under the Euclidean norm without loss of generality,

for all $\theta \in \mathbb{R}^{d'}$ i.e.,

$$\|f(X_1, \theta) - f(X_2, \theta)\| \leq L \|X_1 - X_2\|. \tag{7.2}$$

That is, minor modifications to the inputs of the atomic operation result in correspondingly bounded variation in the outputs. This is expected off physical or biological processes as most physical systems governing behavior in nature are smooth. This is also reflected in the epidemiological and meteorological simulations that are used. For instance, with respect to the OpenMalaria example, the requirement implies that minor changes in policies result in minor changes in outcomes.

In this context we decompose trust into two main components:

- **Validation:** The individual atomic computations of the simulation are guaranteed and accepted to be correct.
- **Verification:** The integrity of the overall simulation process can be checked by other agents in the system.

The two elements ensure local consistency of computation and post-hoc corroboration of audits. Their mathematical characterization is provided in Sec. 7.2.

We consider a multi-agent system where one agent, referred to as the *computing client* (client in short), runs the computational algorithm. The other agents in the system, called *peers*, are aware of the atomic operation $f(\cdot)$ and share the same external randomness and hence can recompute the function. Validation of intermediate states is performed by independent peers referred to as *endorsers* through an iterative recomputation of the reported states from the most recent validated state using the atomic operation $f(\cdot)$. The process of validation is referred to as an *endorsement* of the state. A reported state, $\tilde{X}_t$ is *valid* if it lies within a margin, Δ_{val}, of the state $\hat{X}_t$ as recomputed by the endorser, i.e.,

$$\left\|\tilde{X}_t - \hat{X}_t\right\| = \left\|\tilde{X}_t - f(\tilde{X}_{t-1})\right\| \leq \Delta_{\text{val}}. \tag{7.3}$$

The validation criterion (7.3), without loss of generality, associates equal weight to each component of the state, and can be easily generalized to weighted norms or other notions of distance.

Verification involves checking for integrity of the computational process, which is enabled through the storage of frequent audits of validated states. Thus, if the audits record the states $\left\{\tilde{Y}_1, \tilde{Y}_2, \ldots\right\}$, then verification corresponds to ensuring that the recomputed version, $\hat{Y}_t$, of the state is within a margin, Δ_{ver}, of the recorded version, i.e.,

$$\left\|\hat{Y}_t - \tilde{Y}_t\right\| \leq \Delta_{\text{ver}}. \tag{7.4}$$

Without loss of generality, requirements for validation are stricter than for verification, i.e., $\Delta_{\text{val}} \leq \Delta_{\text{ver}}$. We now construct a system to address these two trust requirements.

7.3 Multi-Agent Blockchain Framework

A naive solution is to validate each step (iteration) of the process using independent recomputation by validating agents. The integrity of the computational process can be verified from an immutable record of validated intermediate states. However, practical simulations are long and involve a large number of iterations. Validation requires communication of the iterates to the endorsers, and recording the validated state on the immutable data structure. This results in significant communication and storage overhead if every state is reported and stored as is, in addition to the computational cost of validation. Such costs limit the adoption of the method to large-scale systems.

It is thus important to make use of the computational struture to reduce the inter-peer communication and storage, and the recomputation costs. In this section we elaborate the design of the Multi-agent Blockchain Framework (MBA), starting with the functional categorization of the network. We then elaborate each functional unit, including the compression at the client, the validation by endorsers, and the role of orderers in adding blocks to the ledger. For ease, let us consider a deterministic iterative algorithm for computation, $X_{t+1} = f(X_t)$.

7.3.1 Peer-to-Peer Network—Functional Decomposition

The peer-to-peer network is functionally categorized into clients, endorsers, and orderers, which function together in computing, validating, ordering, and storing the simulated states on the blockchain ledger. Note that the sequence of operations is analogous to those involved in the Hyperledger Fabric as indicated in Chapter 6. Their functions are:

1. Client runs the computations to iteratively compute states $\{X_t : t \geq 1\}$.
2. The client groups a sequence of states into a frame, compresses, and communicates the frame to a set of endorsers.
3. The endorsers decompress frames, validate states by recomputing them iteratively, and report endorsements to orderers.
4. The orderers subsample and add the frame to the blockchain if it has been validated, and if all prior frames have been validated and added.

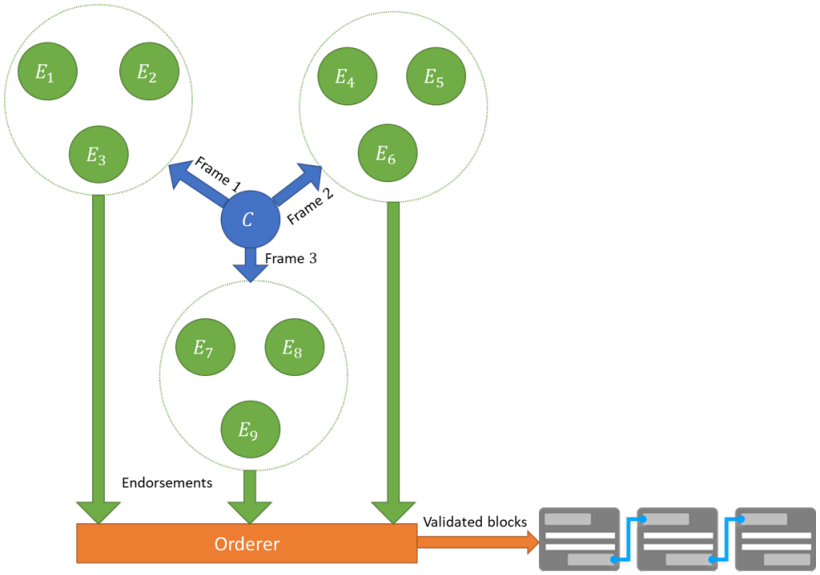


Figure 7.1: Functional categorization of peer-to-peer network: Clients run the iterative algorithm; multiple independent frames are validated in parallel by non-overlapping subsets of endorsers; orderers check consistency and append valid frames to the blockchain.

5. The peers update their copy of the ledger.

This is depicted in Fig. 7.1. The classification is only based on function and each peer can perform different functions across time. Since states are grouped into independent frames, they can be validated by non-overlapping subsets of endorsers in parallel.

7.3.2 Client Operations

Clients performs the computations, construct frames of iterates, compress, and report them to endorsers. We assume there exists an endorser assignment policy.

Owing to the Lipschitz continuity,

$$\|X_{t+1} - X_t\| \leq L \|X_t - X_{t-1}\| .$$

Thus state updates (differences) across iterates are bounded to within a factor of the deviation in the previous iteration. This property can be leveraged to compress state updates using delta encoding [228], where states are represented in the form of differences from the previous state. Then, it suffices to store the state at certain checkpoints of the computational process, with the iterates between checkpoints represented by the updates.

We describe the construction inductively, starting with the initial state X_0, the first checkpoint. Let us assume that the state reported at time t is $\tilde{X}_t$ and the true state is X_t. Then,

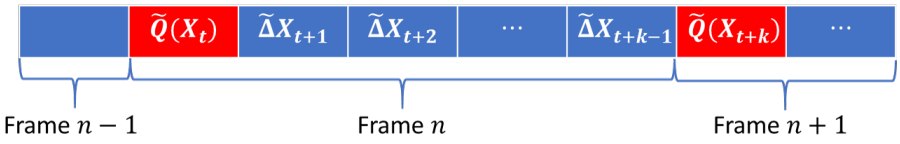


Figure 7.2: Structure of frames: Each frame includes a header followed by compressed updates of successive iterates.

if $X_{t+1} = f(X_t)$, define the update as

$$\Delta X_{t+1} = X_{t+1} - \tilde{X}_t.$$

The cost of communication (for validation) and storage (for verification) of these updates is reduced by performing lossy compression (vector quantization [229]). Let the quantizer be represented by $Q(\cdot)$ and let the maximum quantization error magnitude be ϵ, i.e., if the client reports $\tilde{\Delta}X_t = Q(\Delta X_t)$, then,

$$\left\| \tilde{\Delta}X_t - \Delta X_t \right\| \leq \epsilon. \tag{7.5}$$

Additionally, the checkpoints can also be compressed using a Lempel-Ziv-like dictionary-based lossy compressor. Here, a dictionary of unique checkpoints are maintained. For each new checkpoint, we first check if the state is within a margin ϵ from an entry in the dictionary, and the index of this entry is reported. If not, the state is added to the dictionary and its index is reported. Other universal vector quantizers can also be utilized for compressing checkpoints, and we denote this quantizer by $\tilde{Q}(\cdot)$.

Let Δ_{quant} be the maximum magnitude of a state update within a frame, i.e., if $\|\Delta X_t\| > \Delta_{\text{quant}}$, the client creates a checkpoint at $t+1$ and reports $\tilde{X}_{t+1} = \tilde{Q}(X_{t+1})$. Then X_{t+1} is reported as

$$\tilde{X}_{t+1} = \begin{cases} \tilde{Q}(X_{t+1}), & \text{if } t+1 \text{ is a checkpoint} \\ \tilde{X}_t + \tilde{\Delta}X_{t+1}, & \text{o/w} \end{cases}. \tag{7.6}$$

The resulting sequence of frames is as shown in Fig. 7.2.

Separate from creating new checkpoints adaptively, the system also restricts the maximum size of a frame by a constant $\bar{M}$ to limit the computational overhead of its validation. Figure 7.3 summarizes the tasks performed by the computing client.

The choice of design parameters, $\epsilon, \Delta_{\text{quant}}$, are to be made such that the reports are accurate enough for validation. The optimal design choice is shown in the following results.

Theorem 27. *If $f(\cdot)$ is L-Lipschitz continuous, and $\epsilon \leq \frac{\Delta_{val}}{L+1}$, then, a state $\tilde{X}_t$ is invalidated*

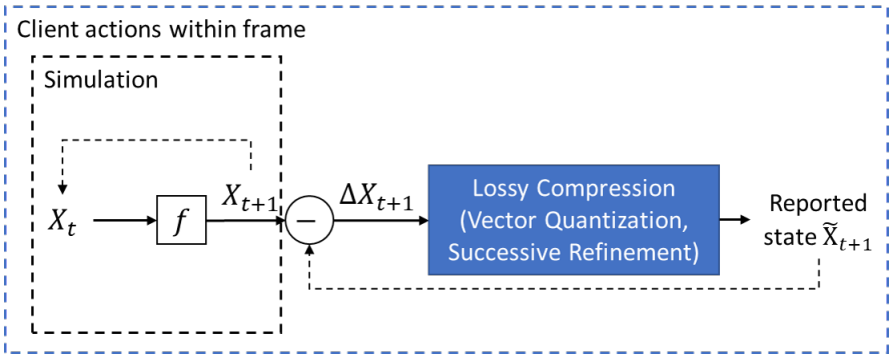


Figure 7.3: Operations performed by the client within a frame. The client computes the states according to the iterative algorithm, performs delta encoding, and communicates the compressed state updates to the endorsers in frames.

by an honest endorser only if there is a computational error of magnitude at least ϵ, i.e., $\left\|\tilde{X}_t - X_t\right\| \geq \epsilon$.

Proof. Let us assume that $\tilde{X}_t$ is a valid state. Then,

$$\begin{aligned}
\left\|\hat{X}_{t+1} - \tilde{X}_{t+1}\right\| &\leq \left\|\hat{X}_{t+1} - X_{t+1}\right\| + \left\|X_{t+1} - \tilde{X}_{t+1}\right\| \\
&\leq L\left\|\tilde{X}_t - X_t\right\| + \left\|\tilde{\Delta}X_t - \Delta X_t\right\| && (7.7) \\
&\leq L\left\|\tilde{\Delta}X_{t-1} - \Delta X_{t-1}\right\| + \epsilon && (7.8) \\
&\leq (L+1)\epsilon \leq \Delta_{\text{val}}, && (7.9)
\end{aligned}$$

where (7.7) follows from the Lipschitz continuity of the atomic operation, (7.8) is from the definition of the compressed state updates, and (7.9) follows from the quantization error bound. □

Corollary 11. *If $\left\|\tilde{X}_t - X_t\right\| \geq \Delta_{val} + L\epsilon$, then $\tilde{X}_t$ is invalidated.*

The necessary and sufficient conditions for invalidation in Thm. 27 and Cor. 11 highlight the fact that computational errors of magnitude less than ϵ are missed, and any error of magnitude at least $\Delta_{\text{val}} + L\epsilon$ is certainly detected. When the approximation error is made arbitrarily small, all errors beyond the tolerance are detected. A variety of vector quantizers, satisfying Thm. 27 can be used for lossy delta encoding—one simple choice is lattice vector quantizers [230].

Theorem 28. *Let $\mathcal{B}(\Delta_{quant}) = \left\{x \in \mathbb{R}^d : \|x\| \leq \Delta_{quant}\right\}$ and let $\Delta X_t \sim Unif(\mathcal{B}(\Delta_{quant}))$. Then, the communication and storage cost per state update is $O\left(d \log\left(\frac{\Delta_{quant}}{\epsilon}\right)\right)$ bits.*

Proof. This follows directly from the covering number of $\mathcal{B}(\Delta_{\text{quant}})$ using $\mathcal{B}(\epsilon)$ balls [231]; a similar cost is incurred for other standard lattices. □

Theorem 29. *For any frame n, with checkpoint at T_n, the maximum number of states in the frame, M_n, is bounded as*

$$M_n \leq \min\left\{\frac{\log\left(\Delta_{quant}-\epsilon\right)-\log\delta_n}{\log L}, \bar{M}\right\}, \tag{7.10}$$

where $\delta_n = \|X_{T_n+1} - X_{T_n}\|$, is the first update in the frame.

Proof. Without loss of generality, let us consider the first frame, i.e., $n = 1, T_n = 0$. Then, $M_n = t$ implies that

$$\begin{aligned}
\Delta_{\text{quant}} &\leq \|\Delta X_t\| = \left\|X_{t+1} - \tilde{X}_t\right\| \\
&\leq \|X_{t+1} - X_t\| + \left\|X_t - \tilde{X}_t\right\| \qquad (7.11)\\
&\leq L^t \|X_1 - X_0\| + \epsilon = L^t\delta_1 + \epsilon, \qquad (7.12)
\end{aligned}$$

where (7.11) follows from the triangle inequality, and (7.12) follows from the Lipschitz continuity and the quantization error. Thus the result follows, for any frame, by direct extension. □

This provides a simple sufficient condition on the size of a frame, in terms of the magnitude of the first iteration in the frame. Naturally a small first iterate implies the possibility of accommodating more iterates in the frame. This lower bound can be used in identifying the typical frame size and the corresponding costs of communication and computation involved, prior to the design of the scheme. We describe a generalization of the compressor to the parameter-unaware setting in Sec. 7.5.

Note that longer frames imply fewer frames communicated to endorsers for validation and storage. This in turn reduces the communication delays, and the storage and communication overheads. However, if a frame is long, then it also implies that all subsequent states in that frame need to be recomputed if the frame is invalidated. Additionally, longer frames also imply larger delays for validation. Shorter frames on the other hand do not make efficient usage of the communication bandwidth.

7.3.3 Endorser and Orderer Operations

We now define the role of an endorser in validating a frame. A summary of the operations is depicted in Fig. 7.4. For preliminary analysis, we assume that endorsers are honest and are

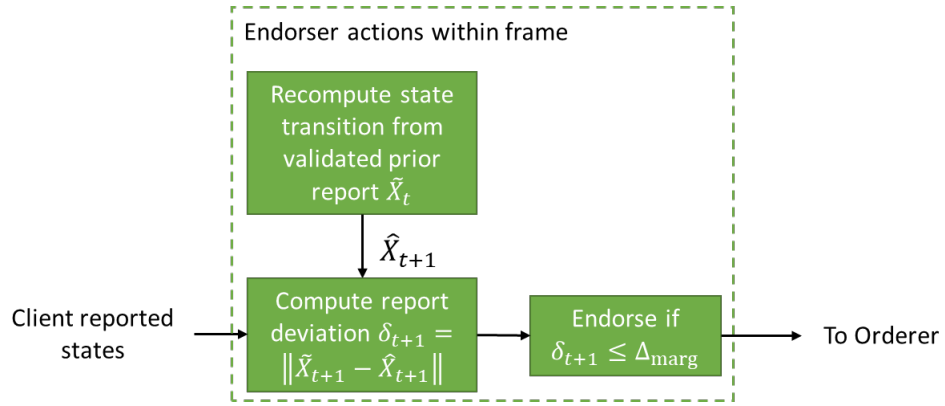


Figure 7.4: Operations performed by the endorser for a single frame: The endorser sequentially validates the states reported by the client by decoding the updates and recomputing the state from the atomic operation.

homogeneous in terms of communication latency and computational capacity. A more refined allocation policy can be designed to account for the case of variabilities in communication and computational costs.

Each endorser involved in validating a frame sequentially checks the state updates by recomputing from the last valid state, i.e., to validate the report $\tilde{X}_{t+1}$, the endorser computes $\hat{X}_{t+1} = f(\tilde{X}_t)$ and checks for the validity criterion (7.3). The frame is reported as valid if all updates are valid in the frame. The endorsements are then reported to the orderer.

Individual update validations can also be performed in parallel and finally verified for sequential consistency. Such parallelism can be performed either at the individual endorser-level, or in the form of the distribution of the sub-frames across endorsers through coded computing. This results in a reduction of the time required for validating a frame.

Upon receiving the endorsements for frames, the orderer checks for consistency of the checkpoints and adds a valid frame to the ledger if all prior frames have been added, and broadcasts it to other peers as in Fig. 7.5.

Since the state updates are stored on the immutable data structure of the blockchain, they provide an avenue for verification of the computations at a later stage. As described in (7.4), the verification requirements are not as strict as the validation requirements. Thus it suffices to subsample the updates in a frame and store only one for every $K = \frac{\Delta_{\text{ver}}}{\Delta_{\text{val}}}$ iterates. Then, the recorded state updates are the sum of the K intermediate updates.

A block stored on the blockchain now contains audits that are either the checkpoints or the cumulative updates corresponding to K successive iterates. The audits $\tilde{Y}_\tau$ are then defined

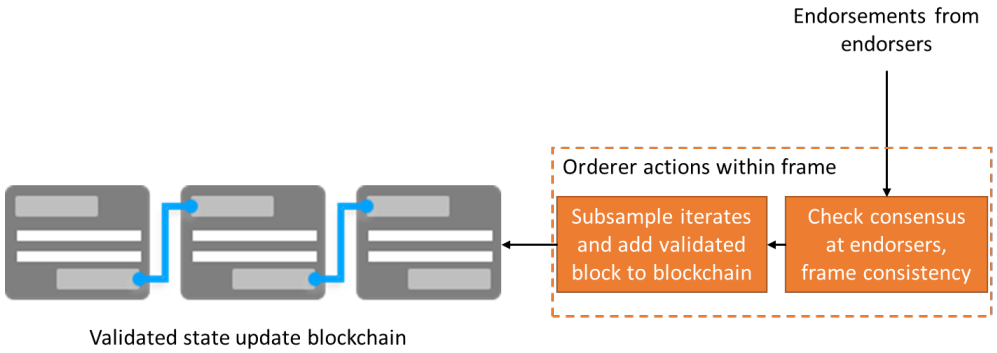


Figure 7.5: Operations performed by the orderer for frames. The orderer sequentially adds valid frames to the blockchain after checking for consistency.

by

$$\tilde{Y}_{\tau+1} = \begin{cases} \tilde{Y}_\tau + \sum \tilde{\Delta} X_{t+1}, & \text{if no checkpoint in next } K \text{ iterates} \\ \tilde{X}_{t'}, & \text{otherwise,} \end{cases} \tag{7.13}$$

where the sum is over the intermediate iterates, and t' is the next checkpoint. These audits are grouped into blocks as in Fig. 7.2 and added to the ledger.

Theorem 30. *For subsampled storage at frequency* $1/K$ *according to* (7.13)*, Lipschitz constant* L *of* $f(\cdot)$*, and quantization error* ϵ*,*

$$\left\| \hat{Y}_{\tau+1} - \tilde{Y}_{\tau+1} \right\| \leq \left(L^K + 1 \right) K\epsilon, \tag{7.14}$$

where $\hat{Y}_{\tau+1} = f^K(\tilde{Y}_\tau)$.

Proof. First, $f^K(\cdot)$ is L^K-Lipschitz continuous. Then,

$$\begin{aligned} \left\| \hat{Y}_{\tau+1} - \tilde{Y}_{\tau+1} \right\| &\leq \left\| f^K(\tilde{Y}_\tau) - f^K(Y_\tau) \right\| + \left\| Y_{\tau+1} - \tilde{Y}_{\tau+1} \right\| && (7.15) \\ &\leq L^K \left\| \tilde{Y}_\tau - Y_\tau \right\| + \left\| \sum_{t \in \mathcal{T}_\tau} \left(\tilde{\Delta} X_t - \Delta X_t \right) \right\| && (7.16) \\ &\leq (L^K + 1) K\epsilon, \end{aligned}$$

where (7.15) follows from the triangle inequality, and (7.16) follows from the Lipschitz inequality and (7.13). □

Thus, a viable subsampling frequency can be determined by finding a K such that $(L^K + 1)K \leq \frac{\Delta_{\text{ver}}}{\epsilon}$. This reduces the storage cost on the blockchain at the expense of accuracy of

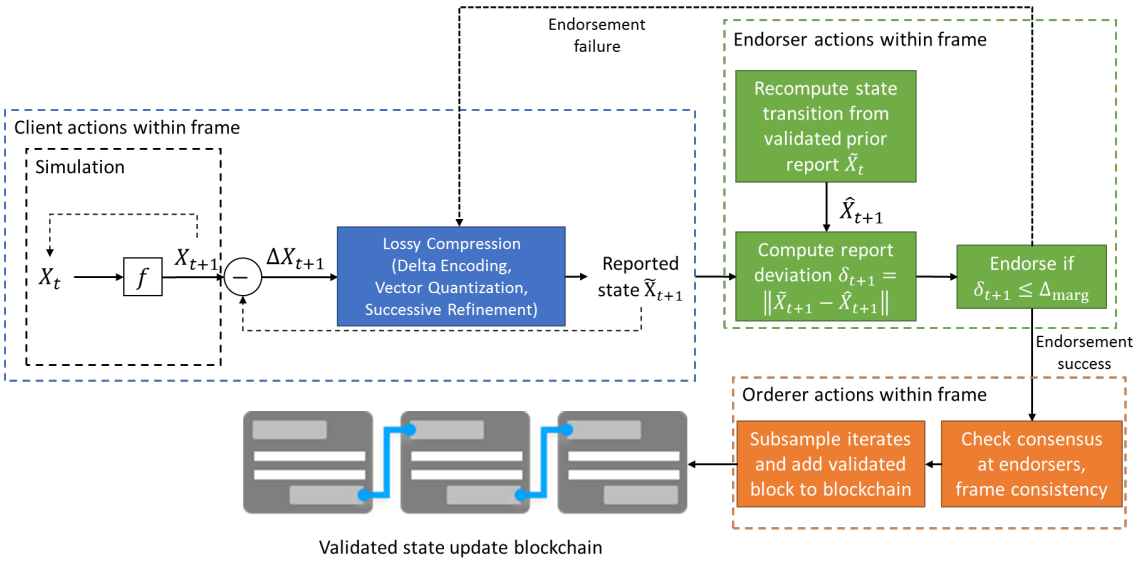


Figure 7.6: Overall computational system model.

recorded audits. If the agents are interested in increasing accuracy of the records over time, then the quantizers can be dynamically adjusted accordingly.

7.3.4 Example Application

Let us now elaborate the design, as shown in Fig. 7.6, from the context of a synthetic example. Consider an agent, the client, in a network that wishes to address a simple classification problem using training data that it has access to. The client aims to train a neural network using backpropagtion based on mini batch stochastic gradient descent (SGD) to solve this classification problem and subsequently share the trained network with other agents that are also interested in the solution.

However, the client is limited by the amount of computational resources available for training, and also does not wish to share the private data used for training. Since it has limited resources for computing gradients, it uses small batch sizes to get faster estimates. However, the peers do not trust the computations performed by the client, not just because of its proclivity toward errors arising from computational limitations, but also possible malicious intent. The peers themselves have access to private datasets, drawn from the same source, but much smaller in size such that they can not train a network on their own.

In such a context, we establish distributed trust using MBA as follows:

1. The client sets up the training with parametric inputs (network architecture, learning rates, batch sizes etc.) and shares them over the blockchain with other peers.

2. The client runs the training algorithm, reports network weights using lossy compression to endorsers for validation.

3. The endorsers rerun steps of mini batch SGD from the validated prior architecture. They compute deviation in network weights, endorse according to (7.3), and communicate validated frames to the orderer.

4. MBA orderer checks for consensus among endorsers, reconstructs subsampled blocks and add blocks sequentially to the blockchain ledger.

5. Client reports the network to peers at the end of training.

Since the experiment is run on the MBA platform, peers are assured of validity of steps of the training, and also have access to the blockchain to verify the computations. Since the private training data is not shared across peers, the endorsement process for revalidation needs to be appropriately adjusted. This is described in Sec. 7.5, and a detailed experimental study of this problem, adapted to the MNIST dataset, is done in Sec. 7.6.

7.4 Design Advantages and Costs

We now perform a cost-benefit analysis of the design. We benchmark the system costs against simpler implementations to emphasize the importance of the different components of the system.

Let us first identify the advantages of the platform.

- **Accountability:** MBA guarantees provenance through the immutable record of computations. Thus, we can not only detect the source of potential conflicts, but also to trace ownership of computations.

- **Transparency:** MBA establishes trust among agents through a transparent record of the validated trajectories of computation.

- **Adaptivity:** The frame design, endorsement, and validation methods adapt according to the state evolution. Further, the validity margins can be altered across time by dynamically varying the quantizers. In convergent simulations/algorithms, the system can thus use monotonically decreasing margins to obtain stricter guarantees at convergence.

- **Generality:** The platform uses fairly general building blocks, and can be easily implemented using existing methods.

- **Computation universality:** The design is agnostic to computational process specifics and can be implemented as long as it is composed of reproducible atomic computations.
- **Scientific reproducibility:** By storing intermediate states MBA guarantees reliable data and model sharing, and collaborative research, facilitating scientific reproducibility in large-scale computations.

To compare system costs, let us consider three different modes:

1. **Transaction Mode:** Here we treat each iteration as a transaction, and validate and store each state transition as a block on the blockchain.
2. **Streaming Mode:** Here each state is independently compressed according to a universal compressor, validated, and stored on the blockchain.
3. **Batch Mode:** This corresponds to the MBA design described here.

Let us assume that the average number of endorsers per frame is $\bar{E}$, the average frame size is $\bar{M}$, and the subsampling frequency is $\nu = 1/K$ in the batch mode. We benchmark costs relative to this average set of $\bar{M}$ iterations, and the same computational redundancy.

First, let us consider the computational overhead involved. Each mode performs $(1 + E)$-times as many computations as the untrusted simulation. The streaming and batch modes additionally incur the cost of compression and decompression of states. The batch mode also includes the cost of subsampling the frames. Thus we can see that the transaction mode incurs the least computational overhead, while the batch mode incurs the most. Informally, the batch mode incurs a cost of

$$C_{\text{batch}}^{(\text{comp.})} = (1 + E)C_{\text{sim}}^{(\text{comp.})} + C_{\text{compression}}^{(\text{comp.})} + C_{\text{sampling}}^{(\text{comp.})}, \tag{7.17}$$

where $C_{\text{sim}}^{(\text{comp.})}, C_{\text{compression}}^{(\text{comp.})}, C_{\text{sampling}}^{(\text{comp.})}$ are the computational costs of the simulation, compression and decompression, and subsampling respectively.

The communication overheads include the state reports and metadata used for validation and coordination respectively. In the transaction mode, as states are uncompressed, the communication cost is significant and is not scalable. On the other hand, the streaming and batch modes reduce these costs through lossy compression. Assuming a bounded set of states, $\mathcal{X}$, such that $\max_{X \in \mathcal{X}} \|X\| = B \gg \Delta_{\text{quant}}$, the worst-case sufficient communication cost in streaming mode using vector quantization for $\bar{M}$ iterations is

$$C_{\text{transaction}}^{(\text{comm.})} = O\left(\bar{M} d \log_2\left(\frac{B}{\epsilon}\right) + \bar{M} C_{\text{meta}}^{(\text{comm.})}\right), \tag{7.18}$$

where $C_{\text{meta}}^{(\text{comm.})}$ is the average metadata cost per communication instance. On the other hand, the batch mode reduces both compression cost, and the metadata, as

$$C_{\text{batch}}^{(\text{comm.})} = O\left(\bar{M} d \log_2\left(\frac{\Delta_{\text{quant}}}{\epsilon}\right) + d \log_2\left(\frac{B}{\epsilon}\right) + C_{\text{meta}}^{(\text{comm.})}\right). \tag{7.19}$$

Costs expressed are sufficient costs in the order sense and more precise estimates can be computed given the compressor and state evolution statistics.

The storage cost for the transaction and streaming modes are the same as their communication costs. The batch mode not only incurs less metadata for storage but also fewer state updates due to subsampling when compared to the streaming mode. To be precise,

$$C_{\text{batch}}^{(\text{storage})} = O\left(\nu \bar{M} d \log_2\left(\frac{\Delta_{\text{quant}}}{\epsilon}\right) + d \log_2\left(\frac{B}{\epsilon}\right) + C_{\text{meta}}^{(\text{storage})}\right). \tag{7.20}$$

Thus the batch mode reduces communication and storage overheads at the expense of added computational cost. Through careful tradeoff analyses we can adopt optimal compression and subsampling mechanisms.

7.5 Extensions of Design

We now describe a couple of avenues for generalization.

7.5.1 Parameter Agnostic Design

In Sec. 7.3 we used vector quantizers based on the Lipschitz constant L. In practice, such parameters of the computation are unknown *a priori*. Underestimating L can result in using a larger quantization error that could cause errors in validation even when the client computes correctly. In such cases, it is essential to be able to identify the cause for the error. One option is to estimate L from computed samples. This translates to estimating the maximum gradient magnitude for the atomic operation, which might be expensive in sample and computational complexity, depending on the application. Thus, we propose an alternative compression scheme.

We draw insight from video compression strategies, and propose the use of successive refinement coding [232] of the state updates. That is, a compression bit stream is generated for each state update such that the accuracy can be improved by sending additional bits from the bit stream. Successive refinement allows the clients to update the reports such

that the state accuracy can be iteratively improved, in the event of invalidation. That is, if a frame is invalidated, the client has two options—checking the computations, and/or refining the reported state through successive refinement. Depending on the computation-communication tradeoff, the client appropriately chooses the more economical alternative. Through successive refinement, the client provides more accurate descriptions of the state vector, and thus reduces the possibility of validation errors caused by report inaccuracy.

One possible efficient compression technique uses lattice vector quantizers [233, 234] to define successive refinement codes. This also reduces the size of the codebook, if the refinement lattices are assumed to be of the same geometry, because the client only needs to communicate the scaling corresponding to the refinement. This allows for improved adaptability in the refinement updates. More efficient quantizers can also be defined if additional information regarding the application and state updates are available.

7.5.2 Computations without Common Randomness

As described in Sec. 7.2, such computational algorithms in practice typically evolve iteratively as a function of the current state X_t, and an external randomness θ_t. When this randomness is not shared across agents, and is inaccessible to the client, reproduction of the reported results by an endorser becomes infeasible and so is validating local computations. This could also emerge in cases where the client is unwilling to share private data associated used by the algorithm with other agents [215].

For instance, in simulations of disease spread using black box models, each run of the simulation adopts a different outcome, depending on the underlying random elements introduced by the model to mimic societal and pathological disease spread factors. Quite often, the client does not have access to all of these random elements. However, the source of such randomness is often common, i.e., $\theta_t \overset{i.i.d.}{\sim} P_\theta$, and P_θ is known. In this context, we redefine validation as guaranteeing (7.3) with probability at least $1-\rho$, i.e.,

$$\mathbb{P}\left[\left\|\tilde{X}_t - \hat{X}_t\right\| \geq \Delta_{\text{marg}}\right] \leq \rho. \tag{7.21}$$

This requirement removes outliers in the computation process and only allows trajectories close to the expected behavior.

Then, we can exploit the law of large numbers to validate reports by their deviation from the average behavior observed across independent endorsers,

$$\hat{X}_{t+1} = \frac{1}{m}\sum_{i=1}^{m} f(\tilde{X}_t, \theta_i),$$

where $\theta_i \overset{\text{i.i.d.}}{\sim} P_\theta$. By choosing a sufficiently large number of endorsers, depending on ρ, we can assure (7.21). The role of the endorsers is appropriately modified and the system calls for higher coordination among the endorsers.

We derive a sufficient condition on the number of endorsers using multi-variate concentration inequalities.

Theorem 31. *Let $\epsilon < \frac{\Delta_{marg}}{L+1}$. For a state at time t, if the average of m endorsers is used for validation,*

$$\mathbb{P}\left[\left\|\tilde{X}_t - \hat{X}_t\right\| \geq \Delta_{marg}\right] \leq \frac{2d\tilde{\lambda}^2}{(\Delta_{marg} - (L+1)\epsilon)^2}\left(1 + \frac{1}{m\tilde{\lambda}}\right)^2, \tag{7.22}$$

where $\tilde{\lambda}$ is the maximum eigenvalue of covariance matrix of the quantized state vector.

Proof. Given a d-dimensional random variable X with mean $\mu = \mathbb{E}[X]$ and covariance matrix $V = \mathbb{E}\left[(X-\mu)(X-\mu)^T\right]$, by the multidimensional Chebyshev inequality [235],

$$\mathbb{P}\left[\|X-\mu\|_{V^{-1}} > t\right] \leq \frac{d}{t^2}.$$

Then, using the fact that $\lambda_{min}\|x\| \leq \|x\|_A \leq \lambda_{max}\|x\|$, for any vector x and matrix A with minimum and maximum eigenvalues $\lambda_{min}, \lambda_{max}$, we have

$$\mathbb{P}\left[\|X-\mu\| > t\right] \leq \frac{\lambda^2 d}{t^2}, \tag{7.23}$$

where λ is the maximum eigenvalue of V.

Further, from the bound on the quantization error, we can observe that for $\tilde{X}_{t+1} = X_{t+1} + Z_t$, where $Z_t \in \mathcal{B}(\epsilon)$. Here, $\mathcal{B}(\epsilon) = \left\{x \in \mathbb{R}^d : \|x\| \leq \epsilon\right\}$. Then,

$$\left\|\mathbb{E}[X_{t+1}] - \mathbb{E}\left[\tilde{X}_{t+1}\right]\right\| \leq \epsilon.$$

Then,

$$\begin{aligned}
\left\|\mathbb{E}\left[\tilde{X}_{t+1}\right] - \mathbb{E}\left[\hat{X}_{t+1}\right]\right\| &= \left\|\mathbb{E}\left[\tilde{X}_{t+1}\right] - \mathbb{E}\left[f(\tilde{X}_t, \theta)\right]\right\| \\
&\leq \left\|\mathbb{E}[X_{t+1}] - \mathbb{E}\left[\tilde{X}_{t+1}\right]\right\| + \left\|\mathbb{E}\left[f(X_t, \theta - f(\tilde{X}_t, \theta)\right]\right\| && (7.24) \\
&\leq \epsilon + L\left\|\mathbb{E}[X_t] - \mathbb{E}\left[\tilde{X}_t\right]\right\| && (7.25) \\
&\leq (L+1)\epsilon, && (7.26)
\end{aligned}$$

where (7.24) follows from the triangle inequality, and (7.25) follows from the quantization error bound, the fact that $\mathbb{E}[\|X\|] \geq \|\mathbb{E}[X]\|$, and Lipschitz continuity.

Finally, for any $\alpha \in (0,1)$,

$$\begin{aligned}
&\mathbb{P}\left[\left\|\tilde{X}_{t+1} - \hat{X}_{t+1}\right\| \geq \Delta_{\text{marg}}\right] \\
&\quad \leq \mathbb{P}\left[\left\|\tilde{X}_{t+1} - \mathbb{E}\left[\tilde{X}_{t+1}\right]\right\| \geq \alpha(\Delta_{\text{marg}} - (L+1)\epsilon)\right] \\
&\qquad + \mathbb{P}\left[\left\|\hat{X}_{t+1} - \mathbb{E}\left[\hat{X}_{t+1}\right]\right\| \geq (1-\alpha)(\Delta_{\text{marg}} - (L+1)\epsilon)\right] \qquad (7.27) \\
&\quad \leq \frac{d}{(\Delta_{\text{marg}} - (L+1)\epsilon)^2}\left(\frac{\tilde{\lambda}^2}{\alpha^2} + \frac{1}{m^2(1-\alpha)^2}\right) \qquad (7.28) \\
&\quad \leq \frac{2d\tilde{\lambda}^2}{(\Delta_{\text{marg}} - (L+1)\epsilon)^2}\left(1 + \frac{1}{m\tilde{\lambda}}\right)^2, \qquad (7.29)
\end{aligned}$$

where (7.27) follows from the triangle inequality and the union bound, and (7.26), and (7.28) from (7.23). Finally, (7.29) is obtained by maximizing the bound over $\alpha \in (0,1)$. □

Corollary 12. *To guarantee validation with probability at least $1-\rho$, for a margin of deviation of Δ_{marg}, where $\rho \leq \frac{2d}{(\Delta_{marg}-(L+1)\epsilon)^2}$, it suffices to use*

$$m = \left\lceil \left[\left(\sqrt{\tfrac{\rho}{2d}}(\Delta_{marg} - (L+1)\epsilon) - \tilde{\lambda} \right) \right]^{-1} \right\rceil \qquad (7.30)$$

endorsers.

This sufficient condition follows directly from Thm. 31.

7.5.3 Enumerative Experiments

Whereas MBA was designed with iterative computations in mind, quite a few computational experiments are enumerative in nature, such as hyperparameter tuning of neural networks and doing "what-if analyses" with various choices of interventions that affect disease spread. Consider an enumerative computational experiment, where we evaluate outputs of a black-box function $f : \mathbb{R}^{d_i} \times \mathbb{R}^{d'} \to \mathbb{R}^{d_o}$ over a set of inputs $\left\{X_i \in \mathbb{R}^{d_i} : i \in [n]\right\}$. Let $Y_i = f(X_i, \theta_i)$, for $i \in [n]$, where $Y_i \in \mathbb{R}^{d_o}$ is the output, and $\theta_i \in \mathbb{R}^{d'}$ is an external source of randomness (noise) used in the computation.

We adopt a similar function model and assume $f(\cdot)$ is L-Lipschitz continuous in the input, for all $\theta \in \mathbb{R}^{d'}$ i.e.,

$$\|f(X_1, \theta) - f(X_2, \theta)\| \leq L \|X_1 - X_2\|. \qquad (7.31)$$

The input-output pairs, (X_i, Y_i), are referred to as *states*. We aim to ensure the computational validity of the outputs.

Algorithm 14 Frame construction, $\mathcal{T} = \Phi\left(\{Z_1, \ldots, Z_n\}, \Delta_q\right)$

$W_{i,j} \leftarrow d(Z_i, Z_j)$, for $i \neq j \in [n]$
Construct weighted complete graph $G = ([n], W)$
Choose $v_1 \in [n]$; $T_1 \leftarrow (\{v_1\}, \emptyset)$, $\mathcal{T} \leftarrow \{T_1\}$, $\mathcal{V} \leftarrow \{v_1\}$
for $r = 2$ **to** n **do**
  $d_{\min} \leftarrow \min\{W_{j,k} : j \in \mathcal{V}, k \in [n] \backslash \mathcal{V}\}$
  $(\tilde{u}, \tilde{v}, \tilde{T}) \leftarrow \arg\min\{W_{j,k} : j \in \mathcal{V} \cap T_i, k \in [n] \backslash \mathcal{V}\}$
  if $d_{\min} \leq \Delta_q$ **then**
    Add vertex $\tilde{v}$ and edge $(\tilde{u}, \tilde{v})$ to tree $\tilde{T}$; $\mathcal{V} \leftarrow \mathcal{V} \cup \{\tilde{v}\}$
  else
    Construct tree $T_{|\mathcal{T}|+1} = (\{\tilde{v}\}, \emptyset)$; $\mathcal{T} \leftarrow \mathcal{T} \cup \{T_{|\mathcal{T}|+1}\}$
  end if
end for
return $\mathcal{T}$

In the iterative experiment the successive states are close to each other, which facilitated delta encoding and lossy compression of the state updates. However, in the enumerative experiments, we only have a set of input-output pairs without any predetermined ordering among them. Thus efficient compression of states requires the construction of good frames that include states that are close to each other. Hence we formulate a frame construction algorithm that sequentially organizes the states into frames that can be compressed efficiently. From the Lipschitz continuity of $f(\cdot)$, we know that any two inputs X_1, X_2 that are close result in spatially close outputs, Y_1, Y_2. This property is leveraged to construct frames of spatially close states, that can be compressed efficiently using a similar lattice-based compression of state differences.

Each frame comprises an ordering of different states and a compressed representation of the differences according to this order. Consider a set of states $\{Z_1, \ldots, Z_n\}$ and let the distance function be $d(\cdot, \cdot)$. The algorithm for frame construction is described in Alg. 14. The pairwise distances between the states are used to construct a forest of trees with the minimum cumulative weight (distances). The algorithm uses a modified version of Prim's algorithm for minimum spanning tree construction.

Lemma 36. *Given the number of frames, the frame construction algorithm Φ minimizes the cumulative weight of the spanning forest for a given Δ_q.*

Proof. By the definition of the greedy frame construction algorithms, the minimum weight trees that constitute the forest minimize the sum of the distances between the data points. Thus the result follows. □

This also naturally implies the following result.

Algorithm 15 Compressor, $\mathcal{F} = \text{ENC}\left(\{Z_1, \ldots, Z_n\}, \mathcal{T}, \epsilon\right)$

for all $i \in |\mathcal{T}|$ **do**
  Choose a root $r \in T_i$, $\tilde{\Delta}_r^{(i)} = \text{ENC}_{LZ}(Z_r, \epsilon)$
  while Depth First Scan(T_i) **do**
    Let scanned edge be (u, v) where u is the parent node
    Delta encode along edge, $\Delta Z_v \leftarrow Z_v - Z_u$
    Lattice code difference $\tilde{\Delta}_v^{(i)} = \text{ENC}_{\mathcal{L}}(\Delta Z_v, \epsilon)$
  end while
  Compressed frame, $F_i \leftarrow \left\{T_i, \left\{\tilde{\Delta}_v^{(i)} : v \in T_i\right\}\right\}$
end for
return $\mathcal{F} \leftarrow \{F_i : i \in [|\mathcal{T}|]\}$

Corollary 13. *For a given set of states* $\{Z_1, \ldots, Z_n\}$, Φ *minimizes the number of frames, subject to the distance constraint imposed by* Δ_q.

Proof. The result follows from the observation that the algorithm constructs the frames by removing the edges with weight exceeding the threshold from the minimum spanning tree, and thus minimizes the number of frames. □

The compression, with approximation error ϵ, is then performed according to Alg. 15. Each tree is compressed into a frame by first choosing a root as checkpoint at random, compressing the checkpoint using the lossy Lempel-Ziv compressor of [204], $\text{ENC}_{LZ}(\cdot)$, as shown in Fig. 7.7. The algorithm then scans edges of the tree using depth first search, delta encodes the states along the edges, and compresses the differences using a successive refinement lattice vector quantizer, according to the lattice $\mathcal{L}$, $\text{ENC}_{\mathcal{L}}(\cdot)$. The compressed frames are then composed of the tree structure and the set of compressed differences.

A direct extension of Thm. 27 indicates that the optimal choice of approximation error for deterministic, L-Lipschitz continuous computations is $\epsilon \leq \frac{\Delta_{val}}{L+1}$. The representation of inter state dependency as trees is optimal as it minimizes the number of bits required to communicate the relationship.

Lemma 37. *If the difference vectors are uniform in* $\mathcal{B}(\Delta_q) = \{x : \|x\| \leq \Delta_q\}$, *then the maximum expected cost of representing a single frame of size* M *is*

$$C_{comm} = O\left(d \log \frac{B}{\epsilon} + (M-1) d \log \frac{\Delta_q}{\epsilon} + (M-1) \log M\right), \tag{7.32}$$

where $d = d_i + d_o$ *and for any state* Z, $\|Z\| \leq B$.

The result follows analogous to the cost analysis for the iterative case. We note that the representation cost of frames as trees incurs a cost of $(M-1)\log M$ bits, as the number

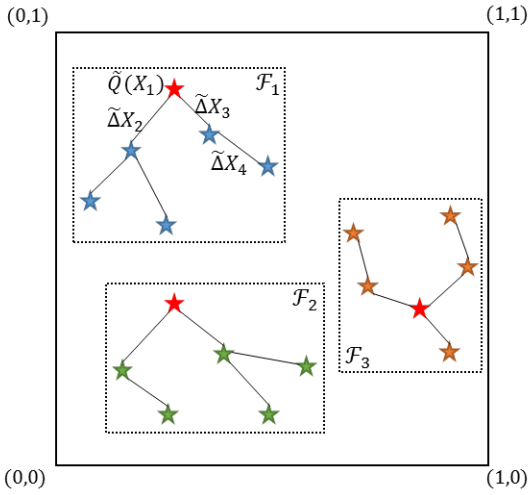


Figure 7.7: Structure of state update frames: Each frame is represented by a rooted tree. Root acts as the checkpoint and is quantized using lossy LZ compression. Other states are compressed using delta encoding and successive refinement lattice vector quantization with respect to their parent node.

of rooted trees on M nodes is given by Cayley's formula as M^{M-1} [185]. This, along with Lem. 36 and Cor. 13 highlights the efficiency, need, and choice of the frame construction algorithm.

The endorser and orderer operations remain as in the iterative case and the overall system model is shown in Fig. 7.8. We perform some experimental evaluation of MBA and the frame construction algorithm in Sec. 7.5.3

7.6 Experiments

To evaluate the system performance and costs of communication and computation, we now consider some simple iterative and enumerative experiments. Since adversarial models for the experiments are not obvious, we consider a simpler computational model under which the agents are noisy but honest. That is, the agents are individually honest but their computations are corrupted by noise. Thus we evaluate the ability of the validation scheme to eliminate the noise in such computations, and identify the tradeoff in costs incurred.

7.6.1 Iterative Experiments with MNIST Training

In this section, we run some simple synthetic experiments using the MNIST database [236], for the scenario described in Sec. 7.3.4, to understand the distributed trust environment

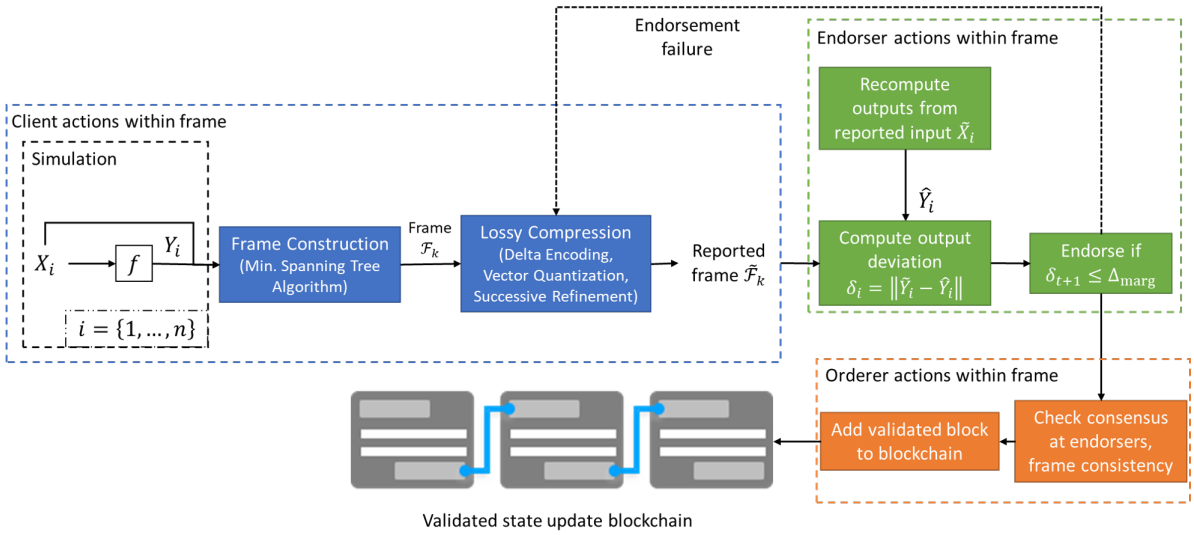


Figure 7.8: Block diagram of computational trust policy.

design, the costs involved, and the benefits of the enforcement. These synthetic experiments were selected to evaluate the efficacy of our approach with a domain that is familiar, and the process of training neural networks that is common in the research community.

Let us consider a simple three-layer neural network, trained on the MNIST database, with 25 neurons in the hidden layer. Consider a client training the neural network using mini-batch SGD, with limited resources such that it is constrained in computing gradients and so uses a small batchsize of 10 samples per iteration and 1000 iterations. The average precision of such a neural network trained with gradient descent is 97.4%. We now wish to establish trust in the training process owing to the limited resources of the client. Whereas this configuration is far from the state of the art on the database, it does help understand the trust environment better.

Since the training process uses stochastic gradients, exact recomputation of the iterates is infeasible. Hence, we compare deviations from the average across $m = 5$ endorsers per state for validation. We evaluate the computation and communication cost of validation as a function of the tolerance chosen for validation. Since the neural network converges to a local minimum according to SGD, we use a tolerance for iteration t as $\Delta_{\text{val}}(t) = \frac{\Delta_{\max}}{\log(t+1)}$. That is, the validation requirements are made stronger with the iterations.

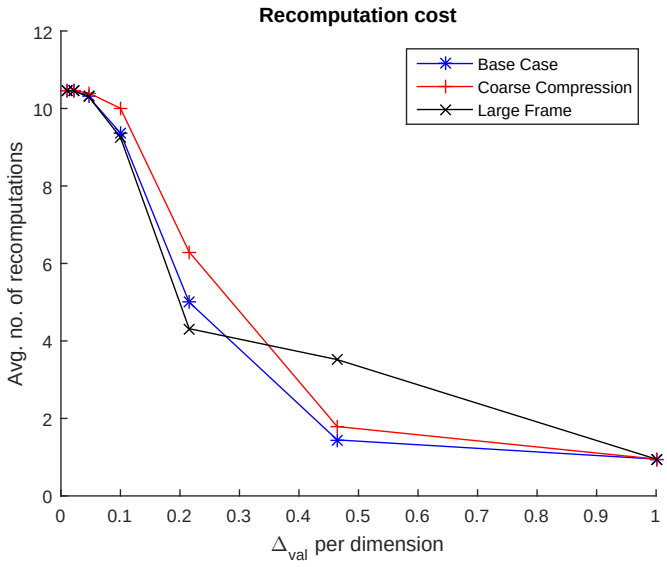
We consider three main cases of the simulation:

1. **Base Case:** Compression error is less than validation tolerance, i.e., $\epsilon \leq \Delta_{\max}$, and maximum frame size is 10% of the total number of iterations.

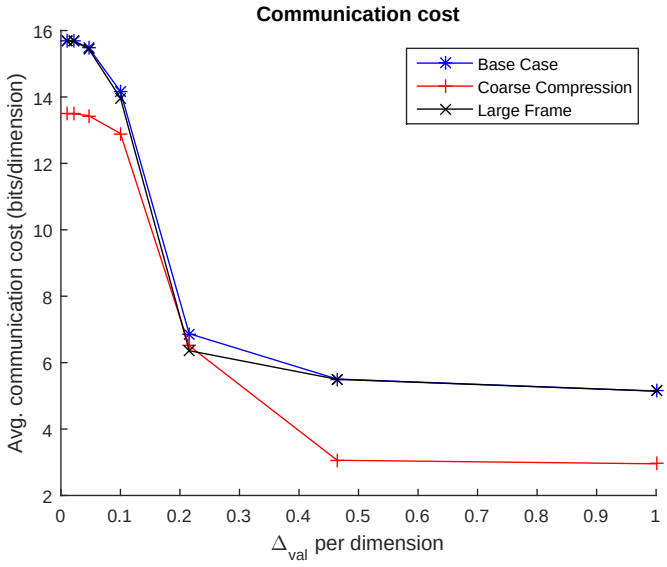
2. **Coarse Compression:** Large compression error, i.e., $\epsilon \geq \Delta_{\max}$ for at least some

instances, and same base $\bar{M}$.

3. **Large Frames:** Same base compression error, and maximum frame size is 20% of total number of iterations.



(a) Avg. recomputation cost tradeoff: Plot depicts the average number of recomputations of gradients per iteration for varying validation requirements.



(b) Avg. communication cost tradeoff: Plot depicts average bits per dimension communicated by clients to endorsers for varying validation tolerance.

Figure 7.9: Cost tradeoff with validation requirements.

In the base case, invalidation from approximation errors are more frequent in later iterations when the tolerance is also lower. However, with increasing iterations, the network weights are also closer to the minima. Thus approximation errors can be eliminated by successive refinement, as gradients estimates by the client also get more accurate. The presence of outliers and smaller batch sizes impact the initial iterations much more, which are reported with comparatively better accuracy, as required by the weaker validation criterion, therein only invalidating computational errors.

In comparison, in the case of coarse compression, approximation errors of the gradients are much more likely, therein resulting in more instances of invalidation. This translates to a higher number of gradient recomputations at the expense of reduced communication overhead on the compressed state updates. On the other hand, in the case of the extremely large frames, the endorsers validate longer sequences of states at once. Thus, each invalidation results in a recomputation of all succeeding states, therein increasing the number of recomputations from the base case. This case however reduces the number of frames and checkpoints, therein reducing the average communication cost in comparison to the base case.

In Fig. 7.9a, the average number of gradient recomputations per iteration is shown for these three cases. As expected, this decays sharply as we increase the tolerance. Note that at either extreme, the three cases converge in the number of recomputations. This is owing to the fact that at one end all gradients are accepted whereas at the stricter end, most gradients are rejected with high probability, irrespective of the compression parameters. In the moderate tolerance range, we observe the tradeoffs as explained above. The corresponding communication cost tradeoff is shown in Fig. 7.9b.

Figure 7.10 shows the precision of the neural network trained under the validation requirement as compared against the networks trained with standard mini batch SGD of batch sizes $10, 30$, and 50. We note that the network trained with trust outperforms the case of vanilla SGD with the same batch size as it eliminates spurious gradients at validation. Increasing trust requirements (decreasing tolerance) results in improved precision of the model. In particular, it is worth noting that the strictest validation criterion results in performance that is almost as good as training with a batch size of 50. This is understandable as the endorsers validate only those gradients that are close to that of the case with mini batch of size 50. In fact, even when the trust requirements are fairly relaxed, just eliminating outliers in the gradients enhances the training significantly.

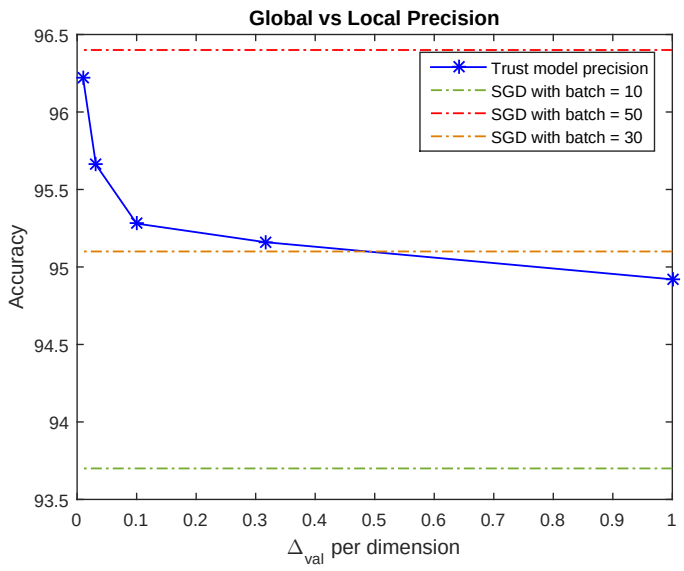


Figure 7.10: Accuracy of neural network vs trust: Plot depicts accuracy of the trained neural network satisfying the local validation criterion. Eliminating spurious gradients through validation enhances training process.

7.6.2 Enumerative Experiments: Openmalaria Simulations

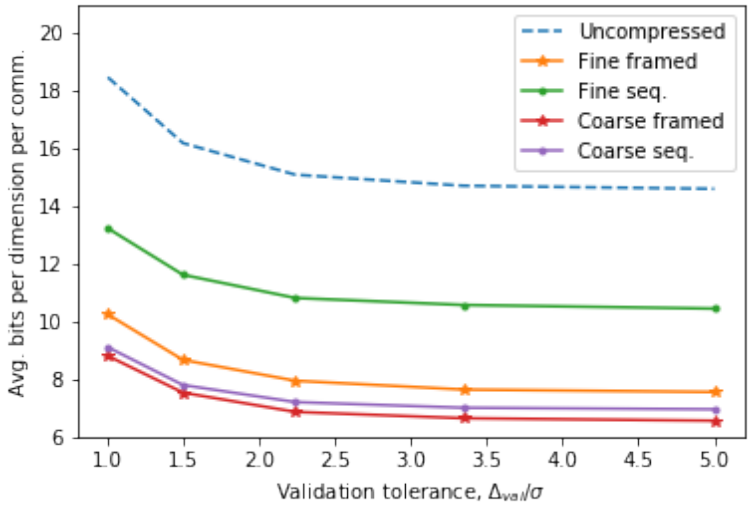
In this section we study the enumerative version of MBA through the example of the OpenMalaria simulation framework [196,210]. In particular, we consider the experiment of identifying the optimal disease intervention policy in terms of the proportion of insecticide treated nets distributed and the fraction of geographical area covered by indoor residual spraying. The client evaluates efficacy of policies in terms of cost-normalized disability adjusted life years, which we refer to as the *reward*.

Consider a set of policies $\{X_1, \ldots, X_n\}$ sampled i.i.d. uniformly at random from $[0,1]^2$ and let the corresponding mean rewards be $\{\bar{Y}_1, \ldots, \bar{Y}_n\}$. The compressed frames are validated, according to (7.3), by subsets of m endorsers. For these experiments we adopt an approximation error of 30% of the maximum deviation, and a maximum frame size of 100.

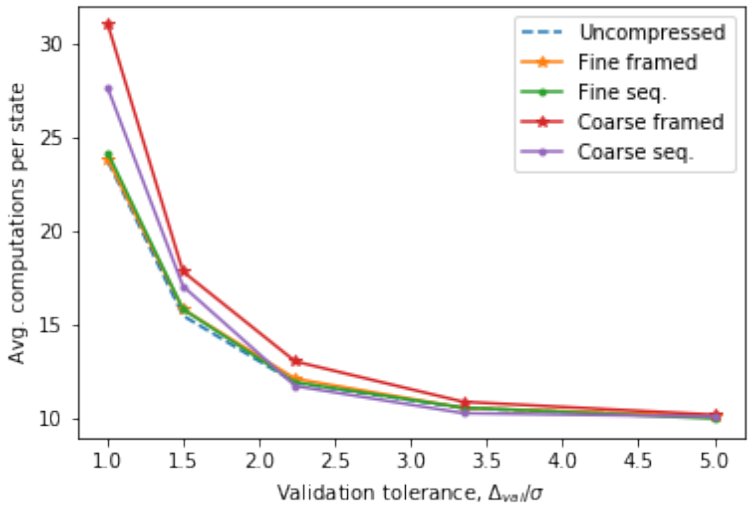
To understand the importance of frame construction toward efficient compression, we consider the communication and compression costs for varying validation tolerance and compression accuracy in Fig. 7.11. For the analysis of costs of trust, we consider validations across $m = 10$ endorsers for each state. First, we note that the baseline is set by the uncompressed communication that incurs significant number bits that effectively amount to communicating an unsigned float per dimension.

We also consider fine and coarse compression as determined by the approximation error using the MST-based frame construction and just sequential framing of states. We observe

that for fine compression, the frame construction algorithm significantly reduces the cost of compression as sequential framing results in more checkpoints and smaller frames. Since approximation error is low, the average number of bits per state, per dimension, per instance of communication also increases with each new uncompressed checkpoint. On the other hand, the frame construction algorithm groups states more efficiently, and the delta encoding and vector quantization functions more efficiently. This however is much less pronounced under coarse compression. When the approximation error is large, the Lempel-Ziv compression [19] of checkpoints suffices to represent frames in terms of prior states, and so the need for frame construction is much less pronounced.



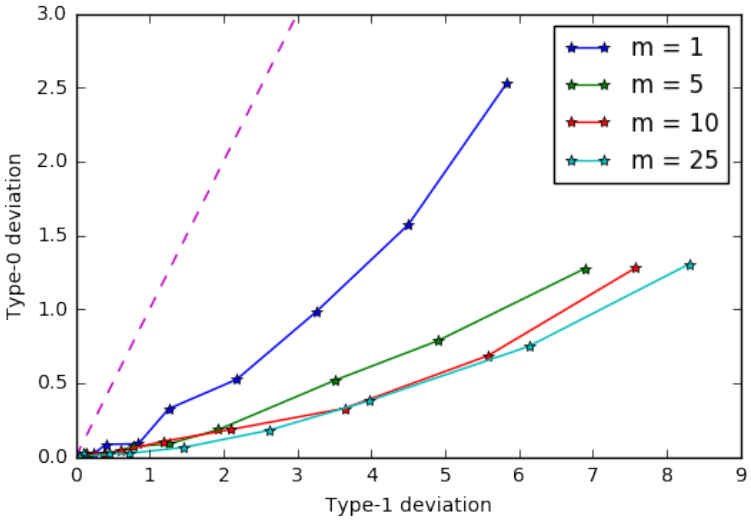
(a) Communication costs of framing.



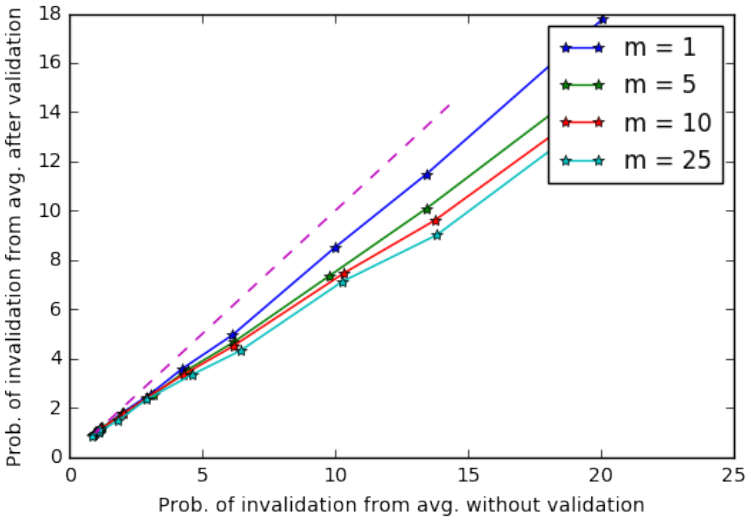
(b) Computation costs with framing.

Figure 7.11: Communication and computation cost tradeoff with tolerance.

The cost of communication is benchmarked against the corresponding cost of computation expressed in terms of the average number of computations across the client and endorsers for validation of a state in Fig. 7.11. The baseline for the computational cost is established by the uncompressed communication case, and as observed in the figure, decreases with increasing validation tolerance magnitude. We note that a fine compression, both with and without frame construction results in comparable computation costs, indicating that the communication costs can be reduced without an increase in the number of computations.



(a) Type errors by deviation: plot of T_0 vs T_1.



(b) Gains from validation.

Figure 7.12: Error performance with and without validation.

On the other hand, in the case of coarse compression, whereas the communication costs

are significantly reduced, the number of computations per state also increases, especially for smaller validation tolerance magnitudes. This is understandable as the coarse compression results in an increased number of invalidations resulting from compression error. Naturally, when the validation tolerance is comparable to the approximation error, the error from the compression is effectively mitigated, resulting in similar computational cost as that of fine compression. Comparing the compression with and without frame construction, we note that the computational cost with the frame construction is larger as the coarse compression error accounts for coarse errors in the deviations across states, therein implying that the deviation in output is comparatively higher, especially when the standard deviation across outputs is small.

To study the effect of the validation scheme on computational accuracy, let us define $\delta_i^0 = \left\|Y_i - \bar{Y}_i\right\|$ and $\delta_i^1 = \left\|\tilde{Y}_i - \bar{Y}_i\right\|$, where $\bar{Y}_i$ is the expected output, $\tilde{Y}_i$ is the validated output, and Y_i is the output generated by the client prior to validation. Define $T_0 = \mathbb{P}\left[\delta_i^1 > \delta_i^0\right]$ and $T_1 = \mathbb{P}\left[\delta_i^1 < \delta_i^0\right]$, i.e., T_0 is the probability that the validation results in larger deviation from the mean reward, and T_1 is the probability with which this deviation from the mean reward is reduced.

In Fig. 7.12a we plot the variation of T_0 with T_1 for different sizes of endorser sets and differing validation tolerance. As the tolerance reduces, the validation mechanism reduces the deviation from the expectation more often. This is observed in Fig. 7.12a and we note that $T_1 > T_0$ implies that the validation improves the computation more often than not. When the tolerance Δ_{val} is large, the number of instances of invalidation is far fewer and so the fraction of computations that are altered are also much fewer.

As the number of number of endorsers increases, the average of recomputed outputs is a robust estimator of the mean of the computation and thus an alignment with these averages for validation also implies reduced deviation from the mean. Thus $T_1 \gg T_0$ when m is large, as seen in Fig. 7.12a, highlighting the returns from the investment in more endorsers for validation.

Another metric to study the computational gains from validation is the probability that the output, as compared against the expected output would be invalidated, with and without the validation framework. To this end, let us define $\rho_0 = \mathbb{P}\left[\delta_i^0 > \Delta_{\text{val}}\right]$ and $\rho_1 = \mathbb{P}\left[\delta_i^1 > \Delta_{\text{val}}\right]$. Naturally we would like to ensure that $\rho_1 < \rho_0$, and in fact make it as small as possible.

Figure 7.12b compares ρ_1 with ρ_0 as achieved by the system for various numbers of endorsers and various tolerance levels. In each case, the validation mechanism ensures that $\rho_1 < \rho_0$. Furthermore, the gains from the validation are far more pronounced upon the use of more endorsers. This is again expected as the average of more independent endorsers is a more robust estimate of the expected rewards. The gains in terms of $\frac{\rho_0}{\rho_1}$ reduce with

increasing tolerance Δ_{val} as the system is more accommodating to deviations from the mean and from the endorser averages.

7.6.3 Anomaly Detection: OpenMalaria Simulations

Since the endorsers check the reports of the client in the validation cycle, it is possible to track anomalous computational sources using the validation statistics corresponding to the agents. To study this feasibility, we again consider the OpenMalaria simulation experiments. The experiment was designed with 144 workers running OpenMalaria executable models distributed across different clusters on IBM Cloud. Each worker performs eight OpenMalaria simulations generating a total of 1152 sources for OpenMalaria results.

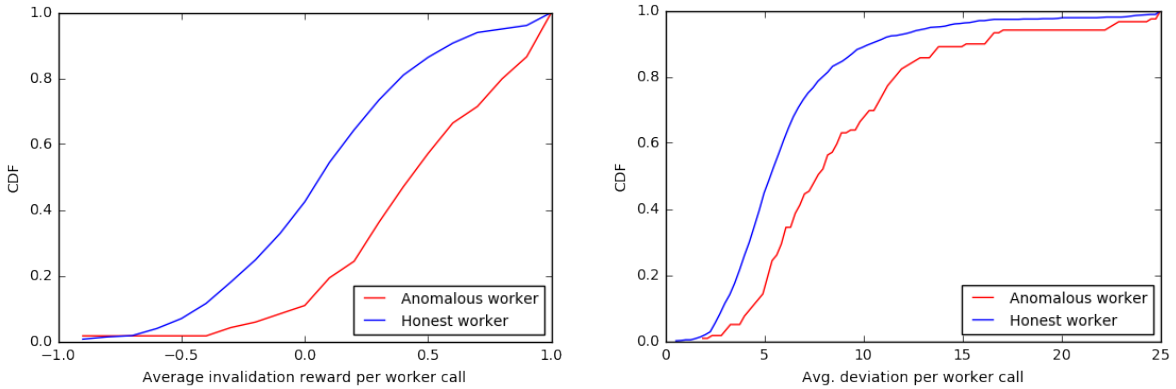
Of these sources, a randomly sampled subset of 10% of the sources are anomalous and generate noisy rewards such that they return a reward that is the true reward plus a noise $N \sim \mathcal{N}(c\sigma(ITN - IRS), \sigma)$. Here c is a scaling constant and σ is the standard deviation of the computations. This represents workers who are biased toward promoting the use of ITNs over IRS, and the extent of this bias is characterized by c. The noise model is inspired by the central limit theorem and proves to be a simple benchmark for the noisy/adversarial agents.

The experiment samples 500 policies, $(ITN, IRS) \in [0, 1]^2$ and queries the system for the corresponding reward. The system employs a worker to evaluate the reward corresponding to the policy and the rewards are validated using MBA. Over the course of the experiment, for each source, $j \in [1152]$, we keep track of the validation statistic V_j and the deviation record Δ_j as follows. Let us assume that the output generated by a source i is being validated by an endorser set $\mathcal{E}_i$. Then, for each $j \in \mathcal{E}_i \cup \{i\}$, if the output is valid, update $V_j \leftarrow V_j - 1$ and if it is invalid, $V_j \leftarrow V_j + 1$. Also, for each $j \in \mathcal{E}_i \cup \{i\}$, record the deviation, $\Delta_j \leftarrow \Delta_j \cup \{\delta\}$, where

$$\delta = \left\| \tilde{Y}_i - \frac{1}{m} \sum_{j \in \mathcal{E}_i} \hat{Y}_j \right\|, \tag{7.33}$$

is the endorsement deviation. That is, it is the Euclidean distance between the state by the client i $\tilde{Y}_i$, and the average state recomputed by the validating agents, $\hat{Y}_j$, for $j \in \mathcal{E}_i$. The higher the deviation, the less reliable the reported state, and therein the more likely of an erroneous report.

We run multiple batches of this experiment and collect the average validation statistics and deviation records for the sources across these batches. We then estimate the probability mass function of the average validation statistic P_{V_j} and that of the deviation P_{D_j} for each source j. We use these distributions to detect anomalous sources. We observe the difference



(a) CDF of average invalidation reward per worker call.

(b) CDF of average deviation per worker call.

Figure 7.13: Invalidation statistics of agents.

Table 7.1: Kolmogorov-Smirnov test statistics between honest and anomalous workers.

	Unbiased, $c = 0$	Biased, $c = 10$
KS stat for P_V	0.11	0.344
p-value for P_V	0.13	$1.9 * 10^{-11}$
KS stat for P_D	0.108	0.407
p-value for P_D	0.165	$3.4 * 10^{-16}$

in the invalidation statistics for the honest and adversarial agents in the form of a separation of the corresponding CDFs, as shown in Figs. 7.13a and 7.13b.

We quantify the distributional dissimilarity using the Kolmogorov-Smirnov (KS) for two main types of anomalies: biased sources ($c > 0$) and unbiased, input-independent anomalies ($c = 0$). The KS test statistic is used to compute the p-value corresponding to the binary hypothesis test with null hypothesis that the two sources are identical and the alternate that they are statistically different sources, as shown in Table 7.1.

We can observe here that in the biased case, both the validation and deviation profiles are starkly different for honest and anomalous workers, as highlighted by the extremely low p-value. On the other hand, for the unbiased anomalies case, we note that the p-values are comparatively much higher, indicating that whilst different, the statistical confidence in separating the two sources is much lower. This is understandable as detection of anomalies with bias is much easier than unbiased anomalies who just behave like noisy sources.

Next, we consider the task of detection of anomalies using $\{(P_{V_j}, P_{D_j}) : j \in [1152]\}$ as the feature representations. Treating the problem of anomaly detection as clustering into two clusters, we used the k-NN classifier to determine honest and anomalous workers. In particular, we used the total variational distances between the probability distributions as

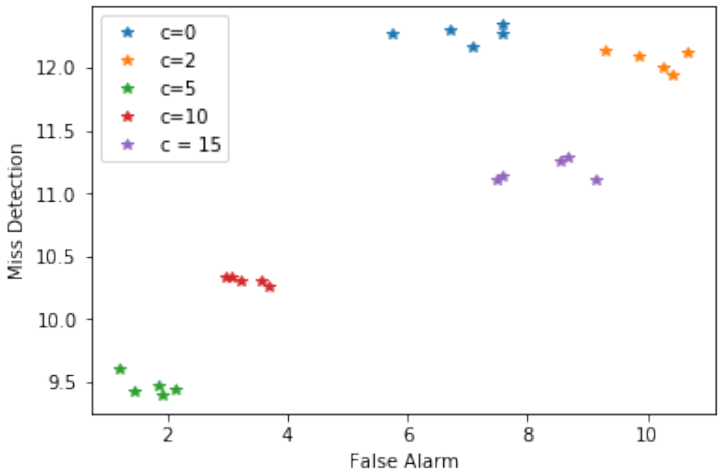


Figure 7.14: Performance of anomaly detection using k-NN classifier.

the notion of distance in the k-NN classifier. Other notions of divergence between probability measures can easily be used without loss of generality.

Let us now study the performance of the classifier for different types of anomalies. The false alarm and miss detection probabilities in percentages is represented in Fig. 7.14. The plot considers the anomaly detection performance for unbiased ($c = 0$) and varying degrees of bias ($c > 0$).

A variety of important characteristics are to be noted:

1. First for $c = 0$, the anomalous sources behave like unbiased noisy compute nodes and thus are harder to detect, as indicated by its high miss detection and false alarm probabilities.

2. For small amounts of bias, such as $c = 2$, the false alarm probability worsens with minimal to no improvement in miss detection probability. This is owing to the fact that the biased anomalies create few more invalidations among honest sources, making the classifier misclassify them as anomalies. However, the small bias also implies that the classifier does not get much more information about the anomalies than the unbiased case—hence the comparable miss detection probabilities.

3. Next, as the bias increases to $c = 5$, we observe that the anomaly detection gets significantly easier as the bias becomes more recognizable through the invalidation profiles.

4. However, when the bias becomes significantly large, for instance $c \in \{10, 15\}$, the anomalous sources are so heavily biased that they end up creating more invalidations,

which make the classifier categorize honest sources as anomalies. Thus, the false alarm probability increases. The miss detection probability also grows as the anomalies are now well-hidden among more honest sources with similar invalidation profiles.

Thus, these simple experiments not only highlights the role of trust in guaranteeing local and global consistency in the computational process, but also the cost tradeoffs involved in establishing them. Appropriate coding schemes can be adopted based on a careful study of these cost-benefit tradeoffs. Additionally, we also note that simple classification algorithms using the invalidation profile can be used to identify anomalous sources.

7.7 Discussion

In this work we considered a multi-agent computational platform and the problem of establishing trust in the computations performed by individual agents in such a system. Using a novel combination of blockchains and distributed consensus through recomputation, we assured validity of local computations and simple verification of computational trajectories. Using efficient, universal compression techniques, we also identified methods to reduce the communication and storage overheads concerned with establishing trust in such systems, therein addressing the scalability challenge posed by blockchain systems.

Creation of such trusted platforms for distributed computation among untrusting agents allows for improved collaboration, and efficient data, model, and result sharing that is critical to establish efficient policy design mechanisms. Additionally they also result in creating unified platforms for sharing results, and in ensuring scientific reproducibility.

CHAPTER 8

CONCLUSION

Data has been called in various venues as the new oil of the current economy, and for good reason. The big data era has ushered in a new age wherein we have been able to collect, interpret, infer, and integrate data-driven learnings into everyday life. In particular, a variety of information processing algorithms have been defined to solve a diverse set of problems in machine learning. Further, large-scale data sharing pipelines have also been designed to store and share data between peers.

In this context, this thesis tries to build understanding of existing algorithms, design new information processing methods, and develop protocols and systems for trusted sharing of data and computations. In particular, the thesis explores three main problems of image registration, unsupervised clustering, and blockchain systems, through the lens of information theory. We develop theoretical insight and practical solutions by drawing on prior information-theoretic studies.

In particular, we considered the image registration problem in Chapters 2 and 3. Here we developed theoretical guarantees on universal image registration algorithms by studying the error exponent using the method of types, and designed multi-image registration algorithms using multivariate information functionals. We built further on the two-image registration problem to establish the finite-sample fundamental limits on the sample complexity-information tradeoff in the channel-aware context. In essence, this part of the thesis considers conventional information-theoretic methods to identify the fundamental requirements for the image registration problem.

Then we consider the problem of unsupervised clustering of random sources in Chapters 4 and 5. Here we considered the clustering problem from three contexts—image registration and clustering, crowdsourced clustering, and analytical independence clustering of sources. We saw the role of multivariate information functionals in such independence clustering problems, defined new and novel multivariate information functionals, and distance and memory properties for universal clustering. This part of the thesis highlights the role of information functionals and develops information processing algorithms with strong theoretical performance guarantees.

Finally in Chapters 6 and 7 we considered the use of blockchain systems in establishing provenance and integrity in data and computations. In particular, in each study we developed the peer-to-peer interaction protocols and compression schema to reduce the costs of communication and storage. In the process we were able to design large-scale blockchain systems that share trusted data and computations between peers in a network. This part of the thesis aims at the more practical side of designing implementable large-scale systems that we have prototyped over the Hyperledger Fabric.

The three parts of this thesis are held together by two main threads. Firstly, the big data era establishes a new set of problems in a variety of fields, interconnected by the interplay of data procurement, storage and sharing, and information processing algorithms that work with such data. Secondly, the thesis also highlights the ubiquitous nature of information theory in the digital age as the mathematics of communication and compression provides theoretical insight into a wide array of problems, and lastly also inspires the design of new and novel methods for practical systems.

8.1 Future Directions

The efficacy of mutual information-based image registration was observed in Chapter 2. The method requires estimation of information and optimization of this estimate over the set of transformations. The variational definition of divergence is given by a maximization over the set of functions of the random variable according to the Donsker-Varadhan characterization [237] and can be used to define mutual information estimation methods [238, 239]. Thus, we can pose the image registration problem as one of joint maximization over the set of transformations and functions of the random variables. To this end, one could propose the use of alternating maximization of the proposed problem to perform image registration. More generally, we could define efficient and consistent image registration algorithms that integrate the optimization over the set of transformations with the information estimation algorithm [240, 241].

Chapter 3 characterizes achivable sample complexities in the channel-aware context. Future explorations could consider converse arguments to establish the necessary conditions on sample sizes by extending strong large deviations results on constant composition codes for communication [94] or extending the Berry-Esseen central limit theorem for hypothesis testing [90]. In the case of universal image registration, future explorations can establish necessary and sufficient conditions on the sample complexity by establishing non-asymptotic results for the estimates of functions of types [103].

The role of multivariate information functionals in clustering problems inspire future explorations on the definition, estimation, and use of multivariate information functionals. In particular, through the study of illum and clustering information, we observed that the functional properties of information can be used to define statistically consistent independence clustering algorithms. An interesting line of study to explore is that of defining computationally efficient clustering algorithms using the submoduularity of entropy [112, 242]. Another direction for future exploration is on the consistent estimation of such multivariate information functionals from data [243–245]. The definition of universal budget-optimal clustering algorithms for crowdsourcing in Chapter 5 examines the fundamental costs of crowdsourcing and future explrations can focus on the definition of efficient clustering algorithms.

The study of distributed storage techniques for blockchain systems in Chapter 6 reveals the potential of coding theory in designing scalable blockchain systems. Further studies directed toward the design of codes that allow repair and fault tolerance would create more efficient blockchain systems [190]. Another interesting direction of exploration would be along the network costs associated with the blockchain system as designing codes cognizant of such costs enhance the capabilities of blockchain systems.

Finally, Chapter 7 reveals the importance of trust in distributed computing systems and how blockchain systems could create the platform for such trust. Further explorations on adaptive compression schemes that are consistent over the distributed network could reduce the costs of the trust. Additionally, exploring cryptographic validation methods, built with specific contexts in mind can help create efficient colloborative computational societies.

Information theory has proven to be the mathematical language for a sound theoretical basis upon which the information age can be built. It certainly inspires many more fundamental theoretical limits, algorithm and large-scale system design, and systemic protocol definitions for the future.

APPENDIX A

PROOFS FOR CHAPTER 2

A.1 Size of Permutation Types

Here we prove Lem. 2, starting with simple permutations.

Lemma 38. *Let $\pi_1, \pi_2 \in \Pi$ be any two non-overlapping permutations and let $\mathbf{x}_{\pi_1}, \mathbf{x}_{\pi_2}$ be the corresponding permutations of $\mathbf{x}$. Let $\pi_1^{-1} \circ \pi_2 \in \Pi$ be a simple permutation. Then, for every $\mathbf{x}$, there exists $\tilde{\mathbf{x}}$, such that*

$$|T^n_{X_{\pi_1},X_{\pi_2}}| = |T^n_{X_0,X_1}|, \quad |T^n_{Y|X_{\pi_1},X_{\pi_2}}(\mathbf{x})| = |T^n_{Y|X_0,X_1}(\tilde{\mathbf{x}})|.$$

Proof. Since permutations are non-overlapping, there is a bijection from $T^n_{X_{\pi_1},X_{\pi_2}}$ to $T^n_{X_0,X_1}$, where $(X_0, X_1) \sim q_{X_{\pi_1},X_{\pi_2}}$. Specifically, consider $\pi \in \Pi$ defined iteratively as $\pi(i+1) = \pi_2(\pi_1^{-1}(\pi(i)))$, with $\pi(1) = \pi_1(1)$. Then, for any $\mathbf{x} \in T^n_{X_{\pi_1},X_{\pi_2}}$, the sequence $\mathbf{x}_\pi \in T^n_{X_0,X_1}$. Further, this map is invertible and so the sets are of equal size.

Result for conditional types follows *mutatis mutandis*. □

Lemma 38 implies $|T^n_{X_{\pi_1},X_{\pi_2}}|$ and $|T^n_{Y|X_{\pi_1},X_{\pi_2}}(\mathbf{x})|$ satisfy (2.12) and (2.13) respectively. We now show that the result of Lem. 38 can be extended to any two permutations $\pi_1, \pi_2 \in \Pi$.

Proof of Lem. 2. Let $\pi = \pi_1^{-1} \circ \pi_2$ and $\kappa = \kappa_\pi$. For $i \in [\kappa]$ let the length of permutation cycle i of π be $\alpha_i n$ for $\alpha_i \in (0, 1]$. Further, $\sum_{i=1}^{\kappa} \alpha_i \leq 1$. Let $\mathcal{I}_\pi$ be the identity block of π and let $\gamma = \gamma_\pi$. Then we have the decomposition

$$q_{X_{\pi_1},X_{\pi_2}}(a_0, a_1) = \sum_{i=1}^{\kappa} \alpha_i q_i(a_0, a_1) + \gamma q_{\mathcal{I}}(a_0, a_1), \tag{A.1}$$

for all $(a_0, a_1) \in \mathcal{X}^2$. Here, q_i is the first-order Markov type defined on the ith permutation cycle of π and $q_{\mathcal{I}}$ is the zeroth-order Markov type corresponding to the identity block of π.

From (A.1), we see that given a valid decomposition of types $\{q_i\}, q_{\mathcal{I}}$, the number of

sequences can be computed as a product of the number of subsequences of each type, i.e.,

$$|T^n_{X_{\pi_1},X_{\pi_2}}| = \sum |T^{\gamma n}_{q_{\mathcal{I}}}| \prod_{i=1}^{\kappa} |T^{\alpha_i n}_{q_i}|,$$

where the sum is over all valid decompositions in (A.1).

Additionally, from Lem. 38 we know the number of valid subsequences of each type. Let q_i' be the marginal corresponding to the first-order Markov type q_i.

Thus, we upper bound the size of the set as

$$|T^n_{X_{\pi_1},X_{\pi_2}}| \leq \sum 2^{\gamma n H(q_{\mathcal{I}})} \prod_{i=1}^{\kappa} |\mathcal{X}| 2^{\alpha_i n \left(H(q_i) - H(q_i')\right)} \tag{A.2}$$

$$\leq |\mathcal{X}|^{\kappa} (\gamma n + 1)^{|\mathcal{X}|} \prod_{i=1}^{\kappa} (\alpha_i n + 1)^{|\mathcal{X}|^2} 2^{n M(q_{X_{\pi_1},X_{\pi_2}})}, \tag{A.3}$$

where

$$M(q_{X_{\pi_1},X_{\pi_2}}) = \max \gamma H(q_{\mathcal{I}}) + \sum_{i=1}^{\kappa} \alpha_i (H(q_i) - H(q_i')),$$

the maximum taken over all valid decompositions in (A.1). Here, (A.2) follows from (2.10) and (2.12), and (A.3) follows since the total number of possible types is polynomial in the length of the sequence. Since $\kappa = o\left(\frac{n}{\log(n)}\right)$,

$$|T^n_{X_{\pi_1},X_{\pi_2}}| \leq 2^{n\left(M(q_{X_{\pi_1},X_{\pi_2}}) + o(1)\right)}.$$

Now, let $q_i''(a_0, a_1) = \frac{1}{|\mathcal{X}|} q_i'(a_1)$, for all $i \in [\kappa]$, and $(a_0, a_1) \in \mathcal{X}^2$. And let $q = q_{X_{\pi_1},X_{\pi_2}}$. Since $\gamma = o(1)$, using Jensen's inequality, we have

$$\begin{aligned}
\sum_{i=1}^{\kappa} \alpha_i (H(q_i) - H(q_i')) &= \sum_{i=1}^{\kappa} \alpha_i \left[\log(|\mathcal{X}|) - D\left(q_i \| q_i''\right)\right] \\
&\leq \log(|\mathcal{X}|) - D\left(q \| q''\right) \\
&= H(q) - H(q').
\end{aligned}$$

Thus,

$$|T^n_{X_{\pi_1},X_{\pi_2}}| \leq 2^{n(H(q) - H(q') + o(1))}.$$

To obtain the lower bound, we note that the total number of sequences is at least the number of sequences obtained from any single valid decomposition. Thus from (2.10) and

(2.12),

$$|T^n_{X_{\pi_1},X_{\pi_2}}| \geq 2^{n\left(M(q_{X_{\pi_1},X_{\pi_2}})+o(1)\right)}.$$

Now, for large n, consider $S = \{i \in \kappa : \alpha_i n = \Omega(n^\beta)\}$ for some $\beta > 0$. Any other cycle of smaller length contributes $o(1)$ to the exponent due to Lem. 1. One valid decomposition of (A.1) is to have $q_i = q$, for all $i \in [\kappa]$. However, the lengths of the subsequences are different and q may not be a valid type of the corresponding length. Nevertheless, for each $i \in S$, there exists a type q_i such that $d_{\mathrm{TV}}(q_i, q) \leq \frac{|\mathcal{X}|}{2\alpha_i n}$, where $d_{\mathrm{TV}}(\cdot)$ is the total variational distance. Further, entropy is continuous in the distribution [246] and satisfies

$$|H(q_i) - H(q)| \leq \frac{|\mathcal{X}|}{\alpha_i n} \log(\alpha_i n) = o(1).$$

This in turn indicates that

$$|T^n_{X_{\pi_1},X_{\pi_2}}| \geq 2^{n(H(q)-H(q')+o(1))}.$$

□

APPENDIX B

PROOFS FOR CHAPTER 5

B.1 Estimating Mutual Information from Samples

Here we briefly describe the maximum likelihood (ML) estimate of mutual information and its convergence properties.

Let $X \sim p$ be a random variable on a discrete space $\mathcal{X}$. Let $X_1, \ldots, X_n \overset{\text{iid}}{\sim} p$ and let $\hat{p}$ be the corresponding empirical distribution. The ML estimate of entropy of X is given by

$$\hat{H}(X) = \mathbb{E}_{\hat{p}} \left[-\log(\hat{p}(X)) \right].$$

The ML estimate of mutual information between random variables X, Y is then given by

$$\hat{I}(X;Y) = \hat{H}(X) + \hat{H}(Y) - \hat{H}(X,Y).$$

The ML estimates of entropy and mutual information have been widely studied [73, 129, 246, 247]. In particular, the following results are notable:

1. For all n, from Jensen's inequality, $\mathbb{E}[\hat{H}(X)] < H(X)$ [247]. For all n, $\epsilon > 0$,

$$\mathbb{P}\left[|\hat{H}(X) - \mathbb{E}[\hat{H}(X)]| > \epsilon \right] \leq 2 \exp\left(\frac{-n\epsilon^2}{2 \log_2^2 n} \right), \tag{B.1}$$

   from McDiarmid's inequality.

2. ML estimate, $\hat{H}$ is negatively biased [129]:

$$b_n(\hat{H}) \triangleq \mathbb{E}_p\left[\hat{H}(X) \right] - H(X) = -\mathbb{E}_p\left[D(\hat{p} \| p) \right] < 0.$$

   Further,

$$-\frac{|\mathcal{X}|}{n} \leq -\log\left(1 + \frac{|\mathcal{X}| - 1}{n} \right) \leq b_n(\hat{H}) \leq 0.$$

3. From [73], we know that the lower bound on sample complexity for estimating entropy

up to an additive error of ϵ is $\frac{|\mathcal{X}|}{\epsilon \log |\mathcal{X}|}$.

4. From [246], we observe that the deviation of the empirical entropy from the entropy of $X \sim P$, is bounded in terms of the total variational distance as

$$|\hat{H}(X) - H(X)| \leq -2\delta(\hat{P}, P) \log \frac{2\delta(P, Q)}{|\mathcal{X}|}. \tag{B.2}$$

Since we deal with finite, constant alphabet sizes, it suffices for us to consider the ML estimates with sufficiently large n, such that the bias is negligible.

Lemma 39. *For fixed alphabet sizes, $|\mathcal{X}|, |\mathcal{Y}|$, the ML estimate of entropy and mutual information are asymptotically consistent and satisfy*

$$\mathbb{P}\left[|\hat{H}(X) - H(X)| > \epsilon\right] \leq 2 \exp\left(\frac{-n\epsilon^2}{2 \log_2^2 n} + o(1)\right) \tag{B.3}$$

$$\mathbb{P}\left[|\hat{I}(X;Y) - I(X;Y)| > \epsilon\right] \leq 6 \exp\left(\frac{-n\epsilon^2}{18 \log_2^2 n} + o(1)\right). \tag{B.4}$$

Proof. The convergence of entropy follows directly by applying the triangle inequality, union bound, and (B.1). The result follows from the fact that the alphabet is of finite, constant size. This implies the convergence result for mutual information. □

Lemma 40. *For fixed alphabet sizes, $|\mathcal{X}|, |\mathcal{Y}|$, the ML estimate of entropy and mutual information are asymptotically consistent and satisfy*

$$\mathbb{P}\left[|\hat{H}(X) - H(X)| > \epsilon\right] \leq (n+1)^{|\mathcal{X}|} \exp\left(-cn\epsilon^4\right), \tag{B.5}$$

$$\mathbb{P}\left[|\hat{I}(X;Y) - I(X;Y)| > \epsilon\right] \leq 3(n+1)^{|\mathcal{X}||\mathcal{Y}|} \exp\left(-\tilde{c}n\epsilon^4\right), \tag{B.6}$$

where $c = (2|\mathcal{X}|^2 \log 2)^{-1}$, $\tilde{c} = (32 \max\{|\mathcal{X}|, |\mathcal{Y}|\}^2 \log 2)^{-1}$.

Proof. We first observe that for all $x > 0$, $\log x < \sqrt{x}$. Thus, for any $X \sim P$, from (B.2), we have

$$|\hat{H}(X) - H(X)| \leq \sqrt{2|\mathcal{X}|\delta(\hat{P}, P)}.$$

Using this and (5.13), the first inequality is obtained. Subsequently, using the triangle inequality and union bound, we obtain the convergence of the empirical mutual information. □

These rates of convergence are used to prove consistency.

B.2 Proof of Lem. 27

In this section we describe the proof of Lem. 27.

Proof. We first note that

$$\begin{aligned} D(Q_i\|Q_j) &= \mathbb{E}_{Q_i}\left[\log\left(\frac{Q_i(Y^\ell)}{Q_j(Y^\ell)}\right)\right] \\ &= \mathbb{E}_{Q_i}\left[\log\left(\frac{W_{Y_{i+1}Y_{i-1}}W_{Y_jY_{j-1}}W_{Y_{j+1}Y_j}}{W_{Y_{i+1}Y_i}W_{Y_iY_{i-1}}W_{Y_{j+1}Y_{j-1}}}\right)\right] \\ &= \mathbb{E}_{Q_{\ell-i}}\left[\log\left(\frac{W_{Y_{\ell-i+1}Y_{\ell-i-1}}W_{Y_{\ell-j}Y_{\ell-j-1}}W_{Y_{\ell-j+1}Y_{\ell-j}}}{W_{Y_{\ell-i+1}Y_{\ell-i}}W_{Y_{\ell-i}Y_{\ell-i-1}}W_{Y_{\ell-j+1}Y_{\ell-j-1}}}\right)\right] \\ &= D(Q_{\ell-i}\|Q_{\ell-j}). \end{aligned}$$

Then we note that

$$\begin{aligned} D(Q_0\|Q_1) &= \sum_{y^\ell}\frac{1}{2}\prod_{i=2}^{\ell}W_{y_iy_{i-1}}\log\left(2W_{y_2y_1}\right) \\ &= 1-h(1/2-\epsilon), \end{aligned}$$

where $h(\cdot)$ is the binary entropy function. Similarly

$$D(Q_1\|Q_0) = -\frac{1}{2}\log\left(1-4\epsilon^2\right).$$

For any $1 < i < \ell$,

$$\begin{aligned} D(Q_0\|Q_i) &= \mathbb{E}_{Q_0}\left[\log\left(\frac{2W_{Y_iY_{i-1}}W_{Y_{i+1}Y_i}}{W_{Y_{i+1}Y_{i-1}}}\right)\right] \\ &= 1+\left(\frac{1}{2}+2\epsilon-\epsilon^2\right)\log\left(\frac{1}{2}+\epsilon\right) \\ &\quad+\left(\frac{1}{2}-2\epsilon+\epsilon^2\right)\log\left(\frac{1}{2}-\epsilon\right). \end{aligned}$$

Similarly,

$$D(Q_i\|Q_0) = -\left(\frac{1}{2}-\epsilon\right)\log\left(\frac{1}{2}+\epsilon\right)+\left(\frac{1}{2}+\epsilon\right)\log\left(\frac{1}{2}-\epsilon\right).$$

Having computed these distances, we make one additional observation. For $1 \le i, j \le \ell$,

$i \neq j$,

$$\begin{aligned} D(Q_i \| Q_j) &= \mathbb{E}_{Q_i}\left[\log\left(\frac{W_{Y_{i+1}Y_{i-1}}}{W_{Y_{i+1}Y_i}W_{Y_iY_{i-1}}}\right) + \log\left(\frac{W_{Y_jY_{j-1}}W_{Y_{j+1}Y_j}}{W_{Y_{j+1}Y_{j-1}}}\right)\right] \\ &= D(Q_0 \| Q_j) + D(Q_i \| Q_0). \end{aligned}$$

Since ϵ is bounded, $D(Q_0 \| Q_i) = O(1)$ and $D(Q_i \| Q_0) = O(1)$. This in turn proves that the KL divergences between any two hypotheses is a constant independent of ℓ. $\square$

REFERENCES

[1] C. E. Shannon, "Bits storage capacity," Manuscript Division, Library of Congress, Handwritten note, July 1949.

[2] M. Weldon, *The Future X Network: A Bell Labs Perspective*. CRC Press, 2015.

[3] M. McCartney, "Margaret McCartney: AI in medicine must be rigorously tested," *Br. Med. J.*, vol. 361, Apr. 2018.

[4] K. Hao, "AI is sending people to jail—and getting it wrong," *Technol. Rev.*, Jan. 2019.

[5] Y. Gil, M. Greaves, J. Hendler, and H. Hirsh, "Amplify scientific discovery with artificial intelligence," *Science*, vol. 346, no. 6206, pp. 171–172, Oct. 2014.

[6] D. Acemoglu and P. Restrepo, "Artificial intelligence, automation and work," National Bureau of Economic Research, Working Paper 24196, Jan. 2018.

[7] Y. Lu, "Artificial Intelligence: A survey on evolution, models, applications and future trends," *J. Manage. Analytics*, vol. 6, no. 1, pp. 1–29, Jan. 2019.

[8] B. Lepri, N. Oliver, E. Letouzé, A. Pentland, and P. Vinck, "Fair, transparent, and accountable algorithmic decision-making processes," *Philos. Technol.*, vol. 31, no. 4, pp. 611–627, Dec. 2018.

[9] N. Diakopoulos, "Accountability in algorithmic decision making," *Commun. ACM*, vol. 59, no. 2, pp. 56–62, Feb. 2016.

[10] D. Boyd and K. Crawford, "Critical questions for big data," *Inf. Commun. Soc.*, vol. 15, no. 5, pp. 662–679, Mar. 2012.

[11] W. Knight, "An AI that writes convincing prose risks mass-producing fake news," *Technol. Rev.*, Feb. 2019.

[12] J. T. Vogelstein, Y. Park, T. Ohyama, R. A. Kerr, J. W. Truman, C. E. Priebe, and M. Zlatic, "Discovery of brainwide neural-behavioral maps via multiscale unsupervised structure learning," *Science*, vol. 344, no. 6182, pp. 386–392, Apr. 2014.

[13] P. Viola and W. M. Wells III, "Alignment by maximization of mutual information," *Int. J. Comput. Vis.*, vol. 24, no. 2, pp. 137–154, 1997.

[14] A. Hyvärinen, J. Karhunen, and E. Oja, *Independent Component Analysis*. John Wiley & Sons, 2004, vol. 46.

[15] W. M. Rand, "Objective criteria for the evaluation of clustering methods," *J. Am. Stat. Assoc.*, vol. 66, no. 336, pp. 846–850, Dec. 1971.

[16] L. Valiant, *Probably Approximately Correct: Nature's Algorithms for Learning and Prospering in a Complex World*. Basic Books, 2013.

[17] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015.

[18] T. Simonite, "The missing link of artificial intelligence," *Technol. Rev.*, Feb. 2016.

[19] J. Ziv and A. Lempel, "Compression of individual sequences via variable-rate coding," *IEEE Trans. Inf. Theory*, vol. IT-24, no. 5, pp. 530–536, Sep. 1978.

[20] R. Gallager, "Variations on a theme by Huffman," *IEEE Trans. Inf. Theory*, vol. IT-24, no. 6, pp. 668–674, Nov. 1978.

[21] J. C. Lawrence, "A new universal coding scheme for the binary memoryless source," *IEEE Trans. Inf. Theory*, vol. IT-23, no. 4, pp. 466–472, July 1977.

[22] J. Ziv, "Coding of sources with unknown statistics—Part I: Probability of encoding error," *IEEE Trans. Inf. Theory*, vol. IT-18, no. 3, pp. 384–389, May 1972.

[23] J. Ziv, "Coding of sources with unknown statistics—Part II: Distortion relative to a fidelity criterion," *IEEE Trans. Inf. Theory*, vol. IT-18, no. 3, pp. 389–394, May 1972.

[24] J. Ziv, "On universal quantization," *IEEE Trans. Inf. Theory*, vol. IT-31, no. 3, pp. 344–347, May 1985.

[25] E. hui Yang and J. C. Kieffer, "Simple universal lossy data compression schemes derived from the Lempel-Ziv algorithm," *IEEE Trans. Inf. Theory*, vol. 42, no. 1, pp. 239–245, Jan. 1996.

[26] V. D. Goppa, "Nonprobabilistic mutual information without memory," *Probl. Control Inf. Theory*, vol. 4, no. 2, pp. 97–102, 1975.

[27] J. Ziv, "Universal decoding for finite-state channels," *IEEE Trans. Inf. Theory*, vol. IT-31, no. 4, pp. 453–460, July 1985.

[28] M. Feder and A. Lapidoth, "Universal decoding for channels with memory," *IEEE Trans. Inf. Theory*, vol. 44, no. 5, pp. 1726–1745, Sep. 1998.

[29] A. Lapidoth and P. Narayan, "Reliable communication under channel uncertainty," *IEEE Trans. Inf. Theory*, vol. 44, no. 6, pp. 2148–2177, Oct. 1998.

[30] J. Rissanen, "Optimal estimation," *IEEE Inf. Theory Soc. Newsletter*, vol. 59, no. 3, pp. 1/6–7, Sep. 2009, 2009 Shannon Lecture.

[31] M. Swan, *Blockchain: Blueprint for a New Economy.* Sebastopol, CA: O'Reilly Media, Inc., 2015.

[32] J. Li, J. Wu, and L. Chen, "Block-secure: Blockchain based scheme for secure p2p cloud storage," *Inf. Sci.*, vol. 465, pp. 219 – 231, Oct. 2018.

[33] A. Azaria, A. Ekblaw, T. Vieira, and A. Lippman, "Medrec: Using blockchain for medical data access and permission management," in *2nd Int. Conf. Open Big Data (OBD)*, Aug. 2016, pp. 25–30.

[34] Q. Xia, E. B. Sifah, K. O. Asamoah, J. Gao, X. Du, and M. Guizani, "Medshare: Trust-less medical data sharing among cloud service providers via blockchain," *IEEE Access*, vol. 5, pp. 14 757–14 767, July 2017.

[35] G. Zyskind, O. Nathan, and A. S. Pentland, "Decentralizing privacy: Using blockchain to protect personal data," in *IEEE Security Privacy Workshops*, May 2015, pp. 180–184.

[36] J. Dai and M. A. Vasarhelyi, "Toward blockchain-based accounting and assurance," *J. Inf. Syst.*, vol. 31, no. 3, pp. 5–21, June 2017.

[37] K. Croman, C. Decker, I. Eyal, A. E. Gencer, A. Juels, A. Kosba, A. Miller, P. Saxena, E. Shi, E. G. Sirer, D. Song, and R. Wattenhofer, "On scaling decentralized blockchains," in *Financial Cryptography and Data Security*, ser. Lecture Notes in Computer Science, J. Clark, S. Meiklejohn, P. Y. A. Ryan, D. Wallach, M. Brenner, and K. Rohloff, Eds. Berlin: Springer, 2016, vol. 9604, pp. 106–125.

[38] M. Vukolić, "The quest for scalable blockchain fabric: Proof-of-work vs. BFT replication," in *Open Problems in Network Security*, J. Camenisch and D. Kesdoğan, Eds. Cham: Springer International Publishing, 2016, pp. 112–125.

[39] M. Zamani, M. Movahedi, and M. Raykova, "Rapidchain: Scaling blockchain via full sharding," in *Proc. ACM SIGSAC Conf. Computer Comm. Security (CCS '18)*, Oct. 2018, pp. 931–948.

[40] R. Neisse, G. Steri, and I. Nai-Fovino, "A blockchain-based approach for data accountability and provenance tracking," in *Proc. 12th Int. Conf. Availability, Reliability and Security (ARES '17)*, Aug. 2017.

[41] X. Liang, S. Shetty, D. Tosh, C. Kamhoua, K. Kwiat, and L. Njilla, "Provchain: A blockchain-based data provenance architecture in cloud environment with enhanced privacy and availability," in *Proc. 17th IEEE/ACM Int. Symp. Cluster, Cloud, Grid Comput. (CCGrid '17)*, May 2017, pp. 468–477.

[42] J. Du, S. Tang, T. Jiang, and Z. Lu, "Intensity-based robust similarity for multimodal image registration," *Int. J. Comput. Math.*, vol. 83, no. 1, pp. 49–57, 2006.

[43] A. Gelman, A. Jarrot, A. He, J. Kherroubi, and R. Laronga, "Borehole image correspondence and automated alignment," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP 2017)*, Mar. 2017, pp. 1807–1811.

[44] Z. Zhao and A. Singer, "Rotationally invariant image representation for viewing direction classification in cryo-EM," *J. Struct. Biol.*, vol. 186, no. 1, pp. 153–166, Apr. 2014.

[45] H. mei Chen, M. K. Arora, and P. K. Varshney, "Mutual information-based image registration for remote sensing data," *Int. J. Remote Sens.*, vol. 24, no. 18, pp. 3701–3706, 2003.

[46] B. Zitova and J. Flusser, "Image registration methods: A survey," *Image Vision Comput.*, vol. 21, no. 11, pp. 977–1000, 2003.

[47] M. E. Leventon and W. E. L. Grimson, "Multi-modal volume registration using joint intensity distributions," in *Int. Conf. Medical Image Comp. Comp.-Assisted Intervention*, 1998, pp. 1057–1066.

[48] H.-M. Chan, A. C. S. Chung, S. C. H. Yu, A. Norbash, and W. M. Wells, "Multi-modal image registration by minimizing Kullback-Leibler distance between expected and observed joint class histograms," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognition (CVPR'03)*, vol. 2, June 2003, pp. II–570.

[49] W. K. Pratt, "Correlation techniques of image registration," *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-10, no. 3, pp. 353–358, May 1974.

[50] J. P. W. Pluim, J. B. A. Maintz, and M. A. Viergever, "Mutual-information-based registration of medical images: A survey," *IEEE Trans. Med. Imag.*, vol. 22, no. 8, pp. 986–1004, Aug. 2003.

[51] H. S. Alhichri and M. Kamel, "Image registration using the Hausdorff fraction and virtual circles," in *Proc. IEEE Int. Conf. Image Process. (ICIP 2001)*, vol. 2, Oct. 2001, pp. 367–370.

[52] H. Neemuchwala, A. Hero, S. Zabuawala, and P. Carson, "Image registration methods in high-dimensional space," *Int. J. Imaging Syst. Tech.*, vol. 16, no. 5, pp. 130–145, Mar. 2006.

[53] M. Xu, H. Chen, and P. K. Varshney, "Ziv-Zakai bounds on image registration," *IEEE Trans. Signal Process.*, vol. 57, no. 5, pp. 1745–1755, May 2009.

[54] C. Aguerrebere, M. Delbracio, A. Bartesaghi, and G. Sapiro, "Fundamental limits in multi-image alignment," *IEEE Trans. Signal Process.*, vol. 64, no. 21, pp. 5707–5722, Nov. 2016.

[55] H. D. Tagare and M. Rao, "Why does mutual-information work for image registration? A deterministic explanation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 6, pp. 1286–1296, June 2015.

[56] L. Zollei and W. M. Wells, "On the optimality of mutual information as an image registration objective function," in *Proc. IEEE Int. Conf. Image Process. (ICIP 2009)*, Nov. 2009, pp. 189–192.

[57] T. Bendory, N. Boumal, C. Ma, Z. Zhao, and A. Singer, "Bispectrum inversion with application to multireference alignment," arXiv:1705.00641, May 2017.

[58] E. Abbe, J. M. Pereira, and A. Singer, "Sample complexity of the Boolean multireference alignment problem," in *Proc. 2017 IEEE Int. Symp. Inf. Theory*, June 2017, pp. 1316–1320.

[59] A. Pananjady, M. J. Wainwright, and T. A. Courtade, "Denoising linear models with permuted data," in *Proc. 2017 IEEE Int. Symp. Inf. Theory*, June 2017, pp. 446–450.

[60] J. Stein, J. Ziv, and N. Merhav, "Universal delay estimation for discrete channels," *IEEE Trans. Inf. Theory*, vol. 42, no. 6, pp. 2085–2093, Nov. 1996.

[61] R. K. Raman and L. R. Varshney, "Universal joint image clustering and registration using partition information," in *Proc. 2017 IEEE Int. Symp. Inf. Theory*, June 2017, pp. 2168–2172.

[62] R. K. Raman and L. R. Varshney, "Universal joint image clustering and registration using multivariate information measures," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 5, pp. 928–943, Oct. 2018.

[63] F. Yasuma, T. Mitsunaga, D. Iso, and S. K. Nayar, "Multispectral image database," Nov. 2008. [Online]. Available: http://www.cs.columbia.edu/CAVE/databases/multispectral/

[64] D. P. Palomar and S. Verdú, "Lautum information," *IEEE Trans. Inf. Theory*, vol. 54, no. 3, pp. 964–975, Mar. 2008.

[65] P. Ngo, Y. Kenmochi, N. Passat, and H. Talbot, "On 2D constrained discrete rigid transformations," *Ann. Math. Artif. Intell.*, vol. 75, no. 1, pp. 163–193, 2015.

[66] L. A. Shepp and S. P. Lloyd, "Ordered cycle lengths in a random permutation," *Trans. Am. Math. Soc.*, vol. 121, no. 2, pp. 340–357, 1966.

[67] R. K. Raman and L. R. Varshney, "Universal clustering via crowdsourcing," arXiv:1610.02276, Oct. 2016.

[68] I. Csiszár, "The method of types," *IEEE Trans. Inf. Theory*, vol. 44, no. 6, pp. 2505–2523, Oct. 1998.

[69] P. Whittle, "Some distribution and moment formulae for the Markov chain," *J. R. Stat. Soc. Ser. B. Methodol.*, vol. 17, no. 2, pp. 235–242, 1955.

[70] T. Kailath, "The divergence and Bhattacharyya distance measures in signal selection," *IEEE Trans. Commun. Technol.*, vol. COM-15, no. 1, pp. 52–60, Feb. 1967.

[71] R. K. Raman, H. Yu, and L. R. Varshney, "Illum information," in *Proc. 2017 Inf. Theory Appl. Workshop*, Feb. 2017.

[72] M. Studený and J. Vejnarová, "The multiinformation function as a tool for measuring stochastic dependence," in *Learning in Graphical Models*, M. I. Jordan, Ed. Dodrecht: Kluwer Academic Publishers, 1998, pp. 261–297.

[73] G. Valiant and P. Valiant, "Estimating the unseen: An $n/\log(n)$-sample estimator for entropy and support size, shown optimal via new CLTs," in *Proc. 43rd Annu. ACM Symp. Theory Comput. (STOC'11)*, 2011, pp. 685–694.

[74] R. K. Raman and L. R. Varshney, "Budget-optimal clustering via crowdsourcing," in *Proc. 2017 IEEE Int. Symp. Inf. Theory*, June 2017, pp. 2163–2167.

[75] F. Yasuma, T. Mitsunaga, D. Iso, and S. K. Nayar, "Generalized assorted pixel camera: Postcapture control of resolution, dynamic range, and spectrum," *IEEE Trans. Image Process.*, vol. 19, no. 9, pp. 2241–2253, Sep. 2010.

[76] L. Weiss, "On the strong converse of the coding theorem for symmetric channels without memory," *Q. Appl. Math.*, vol. 18, no. 3, pp. 209–214, Oct. 1960.

[77] V. Strassen, "Asymptotische abschätzungen in Shannons informationstheorie," in *Transactions of the 3rd Prague Conference on Information Theory, Statistical Decision Functions, Random Processes*. Prague: Pub. House of the Czechoslovak Academy of Sciences, 1962, pp. 689–723.

[78] Y. Polyanskiy, "Channel coding: non-asymptotic fundamental limits," Ph.D. dissertation, Princeton University, Nov. 2010.

[79] V. Kostina and S. Verdú, "Lossy joint source-channel coding in the finite blocklength regime," *IEEE Trans. Inf. Theory*, vol. 59, no. 5, pp. 2545–2575, May 2013.

[80] A. C. Berry, "The accuracy of the Gaussian approximation to the sum of independent variates," *Trans. Am. Math. Soc.*, vol. 49, no. 1, pp. 122–136, 1941.

[81] C.-G. Esseen, "On the liapounoff limit of error in the theory of probability," *Ark. Mat. Astron. Fys.*, no. 9, pp. 1–19, 1942.

[82] M. Hayashi, "Second-order asymptotics in fixed-length source coding and intrinsic randomness," *IEEE Trans. Inf. Theory*, vol. 54, no. 10, pp. 4619–4637, Oct. 2008.

[83] V. Kostina and S. Verdú, "Fixed-length lossy compression in the finite blocklength regime," *IEEE Trans. Inf. Theory*, vol. 58, no. 6, pp. 3309–3338, June 2012.

[84] V. Kostina, "Lossy data compression: nonasymptotic fundamental limits," Ph.D. dissertation, Princeton University, 2013.

[85] I. Kontoyiannis and S. Verdú, "Optimal lossless data compression: Non-asymptotics and asymptotics," *IEEE Trans. Inf. Theory*, vol. 60, no. 2, pp. 777–795, Feb. 2014.

[86] V. Y. F. Tan, "Asymptotic estimates in information theory with non-vanishing error probabilities," *Found. Trends Commun. Inf. Theory*, vol. 11, no. 1-2, pp. 1–184, 2014.

[87] I. Kontoyiannis, "Pointwise redundancy in lossy data compression and universal lossy data compression," *IEEE Trans. Inf. Theory*, vol. 46, no. 1, pp. 136–152, Jan. 2000.

[88] O. Kosut and L. Sankar, "Asymptotics and non-asymptotics for universal fixed-to-variable source coding," *IEEE Trans. Inf. Theory*, vol. 63, no. 6, pp. 3757–3772, June 2017.

[89] I. Kontoyiannis, "Second-order noiseless source coding theorems," *IEEE Trans. Inf. Theory*, vol. 43, no. 4, pp. 1339–1341, July 1997.

[90] Y. Polyanskiy, H. V. Poor, and S. Verdú, "Channel coding rate in the finite blocklength regime," *IEEE Trans. Inf. Theory*, vol. 56, no. 5, pp. 2307–2359, May 2010.

[91] V. Y. F. Tan and P. Moulin, "Second-order capacities of erasure and list decoding," in *Proc. 2014 IEEE Int. Symp. Inf. Theory*, June 2014, pp. 1887–1891.

[92] P. Moulin, "The log-volume of optimal codes for memoryless channels, asymptotically within a few nats," *IEEE Trans. Inf. Theory*, vol. 63, no. 4, pp. 2278–2313, Apr. 2017.

[93] M. Hayashi, "Information spectrum approach to second-order coding rate in channel coding," *IEEE Trans. Inf. Theory*, vol. 55, no. 11, pp. 4947–4966, Nov. 2009.

[94] P. Moulin, "The log-volume of optimal constant-composition codes for memoryless channels, within $o(1)$ bits," in *Proc. 2012 IEEE Int. Symp. Inf. Theory*, July 2012, pp. 826–830.

[95] V. Kostina and S. Verdú, "Channels with cost constraints: Strong converse and dispersion," *IEEE Trans. Inf. Theory*, vol. 61, no. 5, pp. 2415–2429, May 2015.

[96] M. Tomamichel and V. Y. F. Tan, "A tight upper bound for the third-order asymptotics for most discrete memoryless channels," *IEEE Trans. Inf. Theory*, vol. 59, no. 11, pp. 7041–7051, Nov. 2013.

[97] M. Tomamichel and M. Hayashi, "A hierarchy of information quantities for finite block length analysis of quantum tasks," *IEEE Trans. Inf. Theory*, vol. 59, no. 11, pp. 7693–7710, Nov. 2013.

[98] T. S. Han and R. Nomura, "First- and second-order hypothesis testing for mixed memoryless sources with general mixture," in *Proc. 2017 IEEE Int. Symp. Inf. Theory*, June 2017, pp. 126–130.

[99] Y. Huang and P. Moulin, "Strong large deviations for composite hypothesis testing," in *Proc. 2014 IEEE Int. Symp. Inf. Theory*, June 2014, pp. 556–560.

[100] L. Itti and P. Baldi, "Bayesian surprise attracts human attention," *Vis. Res.*, vol. 49, no. 10, pp. 1295–1306, June 2009.

[101] P. Baldi and L. Itti, "Of bits and wows: A Bayesian theory of surprise with applications to attention," *Neural Netw.*, vol. 23, no. 5, pp. 649–666, June 2010.

[102] A. Feinstein, "A new basic theorem of information theory," *IRE Trans. Inf. Theory*, vol. IT-4, no. 4, pp. 2–22, Sep. 1954.

[103] A. Ingber, D. Wang, and Y. Kochman, "Dispersion theorems via second order analysis of functions of distributions," in *2012 46th Annual Conference on Information Sciences and Systems (CISS)*, Mar. 2012.

[104] U. von Luxburg, R. C. Williamson, and I. Guyon, "Clustering: Science or art?" in *Proc. 29th Int. Conf. Mach. Learn. (ICML 2012)*, vol. 27, July 2012, pp. 65–79.

[105] A. Banerjee, S. Merugu, I. S. Dhillon, and J. Ghosh, "Clustering with Bregman divergences," *J. Mach. Learn. Res.*, vol. 6, pp. 1705–1749, Oct. 2005.

[106] D. Dueck and B. J. Frey, "Non-metric affinity propagation for unsupervised image categorization," in *Proc. 11th IEEE Int. Conf. Computer Vision*, Oct. 2007, pp. 1–8.

[107] Z. Kato, J. Zerubia, and M. Berthod, "Unsupervised parallel image classification using Markovian models," *Pattern Recognit.*, vol. 32, no. 4, pp. 591–604, June 1999.

[108] T.-W. Lee and M. S. Lewicki, "Unsupervised image classification, segmentation, and enhancement using ICA mixture models," *IEEE Trans. Image Process.*, vol. 11, no. 3, pp. 270–279, Mar. 2002.

[109] H. Ren and C.-I. Chang, "A generalized orthogonal subspace projection approach to unsupervised multispectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 38, no. 6, pp. 2515–2528, Nov. 2000.

[110] N. Slonim, N. Friedman, and N. Tishby, "Agglomerative multivariate information bottleneck," in *Advances in Neural Information Processing Systems 14*, T. G. Dietterich, S. Becker, and Z. Ghahramani, Eds. Cambridge, MA: MIT Press, 2002, pp. 929–936.

[111] K. Nagano, Y. Kawahara, and S. Iwata, "Minimum average cost clustering," in *Advances in Neural Information Processing Systems 23*, J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, Eds. MIT Press, 2010, pp. 1759–1767.

[112] C. Chan, A. Al-Bashabsheh, J. B. Ebrahimi, T. Kaced, and T. Liu, "Multivariate mutual information inspired by secret-key agreement," *Proc. IEEE*, vol. 103, no. 10, pp. 1883–1913, Oct. 2015.

[113] W. J. McGill, "Multivariate information transmission," *IRE Trans. Inf. Theory*, vol. IT-4, no. 4, pp. 93–111, Sep. 1954.

[114] D. M. Fass, "Human sensitivity to mutual information," Ph.D. dissertation, Rutgers, The State University of New Jersey, New Brunswick, Jan. 2006.

[115] A. Jakulin, "Machine learning based on attribute interactions," Ph.D. dissertation, University of Ljubljana, Slovenia, June 2005.

[116] J. L. Boes and C. R. Meyer, "Multi-variate mutual information for registration," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI'99*, ser. Lecture Notes in Computer Science, C. Taylor and A. Colchester, Eds. Berlin: Springer, 2006, vol. 1679, pp. 606–612.

[117] Y.-M. Zhu, "Volume image registration by cross-entropy optimization," *IEEE Trans. Med. Imag.*, vol. 21, no. 2, pp. 174–180, Feb. 2002.

[118] L. Zollei, "A unified information theoretic framework for pair- and group-wise registration of medical images," Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, MA, Jan. 2006.

[119] W. R. Ashby, "Measuring the internal informational exchange in a system," *Cybernetica*, vol. 8, no. 1, pp. 5–22, 1965.

[120] R. C. Conant, "Laws of information which govern systems," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-6, no. 4, pp. 240–255, Apr. 1976.

[121] G. Chechik, M. J. Anderson, O. Bar-Yosef, E. D. Young, N. Tishby, and I. Nelken, "Reduction of information redundancy in the ascending auditory pathway," *Neuron*, vol. 51, no. 3, pp. 359–368, Aug. 2006.

[122] Y.-S. Liu and B. L. Hughes, "A new universal random bound for the multiple-access channel," *IEEE Trans. Inf. Theory*, vol. 42, no. 2, pp. 376–386, Mar. 1996.

[123] H. Wang and P. Viswanath, "Vector Gaussian multiple description with individual and central receivers," *IEEE Trans. Inf. Theory*, vol. 53, no. 6, pp. 2133–2153, June 2007.

[124] V. Misra, V. K. Goyal, and L. R. Varshney, "Distributed scalar quantization for computing: High-resolution analysis and extensions," *IEEE Trans. Inf. Theory*, vol. 57, no. 8, pp. 5298–5325, Aug. 2011.

[125] M. S. Pinsker, *Information and Information Stability of Random Variables and Processes*. San Francisco: Holden-Day, 1964.

[126] I. Csiszár and Z. Talata, "Context tree estimation for not necessarily finite memory processes, via BIC and MDL," *IEEE Trans. Inf. Theory*, vol. 52, no. 3, pp. 1007–1016, Mar. 2006.

[127] S. Kullback and R. A. Leibler, "On information and sufficiency," *Ann. Math. Stat.*, vol. 22, no. 1, pp. 79–86, 1951.

[128] H. Chernoff, "Large-sample theory: Parametric case," *Ann. Math. Stat.*, vol. 27, no. 1, pp. 1–22, Mar. 1956.

[129] L. Paninski, "Estimation of entropy and mutual information," *Neural Comput.*, vol. 15, no. 6, pp. 1191–1253, June 2003.

[130] A. Kittur, E. H. Chi, and B. Suh, "Crowdsourcing user studies with Mechanical Turk," in *Proc. SIGCHI Conf. Hum. Factors Comput. Syst. (CHI 2008)*, Apr. 2008, pp. 453–456.

[131] P. G. Ipeirotis, F. Provost, and J. Wang, "Quality management on Amazon Mechanical Turk," in *Proc. ACM SIGKDD Workshop Human Comput. (HCOMP'10)*, July 2010, pp. 64–67.

[132] N. Scheiber, "A middle ground between contract worker and employee," *The New York Times*, Dec. 2015.

[133] F. Gino and G. Pisano, "Toward a theory of behavioral operations," *Manuf. Service Oper. Manag.*, vol. 10, no. 4, pp. 676–691, Fall 2008.

[134] D. R. Karger, S. Oh, and D. Shah, "Budget-optimal task allocation for reliable crowdsourcing systems," *Oper. Res.*, vol. 62, no. 1, pp. 1–24, Jan.-Feb. 2014.

[135] V. Misra and T. Weissman, "Unsupervised learning and universal communication," in *Proc. 2013 IEEE Int. Symp. Inf. Theory*, July 2013, pp. 261–265.

[136] H. J. Jung, Y. Park, and M. Lease, "Predicting next label quality: A time-series model of crowdwork," in *Proc. AAAI Conf. Human Comput. and Crowdsourcing (HCOMP'14)*, 2014.

[137] A. Vempaty, L. R. Varshney, and P. K. Varshney, "Reliable crowdsourcing for multi-class labeling using coding theory," *IEEE J. Sel. Topics Signal Process.*, vol. 8, no. 4, pp. 667–679, Aug. 2014.

[138] D. R. Karger, S. Oh, and D. Shah, "Efficient crowdsourcing for multi-class labeling," in *Proc. ACM SIGMETRICS Int. Conf. Meas. Model. Comput. Syst.*, June 2013, pp. 81–92.

[139] N. B. Shah, S. Balakrishnan, and M. J. Wainwright, "A permutation-based model for crowd labeling: Optimal estimation and robustness," arXiv:1606.09632, June 2016.

[140] J. Zhang, V. S. Sheng, J. Wu, and X. Wu, "Multi-class ground truth inference in crowdsourcing with clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 4, pp. 1080–1085, Apr. 2016.

[141] V. Misra, "Universal communication and clustering," Ph.D. dissertation, Stanford University, June 2014.

[142] R. K. Vinayak and B. Hassibi, "Crowdsourced clustering: Querying edges vs triangles," in *Advances in Neural Information Processing Systems 29*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, Eds. Curran Associates, Inc., 2016, pp. 1316–1324.

[143] A. Mazumdar and B. Saha, "Clustering via crowdsourcing," arXiv:1604.01839, Apr. 2016.

[144] L. R. Varshney, P. Jyothi, and M. Hasegawa-Johnson, "Language coverage for mismatched crowdsourcing," in *Proc. 2016 Inf. Theory Appl. Workshop*, Feb. 2016.

[145] Q. Li, A. Vempaty, L. R. Varshney, and P. K. Varshney, "Multi-object classification via crowdsourcing with a reject option," *IEEE Trans. Signal Process.*, vol. 65, no. 4, pp. 1068–1081, Feb. 2017.

[146] S. M. Ali and S. D. Silvey, "A general class of coefficients of divergence of one distribution from another," *J. R. Stat. Soc. Ser. B. Methodol.*, vol. 28, no. 1, pp. 131–142, 1966.

[147] I. Csiszár, "Information-type measures of difference of probability distributions and indirect observations," *Stud. Sci. Math. Hung.*, vol. 2, pp. 299–318, 1967.

[148] S. S. Dragomir, Ed., *Inequalities for Csiszár f-Divergence in Information Theory*, ser. RGMIA Monographs. Victoria University, 2000.

[149] I. Csiszár and Z. Talata, "Context tree estimation for not necessarily finite memory processes, via BIC and MDL," *IEEE Trans. Inf. Theory*, vol. 52, no. 3, pp. 1007–1016, Mar. 2006.

[150] B. Yu, *Assouad, Fano, and Le Cam*. New York, NY: Springer New York, 1997, pp. 423–435.

[151] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," http://bitcoin.org/bitcoin.pdf, 2008.

[152] J. Bonneau, A. Miller, J. Clark, A. Narayanan, J. A. Kroll, and E. W. Felten, "SoK: Research perspectives and challenges for bitcoin and cryptocurrencies," in *Proc. 2015 IEEE Symp. Security Privacy*, May 2015, pp. 104–121.

[153] A. Narayanan, J. Bonneau, E. Felten, A. Miller, and S. Goldfeder, *Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction*. Princeton: Princeton University Press, 2016.

[154] A. Azaria, A. Ekblaw, T. Vieira, and A. Lippman, "Medrec: Using blockchain for medical data access and permission management," in *2nd Int. Conf. Open Big Data (OBD 2016)*, Aug. 2016, pp. 25–30.

[155] M. J. Casey and P. Wong, "Global supply chains are about to get better, thanks to blockchain," Harvard Bus. Rev., Mar. 2017. [Online]. Available: https://hbr.org/2017/03/global-supply-chains-are-about-to-get-better-thanks-to-blockchain

[156] D. Tapscott and A. Tapscott, *Blockchain Revolution: How the Technology behind Bitcoin is Changing Money, Business, and the World*. New York: Penguin, 2016.

[157] M. Iansiti and K. R. Lakhani, "The truth about blockchain," *Harvard Bus. Rev.*, vol. 95, no. 1, pp. 118–127, Jan. 2017.

[158] A. Kosba, A. Miller, E. Shi, Z. Wen, and C. Papamanthou, "Hawk: The blockchain model of cryptography and privacy-preserving smart contracts," in *Proc. 2016 IEEE Symp. Security Privacy*, May 2016, pp. 839–858.

[159] "Blockchain info." [Online]. Available: https://blockchain.info/home

[160] M. Vilim, H. Duwe, and R. Kumar, "Approximate bitcoin mining," in *Proc. 53rd Des. Autom. Conf. (DAC '16)*, June 2016, pp. 97:1–97:6.

[161] P. Fairley, "Blockchain world - Feeding the blockchain beast if bitcoin ever does go mainstream, the electricity needed to sustain it will be enormous," *IEEE Spectr.*, vol. 54, no. 10, pp. 36–59, Oct. 2017.

[162] M. O. Rabin, "The information dispersal algorithm and its applications," in *Sequences*, R. M. Capocelli, Ed. New York: Springer-Verlag, 1990, pp. 406–419.

[163] C. Cachin and S. Tessaro, "Asynchronous verifiable information dispersal," in *24th IEEE Symp. Rel. Distrib. Sys. (SRDS'05)*, Oct. 2005, pp. 191–201.

[164] A. G. Dimakis and K. Ramchandran, "Network coding for distributed storage in wireless networks," in *Networked Sensing Information and Control*, V. Saligrama, Ed. New York: Springer, 2008, pp. 115–136.

[165] K. V. Rashmi, N. B. Shah, P. V. Kumar, and K. Ramchandran, "Explicit construction of optimal exact regenerating codes for distributed storage," in *Proc. 47th Annu. Allerton Conf. Commun. Control Comput.*, Sep. 2009, pp. 1243–1249.

[166] A. S. Rawat, O. O. Koyluoglu, N. Silberstein, and S. Vishwanath, "Optimal locally repairable and secure codes for distributed storage systems," *IEEE Trans. Inf. Theory*, vol. 60, no. 1, pp. 212–236, Jan. 2014.

[167] S. Pawar, S. El Rouayheb, and K. Ramchandran, "Securing dynamic distributed storage systems against eavesdropping and adversarial attacks," *IEEE Trans. Inf. Theory*, vol. 57, no. 10, pp. 6734–6753, Oct. 2011.

[168] R. K. Raman and L. R. Varshney, "Dynamic distributed storage for scaling blockchains," arXiv:1711.07617v2 [cs.IT], Jan. 2018.

[169] A. Shamir, "How to share a secret," *Commun. ACM*, vol. 22, no. 11, pp. 612–613, Nov. 1979.

[170] H. Krawczyk, "Secret sharing made short," in *Advances in Cryptology — CRYPTO '93*, ser. Lecture Notes in Computer Science, D. R. Stinson, Ed. Berlin: Springer, 1994, vol. 773, pp. 136–146.

[171] Q. Yu, N. Raviv, J. So, and A. S. Avestimehr, "Lagrange coded computing: Optimal design for resiliency, security and privacy," *arxiv:1806.00939 [CS.IT]*, June 2018.

[172] S. Li, M. Yu, A. S. Avestimehr, S. Kannan, and P. Viswanath, "Polyshard: Coded sharding achieves linearly scaling efficiency and security simultaneously," *arXiv:1809.10361 [cs.CR]*, Sep. 2018.

[173] V. Bagaria, S. Kannan, D. Tse, G. Fanti, and P. Viswanath, "Deconstructing the blockchain to approach physical limits," *arXiv:1810.08092 [cs.CR]*, Oct. 2018.

[174] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. D. Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich, S. Muralidharan, C. Murthy, B. Nguyen, M. Sethi, G. Singh, K. Smith, A. Sorniotti, C. Stathakopoulou, M. Vukolić, S. W. Cocco, and J. Yellick, "Hyperledger fabric: A distributed operating system for permissioned blockchains," in *Proc. 13th EuroSys Conf.*, ser. EuroSys '18, Apr. 2018, pp. 30:1–30:15.

[175] P. Rogaway and T. Shrimpton, "Cryptographic hash-function basics: Definitions, implications, and separations for preimage resistance, second-preimage resistance, and collision resistance," in *Fast Software Encryption*, B. Roy and W. Meier, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 371–388.

[176] D. R. Stinson, "Some observations on the theory of cryptographic hash functions," *Designs, Codes and Cryptography*, vol. 38, no. 2, pp. 259–277, Feb. 2006.

[177] R. J. McEliece and D. V. Sarwate, "On sharing secrets and Reed-Solomon codes," *Commun. ACM*, vol. 24, no. 9, pp. 583–584, Sep. 1981.

[178] E. Karnin, J. Greene, and M. Hellman, "On secret sharing systems," *IEEE Trans. Inf. Theory*, vol. IT-29, no. 1, pp. 35–41, Jan. 1983.

[179] W. Huang, M. Langberg, J. Kliewer, and J. Bruck, "Communication efficient secret sharing," *IEEE Trans. Inf. Theory*, vol. 62, no. 12, pp. 7195–7206, Dec. 2016.

[180] J. L. Massey, "Minimal codewords and secret sharing," in *Proc. 6th Joint Swedish-Russian Int. Workshop Inf. Theory*, Aug. 1993, pp. 276–279.

[181] C. E. Shannon, "Communication theory of secrecy systems," *Bell Syst. Tech. J.*, vol. 28, pp. 656–715, Oct. 1949.

[182] A. G. Dimakis, K. Ramchandran, Y. Wu, and C. Suh, "A survey on network codes for distributed storage," *Proc. IEEE*, vol. 99, no. 3, pp. 476–489, Mar. 2011.

[183] Y. Wu, A. G. Dimakis, and K. Ramchandran, "Deterministic regenerating codes for distributed storage," in *Proc. 45th Annu. Allerton Conf. Commun. Control Comput.*, Sep. 2007, pp. 242–249.

[184] N. B. Shah, K. V. Rashmi, P. V. Kumar, and K. Ramchandran, "Distributed storage codes with repair-by-transfer and nonachievability of interior points on the storage-bandwidth tradeoff," *IEEE Trans. Inf. Theory*, vol. 58, no. 3, pp. 1837–1852, Mar. 2012.

[185] R. Durrett, *Random Graph Dynamics*. Cambridge: Cambridge University Press, 2007.

[186] J. Gottlieb, B. A. Julstrom, G. R. Raidl, and F. Rothlauf, "Prüfer numbers: A poor representation of spanning trees for evolutionary search," in *Proc. 3rd Ann. Conf Genetic Evolutionary Comput.*, 2001, pp. 343–350.

[187] Z. Baranyai, "The edge-coloring of complete hypergraphs I," *J. Comb. Theory, Ser. B*, vol. 26, no. 3, pp. 276–294, 1979.

[188] R. K. Raman and L. R. Varshney, "Distributed storage meets secret sharing on the blockchain," in *Proc. 2018 Inf. Theory Appl. Workshop*, Feb. 2018.

[189] X. Ma, "On the feasibility of data loss insurance for personal cloud storage," in *Proc. 6th USENIX Conf. Hot Topics Storage File Sys.*, June 2014.

[190] Y. Kim, R. K. Raman, Y. Kim, L. R. Varshney, and N. R. Shanbhag, "Efficient local secret sharing for distributed blockchain systems," *IEEE Commun. Lett.*, vol. 23, no. 2, pp. 282–285, Feb. 2019.

[191] P. Gopalan, C. Huang, H. Simitci, and S. Yekhanin, "On the locality of codeword symbols," *IEEE Trans. Inf. Theory*, vol. 58, no. 11, pp. 6925–6934, Nov. 2012.

[192] N. Silberstein, A. S. Rawat, O. O. Koyluoglu, and S. Vishwanath, "Optimal locally repairable codes via rank-metric codes," in *Proc. 2013 IEEE Int. Symp. Inf. Theory*, July 2013, pp. 1819–1823.

[193] I. Tamo and A. Barg, "A family of optimal locally recoverable codes," *IEEE Trans. Inf. Theory*, vol. 60, no. 8, pp. 4661–4676, Aug. 2014.

[194] D. J. Power, "Data science: Supporting decision-making," *J. Decis. Sys.*, vol. 25, no. 4, pp. 345–356, Apr. 2016.

[195] D. Shah, "Data science and statistics: Opportunities and challenges," *Technol. Rev.*, Sep. 2016.

[196] T. Smith, N. Maire, A. Ross, M. Penny, N. Chitnis, A. Schapira, A. Studer, B. Genton, C. Lengeler, F. Tediosi, D. d. Savigny, and M. Tanner, "Towards a comprehensive simulation model of malaria epidemiology and control," *Parasitology*, vol. 135, no. 13, p. 1507?1516, Aug. 2008.

[197] J. D. Piette, S. L. Krein, D. Striplin, N. Marinec, R. D. Kerns, K. B. Farris, S. Singh, L. An, and A. A. Heapy, "Patient-centered pain care using artificial intelligence and mobile health tools: Protocol for a randomized study funded by the US Department of Veterans Affairs Health Services Research and Development Program," *JMIR Res. Protocols*, vol. 5, no. 2, 2016.

[198] J. Nelson, "The operation of non-governmental organizations (NGOs) in a world of corporate and other codes of conduct," *Corporate Social Responsibility Initiative*, Mar. 2007.

[199] "CDC/ATSDR policy on releasing and sharing data," Sep. 2005. [Online]. Available: https://www.cdc.gov/maso/policy/releasingdata.pdf

[200] W. G. V. Panhuis, P. Paul, C. Emerson, J. Grefenstette, R. Wilder, A. J. Herbst, D. Heymann, and D. S. Burke, "A systematic review of barriers to data sharing in public health," *BMC Public Health*, vol. 14, no. 1, p. 1144, Feb. 2014.

[201] D. R. Wong, S. Bhattacharya, and A. J. Butte, "Prototype of running clinical trials in an untrustworthy environment using blockchain," *Nature Commun.*, vol. 10, no. 1, p. 917, Feb. 2019.

[202] Z. Koticha, "2018: Blockchain scaling > all else," https://medium.com/thunderofficial/2018-blockchain-scaling-all-else-7937b660c08, June 2018.

[203] R. K. Raman and L. R. Varshney, "Dynamic distributed storage for blockchains," in *Proc. 2018 IEEE Int. Symp. Inf. Theory*, July 2018.

[204] R. K. Raman, R. Vaculin, M. Hind, S. L. Remy, E. K. Pissadaki, N. K. Bore, R. Daneshvar, B. Srivastava, and K. R. Varshney, "Trusted multi-party computation and verifiable simulations: A scalable blockchain approach," arXiv:1809.08438 [CS.DC], Sep. 2018.

[205] N. K. Bore, R. K. Raman, I. M. Markus, S. L. Remy, O. Bent, M. Hind, E. K. Pissadaki, B. Srivastava, R. Vaculin, K. R. Varshney, and K. Weldemariam, "Promoting distributed trust in machine learning and computational simulation via a blockchain network," arXiv:1810.11126 [CS.DC], Oct. 2018.

[206] R. K. Raman, R. Vaculin, M. Hind, S. L. Remy, E. K. Pissadaki, N. K. Bore, R. Daneshvar, B. Srivastava, and K. R. Varshney, "A scalabale blockchain approach for trusted computation and verifiable simulation in multi-party collaboration," in *Proc. IEEE Int. Conf. Blockchain Cryptocurrency*, May 2019.

[207] N. K. Bore, R. K. Raman, I. M. Markus, S. L. Remy, O. Bent, M. Hind, E. K. Pissadaki, B. Srivastava, R. Vaculin, K. R. Varshney, and K. Weldemariam, "Promoting distributed trust in machine learning and computational simulation," in *Proc. IEEE Int. Conf. Blockchain Cryptocurrency*, May 2019.

[208] R. K. Raman, K. R. Varshney, R. Vaculin, N. K. Bore, S. L. Remy, E. K. Pissadaki, and M. Hind, "Constructing and compressing frames in blockchain-based verifiable multi-party computation," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, May 2019.

[209] S. L. Remy, O. Bent, and N. Bore, "Reshaping the use of digital tools to fight malaria," *arXiv:1805.05418 [cs.CY]*, May 2018.

[210] O. Bent, S. L. Remy, S. Roberts, and A. Walcott-Bryant, "Novel exploration techniques (NETs) for malaria policy interventions," in *Proc. 32nd AAAI Conf. Artif. Intell.*, Feb. 2018.

[211] O. E. Gundersen and S. Kjensmo, "State of the art: Reproducibility in artificial intelligence," in *Proc. 32nd AAAI Conf. Artif. Intell.*, New Orleans, USA, Feb. 2018.

[212] P. Henderson, R. Islam, P. Bachman, J. Pineau, D. Precup, and D. Meger, "Deep reinforcement learning that matters," *arXiv:1709.06560v2 [cs.LG]*, Nov. 2017.

[213] A. Brinckman, K. Chard, N. Gaffney, M. Hategan, M. B. Jones, K. Kowalik, S. Kulasekaran, B. Ludher, B. D. Mecum, J. Nabrzyski, V. Stodden, I. J. Taylor, M. J. Turk, and K. Turner, "Computing environments for reproducibility: Capturing the "whole tale"," *Future Generation Comput. Sys.*, vol. 94, pp. 854 – 867, May 2019.

[214] J. Singh, J. Cobbe, and C. Norval, "Decision provenance: Capturing data flow for accountable systems," *arXiv:1804.05741 [cs.CY]*, Apr. 2018.

[215] D. Verma, S. Calo, and G. Cirincione, "Distributed AI and security issues in federated environments," in *Proc. Workshop Program 19th Int. Conf. Distrib. Comput. Netw. (ICDN 2018)*, Jan. 2018.

[216] A. Haeberlen, P. Kouznetsov, and P. Druschel, "Peerreview: Practical accountability for distributed systems," *SIGOPS Oper. Syst. Rev.*, vol. 41, no. 6, pp. 175–188, Oct. 2007.

[217] A. Haeberlen, "A case for the accountable cloud," *SIGOPS Oper. Syst. Rev.*, vol. 44, no. 2, pp. 52–57, Apr. 2010.

[218] Z. Xiao and Y. Xiao, "Security and privacy in cloud computing," *IEEE Commun. Surveys Tutorials*, vol. 15, no. 2, pp. 843–859, July 2013.

[219] J. A. Kroll, S. Barocas, E. W. Felten, J. R. Reidenberg, D. G. Robinson, and H. Yu, "Accountable algorithms," *University of Pennsylvania Law Review*, vol. 165, pp. 633–706, Feb. 2017.

[220] R. Falcone, M. Singh, and Y.-H. Tan, *Trust in cyber-societies: Integrating the human and artificial perspectives*. Springer Science & Business Media, 2001, vol. 2246.

[221] S. P. Marsh, "Formalising trust as a computational concept," Ph.D. dissertation, University of Stirling, 1994.

[222] P. Dasgupta, "Trust as a commodity," *Trust: Making and Breaking Cooperative Relations*, vol. 4, pp. 49–72, 2000.

[223] S. D. Ramchurn, D. Huynh, and N. R. Jennings, "Trust in multi-agent systems," vol. 19, no. 1, pp. 1–25, 2004.

[224] D. A. Grier, "Error identification and correction in human computation: Lessons from the WPA," in *Human Computation*, 2011.

[225] M. Vukolić, "Rethinking permissioned blockchains," in *Proc. ACM Workshop Blockchain, Cryptocurrencies, Contracts (BCC '17)*, Apr. 2017, pp. 3–7.

[226] J. Tsai, "Transform blockchain into distributed parallel computing architecture for precision medicine," in *Proc. 38th IEEE Int. Conf. Distrib. Comput. Systems (ICDCS 2018)*, July 2018, pp. 1290–1299.

[227] H. I. Ozercan, A. M. Ileri, E. Ayday, and C. Alkan, "Realizing the potential of blockchain technologies in genomics," *Genome Res.*, 2018.

[228] C. W. Granger and R. Joyeux, "An introduction to long-memory time series models and fractional differencing," *J. Time Ser. Anal.*, vol. 1, no. 1, pp. 15–29, Jan. 1980.

[229] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*. Springer Science & Business Media, 2012, vol. 159.

[230] S. D. Servetto, V. A. Vaishampayan, and N. J. A. Sloane, "Multiple description lattice vector quantization," in *Proc. IEEE Data Compression Conf. (DCC 1999)*, Mar. 1999, pp. 13–22.

[231] D. Pollard, *Section 4: Packing and covering in Euclidean spaces*, ser. Regional Conference Series in Probability and Statistics. Institute of Mathematical Statistics and American Statistical Association, 1990, pp. 14–20.

[232] W. H. R. Equitz and T. M. Cover, "Successive refinement of information," *IEEE Trans. Inf. Theory*, vol. 37, no. 2, pp. 269–275, Mar. 1991.

[233] D. Mukherjee and S. K. Mitra, "Successive refinement lattice vector quantization," *IEEE Trans. Signal Process.*, vol. 11, no. 12, pp. 1337–1348, Dec. 2002.

[234] Y. Liu and W. A. Pearlman, "Multistage lattice vector quantization for hyperspectral image compression," in *Conf. Rec. 41st Asilomar Conf. Signals, Syst. Comput.*, Nov. 2007, pp. 930–934.

[235] A. W. Marshall and I. Olkin, "Multivariate Chebyshev inequalities," *Ann. Math. Stat.*, vol. 31, no. 4, pp. 1001–1014, Dec. 1960.

[236] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[237] M. D. Donsker and S. R. S. Varadhan, "Asymptotic evaluation of certain Markov process expectations for large time. IV," *Communications on Pure and Applied Mathematics*, vol. 36, no. 2, pp. 183–212, 1983.

[238] M. I. Belghazi, A. Baratin, S. Rajeshwar, S. Ozair, Y. Bengio, A. Courville, and D. Hjelm, "Mutual information neural estimation," J. Dy and A. Krause, Eds., vol. 80, July 2018, pp. 531–540.

[239] M. Gabrié, A. Manoel, C. Luneau, J. Barbier, N. Macris, F. Krzakala, and L. Zdeborová, "Entropy and mutual information in models of deep neural networks," in *Proc. 31st Annu. Conf. Neural Inf. Process. Syst. (NIPS)*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., Dec. 2018, pp. 1821–1831.

[240] R. Chen, A. B. Das, and L. R. Varshney, "Registration for image-based transcriptomics: Parametric signal features and multivariate information measures," in *Proc. 53rd Annu. Conf. Inf. Sci. Syst. (CISS 2019)*, Mar. 2019.

[241] A. Sedghi, J. Luo, A. Mehrtash, S. Pieper, C. M. Tempany, T. Kapur, P. Mousavi, and W. M. Wells III, "Deep information theoretic registration," *arXiv:1901.00040 [cs.CV]*, Jan. 2019.

[242] C. Chan, A. Al-Bashabsheh, Q. Zhou, T. Kaced, and T. Liu, "Info-clustering: A mathematical theory for data clustering," *IEEE Trans. Mol. Biol. Multi-Scale Commun.*, vol. 2, no. 1, pp. 64–91, June 2016.

[243] K. Moon and A. Hero, "Multivariate f-divergence estimation with confidence," in *Proc. 27th Annu. Conf. Neural Inf. Process. Syst. (NIPS)*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, Inc., Dec. 2014, pp. 2420–2428.

[244] W. Gao, S. Oh, and P. Viswanath, "Demystifying fixed$k$-nearest neighbor information estimators," *IEEE Trans. Inf. Theory*, vol. 64, no. 8, pp. 5629–5661, Aug. 2018.

[245] A. Rahimzamani, H. Asnani, P. Viswanath, and S. Kannan, "Estimators for multivariate information measures in general probability spaces," in *Proc. 31st Annu. Conf. Neural Inf. Process. Syst. (NIPS)*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., Dec. 2018, pp. 8664–8675.

[246] P. Netrapalli, S. Banerjee, S. Sanghavi, and S. Shakkottai, "Greedy learning of Markov network structure," in *Proc. 48th Annu. Allerton Conf. Commun. Control Comput.*, Sept 2010, pp. 1295–1302.

[247] A. Antos and I. Kontoyiannis, "Convergence properties of functional estimates for discrete distributions," *Rand. Str. & Alg.*, vol. 19, no. 3-4, pp. 163–193, 2001.